ECS Integration Test Cases

Table of Contents

1 BE_82_01 1
1.1 Performance 1
1.1.1 Performance - Aggregate (ECS-ECSTC-91) 1
1.1.2 Performance - Granule Inserts/Deletes (ECS-ECSTC-92) 8
1.1.3 Performance - QA Updates (ECS-ECSTC-93) 11
1.1.4 Performance - Datapool Inserts/Updates/Deletes (ECS-ECSTC-94) 15
1.1.5 Performance - Granule Updates (ECS-ECSTC-95) 17
1.1.6 Performance - Collection Inserts/Updates (ECS-ECSTC-96) 22
1.1.7 Performance - MISBR (ECS-ECSTC-97) 25
1.1.8 Performance - Long Form Verification (ECS-ECSTC-98) 29
1.2 Nominal Collection Export[S-1]: Manual Export (ECS-ECSTC-1) 31
1.3 ECS Collection Additional Metadata[S-01]: ECS CoordinateSystem (ECS-ECSTC-2) . 35
1.4 Nominal Collection Export[S-2a]: Automatic Export: Insert Collection (ECS-ECSTC-3) 39
1.5 Nominal Collection Export[S-2b]: Automatic Export: Delete Collection (ECS-ECSTC-4) 43
1.6 Nominal Collection Export[S-2c]: Automatic Export: Update Collection (ECS-ECSTC-5) 46
1.7 Nominal Collection Export[S-2d]: Automatic Export: Enable Collection Export (ECS-ECSTC-6) 50
1.8 Collection Deletion (ECS-ECSTC-7) 53
1.9 ECS Granule Additional Metadata[S-1]: Backtrack Orbit Metadata (ECS-ECSTC-8) ... 57
1.10 Datapool URLs[S-3] Publish, Unpublish, Delete (ECS-ECSTC-9) 72
1.11 Nominal Granule Export[S-1]: Manual Export (ECS-ECSTC-10) 77
1.12 Nominal Granule Export[S-2a]: Automatic Export: Ingest Granule (ECS-ECSTC-11) .. 84
1.13 Nominal Granule Export[S-2b]: Automatic Export: Logically Delete Granule (ECS-ECSTC-12) 88
1.14 Nominal Granule Export[S-2c]: Automatic Export: Physically Delete Granule (ECS-ECSTC-13) 92
1.15 Nominal Granule Export[S-2d]: Automatic Export: DFA Granule (ECS-ECSTC-14).... 96
1.16 Nominal Granule Export[S-2e]: Automatic Export: Hide a Granule (ECS-ECSTC-15) 100
1.17 Nominal Granule Export[S-2f]: Automatic Export: Restrict Granule (ECS-ECSTC-16) 105
1.18 Nominal Granule Export[S-2g]: Automatic Export: Unrestrict Granule (ECS-ECSTC-17) 109
1.19 Nominal Granule Export[S-2h]: Automatic Export: QAUpdate granule. (ECS-ECSTC-18) 115
1.20 Nominal Granule Export[S-2i]: Automatic Export: Publish Granule (ECS-ECSTC-19) 121
1.21 Nominal Granule Export[S-2j]: Automatic Export: Unpublish Granule (ECS-ECSTC-20) 126
1.22 Nominal Granule Export[S-2k]: Automatic Export: Link to Browse (ECS-ECSTC-21)131

1.23 Nominal Granule Export[S-2l]: Automatic Export: Unlink from Browse (ECS-ECSTC-22) 136
1.24 Nominal Granule Export[S-2n]: Automatic Export: Move Collection (ECS-ECSTC-23) 141
1.25 Nominal Granule Export[S-2o]: Automatic Export: XML Replacement (ECS-ECSTC-24) 146
1.26 QA Granule URLs (ECS-ECSTC-25) 152
1.27 PH Granule URLs (ECS-ECSTC-26) 161
1.28 HDF Map Granule URLs (ECS-ECSTC-27) 170
1.29 Granule Deletion (ECS-ECSTC-28) 177
1.30 QA Update (ECS-ECSTC-29) 181
1.31 ISO Nominal Granule Export: Granule Export by Collection (ECS-ECSTC-30) 186
1.32 ISO Nominal Collection Export (ECS-ECSTC-31) 191
1.33 MISR Browse Links - Science Before Browse (ECS-ECSTC-32) 195
1.34 MISR Browse Links - Browse Before Science (ECS-ECSTC-33) 202
1.35 Nominal Automatic Export (ECS-ECSTC-34) 208
1.36 Automatic Export Event Consolidation (ECS-ECSTC-35) 220
1.37 Unexported Events (ECS-ECSTC-36) 227
1.38 Automatic Export Errors (ECS-ECSTC-37) 229
1.39 Manual Export - Collection Inserts (ECS-ECSTC-38) 235
1.40 Manual Export - Collection Deletes (ECS-ECSTC-39) 237
1.41 Manual Export - "Fake" Collection Delete (ECS-ECSTC-40) 243
1.42 Manual Export - Collection Full Update (ECS-ECSTC-41) 245
1.43 Manual Export - Granule Insert (ECS-ECSTC-42) 247
1.44 Manual Export - Granule Delete (ECS-ECSTC-43) 252
1.45 Manual Export - Insert Time (ECS-ECSTC-44) 255
1.46 Manual Export - Last Update Time (ECS-ECSTC-45) 258
1.47 Manual Export - Manual Granule Updates (ECS-ECSTC-46) 262
1.48 Manual Export - Multiple Options/Select by Group (ECS-ECSTC-47) 268
1.49 Manual Export - Nominal (ECS-ECSTC-48) 271
1.50 Manual Export - Invocation via Command Line[a]: granules w/ granule args (ECS-ECSTC-49) 275
1.51 Automatic/Manual Interaction (ECS-ECSTC-50) 278
1.52 Manual Export Errors (ECS-ECSTC-51) 283
1.53 GUI (ECS-ECSTC-52) 289
1.54 Configuring Collections For Export (ECS-ECSTC-53) 295
1.55 Re-Export Errors (ECS-ECSTC-54) 301
1.56 Ignorable Errors (ECS-ECSTC-55) 306
1.57 Ingest Errors (ECS-ECSTC-56) 309
1.58 Retriable HTTP Level Errors (ECS-ECSTC-57) 314
1.59 Contact ECHO Errors (ECS-ECSTC-58) 317
1.60 Data Related Errors (ECS-ECSTC-59) 320
1.61 Non Data Related Generation Errors (ECS-ECSTC-60) 323
1.62 Alert After Numerous Retries (ECS-ECSTC-61) 327
1.63 Fault Recovery (ECS-ECSTC-62) 331
1.64 Corrective Export (ECS-ECSTC-63) 333

1.65 Long Form Verification - Granule (ECS-ECSTC-64) 344
1.66 Long Form Verification - Collection (ECS-ECSTC-65) 348
1.67 Long Form Verification - Insert Time (ECS-ECSTC-66) 350
1.68 Long Form Verification - Last Update Time (ECS-ECSTC-67) 354
1.69 Long Form Verification - Granule Updates (ECS-ECSTC-68) 357
1.70 Long Form Verification - Multiple Options/Select by Group (ECS-ECSTC-69) 361
1.71 Long Form Verification - Size Limits (ECS-ECSTC-70) 363
1.72 Long Form Verification - Concurrent Operation (ECS-ECSTC-71) 366
1.73 Long Form Verification - Nominal (ECS-ECSTC-72) 369
1.74 Long Form Verification - Invocation via Command Line - a) Multiple granules (ECS-ECSTC-73) 375
1.75 Long Form Verification - Exclude Pending Exports (ECS-ECSTC-74) 378
1.76 Long Form Verification - Errors (ECS-ECSTC-75) 381
1.77 Incremental Verification - Nominal (ECS-ECSTC-76) 387
1.78 Incremental Verification - Reset (ECS-ECSTC-77) 399
1.79 Incremental Verification - Max Granules Exceeded (ECS-ECSTC-78) 402
1.80 Incremental Verification - Concurrent Operation (ECS-ECSTC-79) 404
1.81 Incremental Verification - Errors (ECS-ECSTC-80) 407
1.82 Short Form Verification - Granule (ECS-ECSTC-81) 411
1.83 Short Form Verification - Collection (ECS-ECSTC-82) 413
1.84 Short Form Verification - Concurrent Operation (ECS-ECSTC-83) 414
1.85 Short Form Verification - Nominal (ECS-ECSTC-84) 417
1.86 Short Form Verification - Exclude Pending Exports (ECS-ECSTC-85) 420
1.87 Short Form Verification - Errors (ECS-ECSTC-86) 423
1.88 Connecting to ECHO (ECS-ECSTC-87) 430
1.89 Monitor System (ECS-ECSTC-88) 433
1.90 Artifact Cleanup (ECS-ECSTC-89) 438
1.91 Export Takes Too Long (ECS-ECSTC-90) 441
1.92 ECS Collection Additional Metadata[S-02]: ECS GranuleSpatialRepresentation (ECS-ECSTC-99) 444
1.93 ECS Collection Additional Metadata[S-03]: ISO CoordinateSystem (ECS-ECSTC-100) 448
1.94 ECS Collection Additional Metadata[S-04]: ISO GranuleSpatialRepresentation (ECS-ECSTC-101) 452
1.95 ECS Collection Additional Metadata[S-05]: backtrack (ECS-ECSTC-102) 456
1.96 ECS Collection Additional Metadata[S-06]: no backtrack (ECS-ECSTC-103) 465
1.97 ECS Collection Additional Metadata[S-07]: TwoDCoordinateSystem (ECS-ECSTC-104) 469
1.98 ECS Collection Additional Metadata[S-08]: no TwoDCoordinateSystem (ECS-ECSTC-105) 474
1.99 ECS Collection Additional Metadata[S-09]: Product Specific Attributes (PSAs) (ECS-ECSTC-106) 479
1.100 ECS Collection Additional Metadata[S-10]: DIF ID (ECS-ECSTC-107) 483
1.101 ECS Collection Additional Metadata[S-12]: null ECS CoordinateSystem (ECS-ECSTC-108)487

1.102 ECS Collection Additional Metadata[S-13]: null ECS GranuleSpatialRepresentation (ECS-ECSTC-109) 491
1.103 ECS Granule Additional Metadata[S-2]: No Backtrack Orbit Metadata (ECS-ECSTC-127)495
1.104 ECS Granule Additional Metadata[S-3]: TwoDCoordinateSystem (ECS-ECSTC-128) 499
1.105 ECS Granule Additional Metadata[S-4]: No TwoDCoordinateSystem (ECS-ECSTC-129)504
1.106 ECS Granule Additional Metadata[S-5]: Product Specific Attributes (PSAs) (ECS-ECSTC-130) 507
1.107 ECS Granule Additional Metadata[S-6]: Cloud Cover from Core Metadata (ECS-ECSTC-131) 511
1.108 ECS Granule Additional Metadata[S-7]: Cloud Cover from PSA (ECS-ECSTC-132)514
1.109 ECS Granule Additional Metadata[S-8]: No CloudCover (ECS-ECSTC-133) 518
1.110 ECS Granule Additional Metadata[S-9]: Invalid Cloud Cover Location (ECS-ECSTC-134)521
1.111 ISO Nominal Granule Export: Granule Ingest (ECS-ECSTC-135) 525
1.112 ISO Nominal Granule Export: Logical Delete (ECS-ECSTC-136) 528
1.113 ISO Nominal Granule Export: Physical Delete (ECS-ECSTC-137)..... 531
1.114 ISO Nominal Granule Export: DFA (ECS-ECSTC-138)..... 534
1.115 ISO Nominal Granule Export: Hide (ECS-ECSTC-139)..... 537
1.116 ISO Nominal Granule Export: Restrict (ECS-ECSTC-140) 541
1.117 ISO Nominal Granule Export: Unrestrict (ECS-ECSTC-141)..... 545
1.118 ISO Nominal Granule Export: Publish (ECS-ECSTC-142)..... 548
1.119 ISO Nominal Granule Export: Unpublish (ECS-ECSTC-143) 552
1.120 ISO Nominal Granule Export: Move Collection (ECS-ECSTC-144)..... 556
1.121 Manual Export â€“ Invocation via Command Line[b]: granules w/ granule file (ECS-ECSTC-145) 561
1.122 Manual Export â€“ Invocation via Command Line[c]: granules w/ collection arg (ECS-ECSTC-146) 562
1.123 Manual Export â€“ Invocation via Command Line[d]: granules w/ collection file (ECS-ECSTC-147) 563
1.124 Manual Export â€“ Invocation via Command Line[e]: collections w/ collection args (ECS-ECSTC-148) 564
1.125 Manual Export â€“ Invocation via Command Line[f]: collections w/ collection file (ECS-ECSTC-149) 565
1.126 Manual Export â€“ Invocation via Command Line[g]: metadata w/ group arg (ECS-ECSTC-150) 566
1.127 Manual Export â€“ Invocation via Command Line[h]: metadata w/ group file (ECS-ECSTC-151) 567
1.128 Manual Export â€“ Invocation via Command Line[i]: granule deletions w/ collection arg (ECS-ECSTC-152) 569
1.129 Manual Export â€“ Invocation via Command Line[j]: granule deletions w/ collection file (ECS-ECSTC-153) 570
1.130 Manual Export â€“ Invocation via Command Line[k]: non-deleted granules w/ collection arg (ECS-ECSTC-154)..... 571

1.131 Manual Export â€“ Invocation via Command Line[l]: non-deleted granules w/ collection file (ECS-ECSTC-155) ... 572
1.132 Long Form Verification â€“ Invocation via Command Line - b) Granules in Granule File (ECS-ECSTC-156) ... 573
1.133 Long Form Verification â€“ Invocation via Command Line - c) Granules in Collection (ECS-ECSTC-157) ... 576
1.134 Long Form Verification â€“ Invocation via Command Line - d) Granules in Collection File (ECS-ECSTC-158) ... 578
1.135 Long Form Verification â€“ Invocation via Command Line - e) Multiple Collections (ECS-ECSTC-159) ... 581
1.136 Long Form Verification â€“ Invocation via Command Line - f) Collections in Collection File (ECS-ECSTC-160) ... 583
1.137 Long Form Verification â€“ Invocation via Command Line - g) Datapool Group (ECS-ECSTC-161) ... 586
1.138 Long Form Verification â€“ Invocation via Command Line - h) Group File (ECS-ECSTC-162) ... 589
1.139 Datapool URLs[S-6] EcDlCleanupGranules (ECS-ECSTC-348) ... 591

2 SD_82_01 ... 596
2.1 Invalid Collection/Series Level Schema Referenced (ECS-ECSTC-117) ... 596
2.2 Invalid Collection/Series Metadata Attribute Value (ECS-ECSTC-118) ... 601
2.3 XSD Generation Verification-ISO (ECS-ECSTC-119) ... 604
2.4 XSD Generation Verification - ECS (ECS-ECSTC-120) ... 606
2.5 Data Model Type Attribute Missing (ECS-ECSTC-113) ... 608
2.6 Collection/Series Level XPath File Missing Verification (ECS-ECSTC-114) ... 611
2.7 Granule/Dataset Level XPath File Missing (ECS-ECSTC-115) ... 613
2.8 Collection/Series Level Metadata File Missing (ECS-ECSTC-116) ... 616
2.9 XML metadata files for ISO Distribution (ECS-ECSTC-125) ... 618
2.10 QA Update Failure Verification (ECS-ECSTC-126) ... 620
2.11 Ingest XPath Validation (ECS-ECSTC-121) ... 624
2.12 ISO Ingest Attribute Type Conversion (ECS-ECSTC-122) ... 627
2.13 Invalid Granule/Dataset Metadata Attribute Value (ECS-ECSTC-123) ... 631
2.14 ODL metadata files for ECS Distribution (ECS-ECSTC-124) ... 633
2.15 Invalid Model Type (ECS-ECSTC-112) ... 636
2.16 Install ECS ESDT (ECS-ECSTC-111) ... 638
2.17 Install ISO ESDT (ECS-ECSTC-110) ... 641

3 FULL RESOLUTION BROWSE ... 645
3.1 Install AST_L1T ESDT (ECS-ECSTC-589) ... 645
3.2 Ingest AST_L1T granules into hidden online archive (ECS-ECSTC-590) ... 646
3.3 Export AST_L1T to ECHO (ECS-ECSTC-591) ... 647
3.4 Add AST_L1T subscription with ftp push action (ECS-ECSTC-592) ... 649
3.5 Ingest a public AST_L1T granule (ECS-ECSTC-593) ... 650
3.6 Run Rollup and EMS for AST_L1T (ECS-ECSTC-594) ... 651
3.7 Ingest a replacement AST_L1T granule (ECS-ECSTC-595) ... 652

4 8.3.2 RELEASE ... 654
4.1 NCR8051409 - Provide KML Outputs (ECS-ECSTC-661) ... 654

4.2 NCR8051408 - Provide netCDF Outputs (ECS-ECSTC-662) 655
4.3 NCR8051422 - Provide ASCII Outputs (ECS-ECSTC-664) 657

5 SMAP HDF5 SUBSETTER 658
6 GLAS HDF5 SUBSETTER..... 658
6.1 End to End GLAS HDF5 subsetting - multiple granules, multiple collections (ECS-ECSTC-573) 658
6.2 End to End GLAS HDF5 subsetting - single granule, single collection (ECS-ECSTC-575) 661
6.3 End to End GLAS HDF5 subsetting - multiple granules single collection (ECS-ECSTC-576) 665
6.4 EGI GLAS requests (ECS-ECSTC-578) 667
6.5 ESIR Test Cases for GLAS (ECS-ECSTC-580) 668
6.6 EMS reporting for GLAH requests (ECS-ECSTC-579)..... 670
6.7 Configuring Limits - requests, output file size (ECS-ECSTC-582) 673
6.8 GLAS ESDT Baseline Capability (ECS-ECSTC-585)..... 674
6.9 ESIR timeout of stitching (ECS-ECSTC-596) 675
6.10 Cleanup by GLAS tool adapters on error condition (ECS-ECSTC-586) 676
6.11 Test Cases for subset_glas (ECS-ECSTC-587) 678
6.12 Verify Spatial Search Capability for GLAH01.033 (ECS-ECSTC-588)..... 680
6.13 End to End Performance Test (ECS-ECSTC-597) 681
6.14 ESI GLAS Error Response Handling (ECS-ECSTC-604) 682
6.15 GLAS Parameter Subsetting Test (ECS-ECSTC-633)..... 683
6.16 GLAS Subsetter Unmatched Subset Constraints (ECS-ECSTC-634)..... 684
6.17 EGI Allocation of SubsetDataLayers (ECS-ECSTC-635) 685
6.18 End to End GLAS HDF5 parameter subsetting (ECS-ECSTC-639)..... 687

7 SDPS-477 RESTRICT GRANULE UTILITY 689
7.1 Add a Restriction Flag (ECS-ECSTC-3875)..... 689
7.2 Use the Restrict Granule Utility to add restrictions with Hide = Y (ECS-ECSTC-3876) 690
7.3 Use the Restrict Granule Utility to add restrictions with Hide = N (ECS-ECSTC-3878) 690
7.4 Unrestrict granules using the Restrict Granule Utility (ECS-ECSTC-3879)..... 691
7.5 Restrict Granules: End to End Test (ECS-ECSTC-3880)..... 692

List of Figures

No table of figures entries found.

List of Tables

No table of figures entries found.

1 BE_82_01

1.1 Performance

1.1.1 Performance - Aggregate (ECS-ECSTC-91)

DESCRIPTION:

	S	670	1	**[Performance – Aggregate]** Perform the following operations, ensuring that different granules are used for each: a) 10,000 granule inserts b) 25,000 granule deletions c) 50,000 QA updates d) 10,000 DPL publications e) 25,000 DPL unpublications f) 5,000 other granule updates (e.g. hide/unhide, restriction, etc) g) 50 collection inserts or updates h) 5,000 browse inserts and linkages		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		

#	Action	Expected Result	Notes
3	Ensure the collections are installed and enabled for collection and granule export. This can be confirmed on the BMGT GUI (Collection Configuration tab) or via the database as follows: SELECT granuleexportflag , collectionexportflag FROM bg_collection_configuration WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; If they are not enabled, UPDATE bg_collection_configuration SET granuleexportflag = 'Y', collectionexportflag = 'Y' WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; Allow any export requests generated by these actions to complete.		
4	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
5	Pause the dispatcher using the GUI		
6	Ensure that the automatic driver is running		
7	Setup		#comment
8	S-1 Perform the following operations, ensuring that different granules are used for each:		#comment
9	a) 10,000 granule inserts		#comment
10	Find 10,000 granules.		
11	Write the geoids of all 10,000 granules to a file. Use a psql command and direct the output into a file: >psql -h p4dbl03 -U readonly -d ecs -c "SELECT g.granuleid, g.shortname, g.versionid FROM aim.amgranule g JOIN aim.ambrowsegranulexref x ON g.granuleid=x.granuleid JOIN aim.bg_collection_configuration b ON g.collectionid = b.collectionid WHERE g.shortname ='${C1.ShortName}' AND g.VersionId = ${C1.VersionId} AND g.deleteeffectivedate IS NULL AND b.granuleexportflag = 'Y' AND b.collectionexportflag = 'Y' LIMIT 10000" -q | egrep "[0-9]+" | grep -v rows | awk '{printf "SC:%s.%03d:%d\n", $3, $5, $1;}' > geoids. (substituing the correct values for ${C1.ShortName} and ${C1.VersionId} 		
12	Using MOD10A1.086 for the manual export		#comment
13	Perform a manual export of all 25,000 granule inserts: EcBmBMGTManualStart <MODE> --metg --granulefile <granuleIdFile> 		
14	b) 25,000 granule deletions		#comment
15	Find 25,000 non existent granule IDs. To do this: select max(granuleid)		

#	Action	Expected Result	Notes

	from amgranule
	then multiply this number by 10. This will be the base id to use.
	
	run this perl script:
	
	$base = 100;
	$id = $base;
	$gransleft = 25000;
	while ($gransleft-- > 0){
	print "SC:<SNVI>:" . $id++ ."\n";
	}
	
	Where $base is set to the base ID determined above, and the correct SNVI for collection C2 is subsititued in place of "<SNVI>"
	Redirect the output of this script to a file.		

16	Using MOD11_L2.004 for deletes.		#comment
17	Perform a manual export of all 25,000 granule deletes: EcBmBMGTManualStart <MODE> -del --metg --granulefile <granuleIdFile> 		
18	c) 50,000 QA updates		#comment
19	Find 50,000 distinct granules in the same collection: select granuleId from AmGranule where ShortName = '${C3.ShortName}' and VersionId = ${C3.VersionId} limit 50000		
20	Add the granuleids to a file QA_GRANULEID_FILE		
21	Record the time as QA_START_TIME		
22	See the QAUU 609 Section 4.8.9: http://edhs1.gsfc.nasa.gov/waisdata/eed/pdf/609eed001_Rev01_File_7.pdf		#comment
23	Find one granule's metadata file: select p.path || '/' || f.archivemetfilename from ammetadatafile f join dsmdxmlpath p on f.archivepathid = p.archivepathid where f.granuleid = ${GRANULEID}		
24	Find the name of a measured parameter to update: xpath ${XML_FILE} '//MeasuredParameterContainer/ParameterName/text()'		
25	Find the measured parameter's science quality flag value: xpath ${XML_FILE} "//MeasuredParameterContainer[ParameterName='${PARAMETER_NAME}']/QAFlags/ScienceQualityFlag/text()"		
26	Select a different science quality flag value from the list of valid values in the properties file: grep VALID_SCIENCE_QA_FLAGS /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
27	Note that the above step needs to be done on p5dpl01 in PVC OPS mode.		#comment
28	Found out the that first granule (3085987951) had "Not Investigated" as flag for parameter "Ice Surface		#comment

#	Action	Expected Result	Notes
	Temperature". Will choose "Hold" as new flag.		
29	Create a QA update request file named according to the the QAUU 609: <MODE>_<Site>_QAUPDATE<description>.<YYYY><MM><DD><hh><mm><ss> E.g., OPS_BMGT82_QAUPDATE_BE_82_01_Crit_687.20130522231600		
30	Write a QA update request to update a science flag, using a GranuleUR (replace <TAB> with an actual tab character): First, add the file header: From ${site} begin QAMetadataUpdate Science GranuleUR		
31	Add the file contents: cat QA_GRANULEID_FILE | awk '{print "${shortname}<TAB>${versionid}<TAB>${shortname}.${versionid}:" $1 "<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation}" >> REQUEST_FILE		
32	Actually this works rather than the previous step: > cat QA_GRANULEID_FILE | awk '{printf "%1$s\t86\tSC:%1$s.%2$03d:%3$d\tALL\t%4$s\t%5$s\n", "<shortname>", <version id>, $1, "<quality flag>", "BmgtPerfTest_<request file>";}' > OPS_LaRC_QAUU.20130522231600		#comment
33	Add the file footer: end QAMetadataUpdate		
34	Copy the QA update request file to the QA request directory configured in the properties file: sed -n 's/QA_REQUEST_DIR *= *//p' /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
35	Run QAUU, specifying the filename (but not the path) if there are other request files in the request directory: EcAmQAUUStart ${MODE} -file ${REQUEST_FILE} -noprompt		
36	Verify that the QA Updates resulted in events in the DsMdGrEventHistory table: Select count(1) from DsMdGREventHistory where eventtime > {QA_START_TIME} and eventType = 'GRQAUPDATE' This should show 50,000 events		
37	d) 10,000 DPL publications		#comment
38	select 10,000 non public granules from collection C4 and write their IDs to a file select granuleid from AmGranule where ShortName = ${C4.ShortName} and VersionId = ${C4.VersionId} and		

#	Action	Expected Result	Notes
	coalesce(isOrderOnly,'B') in ('Y', 'H') limit 10000 Write results to a file		
39	Actually the following query should be used to find 10K non-public granules to publish: select granuleid from AmGranule where ShortName = ${C1.ShortName} and VersionId = ${VersionId} and coalesce(isOrderOnly,'B') in ('Y', 'H') and coalesce(DeleteFromArchive, 'N') = 'N' and DeleteEffectiveDate IS NULL and granuleid not in (select granuleid from DsMdGranuleRestriction) limit 10000;		#comment
40	publish the granules. EcDlPublishUtilityStart <MODE> -ecs -file <FileName>		
41			#comment
42	e) 25,000 DPL unpublications		#comment
43	select 25,000 public granules from collection C5 and write their IDs to a file select granuleid from AmGranule where ShortName = ${C5.ShortName} and VersionId = ${C5.VersionId} and coalesce(isOrderOnly,'B') ='B' limit 25000 Write results to a file		
44	unpublish the granules. EcDlUnpublishStart -mode <MODE> -file <FileName>		
45	f) 5,000 other granule updates (e.g. hide/unhide, restriction, etc)		#comment
46	Ensure that there are enough granules in the collection by ensuring that this query returns a number higher than 5000: select count(1) from AmGranule where granuleId not in (select granuleid from DsMdGranuleRestriction) and shortname = ${C6.ShortName} and versionid = ${C6.VersionId} 		
47	Find a restriction flag value to use. valid values are found by running: select * from DsMdRestrictionFlag where restrictionFlag != 255 note one of the RestrictionFlag column values and record it as ${RestrictionFlag} If no rows are returned, insert a new row with restrictionFlag value < 255 and > 0		
48	Insert the restrictions: insert into DsMdGranuleRestriction (granuleid, restrictionflag, unpublishflag, lastupdate) values select granuleid, ${RestrictionFlag}, 'N', lastupdate from AmGranule where granuleId not in (select granuleid from DsMdGranuleRestriction) and shortname = ${C6.ShortName} and versionid = ${C6.VersionId} limit 5000		

#	Action	Expected Result	Notes
49	Actually we used this query: insert into dsmdgranulerestriction (granuleid, restrictionflag, unpublishflag, lastupdate) select granuleid, 254, 'N', lastupdate from amgranule where granuleid not in (select granuleid from DsMdGranuleRestriction) and shortname = 'MYD13Q1' and versionid = 86 limit 5000;		#comment
50	g) 50 collection inserts or updates		#comment
51	Update the DsGeESDTConfiguredType table manually to simulate collection updates by executing the following queries sequentially: update DsGeESDTConfiguredType set esdtstate = 'updating' limit 50; update DsGeESDTConfiguredType set esdtstate = 'installed' where esdtstate = 'updating';		
52	h) 5,000 browse inserts and linkages		#comment
53	Identify 1 browse image to use for this test (since it is not normal to have so many granules linked to the same browse, it would be a good idea to unlink after completing this test)		
54	Add Browse linkages to the browse Xref table: insert into AmBrowseGranuleXref (granuleId, browseId) values select granuleId, ${BrowseId} from AmGranule where ShortName = '${C7.ShortName}' and VersionId = ${C7.VersionId}		
55	We used this query: insert into ambrowsegranulexref (granuleid, browseid) select granuleid, 2035880617 from amgranule where shortname = 'MISQA' and versionid = 3;		#comment
56	Start the export		#comment
57	Ensure that the BMGT Automatic driver is running or start it.		
58	Ensure that export requests have been generated in the bg_export_request table for all of the actions above		
59	Resume BMGT dispatcher using the GUI, recording the time at which the resumption occurs		
60	Verification		#comment
61	V-1 Verify that the operations in S-1 result in the export of the following:		#comment
62	a) 105,000 HTTP PUT requests containing full granule metadata		#comment
63	Inspect the proxy or fake ECHO log to verify that 105,000 Granule HTTP PUT requests were sent.		
64	b) 25,000 HTTP DELETE requests sent to URLs reflecting granule resources, but containing no request body.		#comment
65	Inspect the proxy or fake ECHO log to verify that 25,000 Granule HTTP		

#	Action	Expected Result	Notes
	DELETE requests were sent.		
66	c) 50 HTTP PUT requests containing full collection metadata.		#comment
67	Inspect the proxy or fake ECHO log to verify that 50 Collection HTTP PUT requests were sent.		
68	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation. TBD		#comment
69	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
70	Verify that all 130,050 requests are moved to the SUCCESS state within 1 hour after the recorded dispatcher resumption time.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	670	1	Verify that the operations in S-1 result in the export of the following: a) 105,000 HTTP PUT requests containing full granule metadata b) 25,000 HTTP DELETE requests sent to URLs reflecting granule resources, but containing no request body. c) 50 HTTP PUT requests containing full collection metadata.		
	V	670	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	670	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.1.2 Performance - Granule Inserts/Deletes (ECS-ECSTC-92)

DESCRIPTION:

	S	680	1	**[Performance - Granule Inserts/Deletes]** Perform the following operations, ensuring that different granules are used for each: a) 25,000 granule inserts for granules with ECS metadata which are also linked to browse granules. b) 25,000 granule deletes		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
3	Ensure the collections are installed and enabled for collection and granule export. This can be confirmed on the BMGT GUI (Collection Configuration tab) or via the database as follows: SELECT granuleexportflag , collectionexportflag FROM bg_collection_configuration WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; If they are not enabled, UPDATE bg_collection_configuration SET granuleexportflag = 'Y', collectionexportflag = 'Y' WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; 		
4	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
5	Pause the dispatcher using the GUI (System Status tab)		
6	Setup		#comment
7	S-1 Perform the following operations, ensuring that different granules are used for each: a) 25,000 granule inserts for granules with ECS metadata which are also linked to browse granules.		#comment
8	Find 25,000 granules which have browse linkages.		
9	Write the geoids of all 25,000 granules to a file.		
10	Use a psql command and direct the output into a file: >psql -h p4dbl03 -U readonly -d ecs -c "SELECT g.granuleid, g.shortname,		#comment

#	Action	Expected Result	Notes
	g.versionid FROM aim.amgranule g JOIN aim.ambrowsegranulexref x ON g.granuleid=x.granuleid JOIN aim.bg_collection_configuration b ON g.collectionid = b.collectionid WHERE g.shortname NOT LIKE 'ECS%' AND g.deleteeffectivedate IS NULL AND b.granuleexportflag = 'Y' AND b.collectionexportflag = 'Y' LIMIT 25" -q | egrep "[0-9]+" | grep -v rows | awk '{printf "SC:%s.%03d:%d\n", $3, $5, $1;}' > geoids. Alternatively, a script to find suitable granules is in /tools/common/test/new_scripts/FindDataForBMGT_performance.sh. The result is placed in /usr/ecs/OPS/CUSTOM/utilities with a filename of the form BmgtGeoid_perf_<date/time>.txt.		
11	Perform a manual export of all 25,000 granule inserts: EcBmBMGTManualStart <MODE> --metg --granulefile <granuleIdFile> 		
12	S-1 Perform the following operations, ensuring that different granules are used for each: b) 25,000 granule deletes		#comment
13	Generate 25,000 non existent granule IDs. From a database command prompt, use: SELECT MAX(granuleid) FROM amgranule; to find the largest granule ID, then multiply this number by 10. This will be the base id to use. From a command prompt: > seq 1 25000 | awk '{printf "SC:MOD11_L2.004:%d\n", $1 + <base id>;}' > delete_geoids 		
14	I used MOD11_L2.005. File is in /tools/common/test/new_scripts/BMGTperf/nonexist.out		#comment
15	Perform a manual export of all 25,000 granule deletes: EcBmBMGTManualStart <MODE> -del --metg --granulefile <granuleIdFile>		
16	Resume BMGT dispatcher using the GUI, recording the time at which the resumption occurs		
17	Verification		#comment
18	V-1 Verify that the operations in S-1 result in the export of the following: a) 25,000 HTTP PUT requests containing full granule metadata		#comment
19	Inspect the proxy or fake ECHO log to verify that 25,000 HTTP PUT requests were sent. > cd /L0_buffer/OPS/ProductOutput/FakedEcho82/logs > grep PUT access.log | grep '2013-07-02T15' | wc -l Note: You will need to modify the grep commands to match the data or timestamps used.		
20	V-1 Verify that the operations in S-1 result in the export of the		#comment

#	Action	Expected Result	Notes
	following: b) 25,000 HTTP DELETE requests sent to URLs reflecting granule resources, but containing no request body.		
21	Inspect the proxy or fake ECHO log to verify that 25,000 HTTP DELETE requests were sent. > cd /L0_buffer/OPS/ProductOutput/FakedEcho82/logs > grep DELETE access.log | grep MOD11_L2\.004 | wc -l Note: You will need to modify the grep commands to match the data or timestamps used.		
22	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		#comment
23	TBD		
24	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
25	Verify that all 50,000 requests are moved to the SUCCESS state within 1 hour after the recorded dispatcher resumption time.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	680	1	Verify that the operations in S-1 result in the export of the following: a) 25,000 HTTP PUT requests containing full granule metadata b) 25,000 HTTP DELETE requests sent to URLs reflecting granule resources, but containing no request body.		
	V	680	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	680	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.1.3 Performance - QA Updates (ECS-ECSTC-93)

DESCRIPTION:

	S	690	1	**[Performance - QA Updates]** Perform the following operations, ensuring that different granules are used for each: a) 100,000 QA Updates		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Setup		#comment
3	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
4	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
5	Pause the dispatcher using the GUI		
6	Stop the automatic driver: EcBmBMGTAutoStop <MODE>		
7	S-1 Perform the following operations, ensuring that different granules are used for each: a) 100,000 QA Updates		#comment
8	Find 100,000 distinct granules in the same collection		
9	Add the granuleids to a file QA_GRANULEID_FILE From a command prompt: > psql -h p4dbl03 -U readonly -d ecs -c "SELECT GranuleId FROM aim.amgranule WHERE VersionId = <version id> AND DeleteFromArchive = 'N' AND DeleteEffectiveDate IS NULL AND ShortName = '<shortname>' ORDER BY GranuleId desc LIMIT 100000" -t -q | grep -E -o "[0-9]+" > QA_GRANULEID_FILE Where <shortname> and <version id> are the collection for which we are selecting granules. e.g. MOD10A1 and 86		
10	Script to find is in /tools/common/test/new_scripts/BMGTperf/GenerateQAUU_performance.sh. I used MOD10A1.086 because it is a collection in OPS mode with > 400K granules. Result file is in /usr/ecs/OPS/CUSTOM/data/DSS/QAUU/OPS_LaRC_QAUU.20130701194		#comment

#	Action	Expected Result	Notes
	018 on p5dpl01. Flag was set to "Suspect".		
11	Record the time as QA_START_TIME		
12	See the QAUU 609 Section 4.8.9: http://edhs1.gsfc.nasa.gov/waisdata/eed/pdf/609eed001_Rev01_File_7.pdf		#comment
13	Select the metadata file for one granule: Using one granule ID from the QA_GRANULEID_FILE, from a database command prompt: > SELECT p.path || '/' || f.archivemetfilename FROM ammetadatafile f JOIN dsmdxmlpath p ON f.archivepathid = p.archivepathid WHERE f.granuleid = <granule ID> Note the resulting file name.		
14	Find the name of a measured parameter to update: > xpath <file name> '//MeasuredParameterContainer/ParameterName/text()' where <file name> is the previously noted xml file name. If more than one parameter name ('NODE' in xml terms) is returned, pick one (e.g. 'Day_Tile_Snow_Cover').		
15	Find the measured parameter's science quality flag value: > xpath <file name> "//MeasuredParameterContainer[ParameterName='<parameter name>']/QAFlags/ScienceQualityFlag/text()" where <parameter name> is the previously selected parameter name. Note the current flag value (e.g. 'Suspect')		
16	Select a different science quality flag value from the list of valid values in the properties file: > grep VALID_SCIENCE_QA_FLAGS /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties e.g. 'Passed'		
17	Create a QA update request file named according to the the QAUU 609: <MODE>_<Site>_QAUU.<YYYY><MM><DD><hh><mm><ss> e.g. > touch OPS_LaRC_QAUU.20130522231600		
18	Write a QA update request to update a science flag, using a GranuleUR (replace <TAB> with an actual tab character): First, add the file header: > echo "From GSFC" >> <request file> > echo "begin QAMetadataUpdate Science GranuleUR" >> <request file> where <request file> is the previously selected name for the QA update request file.		

#	Action	Expected Result	Notes
19	Add the file contents: > cat QA_GRANULEID_FILE | awk '{print "${shortname}<TAB>${versionid}<TAB>${shortname}.${versionid}:" $1 "<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation}" >> OPS_LaRC_QAUU.20130522231600 > cat QA_GRANULEID_FILE | awk '{printf "%1$s\t86\tSC:%1$s.%2$03d:%3$d\tALL\t%4$s\t%5$s\n", "<shortname>", <version id>, $1, "<quality flag>", "BmgtPerfTest_<request file>";}' > OPS_LaRC_QAUU.20130522231600 where <quality flag> is the newly selected quality flag value. 		
20	Add the file footer: > echo "end QAMetadataUpdate" >> <request file>		
21	Copy the QA update request file to the QA request directory configured in the properties file: To find the request directory location, from a command prompt: > sed -n 's/QA_REQUEST_DIR *= *//p' /usr/ecs/<mode>/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
22	Run QAUU, specifying the filename (but not the path) if there are other request files in the request directory: EcDsAmQAUUStart <mode> -file <request file> -noprompt -noExitOnError -skipRecovery		
23	Verify that the QA Updates resulted in events in the DsMdGrEventHistory table: From a database command prompt: SELECT count(1) FROM DsMdGREventHistory WHERE eventtime > {QA_START_TIME} AND eventType = 'GRQAUPDATE' This should show 100,000 events.		
24	Restart the Automatic driver: > EcBmBMGTAutoStart <mode> verifying that BMGT.AutoDriver.PollingFrequency is set greater than 100,000		
25	Verify that all the events have been added to the BMGT requests queue: select count(1) from bg_export_request where enqueuetime > {QA_START_TIME} and exportQueue = 'EVENT' and granuleId is not null This should return 100,000		
26	record the time as BMGT_START_TIME		
27	Resume the dispatcher via the GUI		
28	Verification		#comment

#	Action	Expected Result	Notes
29	V-1 Verify that the operations in S-1 result in the export of the following: a) 100,000 HTTP PUT requests containing full granule metadata		#comment
30	Inspect the proxy or fake ECHO log to verify that 100,000 HTTP PUT requests were sent		
31	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		#comment
32	TBD		
33	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
34	Verify that all 100,000 requests are moved to the SUCCESS state within 1 hour after the recorded dispatcher resumption time. select count(1) from bg_export_request where enqueuetime > {QA_START_TIME} and exportQueue = 'EVENT' and granuleId is not null and status = 'SUCCESS' and completionTime < ({BMGT_START_TIME} + interval '1 hour') should return 100,000		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	690	1	Verify that the operations in S-1 result in the export of the following: a) 100,000 HTTP PUT requests containing full granule metadata		
	V	690	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	690	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.1.4 Performance - Datapool Inserts/Updates/Deletes (ECS-ECSTC-94)

DESCRIPTION:

	S	700	1	**[Performance - Datapool Inserts/Updates/Deletes]** Perform the following operations, ensuring that different granules are used for each: a) 40,000 Datapool granule publications b) 40,000 Datapool granule unpublications c) 20,000 Datapool granule updates - perform this by moving one or more collections with a total of 20,000 granules.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
3	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
4	Stop the automatic driver EcBmBMGTAutoStop <MODE>		
5			#comment
6	Setup		#comment
7	S-1 Perform the following operations, ensuring that different granules are used for each: a) 40,000 Datapool granule publications		#comment
8	40,000 non public granules from collection C1 and write their IDs to a file select granuleid from AmGranule where ShortName = ${C1.ShortName} and VersionId = ${VersionId} and coalesce(isOrderOnly,'B') in ('Y', 'H') limit 40000 Write results to a file		
9	Actually the following query should be used to find 40K non-public granules to publish: select granuleid from AmGranule where ShortName = ${C1.ShortName} and VersionId = ${VersionId} and coalesce(isOrderOnly,'B') in ('Y', 'H') and coalesce(DeleteFromArchive, 'N') = 'N' and DeleteEffectiveDate IS NULL and granuleid not in (select granuleid from		#comment

#	Action	Expected Result	Notes
	DsMdGranuleRestriction) limit 40000;		
10	The below step didn't necessarily publish all granules we found by the above query because some of the candidates to be published already had a later version of the same data in the data pool caused by a duplicate granule ingested.		#comment
11	publish the granules. EcDlPublishUtilityStart <MODE> -ecs -file <FileName>		
12	S-1 Perform the following operations, ensuring that different granules are used for each: b) 40,000 Datapool granule unpublications		#comment
13	select 40,000 public granules from collection C2 and write their IDs to a file select granuleid from AmGranule where ShortName = ${C1.ShortName} and VersionId = ${VersionId} and coalesce(isOrderOnly,'B') ='B' limit 40000 Write results to a file		
14	unpublish the granules. EcDlUnPublishUtilityStart -mode <MODE> -file <FileName>		
15	S-1 Perform the following operations, ensuring that different granules are used for each: c) 20,000 Datapool granule updates - perform this by moving one or more collections with a total of 20,000 granules.		#comment
16	Find a collection C3 with 20,000 public granules: Ensure: select count(1) from AmGranule where granuleId not in (select granuleid from DsMdGranuleRestriction) and shortname = ${C3.ShortName} and versionid = ${C3.VersionId}		
17	Move collection C3 to a different filesystem (e.g., from ${FS_OLD} to ${FS_NEW}): EcDlMoveCollection.pl ${MODE} -shortname ${SHORTNAME} -versionid ${VERSIONID} -sourcefs ${FS_OLD} -targetfs ${FS_NEW} -verbose		
18	We ended up using collection MYD13AQ1.086 and moved this collection of 29,021 granules from FS4 to FS2. This took		#comment
19	Note the current Time, $StartTime		
20	Start the auto driver EcBmBMGTAutoStart <MODE> (and ensure that the dispatcher is running by inspecting the GUI or log)		
21	Verification		#comment
22	V-1 Verify that the operations in S-1 result in the export of the following: a) 100,000 HTTP PUT requests containing full granule metadata		#comment
23	Inspect the proxy or fake ECHO log to verify that 100,000 HTTP PUT		

#	Action	Expected Result	Notes
	requests were sent		
24	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		#comment
25	TBD		
26	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
27	Verify that all exports are complete withing 1 hour of $StartTime		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	700	1	Verify that the operations in S-1 result in the export of the following: a) 100,000 HTTP PUT requests containing full granule metadata		
	V	700	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	700	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.1.5 Performance - Granule Updates (ECS-ECSTC-95)

DESCRIPTION:

	S	710	1	**[Performance - Granule Updates]** Perform the following operations, ensuring that different granules are used for each: a) Use the XRU to replace the metadata for 5,000 granules. b) Restrict 5,000 granules. c) Unrestrict 5,000 granules.		

				d) Hide 5,000 granules. e) Unhide 5,000 granules.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
3	Set BMGT.AutoDriver.MaxEvents to 60000 (instead of 300000) on BMGT GUI Configuration tab.		#comment
4	Ensure the collections are installed and enabled for collection and granule export. This can be confirmed on the BMGT GUI (Collection Configuration tab) or via the database as follows: SELECT granuleexportflag , collectionexportflag FROM bg_collection_configuration WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; If they are not enabled, UPDATE bg_collection_configuration SET granuleexportflag = 'Y', collectionexportflag = 'Y' WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; Allow any export requests generated by these actions to complete. 		
5	Restrict 5000 granules:		#comment
6	Ensure that there are enough granules in the collection by ensuring that this query returns a number higher than 5000: select count(1) from AmGranule where granuleId not in (select granuleid from DsMdGranuleRestriction) and shortname = ${C3.ShortName} and versionid = ${C3.VersionId}		
7	Find a restriction flag value to use. valid values are found by running: select * from DsMdRestrictionFlag where restrictionFlag != 255 note one of the RestrictionFlag column values and record it as ${RestrictionFlag} If no rows are returned, insert a new row with restrictionFlag value < 255 and > 0		
8	insert into DsMdGranuleRestriction (granuleid, restrictionflag, unpublishflag, lastupdate) values select granuleid, ${RestrictionFlag}, 'N', lastupdate) from AmGranule where granuleId not in (select granuleid		

#	Action	Expected Result	Notes
	from DsMdGranuleRestriction) and shortname = ${C3.ShortName} and versionid = ${C3.VersionId} limit 5000 		
9	Allow the automatic driver to run and the requests generated for these restrictions to complete		
10	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
11	Pause the dispatcher using the GUI		
12	Setup		#comment
13	S-1 Perform the following operations, ensuring that different granules are used for each: a) Use the XRU to replace the metadata for 5,000 granules.		#comment
14	Get metadata file paths for 5000 granules: select 'cp ' || p.path || '/' || f.archivemetfilename || ' ${stagingDirPath}' from AmGranule g join ammetadatafile f on f.granuleid = g.granuleid join DsMdXmlPath p on f.archivepathid = p.archivepathid where g.ShortName= {$C1.ShortName} and g.VersionId = ${C1.VersionId} limit 5000 Where ${StagingDirPath} is a path where metadata fiels will be copied and modified. direct the results fo this query to a ${ScriptFile}		
15	Actually to find the appropriate collection to modify XML, we used this query: select a.shortname, a.versionid, count(1) from amgranule a, bg_collection_configuration b where a.granuleId not in (select granuleid from DsMdGranuleRestriction) and a.shortname = b.shortname and a.versionid = b.versionid and b.granuleexportflag = 'Y' and b.collectionexportflag = 'Y' and a.deleteeffectivedate IS NULL and deletefromarchive = 'N' group by a.shortname, a.versionid order by count(1); This will avoid finding things that are marked for deletion.		#comment
16	Rewrote the above step as follows: o cpit.out; select 'cp ' || p.path || '/' || f.archivemetfilename || ' ${stagingDirPath}' from AmGranule g join ammetadatafile f on f.granuleid = g.granuleid join DsMdXmlPath p on f.archivepathid = p.archivepathid where g.ShortName= {$C1.ShortName} and g.VersionId = ${C1.VersionId}; Where ${StagingDirPath} is a path where metadata files will be copied and modified. The results will be in script file cpit.out. Edit cpit.out and remove the first two line and last two lines of the file.		#comment
17	Copy metadata files to the statging location: mkdir ${StagingDirPath} chmod 777 ${ScriptFile} ./${ScriptFile}		
18	Modify the metadata: In the ${StagingDirPath} directory, run sed		

#	Action	Expected Result	Notes
	-ie '/LocalVersionID/{s/>\(.*\)</>\1-MODIFIED</g}' *.xml		
19	Actually prior to the above step, you need to cd ${StagingDirPath}. Then the command to modify an xml file in place is to run: sed -i '/LocalVersionID/{s/>\(.*\)</>\1-MODIFIED</g}' *.xml You also need to ensure that the metadata you are modifying does indeed contain "LocalVersionID". This test was tried with AE_Land.002 and that did not have LocalVersionID. We used MYD10A1.086 and MYD11_L2.086.		#comment
20	Replace the metadata: on x4dpl01 run EcDsAmXruStart <MODE> -xmldir ${StagingDirPath}		
21	The above step should say "Replace the metadata: on x5dpl01 run EcDsAmXruStart <MODE> -xmldir ${StagingDirPath}		#comment
22	S-1 Perform the following operations, ensuring that different granules are used for each: b) Restrict 5,000 granules.		#comment
23	Ensure that there are enough granules in the collection by ensuring that this query returns a number higher than 5000: select count(1) from AmGranule where granuleId not in (select granuleid from DsMdGranuleRestriction) and shortname = ${C2.ShortName} and versionid = ${C2.VersionId}		
24	Find a restriction flag value to use. valid values are found by running: select * from DsMdRestrictionFlag where restrictionFlag != 255 note one of the RestrictionFlag column values and record it as ${RestrictionFlag} If no rows are rertuend, insert a new row with restrictionFlag value < 255 and > 0		
25	insert into DsMdGranuleRestriction (granuleid, restrictionflag, unpublishflag, lastupdate) values select granuleid, ${RestrictionFlag}, 'N', lastupdate) from AmGranule where granuleId not in (select granuleid from DsMdGranuleRestriction) and shortname = ${C2.ShortName} and versionid = ${C2.VersionId} limit 5000		
26	S-1 Perform the following operations, ensuring that different granules are used for each: c) Unrestrict 5,000 granules.		#comment
27	delete DsMdGranuleRestriction r join AmGranule g on r.granuleid = g.granuleid where g.shortname = ${C3.ShortName} and g.versionid = ${C3.VersionId}		
28	Actually this works: delete from DsMdGranuleRestriction where granuleid in (select granuleid from amgranule where shortname = 'AE_Land' and versionid = 2);		#comment

#	Action	Expected Result	Notes
29	S-1 Perform the following operations, ensuring that different granules are used for each: d) Hide 5,000 granules.		#comment
30	Ensure that there are enough granules in the collection by ensuring that this query returns a number higher than 5000: select count(1) from AmGranule where DeleteFromArchive != 'H' and shortname = ${C4.ShortName} and versionid = ${C4.VersionId}		
31	update AmGranule set DeleteFromArchive = 'H' where g.shortname = ${C4.ShortName} and g.versionid = ${C4.VersionId} limit 5000		
32	S-1 Perform the following operations, ensuring that different granules are used for each: e) Unhide 5,000 granules.		#comment
33	Ensure that there are enough granules in the collection by ensuring that this query returns a number higher than 5000: select count(1) from AmGranule where DeleteFromArchive = 'H' and shortname = ${C5.ShortName} and versionid = ${C5.VersionId}		
34	update AmGranule set DeleteFromArchive = 'H' where g.shortname = ${C5.ShortName} and g.versionid = ${C5.VersionId} limit 5000		
35	Actually the above query should be: update AmGranule set DeleteFromArchive = 'N' where granuleid in (select granuleid from AmGranule where shortname = ${C5.ShortName} and g.versionid = ${C5.VersionId} limit 5000);		#comment
36	Resume BMGT dispatcher using the GUI, recording the time at which the resumption occurs		
37	Verification		#comment
38	V-1 Verify that the operations in S-1 result in the export of the following: a) 25,000 HTTP PUT requests containing full granule metadata		#comment
39	Inspect the proxy or fake ECHO log to verify that 25,000 HTTP PUT requests were sent		
40	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		#comment
41	TBD		
42	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
43	*Verify that all 25,000 requests are moved to the SUCCESS state within 1 hour after the recorded dispatcher resumption time.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	710	1	Verify that the operations in S-1 result in the export of the following: a) 25,000 HTTP PUT requests containing full granule metadata		
	V	710	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	710	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.1.6 Performance - Collection Inserts/Updates (ECS-ECSTC-96)

DESCRIPTION:

	S	720	1	**[Performance - Collection Inserts/Updates]** Perform the following operations, ensuring that different collections are used for each: a) Insert or update 900 collections. If 900 collections do not exist, then insert or update all of the available collections.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
3	Ensure the collections are installed and enabled for collection and granule export. This can be confirmed on the BMGT GUI (Collection Configuration tab) or via the database as follows: SELECT granuleexportflag ,		

#	Action	Expected Result	Notes
	collectionexportflag FROM bg_collection_configuration WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; If they are not enabled, UPDATE bg_collection_configuration SET granuleexportflag = 'Y', collectionexportflag = 'Y' WHERE shortname = <SHORTNAME> AND versionid = <VERSIONID>; Allow any export requests generated by these actions to complete.		
4	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
5	Pause the dispatcher using the GUI		
6	Setup		#comment
7	S-1 Perform the following operations, ensuring that different collections are used for each: a) Insert or update 900 collections. If 900 collections do not exist, then insert or update all of the available collections.		#comment
8	Update the DsGeESDTConfiguredType table manually to simulate collection updates by executing the following queries sequentially: update DsGeESDTConfiguredType set esdtstate = 'updating'; update DsGeESDTConfiguredType set esdtstate = 'installed';		
9	Actually the above queries should be: update DsGeESDTConfiguredType set esdtstate = 'updating' where configuredName in (select configuredName from DsGeESDTConfiguredType limit 900); & update DsGeESDTConfiguredType set esdtstate = 'installed' where esdtstate = 'updating';		#comment
10	Verify the dsmdgreventhistory table has a CLUPDATE event for each of the updated collections: select count(1) from dsmdgreventhistory where eventType = 'CLUPDATE' and eventTime > '${StartTime};		
11	Resume BMGT dispatcher using the GUI, recording the time at which the resumption occurs.		
12	Record the time at which all the requests were completed.		
13	Verification		#comment
14	V-1 Verify that the operations in S-1 result in the export of the following: a) 900 HTTP PUT requests containing full collection metadata. If there were not 900 collections to export, verify that the number of HTTP PUT requests matches the number of collections which were inserted/updated.		#comment

#	Action	Expected Result	Notes
15	Inspect the proxy or fake ECHO log to verify that one HTTP PUT requests was sent per collection updated.		
16	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		#comment
17	TBD		
18	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
19	Verify that all requests are moved to the SUCCESS state within 1 hour after the recorded dispatcher resumption time.		
20	V-4 Verify that on average, the collection metadata was generated and exported at a rate greater than or equal to 7 collections per second. Note: this equates to a rate of 25,000 per hour.		#comment
21	select count(1) as "Total complete requests", count(1) / (date_part('epoch', '${STOP_TIME}) - date_part('epoch', TIMESTAMP '${START_TIME}')) as "Average requests per second" from bg_export_request where enqueueTime > '${START_TIME}' and itemType = 'CL' and status = 'SUCCESS'; Ensure that the 'average requests per second' is greater than 7		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	720	1	Verify that the operations in S-1 result in the export of the following: a) 900 HTTP PUT requests containing full collection metadata. If there were not 900 collections to export, verify that the number of HTTP PUT requests matches the number of collections which were inserted/updated.		
	V	720	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		

	V	720	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		
	V	720	4	Verify that on average, the collection metadata was generated and exported at a rate greater than or equal to 7 collections per second. Note: this equates to a rate of 25,000 per hour.		

1.1.7 Performance - MISBR (ECS-ECSTC-97)

DESCRIPTION:

	S	730	1	**[Performance – MISBR]** Perform the following operations, ensuring that different granules are used for each: a) Insert 4,300 MISR Level 1 granules, which are related to MISBR granules. b) Insert 4,300 MISR Level 2 granules, which are related to MISBR granules. c) Insert 860 MISBR granules which are related to MISR Level 1 and 2 science granules. d) Delete 4,300 MISR Level 1 granules, which are related to MISBR granules. e) Delete 4,300 MISR Level 2 granules, which are related to MISBR granules. f) Delete 860 MISBR granules which are related to MISR Level 1 and 2 science granules.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure the test MISBR, MISR Level 1, MISR Level 2, and MIB2GEOP collections are installed.		
3	Ensure the test collections are configured to be public on ingest.		
4	Ensure test collections are configured for collection and granule export.		
5	Ensure AIM has no MISR granules that match the test MISBR granules. Use the xpath utility to get the Path Number, Orbit Number, Product		

#	Action	Expected Result	Notes
	Version, and Camera ID from the browse metadata file: xpath "//PSAs/PSA[PSAName='SP_AM_PATH_NO']/PSAValue/text()" misbr.xml xpath '//OrbitCalculatedSpatialDomain/OrbitCalculatedSpatialDomainContainer/OrbitNumber/text()' misbr.xml xpath "//PSAs/PSA[PSAName='SP_AM_MISR_ProductVersion']/PSAValue/text()" misbr.xml xpath '//Platform/Instrument/Sensor/SensorShortName/text()' misbr.xml Query AIM for matching MISR Level 1 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = '1' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and a.productversion = $PRODUCTVERSION and c.cameraid = $CAMERAID Query AIM for matching MISR Level 2 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = '2' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and c.cameraid = 'AN'		
6	Ensure AIM has no MISBR granules that match the test MISR granules. Use the xpath utility to get the Path Number, Orbit Number, Product Version, and Camera ID from the granule metadata files: xpath "//PSAs/PSA[PSAName='SP_AM_PATH_NO']/PSAValue/text()" granule.xml xpath '//OrbitCalculatedSpatialDomain/OrbitCalculatedSpatialDomainContainer/OrbitNumber/text()' granule.xml xpath "//PSAs/PSA[PSAName='SP_AM_MISR_ProductVersion']/PSAValue/text()" xpath '//Platform/Instrument/Sensor/SensorShortName/text()' granule.xml Query AIM for MISBR matching the MISR Level 1 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and		

#	Action	Expected Result	Notes
	g.versionid = p.versionid) where p.type = 'B' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and a.productversion = $PRODUCTVERSION and c.cameraid = $CAMERAID Query AIM for MISBR matching the MISR Level 2 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = 'B' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and c.cameraid = 'AN'		
7	Ensure EcDlInsertUtility.properties has this: MISR_SPECIAL_PROCESSING=Y Bounce DPAD after changing the value: ./EcDlActionDriverStart $MODE		
8	Ensure that a TCP proxy or mock ECHO is capturing BMGT traffic.		
9	Ensure the BMGT dispatcher and auto driver are running: ./EcBmBMGTAppStart $MODE	ps auxww | sed -n '/OPS\/CUSTOM/{;s/.* -D\(Bmgt[^]*\).*/\1/p;}' BmgtComponent=EcBmDispatcher BmgtComponent=EcBmAuto BmgtComponent=EcBmMonitor	
10	Actually we make sure that the Dispatcher is down at this point. Do not bring it up yet.		#comment
11	Setup		#comment
12	S-1 Perform the following operations, ensuring that different granules are used for each:		#comment
13	a) Insert 4,300 MISR Level 1 granules, which are related to MISBR granules.		#comment
14	Perform an ingest on 4,300 MISR L1 granules and their related MISBR		
15	b) Insert 4,300 MISR Level 2 granules, which are related to MISBR granules.		#comment
16	Perform an ingest on 4,300 MISR L2 granules and their related MISBR		
17	c) Insert 860 MISBR granules which are related to MISR Level 1 and 2 science granules.		#comment
18	Ensure that the above MISR science ingests contain at least 860 MISBR granules as well.		

#	Action	Expected Result	Notes
19	d) Delete 4,300 MISR Level 1 granules, which are related to MISBR granules.		#comment
20	Perform a logical or physical delete on 4,300 MISR L1 granules and their related MISBR		
21	e) Delete 4,300 MISR Level 2 granules, which are related to MISBR granules.		#comment
22	Perform a logical or physical delete on 4,300 MISR L2 granules and their related MISBR		
23	f) Delete 860 MISBR granules which are related to MISR Level 1 and 2 science granules.		#comment
24	Ensure that the above MISR science deletes contain at least 860 MISBR granules as well.		
25	Bring up AutoDriver and Dispatcher		#comment
26	Verification		#comment
27	V-1 Verify that the operations in S-1 result in the export of the following: a) 4,300 HTTP PUT requests containing full granule metadata for MISR Level 1 granules. b) 4,300 HTTP PUT requests containing full granule metadata for MISR Level 2 granules. c) 860 HTTP PUT requests containing full granule metadata for MISBR granules. a. In addition, 860 HTTP PUT requests containing the full granule metadata for the MISR science granules related to the inserted MISBR granules. d) 4,300 HTTP DELETE requests containing no body, but pointing to a URL which relates to a MISR level 1 granule. e) 4,300 HTTP DELETE requests containing no body, but pointing to a URL which relates to a MISR level 2 granule. f) 860 HTTP DELETE requests containing no body, but pointing to a URL which relates to a MISBR granule. a. In addition, 860 HTTP PUT requests containing the full granule metadata for the MISR science granules related to the deleted MISBR granules. For a total of 11,180 HTTP PUT requests, and 9.460 HTTP DELETE requests.		
28	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
29	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	730	1	Verify that the operations in S-1 result in the export of the following: a) 4,300 HTTP PUT requests containing full granule metadata for MISR Level 1 granules. b) 4,300 HTTP PUT requests containing full granule metadata for MISR Level 2 granules. c) 860 HTTP PUT requests containing full granule metadata for MISBR granules. a. In addition, 860 HTTP PUT requests containing the full granule metadata for the MISR science granules related to the inserted MISBR granules. d) 4,300 HTTP DELETE requests containing no body, but pointing to a URL which relates to a MISR level 1 granule. e) 4,300 HTTP DELETE requests containing no body, but pointing to a URL which relates to a MISR level 2 granule. f) 860 HTTP DELETE requests containing no body, but pointing to a URL which relates to a MISBR granule. a. In addition, 860 HTTP PUT requests containing the full granule metadata for the MISR science granules related to the deleted MISBR granules. For a total of 11,180 HTTP PUT requests, and 9.460 HTTP DELETE requests.		
	V	730	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	730	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.1.8 Performance - Long Form Verification (ECS-ECSTC-98)

DESCRIPTION:

	S	740	1	**[Performance - Long Form Verification]** Perform a long form verification which contains 100,000 granules.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Configure BMGT to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
3	Ensure that the BMGT system is quiesced, with no requests in the PENDING state.		
4	Pause the BMGT dispatcher via the GUI		
5	Setup		#comment
6	S-1 Perform a long form verification which contains 100,000 granules.		#comment
7	Identify 100,000 granules whose collections have previously been enabled for granule export. Write the granule ids to a file.		
8	Perform a long form verification of these granules: EcBmBMGTManualStart <MODE> -metg -long -gf <File>		
9	Note the current time, $StartTime		
10	Resume the BMGT dispatcher via the GUI		
11	Verification		#comment
12	V-1 Verify that the operations in S-1 result in the export of 100,000 HTTP PUT requests containing full granule metadata.		#comment
13	Inspect the proxy or fake ECHO log to verify that 100,000 HTTP PUT requests were sent		
14	V-2 Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		#comment
15	TBD		
16	V-3 Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		#comment
17	Verify that all exports are complete withing 1 hour of $StartTime		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	740	1	Verify that the operations in S-1 result in the export of 100,000 HTTP PUT requests containing full granule metadata.		
	V	740	2	Inspect the database metrics to ensure that during the process of this test, the database suffers no more than a 10% performance degradation.		
	V	740	3	Verify that all exports resulting from S-1 complete within 1 hour of the start of the test.		

1.2 Nominal Collection Export[S-1]: Manual Export (ECS-ECSTC-1)

DESCRIPTION:

	S	10	1	**[Nominal Collection Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Request the manual export of collection metadata for one of these collections.		
	S	10	2	Find collections which have ECS Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Update an existing collection. d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/current32/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. Should include 'aim_<MODE>'	
4	Ensure test collections C1, C2 exist under /sotestdata/DROP_802/BE_82_01/Criteria/010/010_1.		

#	Action	Expected Result	Notes
5	Ensure that the collections C1 and C2 have been installed in the mode, e.g., the DPL Ingest GUI shows them as configured datatypes.		
6	Ensure a local copy of the ECHO 10 schema files is available for validation (from https://api.echo.nasa.gov/ingest/schema).		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 [Nominal Collection Export] Find two collections with ECS metadata and which share the same short name, but have different version IDs. Request the manual export of collection metadata for one of these collections.		#comment
10	Ensure that the two collections C1 and C2 have the same short name but different version IDs.		
11	Ensure collections C1 and C2 enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <C1_SHORTNAME> and versionid in (<C1_VERSIONID>, <C2_VERSIONID>);		
12	Request manual export of collection metadata for collection C1: EcBmBMGTManualStart <MODE> --metc --collections <C1_SHORTNAME>.<C1_VERSIONID>		
13	V-1 Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but no other collections sharing a short name but with different version IDs).		#comment
14	Verify the bg_export_error table shows no errors referring to collection C1 on or after the time of the manual export.		
15	Verify that the TCP proxy log reports a single HTTP PUT request for collection C1.		
16	Verify that the HTTP PUT request contains collection C1's full collection metadata.		
17	Verify the TCP proxy log reports no PUTs for collection C2.		
18	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		#comment
19	Verify collection C1's exported metadata validates against the ECHO 10 collection schema. (https://api.echo.nasa.gov/ingest/schema/Collection.xsd). xmllint		

#	Action	Expected Result	Notes
	--noout --schema Collection.xsd collection.xml		
20	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		#comment
21	Use an xpath utility to verify the exported metadata has the following elements.		#comment
22	a) Visible = true xpath /Collection/Visible C1.xml	<Visible>true</Visible>	
23	b) Orderable = false xpath /Collection/Orderable C1.xml	<Orderable>false</Orderable >	
24	c) InsertTime = The insert time of the collection recorded in the AIM database. xpath /Collection/InsertTime C1.xml	<InsertTime><YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string returned is equal to what is returned from the query to the aim_<MODE> schema in the ecs database: select inserttime from amcollection where shortname = <C1_ShortName> and versionid = <C1_VersionId>;	
25	d) LastUpdate = The last update time of the collection recorded in the AIM database. xpath /Collection/LastUpdate C1.xml	<LastUpdate><YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query: select lastupdate from amcollection where shortname = <C1_ShortName> and versionid = <C1_VersionId>;	
26	V-6 Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		#comment
27	Verify that collection C1's exported metadata contains version numbers with no leading zeroes: xpath /Collection/VersionId C1.xml	<VersionId>${VERSIONID}</VersionId>	

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
10-S1			2 Collections with same shortname, different versionids (C1, C2)	MOD44W.005 MOD44W.006		2 collections	/sotestdata/DROP_802/BE_82_01/Criteria/010/010_1	

EXPECTED RESULTS:

	V	10	1	Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but no other collections sharing a short name but with different version IDs).		
	V	10	2	Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		
	V	10	3	Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but containing no collection metadata in the request body.		
	V	10	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		
	V	10	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		
	V	10	6	Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		

1.3 ECS Collection Additional Metadata[S-01]: ECS CoordinateSystem (ECS-ECSTC-2)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-1 Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		#comment
8	Update the BMGT collection configuration table, setting the value of the S-1 collection's CoordinateSystem to an invalid value. E.g., update bg_collection_configuration set CollectionCoordinateSystem='POLAR' where ShortName='<SHORT_NAME>' and VersionID=<VERSION_ID>		
9	Request a manual export of the S-1 collection: EcBmBMGTManualStart --mode $MODE --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-1 Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		#comment
12	Verify the manual export fails.		

#	Action	Expected Result	Notes
13	Verify the log file records the export failure.		
14	Verify the log file indicates the invalid CoordinateSystem value caused the export failure.		
15	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
16	No collection metadata should have been exported.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
20 S-1				A collection for which a CoordinateSystem can be configured.				
20 S-2				A collection for which a GranuleSpatialRepresentation can be configured.				
20 S-3				An ISO 19115 collection.				
20 S-4				An ISO 19115 collection.				
20 S-5				For each of the valid backtrack types one collection for which that type can be configured.				
20 S-6				A collection for which the backtrack type can be left unconfigured.				
20 S-7				For each of the valid TwoDCoordinateSystem types, a collection which may be configured for that type.				
20 S-8				A collection for which the TwoDCoordinateSystem can be left unconfigured.				
20 S-9				A collection whose native metadata contains Product Specific Attributes (PSAs).				
20 S-				One collection with a DIF ID defined and one				

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
10, S-11				without.				
20 S-12				A collection for which a CoordinateSystem can be configured.				
20 S-13				One collection for each of the spatial search types, GPolygon, Orbit, Point, Rectangle, NotSupported.				

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element		

				which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.4 Nominal Collection Export[S-2a]: Automatic Export: Insert Collection (ECS-ECSTC-3)

DESCRIPTION:

	S	10	1	**[Nominal Collection Export]** Find two collections with ECS metadata and which share the same short name, but		

				have different version IDs. Request the manual export of collection metadata for one of these collections.		
	S	10	2	Find collections which have ECS Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Update an existing collection. d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/current32/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. Should include 'aim_<MODE>'	
4	Ensure the test collection descriptor file is under /sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_A.		
5	Ensure the ESDT Maintenance GUI shows collection C2 is not installed.		
6	Ensure the BMGT dispatcher and auto driver are up and running: ./EcBmBMGTAppStart <MODE> select propertyvalue from bg_configuration_property where propertyname = 'BMGT.Dispatcher.Running'; propertyvalue should be true; On the BMGT host ps -ef | grep <MODE> | grep EcBmDispatcher should show a single dispatcher process running. On the BMGT host ps -ef | grep <MODE> | grep EcBmAuto should show a single auto driver process running		
7	Ensure a local copy of the ECHO 10 schema files is available for validation (from https://api.echo.nasa.gov/ingest/schema).		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Setup		#comment
10	S-2 Find collections which have ECS Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export).		#comment

#	Action	Expected Result	Notes
11	Install collection C2 into ECS. (Use the ESDT Maintenance GUI, the DPL Maintenance GUI, a DB patch from the command line, and the DPL Ingest GUI.)		

12	Use populate script EcBgPopulateCollections.ksh or a modifed populate script to check if the collection does not exist and insert a row in the bg_collection_configuration table.
	
	IF NOT EXISTS (
	SELECT collectionid
	  from bg_collection_configuration
	where shortname = <C2_ShortName>
	  and versionid = <C2_VersionId>
	) THEN
	  insert into bg_collection_configuration (collectionid, shortname, versionid, longname, granuleexportflag, collectionexportflag)
	  select collectionid, shortname, versionid, longname, 'Y' as granuleexportflag, 'Y' as collectionexportflag
	  from amcollection
	  where shortname = <C2_ShortName>
	  and VersionId = <C2_VersionId>
	-- update the other fields based on esdt type
	ELSE
	-- Enable Collection C2 for Collection and Granule Export .
	  update bg_collection_configuration
	  set granuleexportflag = 'Y' , collectionexportflag = 'Y'
	  where shortname = <C2_ShortName>
	  and versionid = <C2_VersionId>;
	END IF;		

13	V-2 Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		#comment
14	Verify that the TCP proxy log shows a single HTTP PUT request for collection C2.		
15	Verify that the HTTP PUT request contains collection C2's full metadata.		
16	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		#comment
17	Verify that collection C2's exported metadata validates agatinst the ECHO 10 collection schema: xmllint --noout --schema Collection.xsd C2.xml		
18	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		#comment
19	Use an xpath utility to verify the exported metadata has the following elements.		#comment

#	Action	Expected Result	Notes
20	a) Visible = true xpath /Collection/Visible C2.xml	<Visible>true</Visible>	
21	b) Orderable = false xpath /Collection/Orderable C2.xml	<Orderable>false</Orderable >	
22	c) InsertTime = The insert time of the collection recorded in the AIM database. xpath /Collection/InsertTime C2.xml	<InsertTime><YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string returned is equal to what is returned from the query to the aim_<MODE> schema in the ecs database: select inserttime from amcollection where shortname = <C2_ShortName> and versionid = <C2_VersionId>;	
23	d) LastUpdate = The last update time of the collection recorded in the AIM database. xpath /Collection/LastUpdate C2.xml	<LastUpdate><YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query: select lastupdate from amcollection where shortname = <C2_ShortName> and versionid = <C2_VersionId>;	
24	V-6 Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		#comment
25	Verify that collection C2's exported metadata contains version numbers with no leading zeroes: xpath /Collection/VersionId C2.xml	<VersionId>${VERSIONID}</VersionId>	

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
10 S-2		Install and enable new collection for			1 Collection to be installed (C2)		/sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_A	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		automatic export						

EXPECTED RESULTS:

	V	10	1	Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but no other collections sharing a short name but with different version IDs).		
	V	10	2	Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		
	V	10	3	Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but containing no collection metadata in the request body.		
	V	10	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		
	V	10	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		
	V	10	6	Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		

1.5 Nominal Collection Export[S-2b]: Automatic Export: Delete Collection (ECS-ECSTC-4)

DESCRIPTION:

	S	10	1	**[Nominal Collection Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Request the manual export of collection metadata for one of these collections.		
	S	10	2	Find collections which have ECS Metadata:		

				a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Update an existing collection. d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/current32/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. Should include 'aim_<MODE>'	
4	Ensure collection C3's descriptor file is under /sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_B.		
5	Ensure collection C3 has been installed in the mode (e.g., the DPL Ingest GUI shows C3 as a configured datatype).		
6	Ensure the BMGT dispatcher and auto driver are up and running: ./EcBmBMGTAppStart <MODE> select propertyvalue from bg_configuration_property where propertyname = 'BMGT.Dispatcher.Running'; propertyvalue should be true; On the BMGT host ps -ef | grep <MODE> | grep EcBmDispatcher should show a single dispatcher process running. On the BMGT host ps -ef | grep <MODE> | grep EcBmAuto should show a single auto driver process running		
7	Ensure a local copy of the ECHO 10 schema files is available for validation (from https://api.echo.nasa.gov/ingest/schema).		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Setup		#comment
10	S-2 Find collections which have ECS Metadata: b) Delete a collection from the ECS inventory.		#comment
11	Ensure collection C3 is enabled for collection and granule export: select granuleexportflag, collectionexportflag from		

#	Action	Expected Result	Notes
	bg_collection_configuration where shortname = <C3_ShortName> and versionid = <C3_VersionId>; both fields should be set to 'Y'		
12	Use the ESDT Maintenance GUI to delete collection C3.		
13	V-3 Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but containing no collection metadata in the request body.		#comment
14	Verify the TCP proxy log shows a single HTTP DELETE request for collection C3.		
15	Verify the DELETE request has collection C3's dataset ID in the URL (longname + " V" + versionid). E.g., if LongName = "ASTER On-Demand L2 Surface Emissivity" and VersionId = 2, the datasetid = ""ASTER On-Demand L2 Surface Emissivity V002". Some characters, such as spaces, are escaped in the URL (e.g., " " => "%20"). Example of a delete request on testbed with provider ID EDF_DEV02: DELETE /catalog-rest/providers/EDF_DEV02/datasets/ASTER%20On-Demand%20L2%20Surface%20Emissivity%20V002 HTTP/1.1		
16	Verify collection C3's DELETE request has no request body (no Content-Length header and no content following the headers).		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
10 S-2b		A collection to delete (C3)	GLAH02.033		1 collection		/sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_B	

EXPECTED RESULTS:

	V	10	1	Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but no other collections sharing a short name but with different version IDs).		

	V	10	2	Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		
	V	10	3	Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but containing no collection metadata in the request body.		
	V	10	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		
	V	10	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		
	V	10	6	Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		

1.6 Nominal Collection Export[S-2c]: Automatic Export: Update Collection (ECS-ECSTC-5)

DESCRIPTION:

	S	10	1	**[Nominal Collection Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Request the manual export of collection metadata for one of these collections.		
	S	10	2	Find collections which have ECS Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Update an existing collection. d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment

#	Action	Expected Result	Notes
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/current32/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. Should include 'aim_<MODE>'	
4	Ensure the test collections are under /sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_C		
5	Ensure collection C4 is installed (e.g., the DPL Ingest GUI shows C4 as a configured datatype).		
6	Ensure collection C4 is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = <C4_ShortName> and versionid = <C4_VersionId>; They should be set to 'Y'.		
7	Ensure a local copy of the ECHO 10 schema files is available for validation (from https://api.echo.nasa.gov/ingest/schema).		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Setup		#comment
10	S-2 Find collections which have ECS Metadata: c) Update an existing collection.		#comment
11	Use the ESDT Maintenance GUI to update the collection C4 with a new descriptor file, or update the DsGeESDTConfiguredType manually by executing the following queries sequentially: update DsGeESDTConfiguredType set esdtstate = 'updating' where configuredname = <C4_ShortName> and versionid = <C4_VersionId>; update DsGeESDTConfiguredType set esdtstate = 'installed' where configuredname = <C4_ShortName> and versionid = <C4_VersionId>;		
12	Verify the dsmdgreventhistory table has a CLUPDATE event for this collection: select eventtime, dbid, eventtype from dsmdgreventhistory where shortname = <C4_ShortName> and versionid = <C4_VersionId>;		
13	V-2 Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		#comment
14	Verify that the TCP proxy log shows a single HTTP PUT request for collection C4.		

#	Action	Expected Result	Notes
15	Verify that the HTTP PUT request contains collection C4's full metadata.		
16	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		#comment
17	Verify that collection C4's exported metadata validates agatinst the ECHO 10 collection schema: xmllint --noout --schema Collection.xsd C4.xml		
18	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		#comment
19	Use an xpath utility to verify the exported metadata has the following elements.		#comment
20	a) Visible = true xpath /Collection/Visible C4.xml	<Visible>true</Visible>	
21	b) Orderable = false xpath /Collection/Orderable C4.xml	<Orderable>false</Orderable >	
22	c) InsertTime = The insert time of the collection recorded in the AIM database. xpath /Collection/InsertTime C4.xml	<InsertTime><YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string returned is equal to what is returned from the query to the aim_<MODE> schema in the ecs database: select inserttime from amcollection where shortname = <C4_ShortName> and versionid = <C4_VersionId>;	
23	d) LastUpdate = The last update time of the collection recorded in the AIM database. xpath /Collection/LastUpdate C4.xml	<LastUpdate><YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query: select lastupdate from amcollection where shortname = <C4_ShortName> and versionid = <C4_VersionId>;	
24	V-6 Verify that the metadata exported in S-1 and S-2 contains version		#comment

#	Action	Expected Result	Notes
	numbers with no leading zeroes.</i>		
25	Verify that collection C4's exported metadata contains version numbers with no leading zeroes: xpath /Collection/VersionId C4.xml	<VersionId>${VERSIONID}</VersionId>	

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
10 S-2c		Collection to test ESDT type update (C4)	GLAH03.033	2 descriptor files (1 + replacement)	1 collection		/sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_C	

EXPECTED RESULTS:

	V	10	1	Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but no other collections sharing a short name but with different version IDs).		
	V	10	2	Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		
	V	10	3	Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but containing no collection metadata in the request body.		
	V	10	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		
	V	10	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		
	V	10	6	Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		

1.7 Nominal Collection Export[S-2d]: Automatic Export: Enable Collection Export (ECS-ECSTC-6)

DESCRIPTION:

	S	10	1	**[Nominal Collection Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Request the manual export of collection metadata for one of these collections.		
	S	10	2	Find collections which have ECS Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Update an existing collection. d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/current32/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. Should include 'aim_<MODE>'	
4	Ensure collection C5's descriptor file is under /sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_D.		
5	Ensure collection C5 is installed (e.g., the DPL Ingest GUI shows it as a configured datatype).		
6	Ensure a row exists in bg_collection_configuration for collection C5 by running EcBgPopulateCollections.ksh (or a modifed populate script):		
7	Ensure collection C5 is disabled for collection export: update bg_collection_configuration set collectionexportflag = 'N', granuleexportflag = 'N' where shortname = '<C5_ShortName>' and versionid = '<C5_VersionId>'		
8	Ensure the BMGT dispatcher is up and running and the auto driver is		

#	Action	Expected Result	Notes
	started: On the BMGT host: EcBmBMGTAppStart <MODE> select propertyvalue from bg_configuration_property where propertyname = 'BMGT.Dispatcher.Running'; propertyvalue should be true; On the BMGT host: ps -ef | grep <MODE> | grep EcBmDispatcher should show a single dispatcher process running ps -ef | grep <MODE> | grep EcBmAuto should show a single auto driver process running		
9	Ensure a local copy of the ECHO 10 schema files is available for validation (from https://api.echo.nasa.gov/ingest/schema).		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Setup		#comment
12	S-2 Find collections which have ECS Metadata: d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		#comment
13	Enable collection C5 for collection (but not granule) export: update bg_collection_configuration set granuleexportflag = 'N', collectionexportflag = 'Y' where shortname = <C5_ShortName> and versionid = <C5_VersionId>; 		
14	V-2 Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		#comment
15	Verify that the TCP proxy log shows a single HTTP PUT request for collection C5.		
16	Verify that the HTTP PUT request contains collection C5's full metadata.		
17	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		#comment
18	Verify collection C1's exported metadata validates against the ECHO 10 collection schema. (https://api.echo.nasa.gov/ingest/schema/Collection.xsd). xmllint --noout --schema Collection.xsd collection.xml		
19	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		#comment
20	Use an xpath utility to verify the exported metadata has the following		#comment

#	Action	Expected Result	Notes
	elements.		
21	a) Visible = true xpath /Collection/Visible C5.xml	<Visible>true</Visible>	
22	b) Orderable = false xpath /Collection/Orderable C5.xml	<Orderable>false</Orderable >	
23	c) InsertTime = The insert time of the collection recorded in the AIM database. xpath /Collection/InsertTime C5.xml	<InsertTime><YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string returned is equal to what is returned from the query to the aim_<MODE> schema in the ecs database: select inserttime from amcollection where shortname = <C5_ShortName> and versionid = <C5_VersionId>;	
24	d) LastUpdate = The last update time of the collection recorded in the AIM database. xpath /Collection/LastUpdate C5.xml	<LastUpdate><YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query: select lastupdate from amcollection where shortname = <C5_ShortName> and versionid = <C5_VersionId>;	
25	V-6 Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		#comment
26	Verify that collection C5's exported metadata contains version numbers with no leading zeroes: xpath /Collection/VersionId C5.xml	<VersionId>${VERSIONID}</VersionId> where ${VERSIONID} has no leading zeros.	

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
S-		1 collection to	GLAH04.033				/sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_D	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
2d		test enable collection export after collection install (C5)						

EXPECTED RESULTS:

	V	10	1	Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but no other collections sharing a short name but with different version IDs).		
	V	10	2	Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		
	V	10	3	Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but containing no collection metadata in the request body.		
	V	10	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		
	V	10	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the collection recorded in the AIM database. d) LastUpdate = The last update time of the collection recorded in the AIM database.		
	V	10	6	Verify that the metadata exported in S-1 and S-2 contains version numbers with no leading zeroes.		

1.8 Collection Deletion (ECS-ECSTC-7)

DESCRIPTION:

	S	30	1	**[Collection Deletion]** Stop the BMGT automatic driver		

				process so that new events are not picked up.		
	S	30	2	Perform the following for granules in the same collection: a) Granule insert. b) Granule update. c) Granule delete.		
	S	30	3	Delete the collection containing the granules used in S-2.		
	S	30	4	Restart the BMGT automatic driver.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure the test collection is installed in the mode.		
3	Ensure the collection is configured for collection and granule export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a TCP proxy is capturing BMGT traffic.		
5	Setup		#comment
6	S-1 Stop the BMGT automatic driver process so that new events are not picked up.		#comment
7	On the host where the BMGT automatic driver is running (f5oml01v), cd /usr/ecs/<MODE>/CUSTOM/utilities ./EcBmBMGTAutoStop <MODE>		
8	S-2 Perform the following for granules in the same collection: a) Granule insert b) Granule update c) Granule delete		#comment
9	Ingest a granule. Record its GranuleId.		
10	Update the granule, e.g. by changing its DayNightFlag value. <br		

#	Action	Expected Result	Notes
	/>select daynightflag from amgranule where granuleid = $GRANULE_ID -- If 'Y', change to 'N', else change to 'Y' update amgranule set daynightflag = 'Y' where granuleid = $GRANULE_ID		
11	Delete the granule and any others belonging to the collection. [f5oml01v] EcDsBulkDelete.pl [f5dpl01v] EcDlUnpublishStart.pl [f5oml01v] EcDsDeletionCleanup.pl 		
12	V-1 Verify that the actions in S-2 trigger events in the AIM event queue.		#comment
13	Verify the insert, update, and delete events appear in the AIM event queue: select * from dsmdgreventhistory where dbid= $GRANULE_ID	One row for each of GRINSERT, GRUPDATE, GRDELETE, in that chronological order.	
14	S-3 Delete the collection containing the granules used in S-2.		#comment
15	Note: A collection may be deleted from AIM only if no granules belonging to the collection exist in the AIM inventory.		#comment
16	Find the collection's group: select groupid from amcollection where shortname = $SHORT_NAME and versionid = $VERSION_ID		
17	Open the Data Pool Maintenance GUI. Visit Collection Groups. Visit the collection's group. Check the box to delete collection. Click DeleteCollection.		
18	Open the ESDT Maintenance GUI. Select the collection (ESDT). Click "Delete Selected ESDTs".		
19	S-4 Restart the BMGT automatic driver.		#comment
20	ssh f5oml01v cd /usr/ecs/CUSTOM/<MODE>/utilities ./EcBmBMGTAutoStart <MODE>		
21	Verification		#comment
22	V-2 Verify that after BMGT is restarted in S-4, no granule events are exported for the actions in S-2.		#comment
23	Verify the TCP proxy log shows no granule events sent to ECHO for the test granule.		
24	V-3 Verify that a single HTTP DELETE is exported to ECHO, or an ECHO stand-in.		#comment
25	Verify the TCP proxy log shows a single HTTP DELETE for the test collection.		
26	V-4 Verify that the URL to which the DELETE is exported contains an identifier for the deleted collection.		#comment

#	Action	Expected Result	Notes
27	Verify the TCP proxy log shows the URL to which the DELETE is sent includes the collection's dataset ID (URL encoded): ${LongName} V${VersionId} E.g., DELETE /catalog-rest/providers/EDF_DEV02/datasets/ASTER%20On-Demand%20L2%20Surface%20Emissivity%20V002 HTTP/1.1		
28	V-5 Verify that the DELETE request body is empty.		#comment
29	Verify the TCP proxy log shows the collection's DELETE request has an empty body.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
30				One collection and at least one granule, all of which will be deleted during the test.			/sotestdata/DROP_802/BE_82_01/Criteria/030	

EXPECTED RESULTS:

	V	30	1	Verify that the actions in S-2 trigger events in the AIM event queue.		
	V	30	2	Verify that after BMGT is restarted in S-4, no granule events are exported for the actions in S-2.		
	V	30	3	Verify that a single HTTP DELETE is exported to ECHO, or an ECHO stand-in.		
	V	30	4	Verify that the URL to which the DELETE is exported contains an identifier for the deleted collection.		
	V	30	5	Verify that the DELETE request body is empty.		

1.9 ECS Granule Additional Metadata[S-1]: Backtrack Orbit Metadata (ECS-ECSTC-8)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Ensure the test collections exist in AIM: select esdt(shortname, versionid) ESDT from amcollection where (shortname='AE_Land'		

#	Action	Expected Result	Notes
	and versionid=2) or (shortname='AMSR-L1A' and versionid=2) or (shortname='GLAH06' and versionid=33) or (shortname='GLAH07' and versionid=33) or (shortname='GLAH08' and versionid=33) or (shortname='MB2LME' and versionid=2) order by shortname, versionid		
3	Ensure the test collections are configured for BMGT collection and granule metadata export, as well as appropriate backtrack orbit groups: select esdt(shortname, versionid) ESDT, collectionexportflag, granuleexportflag, orbitgroup from bg_collection_configuration where (shortname='AE_Land' and versionid=2) or (shortname='AMSR-L1A' and versionid=2) or (shortname='GLAH06' and versionid=33) or (shortname='GLAH07' and versionid=33) or (shortname='GLAH08' and versionid=33) or (shortname='MB2LME' and versionid=2) order by shortname, versionid After changing anything in bg_collection_configuration, export those collections before exporting their granules.	Backtrack orbit groups by ESDT: AE_Land.002 AMSR-E AMSR-L1A.002 AMSR-A GLAH06.033 GLAS Quarter orbit GLAH07.033 GLAS Two orbit GLAH08.003 GLAS 14 orbit MB2LME.002 MISR	
4	Ensure the test granules have been ingested: select shortname, versionid, granuleid from amgranule where localgranuleid in ('LGID', ...)		
5	Ensure a mock ECHO or TCP proxy is capturing BMGT requests.		
6	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available.		
7	Setup		#comment
8	S-1 For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		#comment
9	Write the backtrack granule IDs to a text file: s1_granule_ids.txt		
10	Note the current time as t0.		
11	Supply the granule IDs to the BMGT manual export script: EcBmBMGTManualStart ${MODE} --metg --granulefile /path/to/s1_granule_ids.txt		
12	Verification		#comment
13	V-1 Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		#comment
14	From the TCP proxy log, save the body of each PUT request (the granule		

#	Action	Expected Result	Notes
	metadata XML), after time t0, to a separate XML file.		
15	Use the Ruby script /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
16	BackTrackOrbitGroup: AMSR-E ESDT: AE_Land.002		#comment
17	Verify the AE_Land.002 granule's exported metadata has a Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
18	Find the granule's first equator crossing datetime: xpath '//EquatorCrossingDateTime/text()' granule.xml | sort		
19	Extract the granule's FIRST_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${first_equator_crossing_datetime}']/EquatorCrossingLongitude/text()" granule.xml		
20	Extract the granule's ASC_DESC_FLAG: xpath "//AdditionalAttribute[Name='AscendingDescendingFlg']/Values/Value/text()" granule.xml		
21	If ASC_DESC_FLAG is 'Ascending' or 'ASCENDING', then verify the granule's Orbit block matches this: <Orbit> <AscendingCrossing>${FIRST_ASC_CROSS}</AscendingCrossing> <StartLat>-81.8</StartLat> <StartDirection>A</StartDirection> <EndLat>81.8</EndLat> <EndDirection>A</EndDirection> </Orbit> If ASC_DESC_FLAG is 'Descending', then verify the granule's Orbit block matches this: <Orbit> <AscendingCrossing>${FIRST_ASC_CROSS} - 167.64</AscendingCrossing> <StartLat>81.8</StartLat> <StartDirection>D</StartDirection> <EndLat>-81.8</EndLat> <EndDirection>D</EndDirection> </Orbit>		
22	BacktrackOrbitGroup: AMSR-A ESDT: AMSR-L1A.002		#comment
23	Verify the AMSR-L1A.002 granule's exported metadata has a Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
24	Find the granule's first equator crossing datetime: xpath '//EquatorCrossingDateTime/text()' granule.xml | sort		

#	Action	Expected Result	Notes
25	Extract the granule's FIRST_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${first_equator_crossing_datetime}']/EquatorCrossingLongitude/text()" granule.xml		
26	Extract the granule's ASC_DESC_FLAG: xpath "//AdditionalAttribute[Name='AscendingDescendingFlg']/Values/Value/text()" granule.xml		
27	If ASC_DESC_FLAG is 'Ascending' or 'ASCENDING', then verify the granule's Orbit block matches this: <Orbit> <AscendingCrossing>${FIRST_ASC_CROSS}</AscendingCrossing> <StartLat>-81.8</StartLat> <StartDirection>A</StartDirection> <EndLat>81.8</EndLat> <EndDirection>A</EndDirection> </Orbit> If ASC_DESC_FLAG is 'Descending', then verify the granule's Orbit block matches this: <Orbit> <AscendingCrossing>${FIRST_ASC_CROSS} - 167.375</AscendingCrossing> <StartLat>81.8</StartLat> <StartDirection>D</StartDirection> <EndLat>-81.8</EndLat> <EndDirection>D</EndDirection> </Orbit>		
28	BacktrackOrbitGroup: GLAS Quarter Orbit ESDT: GLAH06.033		#comment
29	Verify the GLAH06.033 granule's exported metadata has a Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
30	Extract the granule's TRACK_SEGMENT: xpath "//AdditionalAttribute[Name='Track_Segment']/Values/Value/text()" granule.xml		
31	Find the granule's equator crossing datetimes: xpath '//EquatorCrossingDateTime/text()' granule.xml | sort If the TRACK_SEGMENT is 1, 2, or 3, note the earliest (first) datetime. If the TRACK_SEGMENT is 4, note the second earliest datetime.		
32	If the TRACK_SEGMENT is 1, 2, or 3, extract the granule's FIRST_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${first_equator_crossing_datetime}']/EquatorCrossingLongitude/text()" granule.xml If the TRACK_SEGMENT is 4, extract the granule's		

#	Action	Expected Result	Notes
	SECOND_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${second_equator_crossing_datetime}']/EquatorCrossingLongitude/text()" granule.xml		
33	Verify the granule's Orbit block matches one of the following, depending on the TRACK_SEGMENT: \# TRACK_SEGMENT = 1 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>D</EndDirection> </Orbit> \# TRACK_SEGMENT = 2 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>D</StartDirection> <EndLat>-50.0</EndLat> <EndDirection>D</EndDirection> </Orbit> \# TRACK_SEGMENT = 3 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-50.0</StartLat> <StartDirection>D</StartDirection> <EndLat>-50.0</EndLat> <EndDirection>A</EndDirection> </Orbit> \# TRACK_SEGMENT = 4 <Orbit> <AscendingCrossing>SECOND_ASC_CROSS</AscendingCrossing> <StartLat>-50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		
34	BacktrackOrbitGroup: GLAS Two Orbit ESDT: GLAH07.033		#comment
35	Verify the GLAH07.033 granule's exported metadata has a Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
36	Find the granule's first equator crossing datetime: xpath '//EquatorCrossingDateTime/text()' granule.xml | sort		
37	Extract the granule's FIRST_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${first_e		

#	Action	Expected Result	Notes
	quator_crossing_datetime}']/EquatorCrossingLongitude/text()" granule.xml		
38	Verify the granule's Orbit block matches the following: <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		
39	BacktrackOrbitGroup: GLAS 14 Orbit ESDT: GLAH08.003		#comment
40	Verify the GLAH08.033 granule's exported metadata has a Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
41	Find the granule's first equator crossing datetime: xpath '//EquatorCrossingDateTime/text()' granule.xml | sort		
42	Extract the granule's FIRST_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${first_equator_crossing_datetime}']/EquatorCrossingLongitude/text()" granule.xml		
43	Verify the granule's Orbit block matches the following: <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		
44	BacktrackOrbitGroup: MISR ESDT: MB2LME.002		#comment
45	Verify the MB2LME.002 granule's exported metadata has a Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
46	Extract the granule's EQ_CROSS_LONG: xpath '//EquatorCrossingLongitude/text()' granule.xml		
47	Find the granule's first equator crossing datetime: xpath '//EquatorCrossingDateTime/text()' granule.xml | sort		
48	Extract the granule's FIRST_ASC_CROSS: xpath "//OrbitCalculatedSpatialDomain[EquatorCrossingDateTime='${first_equator_crossing_datetime}']/EquatorCrossingLongitude/text()"		

#	Action	Expected Result	Notes
	granule.xml		
49	Extract the granule's START_BLOCK: xpath "//AdditionalAttribute[Name='SP_AM_MISR_StartBlock']/Values/Value/text()" granule.xml		
50	Extract the granule's END_BLOCK: xpath "//AdditionalAttribute[Name='SP_AM_MISR_EndBlock']/Values/Value/text()" granule.xml		
51	Extract the granule's FIRST_EDGE: xpath ("//property[Block='${START_BLOCK}']/FirstEdge/text()" /usr/ecs/OPS/CUSTOM/data/BMGT/config/EcBmBMGTMISRBlockLat.xml		
52	Extract the granule's LAST_EDGE: xpath ("//property[Block='${END_BLOCK}']/LastEdge/text()" /usr/ecs/OPS/CUSTOM/data/BMGT/config/EcBmBMGTMISRBlockLat.xml		
53	Verify the granule's Orbit block matches one of the following, depending on the value of EQ_CROSS_LONG: # EQ_CROSS_LONG > 347.65 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 527.65</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection>D if END_BLOCK between 19 and 162. A otherwise</EndDirection> </Orbit> # EQ_CROSS_LONG < -12.35 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS + 192.35</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection>D if END_BLOCK between 19 and 162. A otherwise</EndDirection> </Orbit> # EQ_CROSS_LONG between -12.35 and 347.65 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 167.65</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A		

#	Action	Expected Result	Notes
	otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise</EndDirection> </Orbit>		
54	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
55	Verify each granule's exported metadata validates against the ECHO 10 schema: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		
56	BE_82_01_AdditionalMetadataDescription.doc		#comment
57	1.1.1.2.7 Backtrack orbit metadata (Granule & Collection)		#comment
58	ECHO provides the ability to perform backtrack orbit searches, provided that the proper metadata is provided for granules. This metadata is mostly contained in ECS granule and collection metadata, but is not in the proper format. It is either contained in the OrbitCalculatedSpatialDomain hierarchy, PSA values, or must be calculated from these values using standard rules and/or ancilliary files. A number of groups have been established, each with different rules for generating granule and collection backtrack metadata. Each collection may be associated with one of these groups via the Collection Configuration database table. Backtrack orbit metadata is not required. #Valid values for BackTrack Orbit Group are (values in parenthesis refer to alternative names used in legacy code. We should use the more descriptive names below): GLAS Quarter Orbit (AKA GLAS1) GLAS Two Orbit (AKA GLAS2) GLAS 14 Orbit (AKA GLAS3) AMSR-E (AKA AMSR1) AMSR-A (AKA AMSR2) MISR (LARC) LPDAAC (This was used at one point but from what I can tell is no longer used. Need to verify)		#comment
59	The rules for each group are defined below, but first is a list of the variables that the rules reference, and where they are obtained from. Values obtained from OrbitCalculatedSpatialDomainContainer FIRST_ASC_CROSS = the EquatorCrossingLongitude value of the OrbitCalculatedSpatialDomainContainer enclosure with the earliest EquatorCrossingTime. SECOND_ASC_CROSS = the EquatorCrossingLongitude value of the OrbitCalculatedSpatialDomainContainer enclosure with the second earliest EquatorCrossingTime. Values obtained from PSAs TRACK_SEGMENT = The value of the 'Track_Segment' PSA.<br		#comment

#	Action	Expected Result	Notes
	/>START_BLOCK = The value of the 'SP_AM_MISR_StartBlock' PSA. END_BLOCK = The value of the 'SP_AM_MISR_EndBlock' PSA. ASC_DESC_FLAG = The value of the ‘AscendingDescendingFlg’ PSA. Values obtained from separate MISRBlockLat.xml file Format: <properties> <property> <Block>180</Block> <FirstEdge>-66.695034</FirstEdge> <LastEdge>-65.502984</LastEdge> </property> … </properties> FIRST_EDGE = The FirstEdge value of the property with Block = START_BLOCK LAST_EDGE = The LastEdge value of the property with Block = END_BLOCK		
60	Group Rules:		#comment
61	GLAS Quarter Orbit		#comment
62	Collection:		#comment
63	<OrbitParameters> <SwathWidth>2.0</SwathWidth> <Period>96.7</Period> <InclinationAngle>94.0</InclinationAngle> <NumberOfOrbits>0.25</NumberOfOrbits> <StartCircularLatitude>50.0</StartCircularLatitude> </OrbitParameters>		#comment
64	Granule:		#comment
65	TRACK_SEGMENT = 1		#comment
66	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>D</EndDirection> </Orbit>		#comment
67	TRACK_SEGMENT = 2		#comment
68	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>D</StartDirection> <EndLat>-50.0</EndLat> <EndDirection>D</EndDirection> </Orbit>		#comment
69	TRACK_SEGMENT = 3		#comment

#	Action	Expected Result	Notes
70	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-50.0</StartLat> <StartDirection>D</StartDirection> <EndLat>-50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
71	TRACK_SEGMENT = 4		#comment
72	<Orbit> <AscendingCrossing>SECOND_ASC_CROSS</AscendingCrossing> <StartLat>-50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
73	GLAS Two Orbit		#comment
74	Collection:		#comment
75	<OrbitParameters> <SwathWidth>2.0</SwathWidth> <Period>96.7</Period> <InclinationAngle>94.0</InclinationAngle> <NumberOfOrbits>2.0</NumberOfOrbits> <StartCircularLatitude>50.0</StartCircularLatitude> </OrbitParameters>		#comment
76	Granule:		#comment
77	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
78	GLAS 14 Orbit		#comment
79	Collection:		#comment
80	<OrbitParameters> <SwathWidth>2.0</SwathWidth> <Period>96.7</Period> <InclinationAngle>94.0</InclinationAngle> <NumberOfOrbits>14.0</NumberOfOrbits> <StartCircularLatitude>50.0</StartCircularLatitude> </OrbitParameters>		#comment

#	Action	Expected Result	Notes
81	Granule:		#comment
82	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
83	AMSR-E		#comment
84	Collection:		#comment
85	<OrbitParameters> <SwathWidth>1450.0</SwathWidth> <Period>98.88</Period> <InclinationAngle>98.15</InclinationAngle> <NumberOfOrbits>0.5</NumberOfOrbits> <StartCircularLatitude>-90.0</StartCircularLatitude> </OrbitParameters>		#comment
86	Granule:		#comment
87	ASC_DESC_FLAG = ‘Ascending’ or ‘ASCENDING’		#comment
88	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-81.8</StartLat> <StartDirection>A</StartDirection> <EndLat>81.8</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
89	ASC_DESC_FLAG = ‘Descending’		#comment
90	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS -167.64</AscendingCrossing> <StartLat>81.8</StartLat> <StartDirection>D</StartDirection> <EndLat>-81.8</EndLat> <EndDirection>D</EndDirection> </Orbit>		#comment
91	AMSR-A		#comment
92	Collection:		#comment
93	<OrbitParameters> <SwathWidth>1600.0</SwathWidth> <Period>101.0</Period> <InclinationAngle>98.62</InclinationAngle> 		#comment

#	Action	Expected Result	Notes
	<NumberOfOrbits>0.5</NumberOfOrbits> <StartCircularLatitude>-90.0</StartCircularLatitude> </OrbitParameters>		
94	Granule:		#comment
95	ASC_DESC_FLAG = ‘Ascending’ or ‘ASCENDING’		#comment
96	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-81.8</StartLat> <StartDirection>A</StartDirection> <EndLat>81.8</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
97	ASC_DESC_FLAG = ‘Descending’ or ‘DESCENDING’		#comment
98	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 167.375</AscendingCrossing> <StartLat>81.8</StartLat> <StartDirection>D</StartDirection> <EndLat>-81.8</EndLat> <EndDirection>D</EndDirection> </Orbit>		#comment
99	MISR		#comment
100	Collection:		#comment
101	<OrbitParameters> <SwathWidth>400.0</SwathWidth> <Period>98.88</Period> <InclinationAngle>98.3</InclinationAngle> <NumberOfOrbits>1.0</NumberOfOrbits> <StartCircularLatitude /> </OrbitParameters>		#comment
102	Granule:		#comment
103	EQ_CROSS_LONG > 347.65		#comment
104	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 527.65</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise </EndDirection> </Orbit>		#comment
105	EQ_CROSS_LONG < -12.35		#comment

#	Action	Expected Result	Notes
106	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS + 192.35</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise </EndDirection> </Orbit>		#comment
107	EQ_CROSS_LONG betweem -12.35 and 347.65		#comment
108	<Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 167.65</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise </EndDirection> </Orbit>		#comment
109	LPDAAC		#comment
110	Collection:		#comment
111	<OrbitParameters> <SwathWidth>1450.0</SwathWidth> <Period>98.88</Period> <InclinationAngle>98.15</InclinationAngle> <NumberOfOrbits>1.0</NumberOfOrbits> <StartCircularLatitude /> </OrbitParameters>		#comment
112	Granule:		#comment
113	NO ORBIT METADATA		#comment

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
50 S-			AE_Land.002 AMSR-	One granule for each of the backtrack	6 granules		/sotestdata/DROP_802/BE_82_01/Criteria/050/050_1	

Crit id	Critccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
1			L1A.002 GLAH06.033 GLAH07.033 GLAH08.033 MB2LME.002	types.				
50S-2			MOD10A1.005	One granule with no backtrack type.	1 granule		/sotestdata/DROP_802/BE_82_01/Criteria/050/050_2	
50S-3		gS	MOD10A2.005	TwoDCoordinateSystem = 'MODIS Tile SIN'	1 granule		/sotestdata/DROP_802/BE_82_01/Criteria/050/050_3/MOD10A2.005	
50S-3		gE	MOD29P1D.005	TwoDCoordinateSystem = 'MODIS Tile EASE'	1 granule		/sotestdata/DROP_802/BE_82_01/Criteria/050/050_3/MOD29P1D.005	
50S-3		gM	MB2LMT.002	TwoDCoordinateSystem = 'MISR'	1 granule		/sotestdata/DROP_802/BE_82_01/Criteria/050/050_3/MB2LMT.002	
50S-4				One granule with no TwoDCoordinateSystem.	1 granule		/sotestdata/DROP_802/BE_82_01/Criteria/050/050_4	
50S-5				One granule with Product Specific Attributes.			/sotestdata/DROP_802/BE_82_01/Criteria/050/050_5	
50S-6				One granule whose cloud cover comes from core metadata.			/sotestdata/DROP_802/BE_82_01/Criteria/050/050_6	
50S-7				One granule whose cloud cover comes from a PSA.			/sotestdata/DROP_802/BE_82_01/Criteria/050/050_7	

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
50S-8				One granule with no cloud cover source.			/sotestdata/DROP_802/BE_82_01/Criteria/050/050_8	
50S-9				One granule whose cloud cover source location (core metadata or PSA) is invalid.			/sotestdata/DROP_802/BE_82_01/Criteria/050/050_9	

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		

	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.10 Datapool URLs[S-3] Publish, Unpublish, Delete (ECS-ECSTC-9)

DESCRIPTION:

	S	60	1	**[Datapool URLs]** Find a collection which is enabled for collection and granule export.		
	S	60	2	Stop the BMGT servers.		
	S	60	3	For the collection identified in S-1: a. Publish a granule, and then delete it. b. Publish another granule, and then unpublish it.		
	S	60	4	Bring up BMGT.		
	S	60	5	For the collection identified in S-1: a. Publish a granule. b. Unpublish another granule.		
	S	60	6	Perform the following Online Archive repair functions: a. Execute EcDlCleanupGranules on a public granule. b. Execute EcDlCleanupGranules on a browse granule linked to a public granule. c. Execute EcDlCleanupGranules on a QA granule linked to a public granule. d. Execute EcDlCleanupGranules on a PH granule linked to a public granule. e. Execute EcDlCleanupGranules on a HDF MAP granule linked to a public granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure the test collection is installed. E.g., the DPL Ingest GUI shows it as configured datatypes.		
6	Ensure the test collection is enabled for collection and granule export: select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure granules g1, g2, and g3 are in the hidden data pool (publishtime is null): select shortname, versionid, granuleid, publishtime from amgranule where granuleid in (${g1_GRANULEID}, ${g2_GRANULEID}, ${g3_GRANULEID}) If needed, unpublish the granules: EcDlUnpublishStart.pl -mode $MODE -g ${g1_GRANULEID},${g2_GRANULEID},${g3_GRANULEID}		
9	Ensure granule g4 is in the public data pool (publishtime is non-null): select shortname, versionid, granuleid, publishtime from amgranule where granuleid = ${g4_GRANULEID} If needed, publish the granule: EcDlPublishStart $MODE -ecs -g ${g4_GRANULEID}		

#	Action	Expected Result	Notes
10	Ensure ECHO has the test granules' metadata. For each of g1, g2, g3, g4, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granules, export them: EcBmBMGTManualStart ${MODE} --metg --granules ${g1_GRANULEID},${g2_GRANULEID},${g4_GRANULEID},${g4_GRANULEID}		
11	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
12	Setup		#comment
13	S-1 Find a collection which is enabled for collection and granule export.		#comment
14	Done in preconditions.		#comment
15	S-2 Stop the BMGT servers.		#comment
16	Stop BMGT: EcBmBMGTAppStop $MODE		
17	S-3 For the collection identified in S-1: a. Publish a granule, and then delete it. b. Publish another granule, and then unpublish it.		#comment
18	Publish granules g1 and g2: EcDlPublishUtilityStart ${MODE} -ecs -g ${g1_GRANULEID},${g2_GRANULEID}		
19	Delete granule g1: Create a geoid file for g1: SC:${ShortName}.${VersionId}:${GranuleId} EcDsBulkDelete.pl -mode ${MODE} -physical -geodfile /path/to/geoid_file		
20	Unpublish granule g2: EcDlUnpublishStart.pl -mode ${MODE} -g ${g2_GRANULEID}		
21	S-4 Bring up BMGT.		#comment
22	Note the current time as t0.		
23	Start BMGT: EcBmBMGTAppStart $MODE		
24	S-5 For the collection identified in S-1: a. Publish a granule. b. Unpublish another granule.		#comment
25	Publish g3: EcDlPublishUtilityStart ${MODE} -ecs -g ${g3_GRANULEID}		
26	Unpublish g4: EcDlUnpublishStart.pl -mode ${MODE} -g ${g4_GRANULEID}		
27	Verification		#comment
28	V-1 Verify that for S-3, subclause a, an HTTP DELETE is exported.		#comment

#	Action	Expected Result	Notes
29	Verify the TCP proxy log indicates a DELETE was sent for granule g1 after time t0.		
30	V-2 Verify that for S-3, subclause b, granule metadata is exported, but contains no online access or online resource URLs.		#comment
31	Verify the TCP log shows a PUT was sent for granule g2 after time t0.		
32	Save the body of granule g2's PUT request to an XML file g2.xml.		
33	Verify granule g2's exported metadata contains no online access URLs: xpath '//OnlineAccessURL' g2.xml		
34	Verify granule g2's exported metadata contains no online resource URLs: xpath '//OnlineResource/URL' g2.xml		
35	V-3 Verify that for S-5, subclause a, full granule metadata is exported, including online access and online resource URLs.		#comment
36	Verify the TCP log shows a PUT was sent for granule g3 after time t0.		
37	Save the body of granule g3's PUT request to an XML file g3.xml.		
38	Verify g3's exported XML contains one or more online access URLs: xpath '//OnlineAccessURL/URL' granule.xml		
39	Verify g3's exported XML contains one or more online resource URLs: xpath '//OnlineResource/URL' granule.xml		
40	V-4 Verify that for S-5, subclause b, granule metadata is exported, but contains no online access or online resource URLs.		#comment
41	Verify the TCP log shows a PUT was sent for granule g4 after time t0.		
42	Save the body of granule g4's PUT request to an XML file g4.xml.		
43	Verify granule g4's exported metadata contains no online access URLs: xpath '//OnlineAccessURL' g4.xml		
44	Verify granule g4's exported metadata contains no online resource URLs: xpath '//OnlineResource/URL' g4.xml		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
60 S-3, S-5			MOP03N.005	3 granules belonging to a collection not configured to be			/sotestdata/DROP_802/BE_82_01/Criteria/060	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
				public on ingest.				
60 S-5, S-6			MOP03N.005 AE_DySno.002	3 granules belonging to a collection configured to be public on ingest.			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6c			AE_DySno.002	1 public granule with an associated browse granule			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6d			AE_DySno.002	1 public granule with an associated QA granule			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6e			AE_DySno.002	1 public granule with an associated PH granule			/sotestdata/DROP_802/BE_82_01/Criteria/060	

EXPECTED RESULTS:

	V	60	1	Verify that for S-3, subclause a, an HTTP DELETE is exported.		
	V	60	2	Verify that for S-3, subclause b, granule metadata is exported, but contains no online access or online resource URLs.		
	V	60	3	Verify that for S-5, subclause a, full granule metadata is exported, including online access and online resource URLs.		
	V	60	4	Verify that for S-5, subclause b, granule metadata is exported, but contains no online access or online resource URLs.		
	V	60	5	Verify that each of the operations in S-6 results in the export of granule metadata for the associated science granule containing updated URLs reflecting the modifications made by EcDlCleanupGranules.		

1.11 Nominal Granule Export[S-1]: Manual Export (ECS-ECSTC-10)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database. E.g., /tools/postgres/current32/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('AIM', $MODE); show search_path;	Should list the schemas installed in the mode. Should include 'aim_<MODE>'	
4	Ensure test data is available		

#	Action	Expected Result	Notes
	/sotestdata/DROP_802/BE_82_01/Criteria/040/040_1.		
5	Ensure a local copy of the ECHO 10 schema files is available for metadata validation.		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	S-1 [Nominal Granule Export] Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		#comment
9	Ensure collections C1 and C2 have same short name but different version IDs.		
10	Ensure collections C1 and C2 are installed (e.g,. the DPL Ingest GUI shows C1 and C2 as configured datatypes).		
11	Ensure test granules exist in AIM for each of collections C1 and C2: select shortname, versionid, granuleid from amgranule where shortname = '<C1_SHORTNAME>' and versionid in (<C1_VERSIONID>, <C2_VERSIONID>)		
12	Ensure collections C1 and C2 are enabled for collection and granule export. select shortname, versionid, granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = '<C1_SHORTNAME>' and versionid in (<C1_VERSIONID>, <C2_VERSIONID>)	They should be set to 'Y'.	
13	Request the manual export of granule metadata for all granules in collection C1. EcBmBMGTManualStart <MODE> --metg --collections <C1_SHORTNAME>.<C1_VERSIONID>		
14	V-1 Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		#comment
15	Verify that the collection and granules exported successfully. Check bg_export_error table		
16	Verify that the TCP proxy log indicates manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection C1.		
17	Verify that the TCP proxy log shows no HTTP PUT requests for granules belonging to collection C2.		

#	Action	Expected Result	Notes
18	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
19	Verify that the metadata exported in S-1 validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd). xmllint --noout -s Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		
20	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = true c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
21	Verify that the metadata exported has the following elements, using an xpath utilty, such as /tools/common/test/BE_82_01/bin/xpath.		#comment
22	a) Visible = true xpath /Granule/Visible granule.xml	Should return <Visible>true</Visible>	
23	b) Orderable = true xpath /Granule/Orderable granule.xml	Should return <Orderable>true</Orderable>	
24	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime granule.xml	Should return <InsertTime><YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = <granuleid>;	
25	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate granule.xml	Should return <LastUpdate><YYYY-MM-DDTHH:mm:SSZ</LastUpdate> where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid =	

#	Action	Expected Result	Notes
		<granuleid>;	

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
40-S1		test manual export of granules	2 Collections with same shortname, different versionids (C1, C2) and multiple science granules	NISE.002 NISE.004			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_1/NISE.002 /sotestdata/DROP_802/BE_82_01/Criteria/040/040_1/NISE.004	
40S-2a			1 granule to ingest (g1)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_A	
40S-2b			1 granule to logically delete (g2)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_B	
40S-2c			1 granule to physically delete (g3)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_C	
40S-2d			1 granule to DFA (g4)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_D	
40S-2e			1 granule to hide (g5)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_E	
40			1 granule to	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_F	

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
S-2f			restrict (g6)					
40S-2g			1 granule to unrestrict (g7)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_G	
40S-2h		Science QA Flag update	1 science granule with a measured parameter which can be updated (g8)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_H	
40S-2i			1 granule publish (g9)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_I	
40S-2j			1 granule to unpublish (g10)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_J	
40S-2k			1 granule (g11) 1 browse (b1) with a linkage file to link it to granule g11	1 science granule and browse granule or 1 browse granule linkage file for an existing science granule			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_K	
40S-2l			1 granule (g12) with an associated	1 browse granule linked to a			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_L	

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			browse (b2) to be unlinked	science granule 1 browse granule linked to a different science granule				
40 S-2 m		OBE	Move granule from one collection to another	2 collections with science granules			OBE	
40 S-2 n			1 granule (g13) in a collection to move to a different filesystem	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_N	
40 S-2 o			1 granule (g14) to undergo XML replacement	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_O	

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the		

				full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		

	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.12 Nominal Granule Export[S-2a]: Automatic Export: Ingest Granule (ECS-ECSTC-11)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment

#	Action	Expected Result	Notes
2	Ensure granule g1's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g1's collection is configured to be public on ingest.		
4	Ensure granule g1's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g1's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
9	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
10	Setup		#comment
11	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory.		#comment
12	Ingest a new granule g1 into the ECS inventory.		
13	Verification		#comment
14	V-2 Verify that the operations in S-2, except subclauses b – d, each result in the export of one or more HTTP PUTs containing the full granule metadata.		#comment
15	NOTE: Each action may result in multiple distinct events being triggered. The BMGT Auto driver will consolidate these events if they are picked up within the same polling interval. If they span multiple intervals, then		#comment

#	Action	Expected Result	Notes
	multiple, redundant exports could result.		
16	Verify the TCP proxy log one or more PUTs for granule g1. There could be 1 PUT after the granule is archived and another when the granule is published.		
17	Verify the TCP proxy log shows that each of granule g1's PUT requests contains the full granule metadata.		
18	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
19	Verify granule g1's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g1.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g1.xml		
20	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = true c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
21	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
22	a) Visible = true xpath /Granule/Visible g1.xml	<Visible>true</Visible>	
23	b) Orderable = true xpath /Granule/Orderable g1.xml	<Orderable>true</Orderable >	
24	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g1.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = <GRANULEID>;	
25	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g1.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid =	

#	Action	Expected Result	Notes
		<GRANULEID>;	

TEST DATA:
See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which		

				was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.13 Nominal Granule Export[S-2b]: Automatic Export: Logically Delete Granule (ECS-ECSTC-12)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule.		

				g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available, connected to the ecs database. /tools/postgres/9.2/bin/psql -U ${username} -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. should include 'aim_<MODE>'.	
4	Ensure test data exists under /sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_B.		
5	Ensure granule g2's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure the DPL Ingest GUI shows granule g2's collection is configured to be public on ingest.		
7	Ensure granule g2's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
8	Ensure ECHO has granule g2's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
9	Ensure granule g2 is in AIM: select shortname, versionid,		

#	Action	Expected Result	Notes
	granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
10	Ensure ECHO has granule g2's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
11	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
12	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
13	Setup		#comment
14	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): b) Logically delete a granule from the ECS inventory.		#comment
15	Add granule g2's geoid to a text file (geoids.txt).		
16	Logically delete granule g2 from AIM: ./EcDsBulkDelete.pl -physical -server ${DBSERVER} -database ecs -user ${DBUSER} -password ${DBPASSWORD} -geoidfile /path/to/geoids.txt		
17	V-3 Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		#comment
18	Verify the TCP proxy log shows a single DELETE for granule g2.		
19	Verify the TCP proxy log shows granule g2's delete has no body (no Content-Length header, no content following the HTTP headers).		
20	Verify the TCP proxy shows granule g2's DELETE includes the URL-encoded geoid in the URL. Example of a DELETE request for provider EDF_DEV02 and geoid SC:MOD10A1.005:3000159312 DELETE /catalog-rest/providers/EDF_DEV02/granules/SC%3AMOD10A1.005%3A3000159312		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		

	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.14 Nominal Granule Export[S-2c]: Automatic Export: Physically Delete Granule (ECS-ECSTC-13)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection.		

				o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available, connected to the ecs database. /tools/postgres/9.2/bin/psql -U ${username} -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. should include 'aim_<MODE>'.	
4	Ensure test data exists under /sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_C.		
5	Ensure granule g3's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure the DPL Ingest GUI shows granule g3's collection is configured to be public on ingest.		
7	Ensure granule g3's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
8	Ensure ECHO has granule g3's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
9	Ensure granule g3 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
10	Ensure ECHO has granule g3's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
11	Ensure a local copy of the ECHO 10 schema files is available for schema		

#	Action	Expected Result	Notes
	validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
12	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
13	Setup		#comment
14	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): c) Physically delete a granule from the ECS inventory.		#comment
15	Physically delete a granule from the ECS inventory.		#comment
16	Add granule g3's geoid to a text file (geoids.txt).		
17	Logically delete granule g3 from AIM: ./EcDsBulkDelete.pl -physical -server ${DBSERVER} -database ecs -user ${DBUSER} -password ${DBPASSWORD} -geoidfile /path/to/geoids.txt		
18	Unpublish granule g3: ./EcDlUnpublishStart.pl -mode ${MODE} -g ${GRANULEID}		
19	Physically delete granule g3 by running deletion cleanup and choosing the most destructive option at every opportunity. ./EcDsDeletionCleanup.pl -mode ${MODE} -user ${DBUSER} -server ${DBSERVER} -database ecs -batch 1000 -grbatch 100 -xmlbatch 1000 -databatch 10000 -logbatch 100		
20	V-3 Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		#comment
21	Verify the TCP proxy log shows a single DELETE for granule g3.		
22	Verify the TCP proxy log shows granule g3's delete has no body (no Content-Length header, no content following the HTTP headers).		
23	Verify the TCP proxy shows granule g3's DELETE includes the URL-encoded geoid in the URL. Example of a DELETE request for provider EDF_DEV02 and geoid SC:MOD10A1.005:3000159312 DELETE /catalog-rest/providers/EDF_DEV02/granules/SC%3AMOD10A1.005%3A3000159312		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		

	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.15 Nominal Granule Export[S-2d]: Automatic Export: DFA Granule (ECS-ECSTC-14)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection.		

				o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available, connected to the ecs database. /tools/postgres/9.2/bin/psql -U ${username} -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. should include 'aim_<MODE>'.	
4	Ensure test data exists under /sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_D.		
5	Ensure granule g4's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure the DPL Ingest GUI shows granule g4's collection is configured to be public on ingest.		
7	Ensure granule g4's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
8	Ensure ECHO has granule g4's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
9	Ensure granule g4 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
10	Ensure ECHO has granule g4's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
11	Ensure a local copy of the ECHO 10 schema files is available for schema		

#	Action	Expected Result	Notes
	validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
12	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
13	Setup		#comment
14	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): d) DFA a granule.		#comment
15	*Add granule g4's geoid to a text file (geoids.txt).		
16	DFA granule g4 in AIM: ./EcDsBulkDelete.pl -dfa -server ${DBSERVER} -database ecs -user ${DBUSER} -password ${DBPASSWORD} -geoidfile /path/to/geoids.txt		
17	V-3 Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		#comment
18	Verify the TCP proxy log shows a single DELETE for granule g4.		
19	Verify the TCP proxy log shows granule g4's delete has no body (no Content-Length header, no content following the HTTP headers).		
20	Verify the TCP proxy shows granule g4's DELETE includes the URL-encoded geoid in the URL. Example of a DELETE request for provider EDF_DEV02 and geoid SC:MOD10A1.005:3000159312 DELETE /catalog-rest/providers/EDF_DEV02/granules/SC%3AMOD10A1.005%3A3000159312		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the		

				full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		

	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.16 Nominal Granule Export[S-2e]: Automatic Export: Hide a Granule (ECS-ECSTC-15)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment

#	Action	Expected Result	Notes
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available, connected to the ecs database. /tools/postgres/9.2/bin/psql -U ${username} -h f4dbl03 -d ecs select public.set_search_path('AIM', MODE); show search_path;	Should list the schemas installed in the mode. should include 'aim_<MODE>'.	
4	Ensure test data exists under /sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_E.		
5	Ensure granule g5's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure the DPL Ingest GUI shows granule g5's collection is configured to be public on ingest.		
7	Ensure granule g5's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
8	Ensure ECHO has granule g5's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
9	Ensure granule g5 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
10	Ensure ECHO has granule g5's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
11	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
12	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
13	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		

#	Action	Expected Result	Notes
14	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
15	Setup		#comment
16	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): e) Hide a granule.		#comment
17	Note the current time as t0.		
18	Hide granule g3 in AIM by hacking the database: update AmGranule set deletefromarchive = 'H' where granuleid = ${GRANULEID}		
19	Verification		#comment
20	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
21	Verify the TCP proxy log shows a single PUT for granule g5 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
22	Verify the TCP proxy log shows that the PUT request includes granule g5's full metadata.		
23	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
24	Verify granule g5's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g5.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g5.xml		
25	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = true c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
26	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
27	a) Visible = true xpath /Granule/Visible g5.xml	<Visible>true</Visible>	
28	b) Orderable = true xpath /Granule/Orderable g5.xml	<Orderable>true</Orderable >	

#	Action	Expected Result	Notes
29	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g5.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = <GRANULEID>;	
30	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g5.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate> where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = <GRANULEID>;	
31	V-6 Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		#comment
32	Verify granule g5's exported metadata has a restriction flag value of 255: xpath //RestrictionFlag g5.xml	<RestrictionFlag>255</RestrictionFlag>	

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		

	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was		

				performed.		

1.17 Nominal Granule Export[S-2f]: Automatic Export: Restrict Granule (ECS-ECSTC-16)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g6's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g6's collection is configured to be public on ingest.		

#	Action	Expected Result	Notes
4	Ensure granule g6's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g6's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
9	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
10	Setup		#comment
11	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): f) Restrict a granule.		#comment
12	dsmdrestrictionflag contains valid values for restriction_flag. New values can be added to dsmdrestrictionflag as long as they are between 0 and 254		#comment
13	Note the current time as t0.		
14	Find a restriction flag between 0 and 254: select * from dsmdrestrictionflag where restrictionflag between 0 and 254; Or, create one: insert into dsmdrestrictionflag values (6, 'BE_82_01 Crit 40 S-2f')		
15	Restrict granule g6 in AIM: update dsmdgranulerestriction set restrictionflag = ${RESTRICTION_FLAG} where granuleid = ${GRANULEID}		
16	Verification		#comment

#	Action	Expected Result	Notes
17	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
18	Verify the TCP proxy log shows a single PUT for granule g6 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
19	Verify the TCP proxy log shows that the PUT request includes granule g6's full metadata.		
20	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
21	Verify granule g6's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g6.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g6.xml		
22	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = true c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
23	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
24	a) Visible = true xpath /Granule/Visible g6.xml	<Visible>true</Visible>	
25	b) Orderable = true xpath /Granule/Orderable g6.xml	<Orderable>true</Orderable >	
26	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g6.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
27	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g6.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query <br	

#	Action	Expected Result	Notes
		/>select lastupdate from amgranule where granuleid = ${GRANULEID};	
28	V-7 Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		#comment
29	Verify granule s6's exported metadata includes the restriction flag set in the test: xpath //RestrictionFlag g6.xml	<RestrictionFlag>${RESTRICTION_FLAG}</RestrictionFlag>	

TEST DATA:
See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		

	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.18 Nominal Granule Export[S-2g]: Automatic Export: Unrestrict Granule (ECS-ECSTC-17)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and		

				granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g7's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g7's collection is configured to be public on ingest.		
4	Ensure granule g7's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g7's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g7 is in AIM: select shortname, versionid,		

#	Action	Expected Result	Notes
	granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure granule g7 has a restriction flag: select * from dsmdgranulerestriction where granuleid = ${GRANULEID} If the granule has no restriction flag, add one: Find a restriction flag between 0 and 254: select * from dsmdrestrictionflag where restrictionflag between 0 and 254; Or, create one: insert into dsmdrestrictionflag values (7, 'BE_82_01 Crit 40 S-2g') Restrict granule g7 in AIM: update dsmdgranulerestriction set restrictionflag = ${RESTRICTION_FLAG} where granuleid = ${GRANULEID}		
8	Ensure ECHO has granule g7's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
9	Ensure ECHO has granule g7's restriction flag: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} > g7.xml /tools/common/test/BE_82_01/bin/xpath '//RestrictionFlag' g7.xml	<RestrictionFlag>${RESTRICTION_FLAG}</RestrictionFlag>	
10	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
11	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
12	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
13	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
14	Setup		#comment
15	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): g) Unrestrict a granule.		#comment
16	Note the time as t0.		

#	Action	Expected Result	Notes
17	Unrestrict granule g7 in AIM: delete from dsmdgranulerestriction where granuleid = ${GRANULEID}		
18	Verification		#comment
19	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
20	Verify the TCP proxy log shows a single PUT for granule g7 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
21	Verify the TCP proxy log shows that the PUT request includes granule g7's full metadata.		
22	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
23	Verify granule g7's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g7.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g7.xml		
24	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
25	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
26	a) Visible = true xpath /Granule/Visible g7.xml	<Visible>true</Visible>	
27	b) Orderable = true xpath /Granule/Orderable g7.xml	<Orderable>true</Orderable >	
28	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g7.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
29	d) LastUpdate = The last update time of the granule recorded in the AIM	<LastUpdate>YYYY-MM-	

#	Action	Expected Result	Notes
	database. xpath /Granule/LastUpdate g7.xml	DDTHH:mm:SSZ</LastUpdate> where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = ${GRANULEID};	
30	V-8 Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		#comment
31	Verify granule g7's exported metadata has no RestrictionFlag element: xpath //RestrictionFlag g7.xml		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database.		

				d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.19 Nominal Granule Export[S-2h]: Automatic Export: QAUpdate granule. (ECS-ECSTC-18)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g8's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g8's collection is configured to be public on ingest.		
4	Ensure granule g8's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname =		

#	Action	Expected Result	Notes
	<SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g8's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g8 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure ECHO has granule g8's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
8	Ensure QAUU is configured with a test site (e.g., BMGT82): Append the following lines to /custom/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties if the are not there: BMGT82_EMAIL_FROM_ADDRESSES=labuser@edn.ecs.nasa.gov BMGT82_EMAIL_REPLY_ADDRESS=labuser@edn.ecs.nasa.gov BMGT82_NOTIFICATION_ON_SUCCESS=N		
9	Ensure granule g8's collection is eligible for QA updates: select * from dsqamutesdtsite where shortname = '${SHORTNAME}' and site = 'BMGT82' If no such row exists, add it: insert into dsqamutesdtsite values ('${SHORTNAME}', 'BMGT82')		
10	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
11	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
12	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
13	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		

#	Action	Expected Result	Notes
14	Setup		#comment
15	Use the Ruby script /tools/common/test/BE_82_01/common/bin/xpath to find XML elements.		#comment
16	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): h) Perform a QAUpdate on a granule.		#comment
17	See the QAUU 609 Section 4.8.9: http://edhs1.gsfc.nasa.gov/waisdata/eed/pdf/609eed001_Rev01_File_7.pdf		#comment
18	Find granule g8's metadata file: select p.path || '/' || f.archivemetfilename from ammetadatafile f join dsmdxmlpath p on f.archivepathid = p.archivepathid where f.granuleid = ${GRANULEID}		
19	Find the name of a measured parameter to update: xpath '//MeasuredParameterContainer/ParameterName/text()' ${g8_XML_FILE}		
20	Find the measured parameter's science quality flag value: xpath "//MeasuredParameterContainer[ParameterName='${PARAMETER_NAME}']/QAFlags/ScienceQualityFlag/text()" ${g8_XML_FILE}		
21	Select a different science quality flag value from the list of valid values in the properties file: grep VALID_SCIENCE_QA_FLAGS /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
22	Create a QA update request file named according to the the QAUU 609: <MODE>_<Site>_QAUPDATE<description>.<YYYY><MM><DD><hh><mm><ss> E.g., OPS_BMGT82_QAUPDATE_BE_82_01_Crit_50_S-2h.20130522231600		
23	Write a QA update request to update a science flag, using a GranuleUR (replace <TAB> with an actual tab character): From ${site} begin QAMetadataUpdate Science GranuleUR ${shortname}<TAB>${versionid}<TAB>${granuleur}<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation} end QAMetadataUpdate		
24	Copy the QA update request file to the QA request directory configured in the properties file: sed -n 's/QA_REQUEST_DIR *= *//p' /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
25	Note the current time as t0.		

#	Action	Expected Result	Notes
26	Run QAUU, specifying the filename (but not the path) if there are other request files in the request directory: EcAmQAUUStart ${MODE} -file ${REQUEST_FILE} -noprompt		
27	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
28	Verify the TCP proxy log shows a single PUT for granule g8 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
29	Verify the TCP proxy log shows that the PUT request includes granule g8's full metadata.		
30	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
31	Verify granule g8's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g8.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g8.xml		
32	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
33	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
34	a) Visible = true xpath /Granule/Visible g8.xml	<Visible>true</Visible>	
35	b) Orderable = true xpath /Granule/Orderable g8.xml	<Orderable>true</Orderable >	
36	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g8.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
37	d) LastUpdate = The last update time of the granule recorded in the AIM	<LastUpdate>YYYY-MM-	

#	Action	Expected Result	Notes
	database. xpath /Granule/LastUpdate g8.xml	DDTHH:mm:SSZ</LastUpdate> where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = ${GRANULEID};	
38	V-9 Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		#comment
39	Verify granule g8's exported metadata contains the updated QA flag value: xpath "//MeasuredParameter[ParameterName='${measured_parameter_name}']/QAFlags/ScienceQualityFlag/text()" g8.xml		
40	Verify granule g8's exported metadata contains the updated QA flag explanation: xpath "//MeasuredParameter[ParameterName='${measured_parameter_name}']/QAFlags/ScienceQualityFlagExplanation/text()" g8.xml		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		

	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.20 Nominal Granule Export[S-2i]: Automatic Export: Publish Granule (ECS-ECSTC-19)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g9's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g9's collection is configured to NOT be public on ingest.		
4	Ensure granule g9's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname =		

#	Action	Expected Result	Notes
	<SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g9's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g9 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure granule g9 is not public: select publishtime from amgranule where granuleid = ${GRANULEID} If publishtime is not null, unpublish the granule: EcDlUnpublishStart.pl --mode ${MODE} -g ${GRANULEID}		
8	Ensure ECHO has granule g9's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
9	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
12	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
13	Setup		#comment
14	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): i) Publish a granule in the datapool.		#comment
15	Note the current time as t0.		
16	Publish granule g9: EcDlPublishStart <MODE> -ecs -g ${GRANULEID}		

#	Action	Expected Result	Notes
17	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
18	Verify the TCP proxy log shows a single PUT for granule g9 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
19	Verify the TCP proxy log shows that the PUT request includes granule g9's full metadata.		
20	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
21	Verify granule g9's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g9.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g9.xml		
22	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
23	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
24	a) Visible = true xpath /Granule/Visible g9.xml	<Visible>true</Visible>	
25	b) Orderable = true xpath /Granule/Orderable g9.xml	<Orderable>true</Orderable >	
26	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g9.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
27	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g9.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query <br	

#	Action	Expected Result	Notes
		/>select lastupdate from amgranule where granuleid = ${GRANULEID};	
28	V-10 Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		#comment
29	Verify that granule g9's exported metadata contains URLs for its science file(s): xpath '//OnlineAccessURL/URL/text()' granule.xml		
30	Verify the URL to the science file is valid by downloading it and comparing it to the science file in the data pool: curl -O ${SCIENCE_URL} diff ${SCIENCE_FILE} /datapool/${MODE}/user${SCIENCE_URL_PATH} where ${SCIENCE_URL_PATH} is the part of the ${SCIENCE_URL} after http://${HOST}.		
31	Verify that granule g9's exported metadata contains URLs for its metadata and associated ancillary file(s) (browse, QA, PH, HDF Map): xpath '//OnlineResource/URL/text()' granule.xml		
32	Verify each OnlineResource URL is valid by downloading it and comparing it to the file in the data pool: curl -O ${URL} diff ${DOWNLOADED_FILE} /datapool/${MODE}/user${URL_PATH} where ${URL_PATH} is the part of the ${URL} after http://${HOST}.		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		

	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the		

				collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.21 Nominal Granule Export[S-2j]: Automatic Export: Unpublish Granule (ECS-ECSTC-20)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g10's collection is installed. E.g., the DPL Ingest GUI shows		

#	Action	Expected Result	Notes
	the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g10's collection is configured to be public on ingest.		
4	Ensure granule g10's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g10's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g10 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure granule g10 is public: select publishtime from amgranule where granuleid = ${GRANULEID} If publishtime is null, publish the granule: EcDlPublishUtilityStart ${MODE} -ecs -g ${GRANULEID}		
8	Ensure ECHO has granule g10's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
9	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
12	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
13	Setup		#comment

#	Action	Expected Result	Notes
14	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): j) Unpublish a granule in the datapool.		#comment
15	Unpublish granule g10: EcDlUnpublishStart.pl -mode ${MODE} -g ${GRANULEID}		
16	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
17	Verify the TCP proxy log shows a single PUT for granule g10 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
18	Verify the TCP proxy log shows that the PUT request includes granule g10's full metadata.		
19	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
20	Verify granule g10's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g10.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g10.xml		
21	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
22	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
23	a) Visible = true xpath /Granule/Visible g10.xml	<Visible>true</Visible>	
24	b) Orderable = true xpath /Granule/Orderable g10.xml	<Orderable>true</Orderable >	
25	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g10.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid =	

#	Action	Expected Result	Notes
		${GRANULEID};	
26	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g10.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate> where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = ${GRANULEID};	
27	V-11 Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		#comment
28	Verify granule g10's exported metadata contains no OnlineAccessURLs: xpath '//OnlineAccessURL' g8.xml		
29	Verify granule g10's exported metadata contains no OnlineResource URLs, except BROWSE: xpath '//OnlineResource' g8.xml > all.txt xpath "//OnlineResource[Type='BROWSE']" g8.xml > browse.txt Verify no differences are found: diff all.txt browse.txt		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule		

				metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.22 Nominal Granule Export[S-2k]: Automatic Export: Link to Browse (ECS-ECSTC-21)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g11's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g11's collection is configured to be public on ingest.		
4	Ensure granule g11's collection is enabled for collection and granule export: update bg_collection_configuration set		

#	Action	Expected Result	Notes
	granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g11's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g11 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure granule g11 is unassociated with any browse granules: select * from ambrowsegranulexref where granuleid = ${GRANULEID} If the granule is associated with a browse granule, physically delete it and ingest it again, without associated browse.		
8	Ensure ECHO has granule g11's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
9	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
12	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
13	Setup		#comment
14	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): k) Link a granule to a browse granule.		#comment
15	Ingest browse granule b1, which has been prepared to link with granule g11.		
16	Verification		#comment

#	Action	Expected Result	Notes
17	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
18	Verify the TCP proxy log shows a single PUT for granule g11 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
19	Verify the TCP proxy log shows that the PUT request includes granule g11's full metadata.		
20	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
21	Verify granule g11's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g11.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g11.xml		
22	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
23	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
24	a) Visible = true xpath /Granule/Visible g11.xml	<Visible>true</Visible>	
25	b) Orderable = true xpath /Granule/Orderable g11.xml	<Orderable>true</Orderable >	
26	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g11.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
27	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g11.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query <br	

#	Action	Expected Result	Notes
		/>select lastupdate from amgranule where granuleid = ${GRANULEID};	
28	V-12 Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs. Verify that the TCP proxy shows that the metadata exported for the browse granule contains a browse linkage URL.		#comment
29	Verify granule g11's exported metadata contains a browse linkage URL: xpath "//OnlineResource[Type='BROWSE']" g11.xml		
30	Verify the browse linkage URL is valid by downloading the browse file and comparing it to the file in the data pool: xpath "//OnlineResource[Type='BROWSE']/URL/text()" g11.xml curl -O ${BROWSE_URL} diff ${BROWSE_FILE} ${BROWSE_URL_PATH} where ${BROWSE_URL_PATH} is the part of ${BROWSE_URL} after http://${HOST}.		
31			

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		

	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.23 Nominal Granule Export[S-2l]: Automatic Export: Unlink from Browse (ECS-ECSTC-22)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use the Ruby script "/tools/common/test/BE_82_01/bin/xpath" to search XML files.		#comment
3	Ensure granule g12's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
4	Ensure the DPL Ingest GUI shows granule g12's collection is configured to be public on ingest.		

#	Action	Expected Result	Notes
5	Ensure granule g12's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
6	Ensure ECHO has granule g12's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
7	Ensure granule g12 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
8	Ensure granule g12 has an associated browse (b2): select bf.onlinedirectorypath || bf.onlinefilename from ambrowsegranulexref bx join ambrowseonlinefile bf on bx.browseid = bf.browseid where bx.granuleid = ${GRANULEID}		
9	Ensure ECHO has granule g12's metadata with the associated browse b2 as an OnlineResource: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} > g12.xml xpath "//OnlineResource[Type='BROWSE']/URL/text()" g12.xml If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}	The BROWSE URL should end in the browse path found in the previous step.	
10	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
11	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
12	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
13	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
14	Setup		#comment
15	S-2 For one of the collections in S-1 (or another collection which has ECS		#comment

#	Action	Expected Result	Notes
	metadata, and is enabled for collection and granule export): L) Unlink a granule from a browse granule.		
16	Note the current time as t0.		
17	Unlink the browse granule b2 in AIM: delete from AmBrowseGranuleXref where BrowseId = ${b2_BROWSEID}		
18	Verification		#comment
19	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
20	Verify the TCP proxy log shows a single PUT for granule g12 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
21	Verify the TCP proxy log shows that the PUT request includes granule g12's full metadata.		
22	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
23	Verify granule g12's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g12.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g12.xml		
24	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
25	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
26	a) Visible = true xpath /Granule/Visible g12.xml	<Visible>true</Visible>	
27	b) Orderable = true xpath /Granule/Orderable g12.xml	<Orderable>true</Orderable >	
28	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g12.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from	

#	Action	Expected Result	Notes
		amgranule where granuleid = ${GRANULEID};	
29	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g12.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate> where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = ${GRANULEID};	
30	V-13 Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		#comment
31	Verify granule g12's exported metadata includes no browse linkage URLs: xpath "//OnlineResource[Type='BROWSE']" g12.xml	Should find no OnlineResource elements.	

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements:		

				a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.24 Nominal Granule Export[S-2n]: Automatic Export: Move Collection (ECS-ECSTC-23)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g13's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g13's collection is configured to be public on ingest.		
4	Ensure granule g13's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname =		

#	Action	Expected Result	Notes
	<SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g13's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g13 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure ECHO has granule g13's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
8	Find the number of granules in g13's collection: select count(1) from amgranule where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
9	Find the number of associated browse granules in g13's collection: select count(1) from amgranule g join ambrowsegranulexref bx on g.granuleid = bx.granuleid where g.shortname = ${SHORTNAME} and g.versionid = ${VERSIONID}		
10	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
11	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
12	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
13	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
14	Setup		#comment
15	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): n) Move a collection.		#comment

#	Action	Expected Result	Notes
16	Note the current time as t0.		
17	Start a new proxy log that will have only requests created by the move: Stop the TCP proxy. Move the log to an archived name (e.g,. by appending a timestamp). Start the proxy.		
18	Move granule g13's collection different filesystem (e.g., from ${FS_OLD} to ${FS_NEW}): EcDlMoveCollection.pl ${MODE} -shortname ${SHORTNAME} -versionid ${VERSIONID} -sourcefs ${FS_OLD} -targetfs ${FS_NEW} -verbose		
19	Ensure that the move collection was successful and all the files, browse links are moved from /datapool/${MODE}/user/${FS_OLD}/${GROUP}/${SHORTNAME}.${VERSIONID} to /datapool/${MODE}/user/${FS_NEW}/${GROUP}/${SHORTNAME}.${VERSIONID}		
20	Ensure that the files in the hidden directory structure are moved from /datapool/${MODE}/user/${FS_OLD}/.orderdata/${ENCRYPTED_COLL_DIR} to /datapool/${MODE}/user/${FS_NEW}/.orderdata/${ENCRYPTED_COLL_DIR}		
21	After activity in the TCP proxy log has stopped, save the log for analysis.		
22	Verification		#comment
23	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
24	Verify the TCP proxy log shows a single PUT for granule g13 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
25	Verify the TCP proxy log shows that the PUT request includes granule g13's full metadata.		
26	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
27	Verify granule g13's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g13.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g13.xml		

#	Action	Expected Result	Notes
28	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
29	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
30	a) Visible = true xpath /Granule/Visible g13.xml	<Visible>true</Visible>	
31	b) Orderable = true xpath /Granule/Orderable g13.xml	<Orderable>true</Orderable >	
32	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g13.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
33	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g13.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = ${GRANULEID};	
34	V-15 Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		#comment
35	Verify the number of granule PUT requests in the log matches the expected count: grep '^PUT .*/granules/' tcp.log | wc -l		
36	Verify granule metadata was exported for each PUT request: grep '<?xml [^>]*><Granule>' tcp.log | wc -l	The number of metadata items should match the number of PUTs.	
37	Verify the number of browse URLs matches the expected number: grep grep '<OnlineResource><URL>[^>]*</URL><Type>BROWSE</Type>' tcp.log | wc -l		
38	Verify granule g13's exported metadata includes the new filesystem in its		

#	Action	Expected Result	Notes
	online access URL: xpath '//OnlineAccessURL/URL/text()' g13.xml For example, http://f4ftl01/${FS_NEW}/...		
39	Verify granule g13's exported metadata includes the new filesystem in its online resource URLs: xpath '//OnlineResource/URL/text()' g13.xml For example, http://f4ftl01/${FS_NEW}/...		

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		

	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		
	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.25 Nominal Granule Export[S-2o]: Automatic Export: XML Replacement (ECS-ECSTC-24)

DESCRIPTION:

	S	40	1	**[Nominal Granule Export]** Find two collections with ECS metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	40	2	For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory.		

				c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Perform a QAUpdate on a granule. i) Publish a granule in the datapool. j) Unpublish a granule in the datapool. k) Link a granule to a browse granule. l) Unlink a granule from a browse granule. m) Change the collection to which a granule belongs. n) Move a collection. o) Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure granule g14's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows granule g14's collection is configured to be public on ingest.		
4	Ensure granule g14's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure ECHO has granule g14's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
6	Ensure granule g14 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure ECHO has granule g14's metadata: curl -k -H Echo-		

#	Action	Expected Result	Notes
	Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
8	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
11	Ensure the BMGT Auto driver is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, start the Auto driver: ./EcBmBMGTAutoStart $MODE		
12	Setup		#comment
13	S-2 For one of the collections in S-1 (or another collection which has ECS metadata, and is enabled for collection and granule export): o) Perform XML replacement on a granule.		#comment
14	Locate granule g14's XML file in the small file archive: select p.path || '/' || m.archivemetfilename from ammetadatafile m join dsmdxmlpath p on m.archivepathid = p.archivepathid where m.granuleid = ${GRANULEID}		
15	Copy granule g14's XML file to a temporary working directory, using the following filename pattern: ${SHORTNAME}.${VERSIONID}.${GRANULEID}.xml		
16	Change the XML such that it remains valid. E.g., add a new MeasuredParameterContainer within the MeasuredParameter block: <MeasuredParameterContainer> <ParameterName>BE_82_01 Crit 40 S-2o</ParameterName> <QAStats> <QAPercentMissingData>0</QAPercentMissingData> </QAStats> <QAFlags> <AutomaticQualityFlag>Suspect</AutomaticQualityFlag> <AutomaticQualityFlagExplanation>Whatever.</AutomaticQualityFlagExplanation> <OperationalQualityFlag>Passed</OperationalQualityFlag> <OperationalQualityFlagExplanation>Why		

#	Action	Expected Result	Notes
	not?</OperationalQualityFlagExplanation> <ScienceQualityFlag>Suspect</ScienceQualityFlag> <ScienceQualityFlagExplanation>Okay</ScienceQualityFlagExplanation> </QAFlags> </MeasuredParameterContainer>		
17	Run the XML Replacement utility on the modified metadata file EcDsAmXruStart ${MODE} -xmlfile ${FILENAME}		just a note - BMGT currently allows configuration to get the metadata from datapool or archive. This test illustrates why we will always need to get it from archive, since I dont believe XRU will propagate to datapool. So we should probably remove that option, but if this test fails, this could be the reason 2/26/2013 -- Goff, Timothy
18	Verification		#comment
19	V-2 Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
20	Verify the TCP proxy log shows a single PUT for granule g14 after time t0. E.g., PUT /catalog-rest/providers/${PROVIDERID}/granules/SC%3MYD14.005%3A123456 HTTP/1.1		
21	Verify the TCP proxy log shows that the PUT request includes granule g14's full metadata.		
22	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		#comment
23	Verify granule g14's exported metadata validates against the ECHO granule metadata schema: xmllint --noout --schema /path/to/Granule.xsd g14.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g g14.xml		
24	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule		#comment

#	Action	Expected Result	Notes
	recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
25	Verify that the exported metadata has the following elements, using the Ruby script /tools/common/test/BE_82_01/bin/xpath:		#comment
26	a) Visible = true xpath /Granule/Visible g14.xml	<Visible>true</Visible>	
27	b) Orderable = true xpath /Granule/Orderable g14.xml	<Orderable>true</Orderable >	
28	c) InsertTime = The insert time of the granule recorded in the AIM database xpath /Granule/InsertTime g14.xml	<InsertTime>YYYY-MM-DDTHH:mm:SSZ</InsertTime> where the date time string return is equal to what is returned from the query to the database select archivetime from amgranule where granuleid = ${GRANULEID};	
29	d) LastUpdate = The last update time of the granule recorded in the AIM database. xpath /Granule/LastUpdate g14.xml	<LastUpdate>YYYY-MM-DDTHH:mm:SSZ</LastUpdate&gt ; where the date time string returned is equal to what is returned from the query select lastupdate from amgranule where granuleid = ${GRANULEID};	
30	V-16 Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		#comment
31	Verify granule g14's exported metadata includes the change applied by the XML replacement. E.g., if a MeasuredParameterContainer was added, as in the example above, xpath "//MeasuredParameters/MeasuredParameter[ParameterName='BE_82_01 Crit 40 S-2o']" g14.xml The contents of the <MeasuredParameter> block should match the added <MeasuredParameterContainer>.		
32			

TEST DATA:

See Test Case 603.

EXPECTED RESULTS:

	V	40	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but no granules belonging to other collections sharing a short name but with a different version ID).		
	V	40	2	Verify that the operations in S-2, except subclauses b – d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	40	3	Verify that the operations in S-2 subclauses b – d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but containing no granule metadata in the request body.		
	V	40	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d, validates against the ECHO granule metadata schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd).		
	V	40	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b – d has the following elements: a) Visible = true b) Orderable = false c) InsertTime = The insert time of the granule recorded in the AIM database. d) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	40	6	Verify that the metadata generated by the action in S-2 subclause e contains a Restriction Flag value of 255.		
	V	40	7	Verify that the metadata generated by the action in S-2 subclause f contains the restriction flag value that was set in the database for that granule.		
	V	40	8	Verify that the metadata generated by the action in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) the restriction flag element is absent.		
	V	40	9	Verify that the metadata generated by the action in S-2 subclause h contains the QA values reflecting the update which was performed.		
	V	40	10	Verify that the metadata generated by the action in S-2 subclause i contains URLs for the science and metadata files, as well as any other ancillary files (e.g. browse) associated with the affected granule.		
	V	40	11	Verify that the metadata generated by the action in S-2 subclause j contains no datapool URLs.		

	V	40	12	Verify that the metadata generated by the action in S-2 subclause k contains a browse linkage URL, in addition to any datapool URLs.		
	V	40	13	Verify that the metadata generated by the action in S-2 subclause l contains no browse linkage URLs.		
	V	40	14	Verify that the metadata generated by the action in S-2 subclause m contains a reference to the newly assigned collection.		
	V	40	15	Verify that the action performed in S-2 subclause n results in the export of full granule metadata for every granule in the collection affected.		
	V	40	16	Verify that the metadata generated by the action in S-2 subclause o contains XML reflecting the replacement which was performed.		

1.26 QA Granule URLs (ECS-ECSTC-25)

DESCRIPTION:

	S	70	1	**[QA Granule URLs]** Find a collection whose granules are linked to public QA granules.		
	S	70	2	Insert into the hidden datapool a science granule in the selected collection. Map the granule to a public QA granule. This granule will be referred to as granule A.		
	S	70	3	Insert into the public datapool a science granule in the selected collection. Do not map this granule to any QA granules. This granule will be referred to as granule B.		
	S	70	4	Find a granule which is public, and is linked to a non-public QA granule. This granule will be referred to as granule C. Ensure that the insert of this granule has already been exported.		
	S	70	5	Wait for the automatic export of the granule inserts in S-2, S-3 and S-4 to complete.		
	S	70	6	Publish granule A.		
	S	70	7	Link granule B to a public QA granule.		
	S	70	8	Publish the QA granule linked to granule C.		

	S	70	9	Wait for the automatic exports of the granule updates in S-6, S-7 and S-8 to complete.		
	S	70	10	Unpublish granule A.		
	S	70	11	Unpublish the QA granule associated with granule B.		
			12	Unlink Granule C from its linked QA granule.		
	S	70	13	Wait for the automatic export of the granules in S-10, S-11, and S-12 to complete.		
	S	70	14	In the database, link granule C to a new QA granule which is not public.		
	S	70	15	Wait for the automatic exports caused by the modification in S-14 to complete (if there are any events caused by the update).		
	S	70	16	Publish the QA granule linked to granule C.		
	S	70	17	Wait for the automatic exports caused by the modification in S-16 to complete.		
	S	70	18	Perform a manual export of granules A, B, and C.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure collection C1 is installed. E.g., the DPL Ingest GUI shows the		

#	Action	Expected Result	Notes
	collection as a configured datatype.		
6	Ensure collection C1 is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly abled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure the DPL Ingest GUI shows QA.001 is configured to be public on ingest.		
9	Ensure science granules A and B are not in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid in ('${A_LOCALGRANULEID}', '${B_LOCALGRANULEID}') If needed, physically delete the granules (bulk delete, unpublish, deletion cleanup).	Should return no rows.	
10	Ensure science granule C is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = '${C_LOCALGRANULEID}' If needed, ingest the granule into the public data pool.	Should return 1 row.	
11	Ensure granule C is linked to a non-public QA granule: select g.shortname, g.versionid, g.granuleid, g.publishtime, q.shortname, q.versionid, q.granuleid "qa_id", q.publishtime "qa_publishtime" from amgranule g join amqagranulexref qx on g.granuleid = qx.scienceid join amgranule q on qx.qaid = q.granuleid where g.granuleid = ${C_GRANULEID} If needed, unpublish the linked QA granule.	Should return 1 row. qa_publishtime should be null.	
12	Ensure ECHO has granule C's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: <br		

#	Action	Expected Result	Notes
	/>EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
13	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
14	Setup		#comment
15	S-1 Find a collection whose granules are linked to public QA granules.		#comment
16	Ensure that there are linkage files for at least 4 QA granules to be ingested and linked to the 3 science granules.		
17	S-2 Insert into the hidden datapool a science granule in the selected collection. Map the granule to a public QA granule. This granule will be referred to as granule A.		#comment
18	Using the DPL Ingest GUI, configure collection C1 to NOT be public on ingest.		
19	Ingest granule A (into the hidden data pool).		
20	Ingest granule A's linked QA granule (into the public data pool).		
21	S-3 Insert into the public datapool a science granule in the selected collection. Do not map this granule to any QA granules. This granule will be referred to as granule B.		#comment
22	Using the DPL Ingest GUI, configure collection C1 to be public on ingest.		
23	Ingest granule B (into the public data pool).		
24	S-4 Find a granule which is public, and is linked to a non-public QA granule. This granule will be referred to as granule C. Ensure that the insert of this granule has already been exported.		#comment
25	Ingested and exported in the pre-conditions.		
26	S-5 Wait for the automatic export of the granule inserts in S-2, S-3 and S-4 to complete.		#comment
27	Verify that the TCP proxy log shows that HTTP PUT requests are generated for granules A, B. Granule C should have been exported already.		
28	V-1 Verify that the export of the insert of granule A in S-5 does not contain any URLs.		#comment
29	Save granule A's exported metadata from the TCP proxy log to an XML file.		
30	Verify that the TCP proxy shows that the metadata exported for granule_A does not contain any URLs xpath /Granule/OnlineAccessURLs granule_A.xml should not have any URLs xpath /Granule/OnlineResources granule_A.xml should not have any URLs		
31	V-2 Verify that the export of the insert of granules B and C in S-5 contains only URLs for the science, metadata, and browse (if applicable)		#comment

#	Action	Expected Result	Notes
	files, but not for a QA granule.		
32	Verify that the TCP proxy shows that the metadata exported for granule_B contains science granule URL granule_B xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
33	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any QA URLs xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'Quality Assurance'		
34	Note: granule C was exported before S-4.		#comment
35			#comment
36	S-6 Publish granule A.		#comment
37	EcDlPublishStart $MODE -ecs -g ${granule_A}		
38	S-7 Ingest a public QA granule which is linked to granule B.		#comment
39	Ingest a QA granule QA_B linked to granule B - default publishing on		
40	S-8 Publish the QA granule linked to granule C.		#comment
41	EcDlPublishStart $MODE -ecs -g ${QA_C}		
42	S-9 Wait for the automatic exports of the granule updates in S-6, S-7 and S-8 to complete.		#comment
43	Wait for the TCP proxy log to show that HTTP PUT requests are exported for granule_A, granule_B, and granule_C.		
44	V-3 Verify that the export of metadata for granules A B, and C in S-9 contain the correct URL for the associated QA.		#comment
45	Save granule A's exported metadata from the TCP proxy log.		
46	Verify granule A's exported metadata contains an OnlineResourceURL with <Type>Quality Assurance</Type>: xpath /Granule/OnlineResources granule_A.xml	Should have block like this: <OnlineResource> <URL>${QA_granule_URL}</URL> <Type>Quality Assurance</Type>	
47	Save granule B's exported metadata from the TCP proxy log.		
48	Verify granule B's exported metadata contains an OnlineResourceURL with <Type>Quality Assurance</Type>: xpath /Granule/OnlineResources granule_B.xml	Should have block like this: <OnlineResource> <URL>${QA_granule_URL}</URL> <Type>Quality Assurance</Type>	

#	Action	Expected Result	Notes
49	Save granule C's exported metadata from the TCP proxy log.		
50	Verify granule C's exported metadata contains an OnlineResourceURL with <Type>Quality Assurance</Type>: xpath /Granule/OnlineResources granule_C.xml	Should have block like this: <OnlineResource> <URL>${QA_granule_URL}</URL> <Type>Quality Assurance</Type>	
51	Verify that the URLs of type "Quality Assurance" for granules A, B, and C all point to the correct URL of the associated QA Granule. The URL may or may not be accessible, depending on the mode and the configuration of the FTP or HTTP server, but the correctness can be determined by turning the URL into a local file path, e.g. http://f4ftl01//FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml becomes /datapool/${MODE}/user/FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml		
52			#comment
53	S-10 Unpublish granule A.		#comment
54	EcDlUnpublishStart.pl -mode $MODE -g ${granule_A}		
55	S-11 Unpublish the QA granule associated with granule B.		#comment
56	EcDlUnpublishStart.pl -mode $MODE -g ${QA_B}		
57	S-12 Unlink Granule C from its linked QA granule in the database xref table.		#comment
58	delete from AmQaGranuleXref where qaid = ${QA_C_qaid}		
59	S-13 Wait for the automatic export of the granules in S-10, S-11, and S-12 to complete.		#comment
60	Wait for the TCP proxy log to show HTTP PUT requests for granules A, B, and C.		
61	V-4 Verify that the export of metadata for granule A in S-13 contains no OnlineAccess or OnlineResource URLs.		#comment
62	Verify that the TCP proxy shows that the metadata exported for granule_A does not contain any URLs xpath /Granule/OnlineAccessURLs granule_A.xml should not have any URLs xpath /Granule/OnlineResources granule_A.xml should not have any URLs		
63	V-5 Verify that the export of metadata for granule B in S-13 contains no QA granule URL.		#comment
64	Verify that the TCP proxy shows that the metadata exported for granule_B		

#	Action	Expected Result	Notes
	contains science granule URL granule_B xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
65	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any QA URLs xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'Quality Assurance'		
66	V-6 Verify that no export of metadata for granule C is spawned by the modifications in S-12 or S-14		
67	Verify that the TCP proxy log show no HTTP PUT requests for granule C.		
68			#comment
69	S-14 In the database xref table, link granule C to a new QA granule which is not public.		#comment
70	insert into AmQaGranuleXref (qaid, granuleid) values QA_C_qaid, granule_C_id		
71	S-15 Wait for the automatic exports caused by the modification in S-14 to complete (if there are any events caused by the update). Wait for an automatic dispatch wait time to pick up any updates		#comment
72	V-6 Verify that no export of metadata for granule C is spawned by the modifications in S-12 or S-14		#comment
73	Verify that the TCP proxy log shows no HTTP PUT requests for granule C.		
74			#comment
75	S-16 Publish the QA granule linked to granule C.		#comment
76	EcDlPublishStart $MODE -ecs -g ${QA_D}		
77	S-17 Wait for the automatic exports caused by the modification in S-16 to complete.		#comment
78	Wait for the TCP proxy log to show an HTTP PUT request for granule C.		If a QA granule that is associated with another science granule is used, HTTP PUTs will be exported for both the science granules that the QA granule is associated with them.
79	V-7 Verify that the export of metadata for granule C in S-17 contains the correct URL for the new QA granule linked in S-14.		#comment
80	Verify that the TCP proxy shows that the metadata exported for granule_C		

#	Action	Expected Result	Notes
	contains contains QA_D URL xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource element of type "Quality Assurance" and the url of QA_D		
81	Verify granule C's exported 'Quality Assurance' URL points to granule C's associated QA granule in the public data pool. Either follow the URL or convert the URL to a local pathname rooted at '/datapool/'.		
82			#comment
83	S-18 Perform a manual export of granules A, B, and C.		#comment
84	EcBmBMGTManualStart $MODE --g $GRANULEID_A,$GRANULEID_B,$GRANULEID_C		
85	V-8 Verify that the manual export of granule metadata in S-18 contains correct URLs for the associated QA granules for granule C, but no QA OnlineAccess or OnlineResource URLs for granule B, and no OnlineAccess or OnlineResource URLs for Granule A. Verify that granules B, and C have Science and Metadata URLs.		#comment
86	Verify that the TCP proxy shows that the metadata exported for granule_A contains no URLs xpath /Granule/OnlineAccessURLs granule_A.xml should not contain any URLs xpath /Granule/OnlineResources granule_A.xml should not contain any URLs		
87	Verify that the TCP proxy shows that the metadata exported for granule_C contains science data URLs and QA URLs xpath /Granule/OnlineAccessURLs granule_C.xml should contain the granule_C URL xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource TYPE element "Quality Assurance" and the url of QA_C		
88	Verify that the TCP proxy shows that the metadata exported for granule_B contains science granule URL granule_B xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
89	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any QA URLs xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'Quality Assurance'		
90	V-9 Verify that all QA URLS in the granule metadata have a URL type of “Quality Assurance”		#comment
91	xpath /Granule/OnlineResources granule_A.xml should contain an OnlineResource element of type "Quality		

#	Action	Expected Result	Notes
	Assurance" and the url of QA_A xpath /Granule/OnlineResources granule_B.xml should contain an OnlineResource element of type "Quality Assurance" and the url of QA_B xpath /Granule/OnlineResources/Type of granule_C.xml should contain an OnlineResource element of type "Quality Assurance" and the url of QA_D		

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
70		Testing QA granule linking and unlinking	1 collection (C1) with 3 science granules (A, B, C) and separate linkage files for 4 QA granules (q1, q2, q3, q4)	AE_DySno.002			/sotestdata/DROP_802/BE_82_01/Criteria/070/AE_DySno.002	

EXPECTED RESULTS:

	V	70	1	Verify that the export of the insert of granule A in S-5 does not contain any URLs.		
	V	70	2	Verify that the export of the insert of granules B and C in S-5 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a QA granule.		
	V	70	3	Verify that the export of metadata for granules A, B and C in S-9 contain the correct URL for the associated QA granules.		
	V	70	4	Verify that the export of metadata for granule A in S-13 contains no OnlineAccess or OnlineResource URLs.		

	V	70	5	Verify that the export of metadata for granules B and C in S-13 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a QA granule.		
	V	70	6	Verify that no export of metadata for granule C is spawned by the modifications in S-14, or that if an export is spawned, it contains either no QA URL or the URL prior to the modification in S-14		
	V	70	7	Verify that the export of metadata for granule C in S-17 contains the correct URL for the new QA granule linked in S-14.		
	V	70	8	Verify that the manual export of granule metadata in S-18 contains correct URLs for the associated QA granules for granules A, B, and C.		
	V	70	9	Verify that all QA URLS in the granule metadata have a URL type of “Quality Assurance”		

1.27 PH Granule URLs (ECS-ECSTC-26)

DESCRIPTION:

	S	71	1	**[PH Granule URLs]** Find a collection whose granules are linked to public PH granules.		
	S	71	2	Insert into the hidden datapool a science granule in the selected collection. Map the collection to a public PH granule. This granule will be referred to as granule A.		
	S	71	3	Insert into the public datapool a science granule in the selected collection. Do not map this granule to any PH granules. This granule will be referred to as granule B.		
	S	71	4	Find a granule which is public, and is linked to a non-public PH granule. This granule will be referred to as granule C. Ensure that the insert of this granule has already been exported.		
	S	71	5	Wait for the automatic export of the granule inserts in S-2, S-3 and S-4 to complete.		
	S	71	6	Publish granule A		
	S	71	7	Link granule B to a public PH granule.		
	S	71	8	Publish the PH granule linked to granule C.		

	S	71	9	Wait for the automatic exports of the granule updates in S-6, S-7 and S-8 to complete.		
	S	71	10	Unpublish granule A.		
	S	71	11	Unpublish the PH granule associated with granule B		
		71	12	Unlink Granule C from its linked PH granule.		
	S	71	13	Wait for the automatic export of the granules in S-10, S-11, and S-12 to complete.		
	S	71	14	In the database, link granule C to a new PH granule which is not public.		
	S	71	15	Wait for the automatic exports caused by the modification in S-14 to complete (if there are any events caused by the update).		
	S	71	16	Publish the PH granule linked to granule C.		
	S	71	17	Wait for the automatic exports caused by the modification in S-16 to complete.		
	S	71	18	Perform a manual export of granules A, B, and C.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure collection C1 is installed. E.g., the DPL Ingest GUI shows the		

#	Action	Expected Result	Notes
	collection as a configured datatype.		
6	Ensure collection C1 is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly abled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure the DPL Ingest GUI shows PH.001 is configured to be public on ingest.		
9	Ensure science granules A and B are not in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid in ('${A_LOCALGRANULEID}', '${B_LOCALGRANULEID}') If needed, physically delete the granules (bulk delete, unpublish, deletion cleanup).	Should return no rows.	
10	Ensure science granule C is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = '${C_LOCALGRANULEID}' If needed, ingest the granule into the public data pool.	Should return 1 row.	
11	Ensure granule C is linked to a non-public PH granule: select g.shortname, g.versionid, g.granuleid, g.publishtime, p.shortname, p.versionid, p.granuleid "ph_id", p.publishtime "ph_publishtime" from amgranule g join amphgranulexref qx on g.granuleid = qx.scienceid join amgranule p on qx.phid = p.granuleid where g.granuleid = ${C_GRANULEID} If needed, unpublish the linked PH granule.	Should return 1 row. ph_publishtime should be null.	
12	Ensure ECHO has granule C's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: <br		

#	Action	Expected Result	Notes
	/>EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
13	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
14	Setup		#comment
15	S-1 Find a collection whose granules are linked to public PH granules.		#comment
16	Ensure that there is a collection C1 with Science granules and public PH granules.		
17	S-2 Insert into the hidden datapool a science granule in the selected collection. Map the collection to a public PH granule. This granule will be referred to as granule A.		#comment
18	Insert a science granule granule A into hidden data pool (turn publishing off during ingest) Insert a PH granule PH_A linked to granule A. (with publishing turned on)		
19	S-3 Insert into the public datapool a science granule in the selected collection. Do not map this granule to any PH granules. This granule will be referred to as granule B.		#comment
20	Insert science granule granule B into public data pool (turn default publishing on)		
21	S-4 Find a granule which is public, and is linked to a non-public PH granule. This granule will be referred to as granule C. Ensure that the insert of this granule has already been exported.		#comment
22	Granule C was exported as part of the preconditions.		#comment
23	S-5 Wait for the automatic export of the granule inserts in S-2, S-3 and S-4 to complete.		#comment
24	Wait for the TCP proxy to show HTTP PUT requestsfor granules A and B.		
25	V-1 Verify that the export of the insert of granule A in S-5 does not contain any URLs.		#comment
26	Verify that the TCP proxy shows that the metadata exported for granule_A does not contain any URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should not have any URLs xpath /Granule/OnlineResources granule_A.xml should not have any URLs		
27	V-2 Verify that the export of the insert of granules B and C in S-5 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a PH granule.		#comment
28	Verify that the TCP proxy shows that the metadata exported for granule_B contains science granule URL granule_B: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science		

#	Action	Expected Result	Notes
	granule URL		
29	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any PH URLs: xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'Production History'		
30	Note: Granule C was already exported as part of the preconditions.		#comment
31	S-6 Publish granule A		#comment
32	EcDlPublishStart <MODE> -ecs -g <granule_A>		
33	S-7 Ingest a public PH granule which is linked to Granule B.		#comment
34	Ingest a PH granule PH_B linked to granule B -default publishing on		
35	S-8 Publish the PH granule linked to granule C.		#comment
36	EcDlPublishStart <MODE> -ecs -g <PH_C>		
37	S-9 Wait for the automatic exports of the granule updates in S-6, S-7 and S-8 to complete.		#comment
38	Wait the TCP proxy log to show HTTP PUT requests for granules A, B, and C.		
39	V-3 Verify that the export of metadata for granules A, B and C in S-9 contain the correct URL for the associated PH granules.		#comment
40	Verify that the TCP proxy log shows that the metadata exported for granule_A contains science data URLs and PH URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should contain the granule_A URL xpath /Granule/OnlineResources granule_A.xml should contain an OnlineResource element of type "Production History" and the URL of PH_A		
41	Verify that the TCP proxy log shows that the metadata exported for granule_B contains science data URLs and PH URLs: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the granule_B URL. xpath /Granule/OnlineResources granule_B.xml should contain an OnlineResource element of type "Production History" and the URL of PH_B.		
42	Verify that the TCP proxy log shows that the metadata exported for granule_C contains science data URLs and PH URLs: xpath /Granule/OnlineAccessURLs granule_C.xml should contain the granule_C URL xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource element of type "Production History" and the URL of PH_C		
43	Verify that the URLs of type "Production History" for granules		

#	Action	Expected Result	Notes
	A, B, and C all point to the correct URL of the associated PH Granule. The URL may or may not be accessible, depending on the mode and the configuration of the FTP or HTTP server, but the correctness can be determined by turning the URL into a local file path, e.g. http://f4ftl01//FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml becomes /datapool/<MODE>/user/FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml		
44	S-10 Unpublish granule A.		#comment
45	EcDlUnpublishStart.pl -mode <MODE> -g <granule_A>		
46	S-11 Unpublish the PH granule associated with granule B		#comment
47	EcDlUnpublishStart.pl -mode <MODE> -g <PH_B>		
48	S-12 Unlink Granule C from its linked PH granule in the database xref table.		#comment
49	delete from AmPhGranuleXref where phid = <PH_C_phid>		
50	S-13 Wait for the automatic export of the granules in S-10, S-11, and S-12 to complete.		#comment
51	Wait for the TCP proxy log to show HTTP PUT requests for granules A and B.		
52	V-4 Verify that the export of metadata for granule A in S-13 contains no OnlineAccess or OnlineResource URLs.		#comment
53	Verify that the TCP proxy log shows that the metadata exported for granule_A does not contain any URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should not have any URLs xpath /Granule/OnlineResources granule_A.xml should not have any URLs		
54	V-5 Verify that the export of metadata for granule B in S-13 contains no PH granule URL.		#comment
55	Verify that the TCP proxy log shows that the metadata exported for granule_B contains science granule URL granule_B: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
56	Verify that TCP proxy log shows that the metadata exported for granule_B does not contain any PH URLs: xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'Production History'		
57	V-6 Verify that no export of metadata for granule C is spawned by the		#comment

#	Action	Expected Result	Notes
	modifications in S-12 or S-14		
58	Verify that the TCP proxy does not show any HTTP PUT request is generated for granule C		
59	S-14 In the database xref table, link granule C to a new PH granule which is not public.		#comment
60	insert into AmPhGranuleXref (granuleid, phid) values granule_C_id, PH_D_id		
61	S-15 Wait for the automatic exports caused by the modification in S-14 to complete (if there are any events caused by the update).Verify that an Wait for automatic dispatch wait time to pick up any updates		#comment
62	V-6 Verify that no export of metadata for granule C is spawned by the modifications in S-14, or that if an export is spawned, it contains either no PH URL or the URL prior to the modification in S-14		#comment
63	Verify that the TCP proxy does not show any HTTP PUT request is generated for granule C		
64	S-16 Publish the PH granule linked to granule C.		#comment
65	EcDlPublishStart <MODE> -ecs -g <PH_D> -publish		
66	S-17 Wait for the automatic exports caused by the modification in S-16 to complete.		#comment
67	Verify that the TCP proxy shows that a HTTP PUT request was generated for granule C		
68	V-7 Verify that the export of metadata for granule C in S-17 contains the correct URL for the new PH granule linked in S-14.		#comment
69	Verify that the TCP proxy log shows that the metadata exported for granule_C contains contains PH_D URL: xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource element of type "Production History" and the URL of PH_D		
70	Verify that the URLs of type "Production History" for granule C points to the correct URL of the associated Production History. The URL may or may not be accessible, depending on the mode and the configuration of teh FTP or HTTP server, but the correctness can be determined by turning the URL into a local file path, e.g. http://f4ftl01//FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml becomes /datapool/<MODE>/user/FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml		

#	Action	Expected Result	Notes
71	S-18 Perform a manual export of granules A, B, and C.		#comment
72	EcBmBMGTManualStart $MODE --metg -g $GRANULEID_A,$GRANULEID_B,$GRANULEID_C		
73	V-8 Verify that the manual export of granule metadata in S-18 contains correct URLs for the associated PH granules for granule C, but no PH OnlineAccess or OnlineResource URLs for granule B, and no OnlineAccess or OnlineResource URLs for Granule A. Verify that granules B, and C have Science and Metadata URLs.		#comment
74	Verify that the TCP proxy log shows that the metadata exported for granule_A contains no science data URLs or PH URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should contain no URLs xpath /Granule/OnlineResources granule_A.xml should contain no URLs		
75	Verify that the TCP proxy log shows that the metadata exported for granule_C contains science data URLs and PH URLs: xpath /Granule/OnlineAccessURLs granule_C.xml should contain the granule_C URL xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource TYPE element "Production History" and the URL of QA_C		
76	Verify that the TCP proxy log shows that the metadata exported for granule_B contains science granule URL granule_B: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
77	Verify that TCP proxy log shows that the metadata exported for granule_B does not contain any PH URLs: xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'Production History'		
78	V-9 Verify that all PH URLS in the granule metadata have a URL type of “Production History”		#comment
79	xpath /Granule/OnlineResources granule_A.xml should contain an OnlineResource element of type "Production History" and the URL of PH_A xpath /Granule/OnlineResources granule_B.xml should contain an OnlineResource element of type "Production History" and the URL of PH_B xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource element of type "Production History" and the URL of PH_D		

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
71		Testing PH granule linking and unlinking	AE_DySno.002	1 collection 3 science granules 1 PH granule			/sotestdata/DROP_802/BE_82_01/Criteria/071/AE_DySno.002	

EXPECTED RESULTS:

	V	71	1	Verify that the export of the insert of granule A in S-5 does not contain any URLs.		
	V	71	2	Verify that the export of the insert of granules B and C in S-5 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a PH granule.		
	V	71	3	Verify that the export of metadata for granules A, B and C in S-9 contain the correct URL for the associated PH granules.		
	V	71	4	Verify that the export of metadata for granule A in S-13 contains no OnlineAccess or OnlineResource URLs.		
	V	71	5	Verify that the export of metadata for granules B and C in S-13 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a PH granule.		
	V	71	6	Verify that no export of metadata for granule C is spawned by the modifications in S-14, or that if an export is spawned, it contains either no PH URL or the URL prior to the modification in S-14		
	V	71	7	Verify that the export of metadata for granule C in S-17 contains the correct URL for the new PH granule linked in S-14.		
	V	71	8	Verify that the manual export of granule metadata in S-18 contains correct URLs for the associated PH granules for granules A, B, and C.		
	V	71	9	Verify that all PH URLS in the granule metadata have a URL type of “Production History”		

1.28 HDF Map Granule URLs (ECS-ECSTC-27)

DESCRIPTION:

	S	72	1	**[HDF Map Granule URLs]** Find a collection whose granules are linked to public HDF Map granules.		
	S	72	2	Insert into the hidden datapool a science granule in the selected collection. Map the collection to a public HDF Map granule. This granule will be referred to as granule A.		
	S	72	3	Insert into the public datapool a science granule in the selected collection. Do not map this granule to any HDF Map granules. This granule will be referred to as granule B.		
	S	72	4	Find a granule which is public, and is linked to a non-public HDF Map granule. This granule will be referred to as granule C. Ensure that the insert of this granule has already been exported.		
	S	72	5	Wait for the automatic export of the granule inserts in S-2, S-3 and S-4 to complete.		
	S	72	6	Publish granule A		
	S	72	7	Link granule B to a public HDF Map granule.		
	S	72	8	Publish the HDF Map granule linked to granule C.		
	S	72	9	Wait for the automatic exports of the granule updates in S-6, S-7 and S-8 to complete.		
	S	72	10	Unpublish granule A.		
	S	72	11	Unpublish the HDF Map granule associated with granule B		
		72	12	Unlink Granule C from its linked HDF Map granule.		
	S	72	13	Wait for the automatic export of the granules in S-10, S-11, and S-12 to complete.		
	S	72	14	Perform a manual export of granules A, B, and C.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure collection C1 is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure collection C1 is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly abled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure the DPL Ingest GUI shows HDF_MAP.001 is configured to be public on ingest.		
9	Ensure science granules A and B are not in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid in ('${A_LOCALGRANULEID}', '${B_LOCALGRANULEID}') If needed, physically delete the granules (bulk delete, unpublish, deletion cleanup).	Should return no rows.	
10	Ensure science granule C is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = '${C_LOCALGRANULEID}' If needed, ingest the granule into	Should return 1 row.	

#	Action	Expected Result	Notes
	the public data pool.		
11	Ensure granule C is linked to a non-public HDF_MAP granule: select g.shortname, g.versionid, g.granuleid, g.publishtime, h.shortname, h.versionid, h.granuleid "hdf_id", h.publishtime "hdf_publishtime" from amgranule g join amhdfmapgranulexref hx on g.granuleid = hx.scienceid join amgranule h on hx.hdfmapid = h.granuleid where g.granuleid = ${C_GRANULEID} If needed, unpublish the linked HDF_MAP granule.	Should return 1 row. hdf_publishtime should be null.	
12	Ensure ECHO has granule C's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
13	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
14	Setup		#comment
15	S-1 Find a collection whose granules are linked to public HDF Map granules.		#comment
16	S-2 Insert into the hidden datapool a science granule in the selected collection. Map the collection to a public HDF Map granule. This granule will be referred to as granule A.		#comment
17	Use the DPL Ingest GUI to configure the test collection to NOT be public on ingest and to NOT generate HDF Map granules on ingest.		
18	Ingest science granule A.		
19	Manually generate an HDF Map granule: EcAmInsertMapGenerationRequest.pl -mode $MODE -g ${granule_A}		
20	Publish the generated HDF Map granule: EcDlPublishStart $MODE -ecs -g ${granule_HDF_A}		
21	S-3 Insert into the public datapool a science granule in the selected collection. Do not map this granule to any HDF Map granules. This granule will be referred to as granule B.		#comment
22	Use the DPL Ingest GUI to configure the test collection to be public on ingest and to NOT generate HDF Map granules on ingest.		
23	Ingest granule B.		
24	S-4 Find a granule which is public, and is linked to a non-public HDF Map granule. This granule will be referred to as granule C. Ensure that the insert of this granule has already been exported.		#comment

#	Action	Expected Result	Notes
25	Exported as part of the preconditions.		
26	S-5 Wait for the automatic export of the granule inserts in S-2, S-3 and S-4 to complete.		#comment
27	Verify that the TCP proxy shows that HTTP PUT requests are generated for granules A and B. Granule C was exported before S-4.		
28	V-1 Verify that the export of the insert of granule A in S-5 does not contain any URLs.		#comment
29	Verify that the TCP proxy shows that the metadata exported for granule_A does not contain any URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should not have any URLs xpath /Granule/OnlineResources granule_A.xml should not have any URLs		
30	V-2 Verify that the export of the insert of granules B and C in S-5 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a HDF Map granule.		#comment
31	Verify that the TCP proxy shows that the metadata exported for granule_B contains science granule URL granule_B: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
32	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any HDF Map URLs: xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'HDF Map'		
33	Note: granule C was exported before S-4, not in S-4.		#comment
34	S-6 Publish granule A		#comment
35	EcDlPublishStart <MODE> -ecs -g <granule_A>		
36	S-7 Generate a public HDF Map granule which is linked to Granule B.		#comment
37	EcAmInsertMapGenerationRequest.pl -mode <MODE> -g <granule_B>		
38	S-8 Publish the HDF Map granule linked to granule C.		#comment
39	EcDlPublishStart <MODE> -ecs -g <HDF_C>		
40	S-9 Wait for the automatic exports of the granule updates in S-6, S-7 and S-8 to complete.		#comment
41	Verify that the TCP proxy shows that HTTP PUT requests are generated for granule_A, granule_B and granule_C		
42	V-3 Verify that the export of metadata for granules A, B and C in S-9 contain the correct URL for the associated HDF Map.		#comment

#	Action	Expected Result	Notes
43	Verify that the TCP proxy shows that the metadata exported for granule_A contains science data URLs and HDF Map URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should contain the granule_A URL xpath /Granule/OnlineResources granule_A.xml should contain an OnlineResource element of type "HDF Map" and the url of HDF_A		
44	Verify that the TCP proxy shows that the metadata exported for granule_B contains science data URLs and HDF Map URLs: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the granule_B URL xpath /Granule/OnlineResources granule_B.xml should contain an OnlineResource element of type "HDF Map" and the url of HDF_B		
45	Verify that the TCP proxy shows that the metadata exported for granule_C contains science data URLs and HDF Map URLs: xpath /Granule/OnlineAccessURLs granule_C.xml should contain the granule_C URL xpath /Granule/OnlineResources granule_C.xml should contain an OnlineResource element of type "HDF Map" and the url of HDF_C		
46	Verify that the URLs of type "HDF Map" for granules A, B, and C all point to the correct URL of the associated HDF map. The URL may or may not be accessible, depending on the mode and the configuration of the FTP or HTTP server, but the correctness can be determined by turning the URL into a local file path, e.g. http://f4ftl01//FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml becomes /datapool/<MODE>/user/FS2/SAG3/g3at.004/2005.01.01/g3a.t.01528320v04.00.00.xml		
47	S-10 Unpublish granule A.		#comment
48	EcDlUnpublishStart.pl -mode <MODE> -g <granule_A>		
49	S-11 Unpublish the HDF Map granule associated with granule B		#comment
50	EcDlUnpublishStart.pl -mode <MODE> -g <HDF_B>		
51	S-12 Unlink Granule C from its linked HDF Map granule in the database xref table.		#comment
52	delete from AmHdfMapGranuleXref where hdfmapid = <HDF_C_hdfmapid>		
53	S-13 Wait for the automatic export of the granules in S-10, S-11, and S-12 to complete.		#comment

#	Action	Expected Result	Notes
54	Wait for the TCP proxy log to show HTTP PUT requests for granules A and B.		
55	V-4 Verify that the export of metadata for granule A in S-13 contains no OnlineAccess or OnlineResource URLs.		#comment
56	Verify that the TCP proxy shows that the metadata exported for granule_A does not contain any URLs: xpath /Granule/OnlineAccessURLs granule_A.xml should not have any URLs xpath /Granule/OnlineResources granule_A.xml should not have any URLs		
57	V-5 Verify that the export of metadata for granule B in S-13 contains no HDF Map granule URL.		#comment
58	Verify that the TCP proxy shows that the metadata exported for granule_B contains science granule URL granule_B: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science granule URL		
59	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any HDF Map URLs: xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'HDF Map'		
60	V-6 Verify that no export of metadata for granule C is spawned by the modifications in S-12		#comment
61	Verify that the TCP proxy does not show any HTTP PUT request is generated for granule C		
62	S-14 Perform a manual export of granules A, B, and C.		#comment
63	EcBmBMGTManualStart $MODE --metg --g $GRANULEID_A,$GRANULEID_B,$GRANULEID_C		
64	V-7 Verify that the manual export of granule metadata in S-14 contains no HDF Map OnlineAccess or OnlineResource URLs for granules A, B or C. Verify that granules B, and C have Science and Metadata URLs.		#comment
65	Verify that the TCP proxy shows that the metadata exported for granule_A does not contain a science granule URL: xpath /Granule/OnlineAccessURLs granule_A.xml should return no results		
66	Verify that TCP proxy shows that the metadata exported for granule_A does not contain any HDF Map URLs: xpath /Granule/OnlineResources granule_A.xml should return no results		
67	Verify that the TCP proxy shows that the metadata exported for granule_B contains science granule URL granule_B: xpath /Granule/OnlineAccessURLs granule_B.xml should contain the science		

#	Action	Expected Result	Notes
	granule URL		
68	Verify that TCP proxy shows that the metadata exported for granule_B does not contain any HDF Map URLs: xpath /Granule/OnlineResources granule_B.xml should not contain any URL in OnlineResource type element 'HDF Map'		
69	Verify that the TCP proxy shows that the metadata exported for granule_C contains science granule URL granule_C: xpath /Granule/OnlineAccessURLs granule_C.xml should contain the science granule URL		
70	Verify that TCP proxy shows that the metadata exported for granule_C does not contain any HDF Map URLs: xpath /Granule/OnlineResources granule_C.xml should not contain any URL in OnlineResource type element 'HDF Map'		
71	V-8 Verify that all HDF Map URLS in the granule metadata have a URL type of “HDF Map”		#comment
72	xpath /Granule/OnlineResources granule_A.xml	Should contain an OnlineResource element of type "HDF Map" and the URL of HDF_A.	
73	xpath /Granule/OnlineResources granule_B.xml	Should contain an OnlineResource element of type "HDF Map" and the URL of HDF_B.	
74	xpath /Granule/OnlineResources granule_C.xml	Should contain an OnlineResource element of type "HDF Map" and the URL of HDF_D.	

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
72		testing HDF map granules publish, unpublish,	AE_DySno.002	1 collection with4 science granules which			/sotestdata/DROP_802/BE_82_01/Criteria/072/AE_DySno.002	

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		link, unlink, and export		support HDF map generation				

EXPECTED RESULTS:

	V	72	1	Verify that the export of the insert of granule A in S-5 does not contain any URLs.		
	V	72	2	Verify that the export of the insert of granules B and C in S-5 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a HDF Map granule.		
	V	72	3	Verify that the export of metadata for granules A, B and C in S-9 contain the correct URL for the associated HDF Map granules.		
	V	72	4	Verify that the export of metadata for granule A in S-13 contains no OnlineAccess or OnlineResource URLs.		
	V	72	5	Verify that the export of metadata for granules B and C in S-13 contains only URLs for the science, metadata, and browse (if applicable) files, but not for a HDF Map granule.		
	V	72	6	Verify that the manual export of granule metadata in S-14 contains correct URLs for the associated HDF Map granules for granules A, B, and C.		
	V	72	7	Verify that all HDF Map URLS in the granule metadata have a URL type of “HDF Map”		

1.29 Granule Deletion (ECS-ECSTC-28)

DESCRIPTION:

	S	80	1	**[Granule Deletion]** Stop the BMGT automatic driver process so that new events are not picked up.		
	S	80	2	Perform the following for a single granule:		

				a) Insert the granule. b) Update the granule. c) Delete the granule.		
	S	80	3	Restart BMGT.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure the test collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure the test collection is enabled to be public on ingest, using DPL Ingest GUI.		
7	Ensure the test collection is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly abled for export in this step, wait for it and its granules to be exported.		
8	Ensure ECHO has the collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections		

#	Action	Expected Result	Notes
	${SHORTNAME}.${VERSIONID} Ensure the test granule is not in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = '${LOCALGRANULEID}' If needed, physically delete the granules (bulk delete, unpublish, deletion cleanup).		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 Stop the BMGT automatic driver process so that new events are not picked up.		#comment
12	Stop BMGT automatic driver: ./EcBmBMGTAutoStop $MODE		
13	S-2 Perform the following for a single granule: a) Insert the granule. b) Update the granule. c) Delete the granule.		#comment
14	V-1 Verify that the actions in S-2 trigger events in the AIM event queue.		#comment
15	a) Insert the granule.		#comment
16	Note the current time as t0.		
17	Ingest the test granule into the public data pool.		
18	Verify dsmdgreventhistory has a GRINSERT event for the test granule enqueued after time t0: select dbid, eventtype, eventtime from dsmdeventhistory where eventtype = 'GRINSERT'; and eventtime > '${TIME_T0}' and dbid = ${GRANULEID}	Should have 1 record. The eventtime should be close to the ingest time.	
19	b) Update the granule.		#comment
20	Note the current time as t1.		

21	Update the test granule. E.g, update the DayNightFlag to a different value:
	
	declare l_daynightflag amgranule.daynightflag%type;
	
	select granuleid, daynightflag
	into l_daynightflag
	from amgranule
	where granuleid = ${GRANULEID};
	
	--(if old value was 'Day');
	if l_daynightflag = 'Day' then
	  update amgranule
	  set daynightflag = 'Night'
	  where granuleid = ${GRANULEID};
	else
	  update amgranule
	  set daynightflag = 'Day'
	  where granuleid = ${GRANULEID};
	end if;
	
	return l_daynightflag;
	
	-- Save the original daynightflag if needed to reset.		

22	Verify dsmdgreventhistory has a GRUPDATE event for the test granule enqueued after time t1: select dbid, eventtype, eventtime from dsmdeventhistory where eventtype = 'GRUPDATE'; and eventtime > '${TIME_T1}' and dbid = ${GRANULEID}	Should have 1 record. The eventtime should be close to the update time.	

#	Action	Expected Result	Notes
23	d) Delete the granule.		#comment
24	Create a geoids.txt file for the test granule: SC:${SHORTNAME}.${VERSIONID}:${GRANULEID}		
25	Note the current time as t2.		
26	Bulk Delete the test granule: ./EcDsBulkDelete.pl -mode $MODE -physical -geoidfile /path/to/geoids.txt -server f4dbl03 -database ecs -user EcDsAmGranuleDeletion -password password		
27	Verify dsmdgreventhistory has a GRDELETE event for the test granule enqueued after time t2: select dbid, eventtype, eventtime from dsmdeventhistory where eventtype = 'GRDELETE'; and eventtime > '${TIME_T2}' and dbid = ${GRANULEID}	Should have 1 record. The eventtime should be close to the bulk delete time.	
28	S-3 Restart BMGT.		#comment
29	./EcBmBMGTAutoStart $MODE		
30	V-2 Verify that after BMGT is restarted in S-3, no granule insert (HTTP PUT)s are exported for the actions in S-2.		#comment
31	Verify that the tcp.log shows no HTTP PUT requests for the test granule corresponding to the Insert and Update events for the test granule.		
32	V-3 Verify that a single HTTP DELETE is exported to ECHO, or an ECHO stand-in. Verify that the URL to which the DELETE is exported contains an identifier for the deleted granule, and the request body is empty.		#comment
33	Verify that that the tcp.log shows an HTTP DELETE request is exported for the test granule after time t2.		
34	Verify that the request body for the HTTP DELETE request is empty.		
35	V-4 Verify that the granule delete export is successful, and that if ECHO reports an error for the deletion of a non existent granule, BMGT ignores this error. It is however acceptable for ECHO to not report such an error.		#comment
36	Verify that BMGT does not report an error afer time t2.		

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
80		1 science granule to ingest, update, and delete	MOD10A1.005		1 granule		/sotestdata/DROP_802/BE_82_01/Criteria/080/MOD10A1.005	

EXPECTED RESULTS:

	V	80	1	Verify that the actions in S-2 trigger events in the AIM event queue.		
	V	80	2	Verify that after BMGT is restarted in S-3, no granule insert (HTTP PUT)s are exported for the actions in S-2.		
	V	80	3	Verify that a single HTTP DELETE is exported to ECHO, or an ECHO stand-in. Verify that the URL to which the DELETE is exported contains an identifier for the deleted granule, and the request body is empty.		
	V	80	4	Verify that the granule delete export is successful, and that if ECHO reports an error for the deletion of a non existent granule, BMGT ignores this error. It is however acceptable for ECHO to not report such an error.		

1.30 QA Update (ECS-ECSTC-29)

DESCRIPTION:

	S	90	1	**[QA Update]** Perform a QA update on one or more granules		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment

#	Action	Expected Result	Notes
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure the BMGT dispatcher and auto driver are running. On the BMGT host: ps auxww | grep $MODE | grep -E 'EcBmDispatcher|EcBmAuto' If needed, start BMGT: ./EcBmBMGTAppStart $MODE		
6	Ensure the test collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
7	Ensure the test collection is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for it and its granules to be exported.		
8	Ensure ECHO has the collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
9	Ensure granules g1, g2, g3, and g4 are in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid in ('${LOCALGRANULEID}', ...) If needed, ingest the granules into the public data pool.	Should return 1 row.	
10	Ensure ECHO has metadata for granules g1, g2, g3, and g4. For each granule, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granules, export them: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID},...		

#	Action	Expected Result	Notes
11	Ensure QAUU is configured with a test site (e.g., BMGT82): Append the following lines to /custom/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties if the are not there: BMGT82_EMAIL_FROM_ADDRESSES=labuser@edn.ecs.nasa.gov BMGT82_EMAIL_REPLY_ADDRESS=labuser@edn.ecs.nasa.gov BMGT82_NOTIFICATION_ON_SUCCESS=N		
12	Ensure the test collection is eligible for QA updates: select * from dsqamutesdtsite where shortname = '${SHORTNAME}' and site = 'BMGT82' If no such row exists, add it: insert into dsqamutesdtsite values ('${SHORTNAME}', 'BMGT82')		
13	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
14	Setup		#comment
15	S-1 [QA Update] Perform a QA update on one or more granules		#comment
16	Find the granules' metadata files: select f.granuleid, p.path || '/' || f.archivemetfilename from ammetadatafile f join dsmdxmlpath p on f.archivepathid = p.archivepathid where f.granuleid in (${g1},${g2},${g3},${g4})		
17	Find the names of measured parameters to update. For each granule, xpath '//MeasuredParameterContainer/ParameterName/text()' ${XML_FILE}		
18	Find the measured parameters' science quality flag values. For each granule, xpath "//MeasuredParameterContainer[ParameterName='${PARAMETER_NAME}']/QAFlags/ScienceQualityFlag/text()" ${XML_FILE}		
19	Select different science quality flag values from the list of valid values in the properties file: grep VALID_SCIENCE_QA_FLAGS /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
20	Create a QA update request file to update science flags, named according to the the QAUU 609: <MODE>_<Site>_QAUPDATE<description>.<YYYY><MM><DD><hh><mm><ss> E.g., OPS_BMGT82_QAUPDATE_BE_82_01_Crit_90_Science.20130529153000		
21	Write a QA update request to update science flags, using GranuleURs (replace <TAB> with an actual tab character): From ${site} begin QAMetadataUpdate Science GranuleUR<br		

#	Action	Expected Result	Notes
	/>${shortname}<TAB>${versionid}<TAB>${g1_granuleur}<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation} ${shortname}<TAB>${versionid}<TAB>${g2_granuleur}<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation} end QAMetadataUpdate		
22	Find the measured parameters' operational quality flag values. For each granule, xpath "//MeasuredParameterContainer[ParameterName='${PARAMETER_NAME}']/QAFlags/OperationalQualityFlag/text()" ${XML_FILE}		
23	Select different operational quality flag values from the list of valid values in the properties file: grep VALID_OPERATIONAL_QA_FLAGS /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
24	Create a QA update request file to update operational flags, named according to the the QAUU 609: <MODE>_<Site>_QAUPDATE<description>.<YYYY><MM><DD><hh><mm><ss> E.g., OPS_BMGT82_QAUPDATE_BE_82_01_Crit_90_Operational.20130529153000		
25	Write a QA update request to update operational flags, using GranuleURs (replace <TAB> with an actual tab character): From ${site} begin QAMetadataUpdate Operational GranuleUR ${shortname}<TAB>${versionid}<TAB>${g3_granuleur}<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation} ${shortname}<TAB>${versionid}<TAB>${g4_granuleur}<TAB>${measured_parameter_name}<TAB>${qa_flag_value}<TAB>${qa_flag_explanation} end QAMetadataUpdate		
26	Find the QA update request directory configured in the properties file: sed -n 's/QA_REQUEST_DIR *= *//p' /usr/ecs/${MODE}/CUSTOM/cfg/EcDsAmQaUpdateUtility.properties		
27	Delete any files in the QA update request directory.		
28	Copy the QA update request files to the QA request directory.		
29	Note the current time as t0.		
30	Run QAUU: EcDsAmQAUUStart ${MODE} -noprompt		
31	Verification		#comment

#	Action	Expected Result	Notes
32	V-1 Verify that for each granule involved in the QA Update, an HTTP PUT is exported to ECHO or an ECHO stand-in.		#comment
33	Verify the TCP proxy log shows one PUT for each granule after time t0.		
34	V-2 Verify that the HTTP PUT contains the full granule metadata, including the updated QA values reflecting the QA Update.		#comment
35	Save each granule's exported metadata from the TCP proxy log to a separate XML file.		
36	Verify granule g1's exported metadata contains the ScienceQualityFlag and the ScienceQualityFlagExplanation associated with the MeasuredParameter specified in the Science QA request file: xpath "//MeasuredParameter[ParameterName='${g1_parametername}']/QAFlags/ScienceQualityFlag/text()" g1.xml xpath "//MeasuredParameter[ParameterName='${g1_parametername}']/QAFlags/ScienceQualityFlagExplanation/text()" g1.xml		
37	Verify granule g2's exported metadata contains the ScienceQualityFlag and the ScienceQualityFlagExplanation associated with the MeasuredParameter specified in the Science QA request file: xpath "//MeasuredParameter[ParameterName='${g2_parametername}']/QAFlags/ScienceQualityFlag/text()" g2.xml xpath "//MeasuredParameter[ParameterName='${g2_parametername}']/QAFlags/ScienceQualityFlagExplanation/text()" g2.xml		
38	Verify granule g3's exported metadata contains the OperationalQualityFlag and the OperationalQualityFlagExplanation associated with the MeasuredParameter specified in the Operational QA request file: xpath "//MeasuredParameter[ParameterName='${g3_parametername}']/QAFlags/OperationalQualityFlag/text()" g3.xml xpath "//MeasuredParameter[ParameterName='${g3_parametername}']/QAFlags/OperationalQualityFlagExplanation/text()" g3.xml		
39	Verify granule g4's exported metadata contains the OperationalQualityFlag and the OperationalQualityFlagExplanation associated with the MeasuredParameter specified in the Operational QA request file: xpath "//MeasuredParameter[ParameterName='${g4_parametername}']/QAFlags/OperationalQualityFlag/text()" g4.xml xpath "//MeasuredParameter[ParameterName='${g4_parametername}']/QAFlags/OperationalQualityFlagExplanation/text()" g4.xml		

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
90		Science QA flag update	MYD10A1.005		2 granules		/sotestdata/DROP_802/BE_82_01/Criteria/090/MYD10A1.005	
90		Operational QA flag update	MYD10A1.005		2 granules		/sotestdata/DROP_802/BE_82_01/Criteria/090/MYD10A1.005	

EXPECTED RESULTS:

	V	90	1	Verify that for each granule involved in the QA Update, an HTTP PUT is exported to ECHO or an ECHO stand-in.		
	V	90	2	Verify that the HTTP PUT contains the full granule metadata, including the updated QA values reflecting the QA Update.		

1.31 ISO Nominal Granule Export: Granule Export by Collection (ECS-ECSTC-30)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule.		

				f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to extract XML elements.		#comment
3	Ensure test collections C1, C2 are installed.		
4	Ensure both collections are configured to be public on ingest.		
5	Ensure test granules have been ingested for each collection.		
6	Ensure the BMGT automatic driver is running.		
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		#comment
10	Ensure both ISO collections are enabled for both collection and granule export: update bg_collection_configuration set granuleexport = 'Y', collectionexport = 'Y' where shortname = $SHORT_NAME and versionid in ($C1_VERSION_ID, $C2_VERSION_ID)		
11	Request a manual export of collection C1: EcBmBMGTManualStart --mode <MODE> --metg --collections <SHORTNAME>.<VERSION_ID>		
12	Verification		#comment
13	V-1 Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the		#comment

#	Action	Expected Result	Notes
	requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
14	Verify the TCP proxy log shows a PUT for each C1 granule.		
15	Verify the TCP proxy log shows no PUTs for any C2 granules.		
16	Save the body of each PUT to a separate XML file.		
17	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log) and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
18	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
19	Verify that each granule's exported metadata validates against TBD ISO schema: /tools/libxml2-2.9.1/bin/xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		
20	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
21	Query amgranule for each exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_IDS)		
22	Use an XPath utility to extract the InsertTime from each granule's exported metadata: xpath "//gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime/text()" granule.xml		
23	Verify each granule's InsertTime matches its archivetime from amgranule.		
24	Use an XPath utility to extract the UpdateTime from each granule's exported metadata: xpath "//gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime/text()" granule.xml		
25	Verify each granule's UpdateTime matches its lastupdate from amgranule.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
100 S-1				2 ISO collections with the same ShortName but different VersionIds (C1, C2).				
100 S-1				Several granules for each of C1 and C2 (3 or more).				
100 S-1a				One new C2 granule.				
100 S-1b				One new C2 granule.				
100 S-1c				One new C2 granule.				
100 S-1d				One new C2 granule.				
100 S-1e				One new C2 granule.				
100 S-1f				One new C2 granule.				
100 S-1g				One new C2 granule.				
100 S-1h				One new C2 granule.				
100 S-1i				One new C2 granule.				
100 S-1j				One new C2 granule.				
100 S-1k				C2 from S-1.				
100				One new C2 granule.				

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
S-1l								

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		

	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.32 ISO Nominal Collection Export (ECS-ECSTC-31)

DESCRIPTION:

	S	105	1	**[ISO Nominal Collection Export]** Find two collections with ISO metadata and which share the same short name, but different version IDs. Request the manual export of collection metadata for one of these collections.		
	S	105	2	Find collections which have ISO Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Perform an update on an existing collection. For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to extract XML elements.		#comment
3	Use /tools/libxml2-2.9.1/bin/xmllint to verify ISO XML against ISO schema.		#comment
4	Ensure test collections C1, C2, C4, C5, and C6 are installed.		
5	Ensure each of collections C1, C2, C4, and C5 is enabled for collection export: update bg_collection_configuration collectionexport = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		

#	Action	Expected Result	Notes
6	Ensure collection C6 is disabled for collection or granule export update bg_collection_configuration collectionexport = 'N', granuleexport = 'N' where shortname = <C3_SHORT_NAME> and versionid = <C3_VERSION_ID>		
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 Find two collections with ISO metadata and which share the same short name, but different version IDs. Request the manual export of collection metadata for one of these collections.		#comment
10	Request a manual export of collection C1: EcBmBMGTManualStart --mode <MODE> --metc --collections <C1_SHORTNAME>.<C1_VERSION_ID>		
11	S-2 Find collections which have ISO Metadata: a) Insert a new collection into the ECS inventory (and enable for collection export). b) Delete a collection from the ECS inventory. c) Perform an update on an existing collection. d) For a collection which is currently disabled for collection export, enable it for collection (but not granule) export.		#comment
12	S-2a		#comment
13	Copy collection C3's descriptor file to /usr/ecs/<MODE>/CUSTOM/data/ESS/ on the same host as the ESDT Maintenance GUI.		
14	Use the ESDT Maintenance GUI to install collection C3. Follow the wiki instructions, stopping before the DPL Maintenance GUI: http://edhs1.gsfc.nasa.gov:40000/bin/view/EDF/AddESDT		
15	Enable collection C3 for collection export: update bg_collection_configuration collectionexport = 'Y' where shortname = <C3_SHORT_NAME> and versionid = <C3_VERSION_ID>		
16	S-2b		#comment
17	Use the ESDT Maintenance GUI to delete collection C4.		
18	S-2c		#comment
19	Copy collection C5's updated descriptor file to /usr/ecs/<MODE>/CUSTOM/data/ESS/ on the same host as the ESDT Maintenance GUI.		
20	Use the ESDT Maintenance GUI to update collection C5.		

#	Action	Expected Result	Notes
21	S-2d		#comment
22	Enable collection C6 for collection export: update bg_collection_configuration collectionexport = 'Y' where shortname = <C6_SHORT_NAME> and versionid = <C6_VERSION_ID>		
23	Verification		#comment
24	V-1 Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but not any other collections sharing a short name but with a different version ID).		#comment
25	Verify the TCP proxy logs a single PUT for collection C1's manual export.		
26	Verify the TCP proxy logs the full metadata for collection C1.		
27	Verify the TCP proxy logs no request for C2 (or any other collection with the same ShortName).		
28	V-2 Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		#comment
29	Verify the TCP proxy logs a single PUT for each collection C3, C5, and C6.		
30	Verify the TCP proxy logs the full metadata for each collection C3, C5, and C6.		
31	V-3 Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but not containing any collection metadata in the request body.		#comment
32	Verify the TCP proxy logs a single DELETE for collection C4.		
33	Verify the URL for collection C4's request includes C4's dataset ID.		
34	Verify collection C4's request has an empty body.		
35	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the TBD ISO schema.		#comment
36	For each PUT, save the request body to a separate XML file.		
37	Verify that each granule's exported metadata validates against ISO schema: xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		
38	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) InsertTime = The insert time of the collection recorded in the AIM database. b) LastUpdate = The last update time of the collection recorded in the AIM database.		#comment
39	Query amcollection for each exported collection's InsertTime and		

#	Action	Expected Result	Notes
	LastUpdate: select inserttime, lastupdate from amcollection where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
40	Use an XPath utility to extract the InsertTime from each granule's exported metadata: xpath "//gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime/text()" granule.xml		
41	Verify each collection's exported metadata InsertTime matches its amcollection inserttime.		
42	Use an XPath utility to extract the UpdateTime from each granule's exported metadata: xpath "//gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime/text()" granule.xml		
43	Verify each collection's exported metadata UpdateTime matches its amcollection lastupdate.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
105 S-1				2 ISO collections with the same ShortName but different VersionIds (C1, C2).				
105 S-2a				1 ISO collection to install during the test (C3).				
105 S-2b				1 ISO collection to delete during the test (C4).				
105 S-2c				1 ISO collection to be updated (C5), plus an extra descriptor file with which to update the collection.				
105 S-2d				1 ISO collection (C6).				

EXPECTED RESULTS:

	V	105	1	Verify that the manual export in S-1 results in a single HTTP PUT request containing the full collection metadata for the requested collection (but not any other collections sharing a short name but with a different version ID).		
	V	105	2	Verify that the operations in S-2 subclauses a, c, and d each result in the export of a single HTTP PUT containing the full collection metadata.		
	V	105	3	Verify that the operation in S-2 subclause b results in the export of a single HTTP DELETE, with the ID of the collection in the URL, but not containing any collection metadata in the request body.		
	V	105	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b, validates against the TBD ISO schema.		
	V	105	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclause b has the following elements: a) InsertTime = The insert time of the collection recorded in the AIM database. b) LastUpdate = The last update time of the collection recorded in the AIM database.		

1.33 MISR Browse Links - Science Before Browse (ECS-ECSTC-32)

DESCRIPTION:

	S	110	1	**[MISR Browse Links – Science Before Browse]** This criterion will test the export of the MISBR science granules and their linkage with previously inserted MISR Level 1 and Level 2 science granules. The test uses the following granules. <u>Matching Granules</u>: Select at least one granule from a MISR Level 1 ESDT, and for each a MISBR granule that can be associated with it (cameraIds match, its temporal coverage intersects that of the MISR Level 1 granule and it has a matching SP_AM_MISR_ProductVersion product specific attribute value). Also select at least one granule from a MISR Level 2 ESDT and at least one MIB2GEOP granule. For each of those granules, select a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MISR Level 2 respectively MIB2GEOP granule). <u>Non-Matching Granules</u>: Select one MISBR with a cameraId that is not 'AN' and a temporal coverage that intersects with that of the MISR Level 2 granule. Select one other MISBR granule with a cameraId of 'AN' whose temporal coverage does not match any of the selected MISR Level 1 and Level 2 granules. Select one more MISBR granule whose camera Id is not 'AN' and matches the cameraId and temporal coverage of one of the MISR Level 1 granules in the 'matching' group, but has a different SP_AM_MISR_ProductVersion. Select at least one other MISR Level 1 and one MISR Level 2 granule that do not match any MISBR selected for the test. None of the MISR Level 1 and Level 2 granules selected for the test shall match up with MISBR granules that already exist in the inventory. None of the MISBR granules selected for the test shall match up with MISR		

				granules that already exist in the inventory. The selected MISR collections must be eligible for granule export to ECHO and to export their browse links to ECHO. The selected collections must also be configured to be public in the datapool.		
	S	110	2	Ingest the MISR Level 1, Level 2, and MIB2GEOP granules referenced in step S-1.		
	S	110	3	Wait for the ingest of the MISR granules in S-2 to be picked up by the automatic polling process and their metadata exported.		
	S	110	4	Ingest the MISBR granules referenced in step S-1.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use the xpath script to extract elements from XML files: /tools/common/test/BE_82_01/bin/xpath		#comment
3	Ensure the test MISBR, MISR Level 1, MISR Level 2, and MIB2GEOP collections are installed.		
4	Ensure the test collections are configured to be public on ingest.		
5	Ensure test collections are configured for collection and granule export.		
6	Ensure AIM has no MISR granules that match the test MISBR granules. Use the xpath utility to get the Path Number, Orbit Number, Product Version, and Camera ID from the browse metadata file: xpath "//PSAs/PSA[PSAName='SP_AM_PATH_NO']/PSAValue/text()" misbr.xml xpath '//OrbitCalculatedSpatialDomain/OrbitCalculatedSpatialDomainContainer/OrbitNumber/text()' misbr.xml xpath "//PSAs/PSA[PSAName='SP_AM_MISR_ProductVersion']/PSAValue/text()" misbr.xml xpath '//Platform/Instrument/Sensor/SensorShortName/text()' misbr.xml Query AIM for matching MISR Level 1 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and		

#	Action	Expected Result	Notes
	g.versionid = p.versionid) where p.type = '1' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and a.productversion = $PRODUCTVERSION and c.cameraid = $CAMERAID Query AIM for matching MISR Level 2 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = '2' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and c.cameraid = 'AN'		
7	Ensure AIM has no MISBR granules that match the test MISR granules. Use the xpath utility to get the Path Number, Orbit Number, Product Version, and Camera ID from the granule metadata files: xpath "//PSAs/PSA[PSAName='SP_AM_PATH_NO']/PSAValue/text()" granule.xml xpath '//OrbitCalculatedSpatialDomain/OrbitCalculatedSpatialDomainContainer/OrbitNumber/text()' granule.xml xpath "//PSAs/PSA[PSAName='SP_AM_MISR_ProductVersion']/PSAValue/text()" xpath '//Platform/Instrument/Sensor/SensorShortName/text()' granule.xml Query AIM for MISBR matching the MISR Level 1 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = 'B' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and a.productversion = $PRODUCTVERSION and c.cameraid = $CAMERAID Query AIM for MISBR matching the MISR Level 2 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = 'B' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and c.cameraid = 'AN'		
8	Ensure EcDlInsertUtility.properties has this: MISR_SPECIAL_PROCESSING=Y Bounce DPAD after		

#	Action	Expected Result	Notes
	changing the value: ./EcDlActionDriverStart $MODE		
9	Ensure that a TCP proxy or mock ECHO is capturing BMGT traffic.		
10	Ensure the BMGT dispatcher and auto driver are running: ./EcBmBMGTAppStart $MODE	ps auxww | sed -n '/OPS\/CUSTOM/{;s/.* -D\(Bmgt[^]*\).*/\1/p;}' BmgtComponent=EcBmDispatcher BmgtComponent=EcBmAuto BmgtComponent=EcBmMonitor	
11	Setup		#comment
12	S-1 This criterion will test the export of the MISBR science granules and their linkage with previously inserted MISR Level 1 and Level 2 science granules. The test uses the following granules. Matching Granules: Select at least one granule from a MISR Level 1 ESDT, and for each a MISBR granule that can be associated with it (cameraIds match, its temporal coverage intersects that of the MISR Level 1 granule and it has a matching SP_AM_MISR_ProductVersion product specific attribute value). Also select at least one granule from a MISR Level 2 ESDT and at least one MIB2GEOP granule. For each of those granules, select a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MISR Level 2 respectively MIB2GEOP granule). Non-Matching Granules: Select one MISBR with a cameraId that is not 'AN' and a temporal coverage that intersects with that of the MISR Level 2 granule. Select one other MISBR granule with a cameraId of 'AN' whose temporal coverage does not match any of the selected MISR Level 1 and Level 2 granules. Select one more MISBR granule whose camera Id is not 'AN' and matches the cameraId and temporal coverage of one of the MISR Level 1 granules in the 'matching' group, but has a different SP_AM_MISR_ProductVersion. Select at least one other MISR Level 1 and one MISR Level 2 granule that do not match any MISBR selected for the test. None of the MISR Level 1 and Level 2 granules selected for the test shall match up with MISBR granules that already exist in the inventory. None of the MISBR granules selected for the test shall match up with MISR granules that already exist in the inventory. The selected MISR collections must be eligible for granule export to ECHO and to export their browse links to ECHO. The selected collections must also be configured to be public in the datapool.		#comment
13	S-2 Ingest the MISR Level 1, Level 2, and MIB2GEOP granules		#comment

#	Action	Expected Result	Notes
	referenced in step S-1.		
14	Note the current time as t0.		
15	Ingest test MISR granules but not the test MISBR granules.		
16	S-3 Wait for the ingest of the MISR granules in S-2 to be picked up by the automatic polling process and their metadata exported.		#comment
17	Wait for one of the following export indicators: The BMGT log shows that the MISR granule metadata has been exported. The BMGT GUI shows the test MISR granule exports succeeded. The TCP proxy logs HTTP PUT requests for all the test MISR granules.		
18	V-1 Verify that the export in S-3 contains the granules ingested in S-2		#comment
19	Already verified by waiting in S-3.		
20	Verify each HTTP PUT request contains full granule metadata for each MISR granule.		
21	V-2 Verify that the metadata exported in S-3 contains no browse URLs for the granules ingested in S-2.		#comment
22	Save each granule's exported metadata from the TCP proxy log to a separate XML file.		
23	Verify no exported granule metadata after time t0 contains OnlineResource URLs with Type BROWSE. For each exported granule, the following XPath should have no URLs: xpath "//OnlineResource[Type='BROWSE']/URL" granule.xml		
24	S-4 Ingest the MISBR granules referenced in step S-1.		#comment
25	Note the current time as t1.		
26	Ingest the test MISBR granules.		
27	V-3 Verify that the granules ingested in S-4 are picked up and exported by the automatic polling process.		#comment
28	Verify the BMGT log shows that the MISBR granule metadata has been exported.		
29	Verify the BMGT GUI shows the test MISBR granule exports succeeded.		
30	Verify the TCP proxy logs HTTP PUT requests for all the test MISBR granules after time t1.		
31	V-4 Verify that the second set of exports, after S-4, contains science granule metadata for both the MISR Level 1 & 2 and MISBR granules ingested in S-4.		#comment
32	Verify, after time t1, the mock ECHO logs a single HTTP PUT for each of the MISBR, MISR Level 1, and MISR Level 2 granules.		

#	Action	Expected Result	Notes
33	Save each of the MISBR, MISR Level 1, and MISR Level 2 granules' metadata, exported after time t1, to a separate XML file.		
34	Verify that each of the granules exported after time t1 validates against the ECHO 10 Granule.xsd schema. For each granule file, /tools/common/test/BE_82_01/bin/echo10_validate_metadata granule.xml		
35	V-5 Verify that the second set of exports, after S-4, also contains the full granule metadata for the MISR Level 1 & 2 granules ingested in S-2 that are linked with the MISBR granules ingested in step S-4 (i.e., for the matching granules), and does not include metadata for the other, non-matching granules. Verify that this granule metadata includes Browse link URLs, and that these URLs are correct.		#comment
36	Verify, after time t1, each HTTP PUT request contains full granule metadata for each of the MISR Level 1 and MISR Level 2 granules associated with the MISBR ingested in S-4.		
37	Verify that after time t1 all exported MISR granules that appear in the TCP proxy log are associated with the MISBR ingested in S-4.		
38	Verify the exported metadata for MISR Level 1 and MISR Level 2 granules includes OnlineResource BROWSE URLs: xpath "//OnlineResource[Type='BROWSE']/URL" granule.xml		
39	Verify each OnlineResource BROWSE URL points to the correct Browse file in the data pool.		
40	V-6 Verify that the URLs exported for the MIB2GEOP granules show them linked with MISBR granules that were selected according to the rules that apply to MISR Level 2 granules.		#comment
41	Verify the OnlineResource URLs for the MIB2GEOP granules point to matching MISBR granules, according to MISR Level 2 matching rules. To find a MISBR matching an ingested MISRSC granule, select b.misrbrid from AmBrowseGranuleXref bgx join ambrowse b on bgx.browseid = b.browseid where bgx.granuleid = <MISRSC_granuleId>		

TEST DATA:

Crit id	Critccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
				At least one granule from a MISR Level 1 ESDT, and for each a MISBR granule that can be associated with it (cameraIds match, its temporal coverage intersects that of the MISR Level 1 granule and it has a matching SP_AM_MISR_ProductVersion product specific attribute value).			/sotestdata/DROP_802/BE_82_01/Criteria/110	
				At least one granule from a MISR Level 2 ESDT and for each a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MISR Level 2).			/sotestdata/DROP_802/BE_82_01/Criteria/110	
				At least one MIB2GEOP granule and for each a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MIB2GEOP granule).			/sotestdata/DROP_802/BE_82_01/Criteria/110	

EXPECTED RESULTS:

	V	110	1	Verify that the export in S-3 contains the granules ingested in S-2		
	V	110	2	Verify that the metadata exported in S-3 contains no browse URLs for the granules ingested in S-2.		

	V	110	3	Verify that the granules ingested in S-4 are picked up and exported by the automatic polling process.		
	V	110	4	Verify that the second set of exports, after S-4, contains science granule metadata for both the MISR Level 1 & 2 and MISBR granules ingested in S-4.		
	V	110	5	Verify that the second set of exports, after S-4, also contains the full granule metadata for the MISR Level 1 & 2 granules ingested in S-2 that are linked with the MISBR granules ingested in step S-4 (i.e., for the matching granules), and does not include metadata for the other, non-matching granules. Verify that this granule metadata includes Browse link URLs, and that these URLs are correct.		
	V	110	6	Verify that the URLs exported for the MIB2GEOP granules show them linked with MISBR granules that were selected according to the rules that apply to MISR Level 2 granules.		

1.34 MISR Browse Links - Browse Before Science (ECS-ECSTC-33)

DESCRIPTION:

	S	115	1	**[MISR Browse Links – Browse Before Science]** This criterion will test the export of the MISBR science granules and their linkage with subsequently inserted MISR Level 1 and Level 2 science granules. The test uses the following granules. Matching Granules: Select at least one granule from a MISR Level 1 ESDT, and for each a MISBR granule that can be associated with it (cameraIds match, its temporal coverage intersects that of the MISR Level 1 granule and it has a matching SP_AM_MISR_ProductVersion product specific attribute value). Also select at least one granule from a MISR Level 2 ESDT and at least one MIB2GEOP granule. For each of those granules, select a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MISR Level 2 respectively MIB2GEOP granule). Non-Matching Granules: Select one MISBR with a cameraId that is not 'AN' and a temporal coverage that intersects with that of the MISR Level 2 granule. Select one other MISBR granule with a cameraId of 'AN' whose temporal coverage does not match any of the selected MISR Level 1 and Level 2 granules. Select one more MISBR granule whose camera Id is not 'AN' and matches the cameraId and temporal coverage of one of the MISR Level 1 granules in the 'matching' group, but has a different SP_AM_MISR_ProductVersion. Select at least one other MISR Level 1 and one MISR Level 2 granule that do not match any MISBR selected for the test. None of the MISR Level 1 and Level 2 granules selected for the test shall match up with MISBR granules that already exist in the inventory. None of the MISBR granules selected for the test shall match up with MISR granules that already exist in the inventory. The selected MISR collections must be eligible for granule export to ECHO and to export their browse links to ECHO. The selected collections must also be configured to be public in the Datapool		

	S	115	2	Ingest the MISBR granules referenced in step S-1.		
	S	115	3	Wait for the ingest of the MISR granules in S-2 to be picked up by the automatic polling process and their metadata exported.		
	S	115	4	Ingest the MISR Level 1, Level 2, and MIB2GEOP granules referenced in step S-1.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure the test MISBR, MISR Level 1, MISR Level 2, and MIB2GEOP collections are installed.		
3	Ensure the test collections are configured to be public on ingest.		
4	Ensure test collections are configured for collection and granule export.		
5	Ensure AIM has no MISR granules that match the test MISBR granules. Use the xpath utility to get the Path Number, Orbit Number, Product Version, and Camera ID from the browse metadata file: xpath "//PSAs/PSA[PSAName='SP_AM_PATH_NO']/PSAValue/text()" misbr.xml xpath '//OrbitCalculatedSpatialDomain/OrbitCalculatedSpatialDomainContainer/OrbitNumber/text()' misbr.xml xpath "//PSAs/PSA[PSAName='SP_AM_MISR_ProductVersion']/PSAValue/text()" misbr.xml xpath '//Platform/Instrument/Sensor/SensorShortName/text()' misbr.xml Query AIM for matching MISR Level 1 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = '1' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and a.productversion = $PRODUCTVERSION and c.cameraid = $CAMERAID Query AIM for matching MISR Level 2 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid =		

#	Action	Expected Result	Notes
	a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = '2' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and c.cameraid = 'AN'		
6	Ensure AIM has no MISBR granules that match the test MISR granules. Use the xpath utility to get the Path Number, Orbit Number, Product Version, and Camera ID from the granule metadata files: xpath "//PSAs/PSA[PSAName='SP_AM_PATH_NO']/PSAValue/text()" granule.xml xpath '//OrbitCalculatedSpatialDomain/OrbitCalculatedSpatialDomainContainer/OrbitNumber/text()' granule.xml xpath "//PSAs/PSA[PSAName='SP_AM_MISR_ProductVersion']/PSAValue/text()" xpath '//Platform/Instrument/Sensor/SensorShortName/text()' granule.xml Query AIM for MISBR matching the MISR Level 1 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = 'B' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and a.productversion = $PRODUCTVERSION and c.cameraid = $CAMERAID Query AIM for MISBR matching the MISR Level 2 granules: select esdt(g.shortname, g.versionid) ESDT, g.granuleid from amgranule g join dsmdmisrattributes a on g.granuleid = a.granuleid join dsmdmisrcamera c on g.granuleid = c.granuleid join dsmdmisrprocessingcriteria p on (g.shortname = p.shortname and g.versionid = p.versionid) where p.type = 'B' and a.orbitnumber = $ORBITNUMBER and a.pathno = $PATHNUMBER and c.cameraid = 'AN'		
7	Ensure EcDlInsertUtility.properties has this: MISR_SPECIAL_PROCESSING=Y		
8	Ensure that a TCP proxy or mock ECHO is capturing BMGT traffic.		
9	Setup		#comment
10	S-1 This criterion will test the export of the MISBR science granules and their linkage with subsequently inserted MISR Level 1 and Level 2 science granules. The test uses the following granules. Matching		#comment

#	Action	Expected Result	Notes
	Granules: Select at least one granule from a MISR Level 1 ESDT, and for each a MISBR granule that can be associated with it (cameraIds match, its temporal coverage intersects that of the MISR Level 1 granule and it has a matching SP_AM_MISR_ProductVersion product specific attribute value). Also select at least one granule from a MISR Level 2 ESDT and at least one MIB2GEOP granule. For each of those granules, select a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MISR Level 2 respectively MIB2GEOP granule). Non-Matching Granules: Select one MISBR with a cameraId that is not 'AN' and a temporal coverage that intersects with that of the MISR Level 2 granule. Select one other MISBR granule with a cameraId of 'AN' whose temporal coverage does not match any of the selected MISR Level 1 and Level 2 granules. Select one more MISBR granule whose camera Id is not 'AN' and matches the cameraId and temporal coverage of one of the MISR Level 1 granules in the 'matching' group, but has a different SP_AM_MISR_ProductVersion. Select at least one other MISR Level 1 and one MISR Level 2 granule that do not match any MISBR selected for the test. None of the MISR Level 1 and Level 2 granules selected for the test shall match up with MISBR granules that already exist in the inventory. None of the MISBR granules selected for the test shall match up with MISR granules that already exist in the inventory. The selected MISR collections must be eligible for granule export to ECHO and to export their browse links to ECHO. The selected collections must also be configured to be public in the Datapool		
11	S-2 Ingest the MISBR granules referenced in step S-1.		#comment
12	Note the current time as t0.		
13	Ingest the test MISBR granules.		
14	S-3 Wait for the ingest of the MISR granules in S-2 to be picked up by the automatic polling process and their metadata exported.		#comment
15	Wait for one of the following export indicators: The BMGT log shows that the MISBR granule metadata has been exported. The BMGT GUI shows the test MISBR granule exports succeeded. The mock ECHO logs HTTP PUT requests for all the test MISBR granules.		
16	V-1 Verify that the export in S-3 contains the full granule metadata for the MISBR granules ingested in S-2.		#comment
17	Verify the TCP proxy log shows that after time t0 a single HTTP PUT request is exported for each MISBR granule ingested.		
18	Verify the body of each MISBR PUT request contains the full granule		

#	Action	Expected Result	Notes
	metadata.		
19	V-2 Verify that the metadata exported in S-3 contains no browse URLs.		#comment
20	Verify the exported MISBR metadata contains no OnlineResource elements: xpath '//OnlineResource' misbr.xml		
21	S-4 Ingest the MISR Level 1, Level 2, and MIB2GEOP granules referenced in step S-1.		#comment
22	Note the current time as t1.		
23	Ingest the remaining test MISR granules (Level 1, Level 2, MIB2GEOP).		
24	V-3 Verify that the granules ingested in S-4 are picked up and exported by the automatic polling process.		#comment
25	Verify the TCP proxy log shows that after time t1 a single HTTP PUT request is exported for each MISR granule ingested after the MISBR granules.		
26	V-4 Verify that the second set of exports, after S-4, contains science granule metadata for the MISR Level 1 & 2 ingested in S-4.		#comment
27	Verify the body of each MISR PUT request contains the full granule metadata.		
28	V-5 Verify that the granule metadata for the ingests in S-4, includes Browse link URLs referring to the appropriate MISBR granules ingested in S-2, and that these URLs are correct.		#comment
29	Verify the body of each MISR PUT request contains an OnlineResource URL: xpath '//OnlineResource/URL' misr.xml		
30	Verify each OnlineResource URL points to the correct MISBR, according to MISR Level 1 and Level 2 matching rules: Find the association first through AmBrowseGranuleXref, then the misbrids in AmBrowse using the browseids, then use ProcGetGrFiles to get the file information, then go to tcp.log to grep the BRWS for the SC granule.		
31	Verify each OnlineResource URL points to the correct browse file in the data pool.		
32	V-6 Verify that the URLs exported for the MIB2GEOP granules show them linked with MISBR granules that were selected according to the rules that apply to MISR Level 2 granules.		#comment
33	Verify each OnlineResource URL in the exported MIB2GEOP metadata points to the correct MISBR, according to MISR Level 2 matching rules. To find a MISBR matching an ingested MISRSC granule, select b.misrbrid from AmBrowseGranuleXref bgx join		

#	Action	Expected Result	Notes
	ambrowse b on bgx.browseid = b.browseid where bgx.granuleid = <MISRSC_granuleId>		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
115				At least one granule from a MISR Level 1 ESDT, and for each a MISBR granule that can be associated with it (cameraIds match, its temporal coverage intersects that of the MISR Level 1 granule and it has a matching SP_AM_MISR_ProductVersion product specific attribute value).				
115				At least one granule from a MISR Level 2 ESDT and for each a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MISR Level 2).				
115				At least one MIB2GEOP granule and for each a MISBR granule that can be associated with it (i.e., has a cameraId of 'AN' and a temporal coverage that intersects with that of the MIB2GEOP granule).				

EXPECTED RESULTS:

	V	115	1	Verify that the export in S-3 contains the full granule metadata for the MISBR granules ingested in S-2.		
	V	115	2	Verify that the metadata exported in S-3 contains no browse URLs.		
	V	115	3	Verify that the granules ingested in S-4 are picked up and exported by the automatic polling process.		
	V	115	4	Verify that the second set of exports, after S-4, contains science granule metadata for the MISR Level 1 & 2 ingested in S-4.		

	V	115	5	Verify that the granule metadata for the ingests in S-4, includes Browse link URLs referring to the appropriate MISBR granules ingested in S-2, and that these URLs are correct.		
	V	115	6	Verify that the URLs exported for the MIB2GEOP granules show them linked with MISBR granules that were selected according to the rules that apply to MISR Level 2 granules.		

1.35 Nominal Automatic Export (ECS-ECSTC-34)

DESCRIPTION:

	S	120	1	**[Nominal Automatic Export]** Start the BMGT core application (Dispatcher) and event poller (Event Driver), specifying the ECS mode in which to run.		
	S	120	2	Ensure that there are no pending or in process BMGT exports, then suspend both BMGT processing (Dispatcher) and polling for automatic export events (Event Driver).		
	S	120	3	Configure BMGT to poll for new events every 30 seconds, and configure the maximum number of events to enqueue at a time to be 100.		
	S	120	4	Perform the following operations, each on a different collection or granule: a) Install 2 new collections b) Delete 1 collection c) Ingest 10 granules d) Publish 10 granules and unpublish 10 more e) Perform 5 of each of the following: a. Browse linkage b. QA Linkage c. PH linkage d. HDF Map Linkage f) Delete 10 granules (5 logically, 5 physically) g) Publish a MISBR granule which is linked to at least 1 MISR Level 1 or 2 granule. h) Publish an MISR Level 1 or 2 granule which is linked to a MISBR granule. Ensure that no other events are performed on these items during the course of this test.		
	S	120	5	Resume polling for automatic export events, but not BMGT export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure that no one else uses the mode during the course of this test.		
3	Ensure BMGT configuration is current and correct (config files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available, connected to the ecs database.		
5	Ensure collections C1, C2 are not yet installed.		
6	Ensure collections C3, C4 are installed.		
7	Ensure collections C3, C4 are enabled for collection and granule export.		
8	Ensure granules g1 .. g10 are not yet in AIM.		
9	Ensure granules g11 .. g20, from collection C4, are in the hidden data pool and their granule IDs are in a text file, such as granuleids_g11-g20.txt.		
10	Ensure granules g21 .. g30, from collection C4, are in the public data pool and their granule IDs are in a text file, such as granuleids_g21-g30.txt.		
11	Ensure granules g31 .. g50, from collection C4, are in the public data pool and their granule IDs are in a text file, such as granuleids_g31-g50.txt.		
12	Ensure granules g51 .. g60 are in the public data pool and their granule IDs are in a text file, such as granuleids_g51-g60.txt.		
13	Ensure collection C4 is configured to not be public on ingest.		
14	Ensure collection C4 is configured to not create HDF maps on ingest.		
15	Ensure MISR granule m1 is linked to MISBR granule b1: select * from dsmdmisrbrowsegranulexref where granuleid = m1_GRANULEID and browseid = b1_GRANULEID		
16	Ensure MISR granule m2 is linked to MISBR granule b2: select * from dsmdmisrbrowsegranulexref where granuleid = m2_GRANULEID and browseid = b2_GRANULEID		
17	Ensure m1, b1, m2, b2 are in the public data pool.		
18	Ensure m1, m2 have associated browse granules. These should have been created when the public b1 and b2 were linked to the public m1 and m2. select * from ambrowsegranulexref where granuleid in (m1_GRANULEID, m2_GRANULEID)		
19	Unpublish b1, m2.		
20	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
21	Note the time before creating events as t0.		

#	Action	Expected Result	Notes
22	Setup		#comment
23	S-1 Start the BMGT core application (Dispatcher) and event poller (Event Driver), specifying the ECS mode in which to run.		#comment
24	ssh to the BMGT host (normal site map: x4oml01; alternate site map: x4spl01).		
25	If testing in a DEV* mode, set an appropriate view: c setview ${BMGT_VIEW}		
26	Determine whether BMGT dispatcher is running: ps auxww | grep $MODE | grep EcBmDispatcher		
27	If the BMGT dispatcher is not running, start it: cd /usr/ecs/$MODE/CUSTOM/utilities ./EcBmBMGTDispatcherStart $MODE		
28	Determine whether BMGT auto driver is running: ps auxww | grep $MODE | grep EcBmAuto		
29	If the BMGT auto driver is not running, start it: cd /usr/ecs/$MODE/CUSTOM/utilities ./EcBmBMGTAutoStart $MODE		
30	S-2 Ensure that there are no pending or in process BMGT exports, then suspend both BMGT processing (Dispatcher) and polling for automatic export events (Event Driver).		#comment
31	Ensure that there are no pending exports by checking the GUI or by querying the database: select * from bg_export_request where status = 'PENDING';		
32	If there are PENDING requests, move them to CANCELED (note that there is only 1 "L"): update bg_export_request set status = 'CANCELED' where status = 'PENDING'		
33	Suspend the dispatcher via the GUI: On the "System Status" tab, on the "Dispatcher" row, click the "Pause" button.		
34	Suspend the auto driver via the GUI: On the "System Status" tab, Pause the EVENT queue.		
35	S-3 Configure BMGT to poll for new events every 30 seconds, and configure the maximum number of events to enqueue at a time to be 100.		#comment
36	In the BMGT GUI, on the "BMGT Configuration" tab, set BMGT.AutoDriver.PollingFrequency to 30000 (30 000 ms).		
37	In the BMGT GUI, on the "BMGT Configuration" tab, set BMGT.AutoDriver.MaxEvents to 100.		
38	S-4 Perform the following operations, each on a different collection or		#comment

#	Action	Expected Result	Notes
	granule:		
39	a) Install 2 new collections		#comment
40	Use the ESDT Maintenance GUI to install collections C1 and C2.		
41	Verify C1 and C2 have been added to AIM: select * from amcollection where (shortname = 'C1_SHORTNAME' and versionid = C1_VERSIONID) or (shortname = 'C2_SHORTNAME' and versionid = C2_VERSIONID)		
42	Verify an event was generated for each of C1 and C2: select * from dsmdgreventhistory where eventtime > (now() - interval '10 minutes') and ((shortname = 'C1_SHORTNAME' and versionid = C1_VERSIONID) or (shortname = 'C2_SHORTNAME' and versionid = C2_VERSIONID)) and eventtype = 'CLINSERT'		
43	b) Delete 1 collection		#comment
44	Use the ESDT Maintenance GUI to delete collection C3.		
45	Verify C3 has been removed from AIM: select * from amcollection where shortname = 'C3_SHORTNAME' and versionid = C3_VERSIONID		
46	Verify a row has been added to dsmdgreventhistory for C3: select * from dsmdgreventhistory where eventtime > (now() - interval '10 minutes') and shortname = 'C3_SHORTNAME' and versionid = C3_VERSIONID and eventtype = 'CLDELETE'		
47	c) Ingest 10 granules		#comment
48	Ingest granules g1 .. g10 and save their granule IDs to at text file: granuleids_g01-g10.txt.		
49	Verify g1..g10 have been added to AIM: select archivetime, esdt(shortname, versionid), granuleid, publishtime, isorderonly, deleteeffectivedate, deletefromarchive from amgranule where granuleid in (${g1_GRANULEID}, ..., ${g10_GRANULEID})		
50	Verify an event was generated for each of g1..g10: select * from dsmdgreventhistory where eventtime > (now() - interval '10 minutes') and dbid in (${g1_GRANULEID}, ..., ${g10_GRANULEID})		
51	d) Publish 10 granules and unpublish 10 more		#comment
52	Publish granules g11 .. g20: ./EcDlPublishUtilityStart ${MODE} -ecs -file /path/to/granuleids_g11-g20.txt		
53	Verify g11..g20 have been published: select archivetime, esdt(shortname, versionid), granuleid, publishtime, isorderonly,		

#	Action	Expected Result	Notes
	deleteeffectivedate, deletefromarchive from amgranule where granuleid in (${g11_GRANULEID}, ..., ${g20_GRANULEID}) and publishtime is not null and isorderonly is null		
54	Verify an event was generated for each of g11..g20: select * from dsmdgreventhistory where eventtime > (now() - interval '10 minutes') and dbid in (${g11_GRANULEID}, ..., ${g20_GRANULEID})		
55	Unpublish granules g21 .. g30: ./EcDlUnpublishStart.pl -mode <MODE> -f /path/to/granuleids_g21-g30.txt		
56	Verify g21..g30 have been unpublished: select archivetime, esdt(shortname, versionid), granuleid, publishtime, isorderonly, deleteeffectivedate, deletefromarchive from amgranule where granuleid in (${g21_GRANULEID}, ..., ${g30_GRANULEID}) and publishtime is null and isorderonly is not null		
57	e) Perform 5 of each of the following: a. Browse linkage b. QA Linkage c. PH linkage d. HDF Map Linkage		#comment
58	Ingest 5 browse linkages, linking browse granules to granules g31 .. g35.		
59	Verify granules g31..g35 are each linked to browse granules: select * from ambrowsegranulexref where granuleid in (${g31_GRANULEID}, ..., ${g35_GRANULEID})		
60	Verify an event was added for each linked browse: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${g31_GRANULEID}, ..., ${g35_GRANULEID}) and h.eventtime > (now() - interval '10 minutes');		
61	Ingest 5 QA linkages, linking QA granules to granules g36 .. g40.		
62	Verify granules g36..g40 are each linked to QA granules: select * from amqagranulexref where granuleid in (${g36_GRANULEID}, ..., ${g40_GRANULEID})		
63	Verify an event was added for each QA linkage: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${g36_GRANULEID}, ..., ${g40_GRANULEID}) and h.eventtime > (now() - interval '10 minutes');		
64	Ingest 5 PH linkages, linking PH granules to granules g41 .. g45.		
65	Verify granules g41..g45 are each linked to QA granules: select * from amphgranulexref where granuleid in		

#	Action	Expected Result	Notes
	(${g40_GRANULEID}, ..., ${g45_GRANULEID})		
66	Verify an event was added for each PH linkage: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${g40_GRANULEID}, ..., ${g45_GRANULEID}) and h.eventtime > (now() - interval '10 minutes');		
67	Generate HDF Maps for granules g46 .. g50: ./EcAmInsertMapGenerationRequest.pl -m <MODE> -f /path/to/granuleids_g46-g50.txt		
68	Verify granules g46..g50 are each linked to HDF MAP granules: select * from amhdfmapgranulexref where granuleid in (${g46_GRANULEID}, ..., ${g50_GRANULEID})		
69	Verify an event was added for each HDF MAP linkage: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${g46_GRANULEID}, ..., ${g50_GRANULEID}) and h.eventtime > (now() - interval '10 minutes');		
70	f) Delete 10 granules (5 logically, 5 physically)		#comment
71	Logically delete granules g51 .. g60: ./EcDsBulkDelete.pl -physical -user EcDsAmGranuleDeletion -geoidfile /path/to/geoids_g51-g60.txt		
72	Verify g51..g60 have been marked deleted: select archivetime, esdt(shortname, versionid), granuleid, publishtime, isorderonly, deleteeffectivedate, deletefromarchive from amgranule where granuleid in (${g51_GRANULEID}, ..., ${g60_GRANULEID}) and deleteeffectivedate is not null		
73	Verify an event was added for each granule delete: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${g51_GRANULEID}, ..., ${g60_GRANULEID}) and h.eventtime > (now() - interval '10 minutes');		
74	Physically delete granules g56 .. g60:		#comment
75	Unpublish granules g56 .. g60: ./EcDlUnpublishStart.pl -mode ${MODE} -f /path/to/granuleids_g56-g60.txt		
76	Run DeletionCleanup, choosing the most destructive option at every opportunity: ./EcDsDeletionCleanup.pl -user EcDsAmGranuleDeletion -mode ${MODE} -server ${DB_SERVER} -		

#	Action	Expected Result	Notes
	database ecs -batch 10000 -grbatch 100 -xmlbatch 1000 -databatch 10000 -logbatch 100		
77	g) Publish a MISBR granule which is linked to at least 1 MISR Level 1 or 2 granule.		#comment
78	Publish MISBR granule b1: ./EcDlPublishUtilityStart ${MODE} -ecs -g ${b1_GRANULEID}		
79	Verify b1 is public: select archivetime, esdt(shortname, versionid), granuleid, publishtime, isorderonly, deleteeffectivedate, deletefromarchive from amgranule where granuleid = ${b1_GRANULEID} and publishtime is not null and isorderonly is null		
80	Verify no event was added for m1 or b1: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${m1_GRANULEID}, ${b1_GRANULEID}) and h.eventtime > (now() - interval '2 minutes');		
81	h) Publish a MISR Level 1 or 2 granule which is linked to a MISBR granule.		#comment
82	Publish MISR granule m2: ./EcDlPublishUtilityStart ${MODE} -ecs -g ${m2_GRANULEID}		
83	Verify m2 is public: select archivetime, esdt(shortname, versionid), granuleid, publishtime, isorderonly, deleteeffectivedate, deletefromarchive from amgranule where granuleid = ${m2_GRANULEID} and publishtime is not null and isorderonly is null		
84	Verify events were added for m2 and b2: select esdt(g.shortname, g.versionid), g.granuleid from dsmdgreventhistory h join amgranule g on h.dbid = g.granuleid where g.granuleid in (${m2_GRANULEID}, ${b2_GRANULEID}) and h.eventtime > (now() - interval '2 minutes');		
85	Ensure that no other events are performed on these items during the course of this test.		#comment
86	S-5 Resume polling for automatic export events, but not BMGT export.		#comment
87	Record time as t_auto_start_time.		
88	In the BMGT GUI, on the "System Status" tab, Resume the EVENT queue.		

#	Action	Expected Result	Notes
89	Verification		#comment
90	V-1 Verify that the BMGT starts polling for events to process following its resumption in S-5.		#comment
91	Verify the BMGT GUI, "Export Requests" tab begins displaying requests for all the events created above.		
92	Verify that bg_export_request table contains the queued requests: select * from bg_export_request where enqueuetime > t0		
93	V-2 Verify that when BMGT starts polling for events, it prints a message to the log, followed by another message when it completes polling. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which polling started and completed, as well as how many events were found and enqueued..		#comment
94	Verify that bmgt_log is written to /usr/ecs/${MODE}/CUSTOM/logs/EcBmBMGTAutomaticDriver.log		
95	Verify that the bmgt_log contains a message indicating the time that BMGT started polling for events.		
96	Verify that the bmgt_log contains a message indicating the time that BMGT completed polling for events.		
97	Verify that the bmgt log contains a message indicating the number of events found in the polling cycle.		
98	Verify that the bmgt log contains a messge indicating the number of events queued in the polling cycle.		
99	V-3 Verify that all of the events in S-4 are added to the BMGT queue within 30 seconds of the polling process being resumed. Verify that they are displayed in the BMGT GUI and listed as automatic export requests resulting from events.		#comment
100	Verify that the BMGT GUI "Export Requests" tab lists all the events generated above.		
101	Verify that the events listed on the BMGT GUI "Export Requests" tab all have "Export Queue" values of "EVENT" (the BMGT GUI calls the auto driver the "Catalog Event Driver").		
102	Verify that the enqueuetime of the final event queued is less than t_dispatcher_start_time + 30 s.		
103	S-6 Resume BMGT Export Processing		#comment
104	In the BMGT GUI on the "System Status" tab, click the		

#	Action	Expected Result	Notes
	"Resume" button on the "Dispatcher" row.		
105	V-4 Verify that all of the events in S-4 are eventually exported to ECHO (or an ECHO stand-in). Verify that inserts are exported as HTTP PUTs and deletes as HTTP DELETEs, appropriately to the type of operation that was performed in S-4.		#comment
106	Verify that the BMGT GUI "Export Request" tab eventually lists all events enqueued during the test as having "SUCCESS" status.		
107	Use the TCP proxy (or mock ECHO) log to verify events were exported as appropriate HTTP actions.		#comment
108	Verify metadata for collections C1, C2 were exported as HTTP PUTs, containing full collection metadata.		
109	Verify an HTTP DELETE was exported for collection C3.		
110	Verify metadata for granules g1..g50 were exported as HTTP PUTs, containing full granule metadata.		
111	Verify HTTP DELETEs were exported for granules g51..g60.		
112	Verify metadata for granules m1, b1, m2, b2 were exported as HTTP PUTs, containing full granule metadata.		
113	V-5 Verify that the database and the log files (at ‘info’ level) contain information on the process of each event through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		#comment
114	Choose a small number of events to spot check.		
115	Verify the Dispatcher log shows when each event was generated.		
116	Verify the Dispatcher log shows when each event was exported to ECHO.		
117	Verify the Dispatcher log shows when each response was received from ECHO.		
118	Verify that the metadata generation time (starttime), export time, and response received time (completiontime) are recorded for each event: select c.shortname, c.versionid, r.granuleid, a.starttime, a.exporttime, a.completiontime from bg_export_activity a join bg_export_request r on a.requestid = r.requestid join bg_collection_configuration c on r.collectionid = c.collectionid where r.granuleid in (${GRANULEIDS}) or r.collectionid in (${COLLECTIONIDS})		
119	V-6 Verify that the BMGT GUI displays the completed export events, indicating that they were the result of an automatic, event driven export. Verify that it indicates that the events were successfully exported and		#comment

#	Action	Expected Result	Notes
	indicates the time of export as well as granule or collection ID.		
120	Verify the BMGT GUI "Export Activity/Activity-Errors" tab lists each event from this test as having "Export Queue" = "EVENT" and "Status" = "SUCCESS".		
121	Verify the BMGT GUI "Export Activity/Activity-Errors" tab lists the "Export DateTime" for each event from this test.		
122	Verify the BMGT GUI "Export Activity/Activity-Errors" tab lists the collection ID for each collection event from this test. E.g., CL:MIL3YAL.005:87801		
123	Verify the BMGT GUI "Export Activity/Activity-Errors" tab lists the granule ID for each granule event from this test. E.g., SC:MOD29P1D.005:123456		
124	V-7 Verify that for each item in S-4, the following was exported: a) 2 HTTP PUT requests containing full collection metadata. b) 1 HTTP DELETE request. c) 10 HTTP PUT requests containing full granule metadata. d) 20 HTTP PUT requests containing full granule metadata. The published granules must have Online Access and Online Resource URLs, and the unpublished must not. e) 20 HTTP PUT requests containing full granule metadata. f) 10 HTTP DELETE requests. g) No PUT or DELETE for the MISBR granule or the MISR science granules linked to the MISBR granule. h) 1 HTTP PUT request containing full granule metadata for the published granule, and containing the correct browse link URL for this linked MISBR.		#comment
125	PUTs, DELETEs, and existence of exported metadata were verified as part of V-4.		#comment
126	d) Verify granules g11..g20 have OnlineAccess URLs, using an xpath utility: xpath '//OnlineAccessURL/URL' granule_g1[1-9].xml granule_g20.xml		
127	d) Verify granules g11..g20 have OnlineResource URLs, using an xpath utility: xpath '//OnlineResource/URL' granule_g1[1-9].xml granule_g20.xml		
128	d) Verify granules g21..g30 have no OnlineAccess or OnlineResource URLs, using an xpath utility: xpath '(//OnlineAccessURL|//OnlineResource)/URL' granule_g2[1-9].xml granule_g30.xml		
129	g) Verify the TCP proxy log shows no PUT for either b1 or b2 (the MISBR granules).		

#	Action	Expected Result	Notes
130	g) Verify the TCP proxy log shows no PUT for either m1 or m2 (the MISR granules).		
131	h) Verify the TCP proxy log shows a PUT for m2 with full granule metadata.		
132	h) Verify m2's exported metadata contains an Online Resource URL, linking it to b2.		

TEST DATA:

Crit 120 test data locations are relative to **/sotestdata/DROP_802/BE_82_01/Criteria/120**

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
S4-a		2 collections						
S4-c		10 granules						
S4-d		10 granules						
S4-e-a		5 browse granules						
S4-e-b		5 QA granules						
S4-e-c		5 PH granules						
S4-e-d		5 science granules which can be enabled for HDF Map generation						
S4-g		1 MISBR granule (b1) linked to a MISR granule (m1)						
S4-h		1 MISR granule (m2) linked to a MISBR granule (b2)						

EXPECTED RESULTS:

	V	120	1	Verify that the BMGT starts polling for events to process following its resumption in S-5.		

	V	120	2	Verify that when BMGT starts polling for events, it prints a message to the log, followed by another message when it completes polling. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which polling started and completed, as well as how many events were found and enqueued..		
	V	120	3	Verify that all of the events in S-4 are added to the BMGT queue within 30 seconds of the polling process being resumed. Verify that they are displayed in the BMGT GUI and listed as automatic export requests resulting from events.		
	V	120	4	Resume BMGT Export processing. Verify that all of the events in S-4 are eventually exported to ECHO (or an ECHO stand-in). Verify that inserts are exported as HTTP PUTs and deletes as HTTP DELETEs, appropriately to the type of operation that was performed in S-4.		
	V	120	5	Verify that the database and the log files (at 'info' level) contain information on the process of each event through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		
	V	120	6	Verify that the BMGT GUI displays the completed export events, indicating that they were the result of an automatic, event driven export. Verify that it indicates that the events were successfully exported and indicates the time of export as well as granule or collection ID.		
	V	120	7	Verify that for each item in S-4, the following was exported: a) 2 HTTP PUT requests containing full collection metadata. b) 1 HTTP DELETE request. c) 10 HTTP PUT requests containing full granule metadata. d) 20 HTTP PUT requests containing full granule metadata. The published granules must have Online Access and Online Resource URLs, and the unpublished must not. a. 20 HTTP PUT requests containing full granule metadata. e) 10 HTTP DELETE requests. f) 1 HTTP PUT request containing the full granule metadata for the MISBR granule (including online Access/Resource URLs) and 1 HTTP PUT request containing full granule metadata for each MISR granule linked to the MISBR granule. This metadata must contain the browse link URL for the published MISBR. g) 1 HTTP PUT request containing full granule metadata for the published granule, and containing the correct browse link URL for this linked MISBR.		

1.36 Automatic Export Event Consolidation (ECS-ECSTC-35)

DESCRIPTION:

	S	130	1	**[Automatic Export Event Consolidation]** Configure the maximum number of events for BMGT to enqueue at a time to be 100.		
	S	130	2	Suspend BMGT polling for new events.		
	S	130	3	Insert a new collection, update that same collection.		
	S	130	4	Insert a new collection, delete that same collection.		
	S	130	5	Ingest a granule, update the same granule (e.g. browse link, publish/unpublish, etc).		
	S	130	6	Ingest a granule, logically delete the same granule.		
	S	130	7	Ingest a granule, physically delete the same granule.		
	S	130	8	Ingest a granule which is in a public collection, manually remove (or mark as already picked up by BMGT) all of the events for this insert. Unpublish the same granule.		
	S	130	9	Ingest a granule which is not in a public collection, manually remove (or mark as already picked up by BMGT) all of the events for this insert. Publish the same granule.		
	S	130	10	Resume BMGT polling for new events to export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure an ingest provider is configured with an active polling location.		
4	Ensure collections C1, C2 are not installed.		

#	Action	Expected Result	Notes
5	Ensure collection for granules g1 .. g4 is configured to be public on ingest.		
6	Ensure the collection for granule g5 is configured to not be public on ingest.		
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 Configure the maximum number of events for BMGT to enqueue at a time to be 100.		#comment
10	Use the BMGT GUI to configure the property "Bmgt.AutoDriver.MaxEvents" to be at least 100. The number must be greater than the number of events that will be created during this test.		
11	S-2 Suspend BMGT polling for new events.		#comment
12	Stop the automatic BMGT driver: EcBmBMGTAutoStop <MODE>		
13	Note the current time as t0.		
14	S-3 Insert a new collection, update that same collection.		#comment
15	Copy collection C1's descriptor file to /usr/ecs/<MODE>/CUSTOM/data/ESS/ on the host running the ESDT Maintenance GUI.		
16	Follow the wiki instructions, stopping before the DPL Maintenace GUI: http://edhs1.gsfc.nasa.gov:40000/bin/view/EDF/AddESDT Use the ESDT Maintenance GUI to install the collection.		
17	Insert or update the row in bg_collection_conguration with collection and granule export flags set to 'Y'. This will export a PUT for C1; it will not create an event.		
18	Copy collection C1's updated descriptor file to /usr/ecs/TS3/CUSTOM/data/ESS/ on the host running the ESDT Maintenance GUI.		
19	Use the ESDT Maintenance GUI to update collection C1. Copy C1's descriptor back to the same directory and modify the Description. Make sure you note your modification so you can look for it later. Then use the Maintenance GUI to update the collection		
20	S-4 Insert a new collection, delete that same collection.		#comment
21	Copy collection C2's descriptor file to /usr/ecs/TS3/CUSTOM/data/ESS/ on the host running the ESDT Maintenance GUI.		
22	Follow the wiki instructions, stopping before the DPL Maintenace GUI: http://edhs1.gsfc.nasa.gov:40000/bin/view/EDF/AddESDT Use the ESDT Maintenance GUI to install the collection.		

#	Action	Expected Result	Notes
23	Insert or update the row in bg_collection_configuration with collection and granule export flags set to 'Y'. This will export a PUT for C2; it will not create an event.		
24	Use the ESDT Maintenance GUI to delete collection C2.		
25	S-5 Ingest a granule, update the same granule (e.g. browse link, publish/unpublish, etc).		#comment
26	Copy granule g1's PDR into the polling location and ensure that publication is enabled for the associated collection.		
27	Wait for the Ingest GUI to indicate granule g1 has completed processing.		
28	Unpublish granule g1: EcDlUnpublishStart.pl -mode <MODE> -g <GRANULE_ID>		
29	S-7 Ingest a granule, physically delete the same granule.		#comment
30	Copy granule g3's PDR into the polling location.		
31	Wait for the Ingest GUI to indicate granule g3 has completed processing.		
32	Create a geoid file for granule g3: SC:<SHORT_NAME>.<VERSION_ID>:<GRANULE_ID>		
33	Move granule g3 to the trash: EcDsBulkDelete.pl -mode <MODE> -physical -geoidfile /path/to/geoid_file		
34	Unpublish granule g3: EcDlUnpublishStart.pl -mode <MODE> -g <GRANULE_ID>		
35	Empty the trash: EcDsDeletionCleanup.pl -mode <MODE> Follow the prompts, always choosing the most destructive option.		
36	S-6 Ingest a granule, logically delete the same granule.		#comment
37	Copy granule g2's PDR into the polling location.		
38	Wait for the Ingest GUI to indicate granule g2 has completed processing.		
39	Create a geoid file for granule g2: SC:<SHORT_NAME>.<VERSION_ID>:<GRANULE_ID>		
40	Logically delete granule g2: EcDsBulkDelete.pl -mode <MODE> -physical -geoidfile /path/to/geoid_file		
41	S-8 Ingest a granule which is in a public collection, manually remove (or mark as already picked up by BMGT) all of the events for this insert. Unpublish the same granule.		#comment
42	Copy granule g4's PDR into the polling location.		

#	Action	Expected Result	Notes
43	Wait for the Ingest GUI to indicate granule g4 has completed processing.		
44	Mark granule g4's events as already picked up by BMGT: update dsmdgreventhistory set bmgtpickuptime = now() where dbid = <GRANULE_ID>		
45	Unpublish granule g4: EcDlUnpublishStart.pl -mode <MODE> -g <GRANULE_ID>		
46	S-9 Ingest a granule which is not in a public collection, manually remove (or mark as already picked up by BMGT) all of the events for this insert. Publish the same granule.		#comment
47	Copy granule g5's PDR into the polling location.		
48	Wait for the Ingest GUI to indicate granule g5 has completed processing.		
49	Mark granule g5's events as already picked up by BMGT: update dsmdgreventhistory set bmgtpickuptime = now() where dbid = <GRANULE_ID>		
50	Publish granule g5: EcDlPublishUtilityStart <MODE> -ecs -g <GRANULE_ID>		
51	S-10 Resume BMGT polling for new events to export.		#comment
52	Start the automatic BMGT driver: EcBmBMGTAutoStart <MODE>		
53	Verification		#comment
54	V-1 Verify that events were generated in the AIM event table for each of the actions performed above. Each granule or collection used (with the possible exception of S-8 and S-9) should have at least 2 events generated.		#comment
55	Verify dsmdgreventhistory has 1 CLINSERT and 1 CLUPDATE event for collection C1.		
56	Verify dsmdgreventhistory has 1 CLINSERT and 1 CLDELETE event for collection C2.		
57	Verify dsmdgreventhistory has 1 GRINSERT, 1 GRURLINS, and 1 GRURLDEL event for granule g1.		
58	Verify dsmdgreventhistory has 1 GRINSERT, 1 GRURLINS, and 1 GRDELETE event for granule g2.		
59	Verify dsmdgreventhistory has 1 GRINSERT, 1 GRURLINS, and 1 GRDELETE event for granule g3.		
60	Verify dsmdgreventhistory has 1 GRURLDEL event for granule g4.		
61	Verify dsmdgreventhistory has 1 GRURLINS event for granule g5.		

#	Action	Expected Result	Notes
62	V-2 Verify that the events have been picked up by BMGT (once the polling interval kicks off). Verify that they are marked as picked up by BMGT in the AIM Event table.		#comment
63	Verify the BMGT log shows that events in S-3 through S-9 are picked up.		
64	Verify the BMGT event rows in dsmdgreventhistory are updated to indicate the events have been picked up (bmgtpickuptime should be set): select dbid, eventtype, bmgtpickuptime from dsmdgreventhistory where eventtype is not null and (dbid in (<GRANULE_IDS>) or collectionid in (<COLLECTION_IDS))		
65	V-3 Verify that each of the items (collection or granule) for which there were events appears exactly once in the BMGT export request queue (viewed through the GUI).		#comment
66	Verify the BMGT GUI shows one export for each collection C1, C2 after time t0, excluding the insert request triggered by enabling the collections for export.		
67	Verify the BMGT GUI shows exactly one export for each granule g1 ... g5 after time t0.		
68	V-4 Verify that each of the enqueued export requests results in exactly one export to ECHO (or an ECHO stand-in).		#comment
69	Verify the TCP proxy records exactly 1 HTTP request for each corresponding request that appears in the BMGT GUI after time t0.		
70	V-5 For the collection in S-3, verify that the export is in the form of an HTTP PUT request containing the entire collection metadata, reflecting the updates that were performed.		#comment
71	Verify collection C1's second request was a PUT.		
72	Verify collection C1's second request body contained complete XML metadata.		
73	Verify collection C1's exported metadata includes the update. Check for the updated Description in the tcp.log.		
74	V-6 For the collection in S-4, verify that the export is in the form of an HTTP DELETE request containing no metadata. Note that despite the fact that the collection was never inserted, we export a deletion anyway, as it simplifies the logic and does not cause an error.		#comment
75	Verify collection C2's second request was a DELETE.		
76	Verify collection C2's second request has an empty body.		
77	V-7 For the granule in S-5, verify that the export is in the form of an		#comment

#	Action	Expected Result	Notes
	HTTP PUT request containing the entire granule metadata reflecting the granule state after the update was made.		
78	Verify granule G1's request was a PUT.		
79	Verify granule G1's request body contained complete XML metadata.		
80	Verify granule G1's exported metadata includes the update. Make sure the OnlineAccess or OnlineResource Tags are not present.		
81	V-8 For the granule in S-6 and S-7, verify that the export is in the form of an HTTP DELETE request containing no metadata. Note that despite the fact that the granule was never inserted, we export a deletion anyway, as it simplifies the logic and does not cause an error.		#comment
82	Verify granule G2's request was a DELETE.		
83	Verify granule G2's request has an empty body.		
84	Verify granule G3's request was a DELETE.		
85	Verify granule G3's request has an empty body.		
86	V-9 For the granule in S-8, verify that the export is in the form of an HTTP PUT containing the entire granule metadata. Verify that it does not contain any datapool URLs as the granule has been unpublished.		#comment
87	Verify granule G4's request was a PUT.		
88	Verify granule G4's request body contained complete XML metadata.		
89	Verify granule G4's exported metadata contains no online access or online resource URLs: xpath '//OnlineResource/URL|//OnlineAccessURL/URL' granule.xml		
90	V-10 For the granule in S-9, verify that the export is in the form of an HTTP PUT containing the entire granule metadata. Verify that it contains datapool URLs.		#comment
91	Verify granule G5's request was a PUT.		
92	Verify granule G5's request body contained complete XML metadata.		
93	Verify granule G5's exported metadata contains online access or online resource URLs pointing to the data pool: xpath '//OnlineResource/URL|//OnlineAccessURL/URL' granule.xml		

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		Collection C1 to install and update					/sotestdata/DROP_802/BE_82_01/Criteria/130/GLAH09.033	
		A modified descriptor file with which to update C1 after installing it					/sotestdata/DROP_802/BE_82_01/Criteria/130/GLAH09.033/replacement	
		Collection C2 to install and delete					/sotestdata/DROP_802/BE_82_01/Criteria/130/GLAH10.033	
		5 granules g1..g5					/sotestdata/DROP_802/BE_82_01/Criteria/130/MOD29P1D.005 /sotestdata/DROP_802/BE_82_01/Criteria/130/MOD29P1N.005	

EXPECTED RESULTS:

	V	130	1	Verify that events were generated in the AIM event table for each of the actions performed above. Each granule or collection used (with the possible exception of S-8 and S-9) should have at least 2 events generated.		
	V	130	2	Verify that the events have been picked up by BMGT (once the polling interval kicks off). Verify that they are marked as picked up by BMGT in the AIM Event table.		
	V	130	3	Verify that each of the items (collection or granule) for which there were events appears exactly once in the BMGT export request queue (viewed through the GUI).		

	V	130	4	Verify that each of the enqueued export requests results in exactly one export to ECHO (or an ECHO stand-in).		
	V	130	5	For the collection in S-3, verify that the export is in the form of an HTTP PUT request containing the entire collection metadata, reflecting the updates that were performed.		
	V	130	6	For the collection in S-4, verify that the export is in the form of an HTTP DELETE request containing no metadata. Note that despite the fact that the collection was never inserted, we export a deletion anyway, as it simplifies the logic and does not cause an error.		
	V	130	7	For the granule in S-5, verify that the export is in the form of an HTTP PUT request containing the entire granule metadata reflecting the granule state after the update was made.		
	V	130	8	For the granule in S-6 and S-7, verify that the export is in the form of an HTTP DELETE request containing no metadata. Note that despite the fact that the granule was never inserted, we export a deletion anyway, as it simplifies the logic and does not cause an error.		
	V	130	9	For the granule in S-8, verify that the export is in the form of an HTTP PUT containing the entire granule metadata. Verify that it does not contain any datapool URLs as the granule has been unpublished.		
	V	130	10	For the granule in S-9, verify that the export is in the form of an HTTP PUT containing the entire granule metadata. Verify that it contains datapool URLs.		

1.37 Unexported Events (ECS-ECSTC-36)

DESCRIPTION:

	S	140	1	**[Unexported Events]** Suspend BMGT polling for new events.		
	S	140	2	Add to the AIM event queue events which are well in the past (e.g. 1 day) and which precede existing events which have been picked up and exported. You could do this by modifying the event times of existing events and marking them as not having been picked up by BMGT. Ensure that both Granule and collection events are included, as well as inserts, updates, and deletions.		
	S	140	3	Resume BMGT polling for new events.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collections C1, C2 are installed.		
3	Ensure collections C1, C2 are configured for collection and granule export.		
4	Ensure granules G1, G2, G3, and G4 have been ingested.		
5	Ensure granules G2, G4 have been logically deleted.		
6	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
7	Setup		#comment
8	S-1 Suspend BMGT polling for new events.		#comment
9	Stop the automatic BMGT driver: EcBmBMGTAutoStop <MODE>		
10	S-2 Add to the automatic export request queue events which are well in the past (e.g. 1 day) and which precede existing events which have been exported. Ensure that both Granule and collection events are included, as well as inserts, updates, and deletions.		#comment
11	Update the event history table to make it appear that collection C1 and granules G1, G2 were queued 2 days ago but have not yet been picked up: For each collection do: insert into bg_export_request (enqueuetime, collectionid, itemtype, exporttype, exportqueue, status) values (now() - '2 days'::interval, <COLLECTION_ID>, 'CL', 'OPEN', 'EVENT', 'PENDING') and for each granule: insert into bg_export_request (enqueuetime, collectionid, granuleid, itemtype, exporttype, exportqueue, status) values (now() - '2 days'::interval, <COLLECTION_ID>, <GRANULE_ID>, 'SC', 'OPEN', 'EVENT', 'PENDING') 		
12	Ensure that there are export requests with eventTime > the time set in the previous step, with a status of SUCCESS.		
13	S-3 Resume BMGT polling for new events.		#comment
14	Start the automatic BMGT driver: EcBmBMGTAutoStart <MODE>		
15	Verification		#comment
16	V-1 Ensure that the unexported requests are exported, with the appropriate action for the referenced items current state (i.e. DELETE for a		#comment

#	Action	Expected Result	Notes
	deleted item, PUT for a non deleted item)..</i>		
17	Verify the TCP proxy log shows an HTTP PUT request for collection C1.		
18	Verify the TCP proxy log shows collection C1's PUT included complete collection metadata.		
19	Verify the TCP proxy log shows an HTTP PUT request for granule G1		
20	Verify the TCP proxy log shows granule G1's PUT included comlete granule metadata.		
21	Verify the TCP proxy log shows an HTTP DELETE request for granule G2.		
22	Verify the TCP proxy log shows granule G2's request had no body.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
140				2 collections				
140				4 granules				

EXPECTED RESULTS:

	V	140	1	Ensure that the BMGT log indicates that unexported events were found and have been enqueued.		
	V	140	2	Ensure that the item referenced by each of the inserted events is exported to ECHO (or an ECHO stand-in) with the appropriate action for its current state (i.e. DELETE for a deleted item, PUT for a non deleted item).		

1.38 Automatic Export Errors (ECS-ECSTC-37)

DESCRIPTION:

	S	145	1	**[Automatic Export Errors]** Modify the XML metadata for a granule and the ODL descriptor for a collection such that they are invalid and their export attempt will cause a ‘data related error’. For instance, remove the start or end of an XML or ODL group so that the file fails basic validation.		
	S	145	2	Modify the XML metadata for a granule and the ODL descriptor file for a collection such that the metadata will validate but will be invalid for ingest into ECHO. For instance, cause the granule or collection start date to be after the end date.		

				(Alternatively, if using an ECHO stand-in, ensure that it returns error responses on ingest of these items.)		
	S	145	3	Move the metadata file for a granule, such that it will not be found at the location indicated by its database record.		
	S	145	4	Move the descriptor file for a collection, such that it will not be found at the location indicated by its database record.		
	S	145	5	Update the granules and collections in S-1 through S-4 so that events are generated and picked up by the automatic export polling process.		
	S	145	6	Replace the files moved in S-3 and S-4.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a mock ECHO is capturing BMGT traffic.		
3	Setup		#comment
4	S-1 Modify the XML metadata for a granule and the ODL descriptor for a collection such that they are invalid and their export attempt will cause a ‘data related error’. For instance, remove the start or end of an XML or ODL group so that the file fails basic validation.		#comment
5	Save off the metadata files for Granule G1 and Collection C1		
6	In granule G1's small file archive XML file, remove the final closing tag. /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
7	In collection C1's ODL descriptor file, comment out the beginning tag for GROUP=EVENT (e.g., /* GROUP = EVENT */) /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
8	S-2 Modify the XML metadata for a granule and the ODL descriptor file for a collection such that the metadata will validate but will be invalid for ingest into ECHO. For instance, cause the granule or collection start date to be after the end date. (Alternatively, if using an ECHO stand-in,		#comment

#	Action	Expected Result	Notes
	ensure that it returns error responses on ingest of these items.)		
9	Save off the metadata files for Granule G2 and Collection C2		
10	In granule G2's small file archive XML file, change the value of RangeBeginningDate to have a month > 12 and day > 31. /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
11	In collection C2's ODL descriptor file, change the value of a RangeBeginningDate to be 1 year after its corresponding RangeEndingDate. /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
12	S-3 Move the metadata file for a granule, such that it will not be found at the location indicated by its database record.		#comment
13	Move granule G3's small file archive XML file: cd /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH> mkdir tmp mv <SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml tmp		
14	S-4 Move the descriptor file for a collection, such that it will not be found at the location indicated by its database record.		#comment
15	Move collection C3's small file archive descriptor file: cd /stornext/smallfiles/<MODE>/descriptor mkdir tmp mv *<SHORT_NAME>.<VERSION_ID>.desc tmp		
16	S-5 Update the granules and collections in S-1 through S-4 so that events are generated and picked up by the automatic export polling process.		#comment
17	Update the DayNightFlag values of granules G1, G2, G3: update amgranule set daynightflag = 'Night'		
18	Cause update events to be queued for colelctions C1, C2, and C3 update DsGeEsdtConfiguredType set esdtstate = 'updating' where (configuredName = <C1.ShortName> and versionId = <C1.VersionId>) or (configuredName = <C2.ShortName> and versionId = <C3.VersionId>) or (configuredName = <C3.ShortName> and versionId = <C3.VersionId>) update DsGeEsdtConfiguredType set esdtstate = 'installed' where (configuredName = <C1.ShortName> and versionId = <C1.VersionId>) or (configuredName = <C2.ShortName>		

#	Action	Expected Result	Notes
	and versionId = <C3.VersionId>) or (configuredName = <C3.ShortName> and versionId = <C3.VersionId>)		
19	S-6 Replace the files moved in S-3 and S-4.		#comment
20	Restore granule G3's small file archive XML file: cd /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH> mv tmp/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml . rmdir tmp		
21	Restore collection C3's small file archive XML file: cd /stornext/smallfiles/<MODE>/descriptor mv tmp/*<SHORT_NAME>.<VERSION_ID>.desc . rmdir tmp		
22	Verification		#comment
23	V-1 Verify that the exports in S-1 and S-2 encounter errors.		#comment
24	Verify the BMGT log indicates collections C1, C2 and granules G1, G2 encounter errors. Look in /usr/ecs/${MODE}/CUSTOM/logs/EcBmBMGTGenerator.log		
25	V-2 Verify that the export requests for the items in S-1 and S-2 are marked as ‘failed’ in the GUI and that the associated errors can be viewed for each item.		#comment
26	Verify the BMGT GUI shows export attempts for collections C1, C2 and granules G1, G2.		
27	Verify in the BMGT GUI exports are maked 'failed' for collections C1, C2 and granules G1, G2. Export Request Status: SKIPPED or BLOCKED Corresponding Activity status should be ERROR for collections and granules.		
28	Verify the BMGT GUI allows viewing of errors associated with exports for collections C1, C2 and granules G1, G2.		
29	V-3 Verify that an email is sent indicating that the items in S-1 failed due to a data related error and were skipped.		#comment
30	Verify an email is sent to the the address configured in the BMGT GUI.		
31	Verify the email says that collection C1 and granule G1 failed due to data related errors and were skipped.		
32	V-4 Verify that an email is sent indicating that the items in S-2 failed due to an ECHO ingest error. It is acceptable for this to be in the same email as the errors for S-1.		#comment
33	Verify the email says that collection C2 and granule G2 failed due to ECHO ingest errors.		

#	Action	Expected Result	Notes
34	V-5 Verify that the email messages above list the error messages and the number of items which encountered each error.		#comment
35	Verify the email lists the error messages from the BMGT GUI.		
36	Verify the email lists the number of items for each error.		
37	V-6 Verify that messages are printed to the log file for the errors in S-1 and S-2, including the item (collection or granule) ID, and the reason for the error.		#comment
38	Verify errors for collections C1, C2 and granules G2, G2 appear in the BMGT log file.		
39	Verify the logged errors include collection IDs for C1, C2.		
40	Verify the logged errors include granule IDs for G1, G2.		
41	Verify the logged errors include the reason for each error.		
42	V-7 Verify that in the GUI it is possible to list only those items which encountered data related errors.		#comment
43	Verify the BMGT GUI allows filtering on only data related errors. In Export Requests sort on Status and look for SKIPPED.		
44	V-8 Verify that in the GUI it is possible to list only those items which encountered ECHO ingest errors.		#comment
45	Verify the BMGT GUI allows filtering on only ECHO ingest errors. In Export Requests sort on Status and look for BLOCKED.		
46	V-9 Verify that no HTTP export request was made for the items in S-1, as their failure occurred prior to export and export was therefore ‘skipped’		#comment
47	Verify the mock ECHO records no requests for C1 or G1.		
48	V-10 Verify that the export of the granule and collection whose files were moved in S-3 and S-4 encounter an error, which is printed in the log.		#comment
49	Verify the BMGT log indicates G3 and C3 encountered an error.		
50	V-11 Verify that the export of the granule and collection in S-3 and S-4 is retried until the files are replaced, at which point the export succeeds.		#comment
51	Verify the BMGT log indicates G3 and C3 exports are retried until the metadata and descriptor files are restored.		
52	Verify the BMGT log indicates G3 and C3 exports succeed after the metadata and descriptor files are restored.		
53	V-12 Verify that no email is sent for the errors related to the granule and collection in S-3 and S-4.		#comment
54	Verify no email is sent to the configured email address for G3 or C3.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
145				3 collections (C1, C2, C3)				
145				3 granules (G1, G2, G3)				

EXPECTED RESULTS:

	V	145	1	Verify that the exports in S-1 and S-2 encounter errors.		
	V	145	2	Verify that the export requests for the items in S-1 and S-2 are marked as 'failed' in the GUI and that the associated errors can be viewed for each item.		
	V	145	3	Verify that an email is sent indicating that the items in S-1 failed due to a data related error and were skipped.		
	V	145	4	Verify that an email is sent indicating that the items in S-2 failed due to an ECHO ingest error. It is acceptable for this to be in the same email as the errors for S-1.		
	V	145	5	Verify that the email messages above list the error messages and the number of items which encountered each error.		
	V	145	6	Verify that messages are printed to the log file for the errors in S-1 and S-2, including the item (collection or granule) ID, and the reason for the error.		
	V	145	7	Verify that in the GUI it is possible to list only those items which encountered data related errors.		
	V	145	8	Verify that in the GUI it is possible to list only those items which encountered ECHO ingest errors.		
	V	145	9	Verify that no HTTP export request was made for the items in S-1, as their failure occurred prior to export and export was therefore 'skipped'		
	V	145	10	Verify that the export of the granule and collection whose files were moved in S-3 and S-4 encounter an error, which		

				is printed in the log.		
	V	145	11	Verify that the export of the granule and collection in S-3 and S-4 is retried until the files are replaced, at which point the export succeeds.		
	V	145	12	Verify that no email is sent for the errors related to the granule and collection in S-3 and S-4.		

1.39 Manual Export - Collection Inserts (ECS-ECSTC-38)

DESCRIPTION:

	S	150	1	**[Manual Export - Collection Inserts]** Request the manual export of a collection's metadata.		
	S	150	2	Repeat S-1 for a different collection, but specify that the operation should be an insert only export.		
	S	150	3	Repeat S-1 for a different collection, but specify that the operation should be a delete only export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	Ensure Collections C1, C2, C3 have been installed in the mode. (ESDT verification script)		
9	Verify Collections C1,C2,C3 are enabled for Collection and Granule Export.		

#	Action	Expected Result	Notes
10	S-1 Request the manual export of a collection's metadata.		#comment
11	EcBmBMGTManualStart <MODE> --metc --collections <C1_shortname>.<C1_versionid>		
12	V-1 Verify that the operation in S-1 results in the export of a single HTTP PUT containing the full collection metadata of the requested collections.		#comment
13	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request. (There may be more than one HTTP request, e.g., if there are network issues.)		
14	Verify that the TCP proxy shows that the request contains the full collection metadata for Collection C1.		
15	S-2 Repeat S-1 for a different collection, but specify that the operation should be an insert only export.		#comment
16	EcBmBMGTManualStart <MODE> --insertonly --metc --collections <C2_shortname>.<C2_versionid>		
17	V-2 Verify that the operation in S-2 results in the export of a single HTTP PUT containing the full metadata of the requested collection.		#comment
18	Verify that the TCP proxy log shows an HTTP PUT request. (There may be more than one HTTP request, e.g., if there are network issues.)		
19	Verify that the TCP proxy shows that the request contains the full collection metadata for Collection C2.		
20	S-3 Repeat S-1 for a different collection, but specify that the operation should be a delete only export.		#comment
21	EcBmBMGTManualStart <MODE> --deleteonly --metc --collections <C3_shortname>.<C3_versionid>		
22	V-3 Verify that the operation in S-3 results in no export (but a warning message is printed indicating that the export could not be completed as the collection was not deleted).		#comment
23	Verify that no export request was enqueued.		
24	Verify that the Manual driver log indicates that no request was generated as the collection was not deleted.		
25	Verify that the TCP proxy shows no export request sent		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		testing manual export	3 collections					

EXPECTED RESULTS:

	V	150	1	Verify that the operation in S-1 results in the export of a single HTTP PUT containing the full collection metadata of the requested collections.		
	V	150	2	Verify that the operation in S-2 results in the export of a single HTTP PUT containing the full metadata of the requested collection.		
	V	150	3	Verify that the operation in S-3 results in no export (but the associated export request is in a terminal state, indicating that it was picked up and processed, but caused no export).		

1.40 Manual Export - Collection Deletes (ECS-ECSTC-39)

DESCRIPTION:

	S	160	1	**[Manual Export - Collection Deletes]** Find: a. A collection which is enabled for collection metadata export but which is not in the 'installed' state in AIM. b. A collection which is in the 'installed' state in AIM, but which is not enabled for collection metadata export. c. A collection which does not exist in AIM and is not enabled for collection metadata export (i.e. a completely made up short name and version ID). d. A collection which is in the 'installed' state and is enabled for collection metadata export. Request the manual export of all 4 collections.		
	S	160	2	Repeat S-1, but specify that the operation should be an insert only export.		
	S	160	3	Repeat S-1, but specify that the operation should be a delete only export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
4	Ensure test data exists under /sotestdata/DROP_802/BE_82_01/Criteria/160.		
5	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
6	Pause the EVENT and NEW queues in the BMGT GUI to prevent any database updates from causing exports.		
7	Setup		#comment
8	S-1 Find: a) A collection which is enabled for collection metadata export which is present (in AmCollection) but is not in the ‘installed’ state in AIM (DsGeESDTConfiguredType).		#comment
9	Ensure collection C1 exists in AIM: select * from amcollection where shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>		
10	Ensure collection C1 is not "installed": update DsGeESDTConfiguredType set esdtstate = 'installing' where configuredname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>		
11	Verify collection C1 is enabled for collection export: update bg_collection_configuration set collectionexportflag = 'Y' where shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>		
12	b) A collection which is enabled for collection metadata export which is not present (in AmCollection) and is not in the ‘installed’ state in AIM (DsGeESDTConfiguredType).		#comment
13	Ensure collection C2 is not in AIM: select * from amcollection where shortname = '<C2_SHORTNAME>' and versionid = <C2_VERSIONID>	0 rows	
14	Ensure collection C2 is either not in DsGeESDTConfiguredType or has an esdtstate value other than "installed": select esdtstate from DsGeESDTConfiguredType where configuredname = '<C2_SHORTNAME>' and versionid =	esdtstate != 'installed'	

#	Action	Expected Result	Notes
	<C2_VERSIONID>		
15	Ensure that collection C2 is enabled for collection export: update bg_collection_configuration set collectionexportflag = 'Y' where shortname = '<C2_SHORTNAME>' and versionid = <C2_VERSIONID>		
16	c) A collection which is present (in AmCollection), and in the ‘installed’ state in AIM (DsGeESDTConfiguredType), but which is not enabled for collection metadata export.		#comment
17	Ensure collection C3 exists in AIM: select * from amcollection where shortname = '<C3_SHORTNAME>' and versionid = <C3_VERSIONID>		
18	Ensure collection C3 is "installed": select esdtstate from DsGeESDTConfiguredType where configuredname = '<C3_SHORTNAME>' and versionid = <C3_VERSIONID>	esdtstate == 'installed'	
19	Verify collection C3 is not enabled for collection export: update bg_collection_configuration set collectionexportflag = 'N' where shortname = '<C3_SHORTNAME>' and versionid = <C3_VERSIONID>		
20	d) A collection which does not exist in AIM (AmCollection and DsGeESDTConfiguredType) and is not enabled for collection metadata export (i.e. a completely made up short name and version ID).		#comment
21	Ensure collection C4 does not exist in AIM: select * from amcollection where shortname = '<C4_SHORTNAME>' and versionid = <C4_VERSIONID>	0 rows	
22	Ensure collection C4 does not exist inDsGeESDTConfiguredType: select esdtstate from DsGeESDTConfiguredType where configuredname = '<C4_SHORTNAME>' and versionid = <C4_VERSIONID>	0 rows	
23	Ensure collection C4 is not enabled for collection export: delete from bg_collection_configuration where shortname = '<C4_SHORTNAME>' and versionid = <C4_VERSIONID>		
24	e) A collection which is present (in AmCollection), in the ‘installed’ state (DsGeESDTConfiguredType), and is enabled for collection metadata export.		#comment
25	Ensure collection C5 is in AIM: select * from amcollection where shortname = '<C5_SHORTNAME>' and versionid = <C5_VERSIONID>		

#	Action	Expected Result	Notes
26	Ensure collection C5 is "installed": select esdtstate from DsGeESDTConfiguredType where configuredname = '<C5_SHORTNAME>' and versionid = <C5_VERSIONID>	esdtstate == 'installed'	
27	Ensure that collection C5 is enabled for collection export: update bg_collection_configuration set collectionexportflag = 'Y' where shortname = '<C5_SHORTNAME>' and versionid = <C5_VERSIONID>		
28	Request the manual export of all 5 collections.		#comment
29	Ensure all 5 collections are listed in a text file (e.g., collections.txt): <C1_SHORTNAME>.<C1_VERSIONID> <C2_SHORTNAME>.<C2_VERSIONID> <C3_SHORTNAME>.<C3_VERSIONID> <C4_SHORTNAME>.<C4_VERSIONID> <C5_SHORTNAME>.<C5_VERSIONID>		
30	Manually export all 5 collections: ./EcBmBMGTManualStart <MODE> --metc --collectionfile /path/to/collections.txt		
31	V-1 Verify that the manual export request in S-1 results in the export of an HTTP DELETE for collection b, an HTTP PUT containing metadata for collection e, and no export for collections a, c and d.		#comment
32	Verify the TCP proxy (or mock ECHO) log shows no export of collection C1.		
33	Verify the TCP proxy (or mock ECHO) log shows an HTTP DELETE for collection C2. (There may be more than one HTTP request, e.g., if there are network issues.)		
34	Verify the TCP proxy (or mock ECHO) log shows no export of collection C3.		
35	Verify the TCP proxy (or mock ECHO) log shows no export of collection C4.		
36	Verify the TCP proxy (or mock ECHO) log shows an HTTP PUT for collection C5. (There may be more than one HTTP request, e.g., if there are network issues.)		
37	Verify the TCP proxy (or mock ECHO) log shows the PUT for collection C5 includes full collection metadata.		
38	V-4 Verify that the operations in S-1 – S-3 result in a message printed to the log indicating that collections c and d cannot be exported as they are not enabled or are not installed.		#comment
39	Verify the BMGT Manual log includes a message indicating that collection C3 could not be not exported because it was not enabled or not installed.		
40	Verify the BMGT Manual log includes a message indicating that collection C4 could not be not exported because it was not enabled or not installed.		

#	Action	Expected Result	Notes
41	S-2 Repeat S-1, but specify that the operation should be an insert only export.		#comment
42	Manually export all 5 collections as insert only: ./EcBmBMGTManualStart <MODE> --metc --insertonly --collectionfile /path/to/collections.txt		
43	V-2 Verify that the operation in S-2 results in no export for collections a-d, and an HTTP PUT containing metadata for collection e.		#comment
44	Verify the TCP proxy (or mock ECHO) log shows no export of collection C1.		
45	Verify the TCP proxy (or mock ECHO) log shows no export of collection C2.		
46	Verify the TCP proxy (or mock ECHO) log shows no export of collection C3.		
47	Verify the TCP proxy (or mock ECHO) log shows no export of collection C4.		
48	Verify the TCP proxy (or mock ECHO) log shows an HTTP PUT for collection C5. (There may be more than one HTTP request, e.g., if there are network issues.)		
49	Verify the TCP proxy (or mock ECHO) log shows the PUT for collection C5 includes full collection metadata.		
50	V-4 Verify that the operations in S-1 – S-3 result in a message printed to the log indicating that collections a, c and d cannot be exported as they are either not enabled or not installed.		#comment
51	Verify the BMGT Manual log includes a message indicating that collection C3 could not be not exported because it was not installed.		
52	Verify the BMGT Manual log includes a message indicating that collection C4 could not be not exported because it was not installed.		
53	S-3 Repeat S-1, but specify that the operation should be a delete only export.		#comment
54	Manually export all 5 collections as delete only: ./EcBmBMGTManualStart <MODE> --metc --deleteonly --collectionfile /path/to/collections.txt		
55	V-3 Verify that the operation in S-3 results in the export of an HTTP DELETE for collection b, and no exports for collections a, and c-e.		#comment
56	Verify the TCP proxy (or mock ECHO) log shows no export of collection C1.		
57	Verify the TCP proxy (or mock ECHO) log shows an HTTP DELETE for collection C2. (There may be more than one HTTP request, e.g., if there are network issues.)		
58	Verify the TCP proxy (or mock ECHO) log shows no export of collection C3.		
59	Verify the TCP proxy (or mock ECHO) log shows no export of collection C4.		

#	Action	Expected Result	Notes
60	Verify the TCP proxy (or mock ECHO) log shows no export of collection C5.		
61	V-4 Verify that the operations in S-1 – S-3 result in a message printed to the log indicating that collections a, c and d cannot be exported as they are either not enabled or not installed.		#comment
62	Verify the BMGT Manual log includes a message indicating that collection C1 could not be not exported because it was either not enabled or not installed.	collectionid | subtype | granuleexportflag | collectionexportflag | DsGeESDTConfiguredType:esdtstate ----------+----------+----------+----------+----------+---------- 397623 | MCD43D01.006 | N | Y | updating The EcBmBMGTManualDriver.log reports 2014-09-10 12:02:04,180 INFO [THREAD Thread-0] BmgtManualExport - Collection MCD43D01.006 cannot be added as a Delete request since it exists in AmCollection.	
63	Verify the BMGT Manual log includes a message indicating that collection C3 could not be not exported because it was either not enabled or not installed.		
64	Verify the BMGT Manual log includes a message indicating that collection C4 could not be not exported because it was either not enabled or not installed.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			4 collections					

EXPECTED RESULTS:

	V	160	1	Verify that the manual export request in S-1 results in the export of HTTP DELETEs for collections a-c, and an HTTP PUT containing metadata for collection d.		
	V	160	2	Verify that the operation in S-2 results in no export for collections a-c, and an HTTP PUT containing metadata for collection d.		
	V	160	3	Verify that the operation in S-3 results in the export of HTTP DELETEs for collections a-c.		

1.41 Manual Export - "Fake" Collection Delete (ECS-ECSTC-40)

DESCRIPTION:

	S	165	1	**[Manual Export – 'Fake' Collection Delete]** Request the manual export of a collection which is in the 'installed' state and is enabled for collection metadata export, specifying delete-only export and that the delete should be forced regardless of current collection state.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	Ensure Collections C1 has been installed in the mode. (ESDT verification script)		

#	Action	Expected Result	Notes
9	Verify Collection C1 is enabled for Collection and Granule Export.		
10	S-1 Request the manual export of a collection which is in the 'installed' state and is enabled for collection metadata export, specifying delete-only export and that the delete should be forced regardless of current collection state.		#comment
11	EcBmBMGTManualStart <MODE> --forcedelete --collections <C1_shortname>.<C1_versionid>		
12	V-1 Verify that the manual export request in S-1 results in the export of an HTTP DELETE.		#comment
13	Verify that the TCP proxy shows a single HTTP DELETE request for collection C1		
14	V-2 Verify that the collection in S-1 is automatically disabled for both collection and granule metadata export.		#comment
15	Verify that the collection C1 is disabled for Collection and granule metadata export (in the GUI and database). select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = <C1_shortname> and versionid = <C1_versionid> Both should be 'N'.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	165	1	Verify that the manual export request in S-1 results in the export of an HTTP DELETE.		
	V	165	2	Verify that the collection in S-1 is automatically disabled for both collection and granule metadata export.		

1.42 Manual Export - Collection Full Update (ECS-ECSTC-41)

DESCRIPTION:

	S	170	1	**[Manual Export - Collection Full Update]** Request the manual export of a collection's metadata,specifing that the export shall be a 'full collection update'.		
	S	170	2	Allow the export request initiated in S-1 to reach a complete state.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure Collections C1 has been installed in the mode. (ESDT verification script)		
9	Verify Collection C1 is enabled for Collection and Granule Export.		
10	S-1 Request the manual export of a collection's metadata, specifing that the export shall be a 'full collection update'.		#comment
11	EcBmBMGTManualStart <MODE> --collupd --collections <C1_shortname>.<C1_versionid>		
12	S-2 Allow the export request initiated in S-1 to reach a complete state.		#comment
13	Verify that the GUI displays the request status queued and completed.		
14	V-1 Verify that the operation in S-1 resulted in the export of: a) An HTTP DELETE (to the URL representing the specified collection)		#comment

#	Action	Expected Result	Notes
15	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP DELETE request for collection C1. (There may be more than one HTTP request, e.g., if there are network issues.)		
16	b) An HTTP PUT containing the full metadata for the specified collection.		#comment
17	Verify that the TCP proxy log shows an HTTP PUT request for Collection C1. (There may be more than one HTTP request, e.g., if there are network issues.)		
18	Verify that the TCP proxy log shows that the request contains the full collection metadata for Collection C1.		
19	c) An HTTP PUT for each granule in the specified collection, containing the full granule metadata for that granule.		#comment
20	Verify that the TCP proxy shows an HTTP PUT request for each granule in Collection C1. (There may be more than one HTTP request, e.g., if there are network issues.)		
21	Verify that the TCP proxy log shows that each request has complete granule metadata for each granule in Collection C1.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 collection with granules					

EXPECTED RESULTS:

	V	170	1	Verify that the operation in S-1 resulted in the export of: a) An HTTP DELETE (to the URL representing the specified collection) b) An HTTP PUT containing the full metadata for the specified collection. c) An HTTP PUT for each granule in the specified		

				collection, containing the full granule metadata for that granule.		

1.43 Manual Export - Granule Insert (ECS-ECSTC-42)

DESCRIPTION:

	S	180	1	**[Manual Export - Granule Insert]** Request the manual export of metadata for a single granule, specifying the granule ID.		
	S	180	2	Request the manual export of metadata fro granules in a specified collection. Ensure that the collection contains at least one logically deleted granule.		
	S	180	3	Request, in a single manual operation, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId – NOTE: it is easiest to use a valid ShortName and VersionID but a made up granuleID)		
	S	180	4	Request, in a single manual operation, specifying the export of inserts-only, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId– NOTE: it is easiest to use a valid ShortName and VersionID but a made up granuleID)		
	S	180	5	Request, in a single manual operation, specifying the export of deletes-only, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId– NOTE: it is easiest to use a valid ShortName and VersionID but a made up granuleID)		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment

#	Action	Expected Result	Notes
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	Ensure Collections C1, C2, C3 has been installed in the mode.		
9	Ensure Collections C1, C2, C3 are enabled for Collection and Granule Export.		
10	Ensure the Collections C1, C2, C3 have a few ingested granules.		
11	Ensure collections C1, C2, C3 have been exported to ECHO.		
12	Ensure granules in collections C1, C2, C3 have been exported to ECHO.		
13	S-1 Request the manual export of metadata for a single granule, specifying the granule ID.		#comment
14	Identify a granule g1 in Collection C1 that can be exported.		
15	EcBmBMGTManualStart <MODE> --metg -g <g1_GRANULEID>		
16	V-1 Verify that the operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata of the requested granule.		#comment
17	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		
18	Verify that the TCP proxy log shows that the request contains the full granule metadata for granule g1.		
19	S-2 Request the manual export of metadata for granules in a specified collection. Ensure that the collection contains at least one logically deleted granule.		#comment
20	Identify granules g2, g3 in Collection C2 that can be logically deleted.		
21	Logically delete a granule g2, g3: ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g2_g3.txt		

#	Action	Expected Result	Notes
22	Manually Export granules in Collection C2 ./EcBmBMGTManualStart <MODE> --metg -c <C2_SHORTNAME>.<C2_VERSIONID>		
23	V-2 Verify that the operation in S-2 results in the export of a single HTTP PUT containing the full granule metadata of the requested granule for each granule in the specified collection. Verify that an HTTP DELETE is also exported for any logically deleted granules in the collection.		#comment
24	Verify that the TCP proxy log shows an HTTP PUT request for each granule in Collection C2 except for g2 and g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
25	Verify that the TCP proxy shows that the request contains the full granule metadata for each granule in Collection C2 except for g2 and g3.		
26	Verify that an HTTP DELETE is exported for each of granules g2 and g3 (There may be more than one HTTP request, e.g., if there are network issues.)		
27	S-3 Request, in a single manual operation, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId – NOTE: it is easiest to use a valid ShortName and VersionID but a made up granuleID)		#comment
28	Identify granule g4 in Collection C3 as a normal science granule that can be exported.		
29	Identify granule g5 that is logically deleted in Collection C3 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g5.txt		
30	Identify granule g6 which is physically deleted in Collection C4 or make up a nonexistent granuleid in Collection C3 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g6.txt ./EcDlUnpublishStart.pl -mode <MODE> -granules <g6> ./EcDeletionCleanup.pl -user <db_user> ./EcBmBMGTManualStart --mode <MODE> --metg -g <g4_granuleid>,<g5_granuleid>,<g6_granuleid>		
31	V-3 Verify that the operation in S-3 is in a complete state and the following are exported for each respective operation: a) HTTP PUT for containing the full granule metadata. b) HTTP DELETE c) HTTP DELETE		#comment
32	Verify that the TCP proxy log shows an HTTP PUT request for granule g4. (There may be more than one HTTP request, e.g., if there are		

#	Action	Expected Result	Notes
	network issues.)		
33	Verify that the TCP proxy log shows that the request contains the full granule metadata for granule g4.		
34	Verify that the TCP proxy log shows an HTTP DELETE exported for granule g5. (There may be more than one HTTP request, e.g., if there are network issues.)		
35	Verify that the TCP proxy log shows an HTTP DELETE exported for granule g6. (There may be more than one HTTP request, e.g., if there are network issues.)		
36	S-4 Request, in a single manual operation, specifying the export of inserts-only, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId– NOTE: it is easiest to use a valid ShortName and VersionID but a made up granuleID)		#comment
37	./EcBmBMGTManualStart --mode <MODE> --insertonly --metg -g <g4_granuleid>,<g5_granuleid>,<g6_granuleid>		
38	V-4 Verify that the operation in S-4 is in a complete state and the following are exported for each respective operation: a) HTTP PUT for containing the full granule metadata. b) No export c) No export		#comment
39	Verify that the TCP proxy log shows an HTTP PUT request for granule g4. (There may be more than one HTTP request, e.g., if there are network issues.)		
40	Verify that the TCP proxylog shows that the request contains the full granule metadata for granule g4.		
41	Verify that the TCP proxy log does not show any export request for granule g5.		
42	Verify that the TCP proxy log does not show any export request for granule g6.		
43	S-5 Request, in a single manual operation, specifying the export of deletes-only, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId– NOTE: it is easiest to use a valid ShortName and VersionID but a made up granuleID)		#comment
44	./EcBmBMGTManualStart <MODE> --deleteonly --metg -g <g4_granuleid>,<g5_granuleid>,<g6_granuleid>		
45	V-5 Verify that the operation in S-5 is in a complete state and the		#comment

#	Action	Expected Result	Notes
	following are exported for each respective operation: a) No export b) HTTP DELETE c) HTTP DELETE</i>		
46	Verify that the TCP proxy log shows no HTTP export request for granule g4.		
47	Verify that the TCP proxy log shows an HTTP DELETE exported for granule g5. (There may be more than one HTTP request, e.g., if there are network issues.)		
48	Verify that the TCP proxy log shows an HTTP DELETE exported for granule g6. (There may be more than one HTTP request, e.g., if there are network issues.)		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			3 collections with at least 3 granules in each					

EXPECTED RESULTS:

	V	180	1	Verify that the operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata of the requested granule.		
	V	180	2	Verify that the operation in S-2 results in the export of a single HTTP PUT containing the full granule metadata of the requested granule for each granule in the specified collection. Verify that an HTTP DELETE is also exported for any logically deleted granules in the collection.		
	V	180	3	Verify that the operation in S-3 is in a complete state and the following are exported for each respective operation: a) HTTP PUT for containing the full granule metadata. b) HTTP DELETE c) HTTP DELETE		
	V	180	4	Verify that the operation in S-4 is in a complete state and the following are exported for each respective operation: a) HTTP PUT for containing the full granule metadata. b) No export c) No export		

	V	180	5	Verify that the operation in S-5 is in a complete state and the following are exported for each respective operation: a) No export b) HTTP DELETE c) HTTP DELETE		

1.44 Manual Export - Granule Delete (ECS-ECSTC-43)

DESCRIPTION:

	S	190	1	**[Manual Export - Granule Delete]** Request the manual export of metadata for a single granule which is logically or physically deleted.		
	S	190	2	Request the manual export of metadata fro granules in a specified collection. Ensure that the collection contains at least one logically deleted granule.		
	S	190	3	Request, in a single manual operation, specifying the export of deletes-only, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId)		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment

#	Action	Expected Result	Notes
8	Ensure Collections C1, C2, C3 has been installed in the mode.		
9	Ensure Collections C1, C2, C3 are enabled for Collection and Granule Export.		
10	Ensure the Collections C1, C2, C3 have a few ingested granules.		
11	S-1 Request the manual export of metadata for a single granule which is logically or physically deleted.		#comment
12	Identify a granule g1 that can be logically deleted in Collection C1 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g1.txt EcBmBMGTManualStart <MODE> --metg -g <g1_granuleid>		
13	V-1 Verify that the export operation in S-1 results in the export of a single HTTP DELETE.		#comment
14	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows that an HTTP DELETE is exported for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		
15	S-2 Request the manual export of metadata for granules in a specified collection. Ensure that the collection contains at least one logically deleted granule.		#comment
16	Identify granules g2, g3 in Collection C2 that can be logically deleted.		
17	Logically delete a granule g2,g3 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g2_g3.txt		
18	Manually Export granules in Collection C2 ./EcBmBMGTManualStart <MODE> --metg -c <C2_SHORTNAME>.<C2_VERSIONID>		
19	V-2 Verify that the export operation in S-2 results in the export of a single HTTP DELETE for each logically deleted granule in the specified collection, as well as HTTP PUTs for each non deleted granule.		#comment
20	Verify that the TCP proxy log shows an HTTP PUT request for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		
21	Verify that the TCP proxy log shows that the request contains the full granule metadata for granule g1.		
22	Verify that the TCP proxy log shows an HTTP DELETE is exported for each of granules g2 and g3. (There may be more than one HTTP request, e.g., if there are network issues.)		

#	Action	Expected Result	Notes
23	S-3 Request, in a single manual operation, specifying the export of deletes-only, the export of metadata for the following: a) Normal granule b) Logically deleted granule. c) Physically deleted granule (specify Short Name, VersionId and GranuleId)		#comment
24	Identify granule g4 in Collection C3 as a normal science granule that can be exported.		
25	Identify granule g5 that is logically deleted in Collection C3 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g5.txt		
26	Identify granule g6 which is physically deleted in Collection C4 or make up a nonexistent granuleid in Collection C3 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g6.txt ./EcDlUnpublishStart.pl -mode <MODE> -granules <g6> ./EcDeletionCleanup.pl -user <db_user> ./EcBmBMGTManualStart <MODE> --deleteonly --metg -g <g4_granuleid>,<g5_granuleid>,<g6_granuleid>		
27	V-3 Verify that the operation in S-3 is in a complete state and the following are exported for each respective operation: a) No export b) HTTP DELETE c) HTTP DELETE		#comment
28	Verify that the TCP proxy log shows no HTTP PUT request for granule g4.		
29	Verify that the TCP proxy log shows an HTTP DELETE exported for granule g5. (There may be more than one HTTP request, e.g., if there are network issues.)		
30	Verify that the TCP proxy log shows an HTTP DELETE exported for granule g6. (There may be more than one HTTP request, e.g., if there are network issues.)		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		3 collections with at least 3 granules						

EXPECTED RESULTS:

	V	190	1	Verify that the export operation in S-1 results in the export of a single HTTP DELETE.		
	V	190	2	Verify that the export operation in S-2 results in the export of a single HTTP DELETE for each logically deleted granule in the specified collection, as well as HTTP PUTs for each non deleted granule.		
	V	190	3	Verify that the operation in S-3 is in a complete state and the following are exported for each respective operation: a) No export b) HTTP DELETE c) HTTP DELETE		

1.45 Manual Export - Insert Time (ECS-ECSTC-44)

DESCRIPTION:

	S	200	1	**[Manual Export - Insert Time]** Identify two granules within a collection and their insert times. Ensure that there are other granules inserted in between those times within the same collection. Request the export of granule metadata for the collection, specifying the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		
	S	200	2	Identify two collections and their insert times. Ensure that there are other collections inserted between those times. Request the export of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS		

#	Action	Expected Result	Notes
	metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure Collections C1, C2, C3, C4 has been installed in the mode.		
9	Ensure Collections C1, C2, C3, C4 are enabled for Collection and Granule Export.		
10	Ensure the Collections C1, C2, C3, C4 have a few ingested granules.		
11	S-1 Identify two granules within a collection and their insert times. Ensure that there are other granules inserted in between those times within the same collection. Request the export of granule metadata for the collection, specifying the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		#comment
12	select granuleid, archivetime from AmGranule where shortname = <C1_shortname> and versionid = <C1_versionid> order by archivetime limit 4; Assuming there are 4 granules returned g1,g2,g3,g4.		
13	Let g1 be the first granule in the list. Let gLast be the last granule in the list		
14	g1_insert_time = archivetime of g1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. g4_insert_time = archivetime of g4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
15	EcBmBMGTManualStart <MODE> --starttime <g1_insert_time> --endtime <g4_insert_time> --metg --collections <C1_shortname>.<C1_versionid>		
16	V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata - or HTTP DELETE, depending on the granule's current state - for each granule in the specified collection which was inserted during the specified time range (with the exception noted in V-2).		#comment
17	V-2 Verify that the granule which was inserted at the start time of the specified range is exported, but that the granule inserted at the end of the range is not.		#comment
18	Verify that the TCP proxy shows an HTTP PUT request each for granules g1, g2, g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
19	Verify that the TCP proxy shows that the export request contains the full		

#	Action	Expected Result	Notes
	granule metadata for each of the granules g1, g2, g3.		
20	S-2 Identify two collections and their insert times. Ensure that there are other collections inserted between those times. Request the export of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		#comment
21	select collectionid, shortname, versionid, inserttime from AmCollection order by inserttime limit 4;		
22	Let C1 be the first Collection in the list. Let C4 be the last granule in the list		
23	C1_insert_time = inserttime of C1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. C4_insert_time = inserttime of C4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
24	EcBmBMGTManualStart <MODE> --starttime <C1_insert_time> --endtime <C4_insert_time> --metc		
25	V-3 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata - for each collection which was inserted during the specified time range (with the exception noted in V-4).		#comment
26	V-4 Verify that the collection which was inserted at the start time of the specified range is exported, but that the collection inserted at the end of the range is not.		#comment
27	Verify that the TCP proxy shows an HTTP PUT request each for Collection C1, C2, C3. (There may be more than one HTTP request, e.g., if there are network issues.)		
28	Verify that the TCP proxy shows that the export request contains the full collection metadata for each of the Collections C1, C2, C3.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			4 collections with granules					

EXPECTED RESULTS:

	V	200	1	Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata - or HTTP DELETE, depending on the granule's current state - for each granule in the specified collection which was inserted during the specified time range (with the exception noted in V-2).		
	V	200	2	Verify that the granule which was inserted at the start time of the specified range is exported, but that the granule inserted at the end of the range is not.		
	V	200	3	Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata - for each collection which was inserted during the specified time range (with the exception noted in V-4).		
	V	200	4	Verify that the collection which was inserted at the start time of the specified range is exported, but that the collection inserted at the end of the range is not.		

1.46 Manual Export - Last Update Time (ECS-ECSTC-45)

DESCRIPTION:

	S	210	1	**[Manual Export - Last Update Time]** Identify two granules within a collection and their last update times. Ensure that there are other granules updated in between those times within the same collection. Request the export of granule metadata for the collection, specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.		
	S	210	2	Identify two collections and their update times. Ensure that there are other collections updated between those times. Request the export of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		

#	Action	Expected Result	Notes
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure Collections C1, C2, C3, C4 has been installed in the mode.		
9	Ensure Collections C1, C2, C3, C4 are enabled for Collection and Granule Export.		
10	Ensure the Collections C1, C2, C3, C4 have a few ingested granules.		
11	S-1 Identify two granules within a collection and their last update times. Ensure that there are other granules updated in between those times within the same collection. Request the export of granule metadata for the collection, specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.		#comment
12	select granuleid, lastupdate from AmGranule where shortname = <C1_shortname> and versionid = <C1_versionid> order by lastupdate limit 4; Assuming there are 4 granules returned g1,g2,g3,g4.		
13	Let g1 be the first granule in the list. Let gLast be the last granule in the list		
14	g1_lastupdate_time = lastupdate of g1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. g4_lastupdate_time = lastupdate of g4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
15	EcBmBMGTManualStart <MODE> --starttime <g1_lastupdate_time> --endtime <g4_lastupdate_time> --lastupdate --metg --collections <C1_shortname>.<C1_versionid>		
16	V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata - or HTTP DELETE, depending on the granule's current state - for each granule in the specified collection which was updated during the specified time range (with the exception noted in V-2).		#comment
17	V-2 Verify that the granule which was updated at the start time of the specified range is exported, but that the granule updated at the end of the		#comment

#	Action	Expected Result	Notes
	range is not.		
18	Verify that the TCP proxy shows an HTTP PUT request each for granules g1, g2, g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
19	Verify that the TCP proxy does not show any request for granule g4.		
20	Verify that the TCP proxy shows that the export request contains the full granule metadata for each of the granules g1, g2, g3.		
21	S-2 Identify two collections and their update times. Ensure that there are other collections updated between those times. Request the export of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.		#comment
22	select collectionid, shortname, versionid, lastupdate from AmCollection order by lastupdate limit 4;		
23	Let C1 be the first collection in the list. Let C4 be the last collection in the list.		
24	C1_lastupdate_time = lastupdate of C1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. C4_lastupdate_time = lastupdate of C4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
25	EcBmBMGTManualStart <MODE> --starttime <C1_lastupdate_time> --endtime <C4_lastupdate_time> --metc --lastupdate		
26	V-3 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was updated during the specified time range (with the exception noted in V-4).		#comment
27	V-4 Verify that the collection which was updated at the start time of the specified range is exported, but that the collection updated at the end of the range is not.		#comment
28	Verify that the TCP proxy shows an HTTP PUT request each for Collection C1, C2, C3. (There may be more than one HTTP request, e.g., if there are network issues.)		
29	Verify that the TCP proxy does not show any request for collection C4.		
30	Verify that the TCP proxy shows that the export request contains the full collection metadata for each of the Collections C1, C2, C3.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			4 collections with granules					

EXPECTED RESULTS:

#
Pre-Conditions
#
Setup
#S-1 Identify two granules within a collection and their last update times.
Ensure that there are other granules updated in between those times within the same collection.
Request the export of granule metadata for the collection, specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.
#S-2 Identify two collections and their update times.
Ensure that there are other collections updated between those times.
Request the export of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.
#
Verification
#V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata - or HTTP DELETE, depending on the granule's current state - for each granule in the specified collection which was updated during the specified time range (with the exception noted in V-2).
#V-2 Verify that the granule which was updated at the start time of the specified range is exported, but that the granule updated at the end of the range is not.
#V-3 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was updated during the specified time range (with the exception noted in V-4).
#V-4 Verify that the collection which was updated at the start time of the specified range is exported, but that the collection updated at the end of the range is not.#
Pre-Conditions
#
Setup
#S-1 Identify two granules within a collection and their last update times.
Ensure that there are other granules updated in between those times within the same collection.

Request the export of granule metadata for the collection, specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.
#S-2 Identify two collections and their update times.
Ensure that there are other collections updated between those times.
Request the export of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified update times and specifying that the range shall apply to update times.
#
Verification
#V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata - or HTTP DELETE, depending on the granule's current state - for each granule in the specified collection which was updated during the specified time range (with the exception noted in V-2).
#V-2 Verify that the granule which was updated at the start time of the specified range is exported, but that the granule updated at the end of the range is not.
#V-3 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was updated during the specified time range (with the exception noted in V-4).
#V-4 Verify that the collection which was updated at the start time of the specified range is exported, but that the collection updated at the end of the range is not.

	V	210	1	Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata - or HTTP DELETE, depending on the granule's current state - for each granule in the specified collection which was updated during the specified time range (with the exception noted in V-2).		
	V	210	2	Verify that the granule which was updated at the start time of the specified range is exported, but that the granule updated at the end of the range is not.		
	V	210	3	Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was updated during the specified time range (with the exception noted in V-4).		
	V	210	4	Verify that the collection which was updated at the start time of the specified range is exported, but that the collection updated at the end of the range is not.		

1.47 Manual Export - Manual Granule Updates (ECS-ECSTC-46)

DESCRIPTION:

	S	220	1	**[Manual Export - Manual Granule Updates]** Record the start time of the criteria. Identify the following public collections for use in this test: a. AMSR collection configured for QA and PH.		

				b. Collection configured for HDF MAP. c. Collection with browse. d. MISR Level 1 collection. e. MISBR		
	S	220	2	Publish a hidden QA granule linked to a science granule in one of the selected collections.		
	S	220	3	Publish a hidden PH granule linked to a science granule in one of the selected collections.		
	S	220	4	Regenerate the HDF_MAP for a science granule in one of the selected collections.		
	S	220	5	Modify the browse link for a science granule in one of the selected collections.		
	S	220	6	Publish a MISBR granule which is associated with a MISR Level 1 granule, as well as a MISBR granule (possibly the same granule) which is associated with a MISR Level 2 granule.		
	S	220	7	Using the command line utility or GUI, request the export of granule metadata for the chosen collections, specifying a start time equal to the beginning time of the test (leave end time blank), and specifying selection by update time.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
5	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
6	Verify test data is avilable under /sotestdata/DROP_802/BE_82_01/Criteria/220.		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Setup		#comment
9	Ensure Collections C1, C2, C3, C4, C5 has been installed in the mode.		

#	Action	Expected Result	Notes
10	Ensure Collections C1, C2, C3, C4, C5 are enabled for Collection and Granule Export.		
11	S-1 Record the start time of the criteria. Identify the following public collections for use in this test: a) AMSR collection configured for QA and PH. b) Collection configured for HDF MAP. c) Collection with browse. d) MISR Level 1 collection. e) MISBR		#comment
12	Ensure the Collection C1 is an AMSR collection configured to have QA and PH ancillary granules.		
13	Ensure that Collection C2 is a collection that can be configured for HDF Map.		
14	Ensure that Collection C3 is a Collection with browse granules.		
15	Ensure that Collection C4 is a MISR Level 1 Collection		
16	Ensure that Collection C5 has MISBR granules.		
17	Ensure that all science granules used in the test have been ingested into the public data pool.		
18	Ensure that all science granules used in this test have been exported to ECHO.		
19	Ensure the QA, PH, and MISBR granules are in the hidden data pool.		
20	Record start time as t_start_time		
21	S-2 Publish a hidden QA granule linked to a science granule in one of the selected collections.		#comment
22	Ingest granules g1 and g2.		
23	Ingest QA granule qa_1 associated to public Science granule g1 with default publishing turned off into Collection C1		
24	Publish the hidden QA granule qa_1 EcDlPublishStart <MODE> -ecs -g <granuleid_qa_1>		
25	S-3 Publish a hidden PH granule linked to a science granule in one of the selected collections.		#comment
26	Ingest PH granule ph_1 linked to Science granule g2 with default publishing turned off into Collection C1		
27	Publish the hidden PH granule ph_1 EcDlPublishStart <MODE> -ecs -g <granuleid_ph_1>		
28	S-4 Regenerate the HDF_MAP for a science granule in one of the selected collections.		#comment
29	Ingest Science granule g3 that can be enabled for map generation into		

#	Action	Expected Result	Notes
	Collection C2		
30	Generate HDF Map granule hdf_1 manually and publish the granule EcAmInsertMapGenerationRequest.pl -mode <MODE> -g <granule_A> EcDlPublishStart <MODE> -ecs -g <granule_hdf_1>		
31	S-5 Modify the browse link for a science granule in one of the selected collections.		#comment
32	Ingest a browse granule br_1 linked to science granule g4 into Collection C3		
33	Ingest the browse granule br_1 with a new linkage file into Collection C3		
34	S-6 Publish a MISBR granule which is associated with a MISR Level 1 granule, as well as a MISBR granule (possibly the same granule) which is associated with a MISR Level 2 granule.		#comment
35	Ensure that in EcDlInsertUtility.properties file, MISR_SPECIAL_PROCESSING=Y		
36	Ingest MISR level 1 or 2 granule gr_misr_1 to Collection C4. dsmdmisrprocessingcriteria table should have entries for the science and the misrbrowse		
37	Ingest and publish a MISBR granule gr_misrbr_1 linked to gr_misr_1 to Collection C4		
38	Ingest a MISBR granule gr_misbr_2 to Collection C5		
39	Ingest and publish a MISR granule gr_misr_2 linked to gr_misbr_2 to Collection C5		
40	S-7 Using the command line utility or GUI, request the export of granule metadata for the chosen collections, specifying a start time equal to the beginning time of the test (leave end time blank), and specifying selection by update time.		#comment
41	EcBmBMGTManualStart <MODE> --starttime <t_start_time> --lastupdate --metg --collections <C1_shortname>.<C1_versionid>,<C2_shortname>.<C2_versionid>,<C3_shortname>.<C3_versionid>,<C4_shortname>.<C4_versionid>,<C5_shortname>.<C5_versionid>		
42	Verification		#comment
43	V-1 Verify that the granules used for S-1 – S-4 are exported by BMGT.		#comment
44	V-2 Verify that the exports of the granules in S-1 – S-4 contain the full granule metadata, including, as applicable, the QA, PH, and HDF map URLS (reflecting the correct URLs after the update).		#comment

#	Action	Expected Result	Notes
45	Verify that the TCP proxy shows an HTTP PUT request for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		
46	Verify that the TCP proxy shows that the HTTP PUT request contains the full granule metadata for g1.		
47	Verify that the TCP proxy shows that the metadata exported for granule g1 contains the URL for QA granule qa_1. xpath /Granule/OnlineResources g1.xml should contain an OnlineResource element of type "Quality Assurance" and the URL of qa_1.		
48	Verify that the TCP proxy shows an HTTP PUT request for granule g2. (There may be more than one HTTP request, e.g., if there are network issues.)		
49	Verify that the TCP proxy shows that the HTTP PUT request contains the full granule metadata for g2.		
50	Verify that the TCP proxy shows that the metadata exported for granule g2 contains the URL for PH granule ph_1: xpath /Granule/OnlineResources g2.xml should contain an OnlineResource element of type "Production History" and the URL of ph1		
51	Verify that the TCP proxy shows an HTTP PUT request for granule g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
52	Verify that the TCP proxy shows that the HTTP PUT request contains the full granule metadata for g3		
53	Verify that the TCP proxy shows that the metadata exported for granule g3 contains the URL for HDF Map granule hdf_1: xpath /Granule/OnlineResources g3.xml should contain an OnlineResource element of type "HDF Map" and the URL of hdf_1		
54	Verify that the TCP proxy shows an HTTP PUT request for granule g4. (There may be more than one HTTP request, e.g., if there are network issues.)		
55	Verify that the TCP proxy shows that the HTTP PUT request contains the full granule metadata for g4.		
56	Verify that the TCP proxy shows that the metadata exported for granule g4 contains the URL for browse granule br_1: xpath /Granule/OnlineResources g4.xml should contain an OnlineResource element of type "BROWSE" and the URL of br_1.		
57	Verify that the TCP proxy shows an HTTP PUT request for granule		

#	Action	Expected Result	Notes
	gr_misr_1. (There may be more than one HTTP request, e.g., if there are network issues.)		
58	Verify that the TCP proxy shows that the HTTP PUT request contains the full granule metadata for gr_misr_1.		
59	Verify that the TCP proxy shows that the metadata exported for granule gr_misr_1 contains the URL for browse granule gr_misbr_1: xpath /Granule/OnlineResources gr_misr_1.xml should contain an OnlineResource element of type "BROWSE" and the URL of gr_misbr_1		
60	Verify that the TCP proxy shows an HTTP PUT request for granule gr_misr_2. (There may be more than one HTTP request, e.g., if there are network issues.)		
61	Verify that the TCP proxy shows that the HTTP PUT request contains the full granule metadata for gr_misr_2.		
62	Verify that the TCP proxy shows that the metadata exported for granule gr_misr_2 contains the URL for browse granule gr_misrbr_2: xpath /Granule/OnlineResources gr_misr_2.xml should contain an OnlineResource element of type "BROWSE" and the URL of gr_misbr_2		
63	V-3 Verify that the exports of the granules in S-5 – S-6 contain the full granule metadata, including the correct browse linkage URLs, in addition to SCIENCE, METADATA, etc. URLs.		#comment
64	Verify that the TCP proxy shows in the HTTP PUT request for each granule: xpath /Granule/OnlineAccessURLs/OnlineAccessURL with the correct url to the granule data file. xpath /Granule/OnlineResources/OnlineResource with Type = METADATA and the correct URL to the granule metadata file.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
220		AMSR Collection C1	1 PH granule, 1 QA granule				/sotestdata/DROP_802/BE_82_01/Criteria/220	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
220		Collection C2	HDF Map granule generation capable				/sotestdata/DROP_802/BE_82_01/Criteria/220	
		Collection C3	with browse granules				/sotestdata/DROP_802/BE_82_01/Criteria/220	
		Collection C4 MISR -1	MISR-1 and MISRBR-1 granules				/sotestdata/DROP_802/BE_82_01/Criteria/220	
		Collection C5	MISR-2 and MISRBR-2 granules				/sotestdata/DROP_802/BE_82_01/Criteria/220	

EXPECTED RESULTS:

	V	220	1	Verify that the granules used for S-1 – S-4 are exported by BMGT.		
	V	220	2	Verify that the exports of the granules in S-1 – S-4 contain the full granule metadata, including, as applicable, the QA, PH, and HDF map URLS (reflecting the correct URLs after the update).		
	V	220	3	Verify that the exports of the granules in S-5 – S-6 contain the full granule metadata, including the correct browse linkage URLs, in addition to SCIENCE, METADATA, etc. URLs.		

1.48 Manual Export - Multiple Options/Select by Group (ECS-ECSTC-47)

DESCRIPTION:

	S	230	1	**[Manual Export - Multiple Options/Select by Group]** Request the manual export of both granule and collection metadata for all collections in a particular Datapool Group. Specify a time range and indicate that the range shall apply to update time. Also specify the export of inserts only.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Preconditions		#comment
2	Ensure BMGT configuration is current, complete, and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
4	Ensure the test collections are installed. E.g., the DPL Ingest GUI shows the collections as configured data types.		
5	Ensure the test collections are all in the same data pool group, either in the Data Pool Maintenance GUI or by a query: select shortname, versionid, groupid from amcollection where shortname in ('MCD15A2', 'MCD15A3', 'MCD43A1', 'MCD43A2') and versionid = 5;	All collections should have the same value for groupid.	
6	Ensure the tests collections are enabled for collection and granule export. For each test collection, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each test collection, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Setup		#comment
9	S-1 Request the manual export of both granule and collection metadata for all collections in a particular Datapool Group. Specify a time range and indicate that the range shall apply to update time. Also specify the		#comment

#	Action	Expected Result	Notes
	export of inserts only.		
10	Ingest granules g1, g2 into Collection C1		
11	Record the update start time t_start_time		
12	Ingest granules g3, g4 into Collection C2		
13	Ingest granules g5, g6 into Collection C3		
14	update C4: update amcollection set lastupdate = now() where Shortname = <C4.ShortName> and VersionId = <C4.VersionId>;		
15	Logically Delete granule g5: ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g5.txt		
16	Record the update end time t_end_time		
17	Ingest granules g7, g8 into Collection C4		
18	Wait for all granule and collection inserts and deletes to be exported.		
19	EcBmBMGTManualStart <MODE> --metc --metg --groups <G1> --starttime <t_start_time> --endtime <t_end_time> --lastupdate --insertonly		
20	V-1 Verify that the operation in S-1 results in the export of an HTTP PUT containing the metadata for each collection, and each granule in each collection, which belongs to the specified datapool collection group, was updated within the specified range, and is not logically deleted. Verify that no HTTP DELETEs are exported.		#comment
21	Verify that the TCP proxy log shows an HTTP PUT request for Collection C4. (There may be more than one HTTP request, e.g., if there are network issues.)		
22	Verify that the TCP proxy log shows an HTTP PUT request for granules g3, g4 and g6. (There may be more than one HTTP request, e.g., if there are network issues.)		
23	Verify that the TCP proxy log shows that each request has complete granule metadata for granules g3, g4, g6.		
24	Verify the TCP proxy log shows no PUTs for granules g1, g2, g7, g8 ingested outside the specified range.		
25	Verify that the TCP proxy log shows no HTTP DELETE requests after t_end_time.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
230		4 collections in the same data pool group with 2 science granules each	MCD15A2.005 MCD15A3.005 MCD43A1.005 MCD43A2.005				/sotestdata/DROP_802/BE_82_01/Criteria/230	

EXPECTED RESULTS:

	V	230	1	Verify that the operation in S-1 results in the export of an HTTP PUT containing the metadata for each collection, and each granule in each collection, which belongs to the specified datapool collection group, was updated within the specified range, and is not logically deleted. Verify that no HTTP DELETEs are exported.		

1.49 Manual Export - Nominal (ECS-ECSTC-48)

DESCRIPTION:

	S	240	1	**[Manual Export - Nominal]** Via the BMGT GUI, suspend the processing of manual exports.		
	S	240	2	Use the command line or GUI interface to request a manual export of granule and/or collection metadata. Ensure that the ECS mode is specified either on the command line or in the GUI URL.		
	S	240	3	Via the BMGT GUI resume the processing of manual exports.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings		

#	Action	Expected Result	Notes
	validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/240 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure a Collections C1 has been installed in the mode.		
9	Verify Collection C1 enabled for Collection and Granule Export.		
10	S-1 Via the BMGT GUI, suspend the processing of manual exports.		#comment
11	Suspend the Manual Export (MAN) queue on BMGT GUI.		
12	S-2 Use the command line or GUI interface to request a manual export of granule and/or collection metadata. Ensure that the ECS mode is specified either on the command line or in the GUI URL.		#comment
13	EcBmBMGTManualStart <MODE> --metc --collections <C1_shortname>.<C1_versionid>		
14	EcBmBMGTManualStart <MODE> --metg --collections <C1_shortname>.<C1_versionid>		
15	V-1 Verify that prior to resumption of processing in S-3, the requested operations are visible on the queue, but are not exported		#comment
16	Verify that the GUI displays the requests in the "PENDING" state.		
17	V-2 Verify that each of the requested items (collection or granule) appears in the BMGT export request queue (viewed through the GUI), and indicates that it is a manual export.		#comment
18	Verify that the requests shown on the GUI indicate they are manual export requests.		
19	S-3 Via the BMGT GUI resume the processing of manual exports.		#comment
20	Resume the manual (MAN) queue on BMGT GUI.		
21	V-3 Verify that each of the enqueued export requests results in exactly one export to ECHO (or an ECHO stand-in).		#comment
22	Verify that the TCP proxy shows an HTTP request for each request shown queued in the GUI. (There may be more than one HTTP request,		

#	Action	Expected Result	Notes
	e.g., if there are network issues.)		
23	V-4 Verify that when the manual process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which initiation of the manual export started and completed, as well as how many items were enqueued for export.		#comment
24	Verify that the EcBmBMGTManualDriver.log is under usr/ecs/<MODE>/CUSTOM/logs/<BMGT_LOG>		
25	Verify that the EcBmBMGTManualDriver.log contains a message that manual export process was started		
26	Verify that the EcBmBMGTManualDriver.log contains a message that all requests have been added to the queue		
27	Verify that the EcBmBMGTDispatcher.log contains the time of start of manual export		
28	Verify that the EcBmBMGTDispatcher.log contains the time of completion of the manual export		
29	Verify that the EcBmBMGTManualDriver.log contains a message with the number of items enqueued for export		
30	V-5 Verify that the options specified on the command line (or in the GUI) for the initiation of a manual export are printed to the log file.		#comment
31	Verify that the EcBmBMGTManualDriver.log contains a message showinfg the options used to start the manual export		
32	V-6 Verify that the database and the log files contain information on the progress of each request through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		#comment
33	Verify that bg_export_request contains requests for each of the granules and the collection C1		
34	Verify that the bg_export_activity table shows the requests for each of the granules and the collection C1		
35	Verify that the bg_export_error table shows any errors in the export requests		
36	Verify that the EcBmBMGTGenerator.log contains a message showing when the metadata was generated for each request		
37	Verify that the EcBmBMGTExporter.log contains a message showing when the export was sent to ECHO for each request		
38	Verify that the EcBmBMGTExporter.log contains a message showing when		

#	Action	Expected Result	Notes
	the response was received from ECHO for each request		
39	V-7 Verify that the BMGT GUI displays the completed export events, indicating that they were the result of a manual export. Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or collection ID.		#comment
40	Verify that the bmgt GUI indicates that the exports have completed.		
41	Verify that the bmgt GUI indicates that the export requests were a result of a manual export		
42	Verify that the bmgt GUI indicates that the requests were sucessful.		
43	Verify that the bmgt GUI lists the time of export for each request		
44	Verify that the bmgt lGUI lists the granule or collection ID for each request		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		a collection with granules to test manual export						

EXPECTED RESULTS:

	V	240	1	Verify that prior to resumption of processing in S-3, the requested operations are visible on the queue, but are not exported		
	V	240	2	Verify that each of the requested items (collection or granule) appears in the BMGT export request queue (viewed through the GUI), and indicates that it is a manual export.		
	V	240	3	Verify that each of the enqueued export requests results in exactly one export to ECHO (or an ECHO stand-in).		
	V	240	4	Verify that when the manual process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which initiation of the manual export started and completed, as well as how many items were enqueued for export.		
	V	240	5	Verify that the options specified on the command line (or in the GUI) for the initiation of a manual export are printed to the		

				log file.		
	V	240	6	Verify that the database and the log files contain information on the progress of each request through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc[TR1][TG2].		
	V	240	7	Verify that the BMGT GUI displays the completed export events, indicating that they were the result of a manual export. Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or collection ID.		

1.50 Manual Export - Invocation via Command Line[a]: granules w/ granule args (ECS-ECSTC-49)

DESCRIPTION:

	S	242	1	**[Manual Export – Invocation via Command Line]** From the command line invoke manual export of metadata for the following: a. Granule metadata for multiple granules, specified on the command line. b. Granule metadata for multiple granules, specified in an input file. c. Granule metadata for all granules in a collection, specified on the command line. d. Granule metadata for all granules in a collection, specified in an input file. e. Collection metadata for multiple collections, specified on the command line. f. Collection metadata for multiple collections, specified in an input file. g. Granule and collection metadata for all collections in a Datapool Group, specified on the command line. h. Granule and collection metadata for all collections in a Datapool Group, specified in an input file. i. Granule deletions for all logically deleted granules in a collection, specified on the command line.		

				j. Granule deletions for all logically deleted granules in a collection, specified in an input file. k. Granule metadata for all non-deleted granules in a collection, specified on the command line. l. Granule metadata for all non-deleted granules in a collection, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test granules' collection is installed. (ESDT Verification script)		
4	Ensure the test granules' collection is configured for collection and granule export.		
5	Ensure the test granules are ingested.		
6	Setup		#comment
7	S-1 From the command line invoke manual export of metadata for a) Granule metadata for multiple granules, specified on the command line.		#comment
8	Run BMGT manual export on granules G1, G2, G3, using GranuleIds for G1, G3, and geoid for G2: EcBmBMGTManualStart --mode <MODE> --metg --granules <G1_GRANULE_ID>,SC:<G2_ESDT>:<G2_GRANULE_ID>,<G3_GRANULE_ID>		
9	Verification		#comment
10	V-1 Verify that the operation in S-1 results in a) Single HTTP PUT for each granule.		#comment
11	Verify a single HTTP PUT is sent for each granule G1, G2, G3. from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
12	Repeat the test, replacing the --granules option with the short form: -g		

TEST DATA:

Test data is under /sotestdata/DROP_802/BE_82_01/Criteria/242

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
242 S-1a				3 granules (G1 ... G3)			242_1_A	
242 S-1b				3 granules (G4 ... G6)			242_1_B	
242 S-1c				1 collection (C1) with 3 granules (G7 ... G9)			242_1_C	
242 S-1d				1 collection (C2) with 3 granules (G10 ... G12)			242_1_D	
242 S-1e				3 collections (C3 ... C5)			242_1_E	
242 S-1f				3 collections (C6 ... C8)			242_1_F	
242 S-1g				1 collection (C9) with 3 granules (G13 ... G15) 1 collection (C10) with 3 granules (G16 ... G18)			242_1_G	
242 S-1h				1 collection (C11) with 3 granules (G19 ... G21)			242_1_H	
242 S-1i				1 collection (C12) with 6 granules (G22 ... G27)			242_1_I	
242 S-1j				1 collection (C13) with 6 granules (G28 ... G33)			242_1_J	
242 S-1k				1 collection (C14) with 6 granules (G34 ... G39)			242_1_K	
242 S-1l				1 collection (C15) with 6 granules (G40 ... G45)			242_1_L	

EXPECTED RESULTS:

	V	242	1	Verify that the operation in S-1 results in the following exports: c. Single HTTP PUT. d. Single HTTP PUT. e. HTTP PUT for each granule in the collection. f. HTTP PUT for each granule in the collection. g. HTTP PUT for each specified collection. h. HTTP PUT for each specified collection. i. HTTP PUT for each granule and collection in the specified group. j. HTTP PUT for each granule and collection in the specified group. k. HTTP DELETE for each logically deleted granule in the specified collection. l. HTTP DELETE for each logically deleted granule in the specified collection. m. HTTP PUT for each non-deleted granule in the specified collection. n. HTTP PUT for each non-deleted granule in the specified collection.		

1.51 Automatic/Manual Interaction (ECS-ECSTC-50)

DESCRIPTION:

	S	250	1	**[Automatic/Manual Interaction]** Execute the manual, automatic, and verification drivers in such a way that they run concurrently.		
	S	250	2	Stop the automatic driver polling process.		
	S	250	3	Update a granule.		
	S	250	4	Initiate a manual export for the granule that was updated in S-3.		
	S	250	5	Start the automatic driver polling process.		

	S	250	6	Pause the processing of export requests in BMGT.		
	S	250	7	Perform the following in the order specified: Create at least 1000 events in the database and allow them to be picked up by the automatic poller (i.e. added to the BMGT Export Request table). Initiate a manual export for at least 1000 granules, initiate a long form verification for at least 1000 granules, and start an incremental verification with the max number of granules set to at least 1000.		
	S	250	8	Resume processing of export requests in BMGT.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection is installed.		
4	Ensure the test collection is enabled for collection and granule export.		
5	Ensure the test granules are ingested.		
6	Setup		#comment
7	S-2 Stop the automatic driver polling process.		#comment
8	EcBmBMGTAutoStop <MODE>		
9	S-3 Update a granule.		#comment
10	Modify a granule's DayNightFlag: update amgranule set daynightflag = 'Both' where granuleid = <GRANULE_ID>		Note that the dayNightFlag is not extracted from the DB by BMGT, so this will not really result in an update. It probably will however cause lastupdatetime to be incremented and an event to be generated, but the resulting export will not reflect the new dayNightFlag. 3/25/2013 -- Goff, Timothy
11	S-4 Initiate a manual export for the granule that was updated in S-3.		#comment

#	Action	Expected Result	Notes
12	Manually export the updated granule: EcBmBMGTManualStart $MODE --metg --granules $GRANULEID		
13	S-5 Start the automatic driver polling process.		#comment
14	EcBmBMGTAutoStart <MODE>		
15	S-6 Pause the processing of export requests in BMGT.		#comment
16	In the BMGT GUI system status tab, press the 'pause' button next to 'Dispatcher'		
17	S-7 Perform the following in the order specified: Create at least 1000 events in the database and allow them to be picked up by the automatic poller (i.e. added to the BMGT Export Request table). Initiate a manual export for at least 1000 granules. Initiate a long form verification for at least 1000 granules. Start an incremental verification with the max number of granules set to at least 1000.		#comment
18	Prepare an input file, manual.txt, with 1000 granule IDs for manual export.		
19	Prepare an input file, verification.txt, with 1000 different granule IDs for a long form verification.		
20	In the BMGT GUI, BMGT Configuration tab, set BMGT.Incremental.MaxGranules to at least 1000.		
21	Modify 1000 granules' DayNightFlag values: update amgranule set daynightflag = 'Both' where granuleid in (<GRANULE_ID>,...)		See note above. If the intention is just to make a chaneg which will cause an export this will probably work, but need to verify. 3/25/2013 -- Goff, Timothy
22	Wait for the update events to be added to the BMGT Export Request table.		
23	In one process, start a manual export: EcBmBMGTManualStart <MODE> --metg --granulefile manual.txt		
24	In a separate process, start a long form verification: EcBmBMGTManualStart <MODE> --metg --long --granulefile verification.txt If given a warning about another process running, choose to continue.		
25	In a separate process, start an incremental verification: EcBmBMGTManualStart <MODE> --metg --incremental If given a warning about another process running, choose to continue.		
26	V-1 Verify that all of the export requests enqueued in S-7 are eligible for export at the same time by viewing the export request table prior to resuming		#comment

#	Action	Expected Result	Notes
	processing of export requests in BMGT.		
27	From the logs, determine the batch id associated with the manual and verification exports. then: select count(1) from bm_export_requests where granuleid in (<UPDATED_GRANULE_IDS>) or batchid = <MANUAL_BATCH_ID> or batchid =<VERIFICATION_BATCH_ID> or batchid= <INCREMENTAL_BATCH_ID> and status = 'PENDING' Verify the count is 4000.		
28	S-8 Resume processing of export requests in BMGT.		#comment
29	In the BMGT GUI system status tab, press the 'resume' button next to 'Dispatcher'		
30	Verification		#comment
31	V-2 Verify that the drivers started in S-1 all have processing times which overlap (look at start and stop times in the log).		#comment
32	Verify, by inspecting the logs, that the latest start time precedes the earliest end time of the manual, long form verification, incremental verification, and automatic drivers.		
33	V-3 Verify that there are no errors caused by the concurrent execution and that none of the drivers are made to wait for another to complete.		#comment
34	Verify no errors appear in any of the BMGT logs while the 4 drivers are running.		
35	Verify that each driver starts and stops independently of the others by verifying that each was instatiated in a separate process and that the drivers execute concurrently.		
36	V-4 Verify that the exports started in S-1 complete successfully.		#comment
37	Verify the logs indicate that the manual export completes successfully.		
38	Verify the logs indicate that the long form verification completes successfully.		
39	Verify the logs indicate that the incremental verification completes successfully.		
40	Verify the logs indicate that the automatic updates complete successfully.		
41	V-5 Verify that the granule updated in S-3 is successfully exported both by the manual export in S-4 and by the automatic export in S-5 by viewing that both an automatic and manual export of the granule are listed in the GUI for the time frame of this test and are in a terminal and successful state.		#comment
42	Verify the BMGT GUI export request tab shows that the automatic export of		

#	Action	Expected Result	Notes
	the updated granule is complete and successful.		
43	Verify the BMGT GUI export requests tab shows that the manual export of the updated granule is complete and successful.		
44	V-6 Verify that the requests enqueued in S-7 are worked off in such a way that the manual requests, for example do not have to wait for the automatic requests to complete. Each type of export must be able to work off its requests independently of the other types and no type shall have to wait for the requests of another type to complete before processing.		#comment
45	Verify the logs and the bg_export_requests table show that manual requests and automatic requests are interleaved, with start and stop times indicating that manual requests are processed concurrently with automatic requests.		
46	V-7 Verify that the exports started in S-7 all complete successfully.		#comment
47	Verified in V-4.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
					4000 granules			

EXPECTED RESULTS:

	V	250	1	Verify that all of the export requests enqueued in S-7 are eligible for export at the same time by viewing the export request table prior to resuming processing of export requests in BMGT.		
	V	250	2	Verify that the drivers started in S-1 all have processing times which overlap (look at start and stop times in the log).		
	V	250	3	Verify that there are no errors caused by the concurrent execution and that none of the drivers are made to wait for another to complete.		
	V	250	4	Verify that the exports started in S-1 complete successfully.		
	V	250	5	Verify that the granule updated in S-3 is successfully exported both by the manual export in S-4 and by the automatic export in S-5 by viewing that both an automatic and manual export of the granule are listed in the GUI for the time frame of this test and are in a terminal and successful state.		

	V	250	6	Verify that the requests enqueued in S-7 are worked off in such a way that the manual requests, for example do not have to wait for the automatic requests to complete. Each type of export must be able to work off its requests independently of the other types and no type shall have to wait for the requests of another type to complete before processing.		
	V	250	7	Verify that the exports started in S-7 all complete successfully.		

1.52 Manual Export Errors (ECS-ECSTC-51)

DESCRIPTION:

	S	260	1	**[Manual Export Errors]** Modify the XML metadata for a granule and the ODL descriptor for a collection such that they are invalid and their export attempt will cause a 'data related error'. For instance, remove the start or end of an XML or ODL group so that the file fails basic validation.		
	S	260	2	Modify the XML metadata for a granule and the ODL descriptor file for a collection such that the metadata will validate but will be invalid for ingest into ECHO. For instance, cause the granule or collection start date to be after the end date. (Alternatively, if using an ECHO stand-in, ensure that it returns error responses on ingest of these items.)		
	S	260	3	Perform a manual export of the granules and collections in S-1 and S-2.		
	S	260	4	Request the manual export of granule and collection metadata for a collection which is not enabled for export.		
	S	260	5	Move the metadata file for a granule, such that it will not be found at the location indicated by its database record.		
	S	260	6	Move the descriptor file for a collection, such that it will not be found at the location indicated by its database record.		
	S	260	7	Perform a manual export of the granules and collections in S-5 and S-6.		
	S	260	8	Replace the files moved in S-5 and S-6.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment

#	Action	Expected Result	Notes
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collections C1, C2, C3, C4 are installed.		
4	Ensure the test collections C1, C2, C4 are enabled for collection and granule export.		
5	Ensure the test collection C3 is not enabled for collection or granule export.		
6	Ensure the test granules are ingested.		
7	Configure the email address in the bg_configuration_property table to an email where verification can be done.		
8	Setup		#comment
9	S-1 Modify the XML metadata for a granule and the ODL descriptor for a collection such that they are invalid and their export attempt will cause a ‘data related error’. For instance, remove the start or end of an XML or ODL group so that the file fails basic validation.		#comment
10	Find granule G1's metadata file in the small file archive: /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
11	Copy the file so that it can be restored later.		
12	Remove the final closing tag.		
13	Find collection C1's descriptor file: /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
14	Copy the file so it can be restored later.		
15	Remove the final END_GROUP.		
16	S-2 Modify the XML metadata for a granule and the ODL descriptor file for a collection such that the metadata will validate but will be invalid for ingest into ECHO. For instance, cause the granule or collection start date to be after the end date. (Alternatively, if using an ECHO stand-in, ensure that it returns error responses on ingest of these items.)		#comment
17	Find granule G2's metadata file in the small file archive: /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
18	Copy the file so it can be restored later.		
19	Change the RangeEndingDate to be 1 year before the RangeBeginningDate. Some granules do not have RangeBeginningDate and RangeEndingDate. If it has a		

#	Action	Expected Result	Notes
	SingleDateTime, change the date to be outside the collection's beginning and ending date.		
20	Find collection C2's descriptor file: /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
21	Copy the file so it can be restored later.		
22	Change the RangeEndingDate to be 1 year before the RangeBeginningDate.		
23	S-3 Perform a manual export of the granules and collections in S-1 and S-2.		#comment
24	EcBmBMGTManualStart <MODE> --metc --collections <C1>,<C2>		
25	Correct the errors in the collection in before request the granule export. If the collection export has an error and is blocked the granule will remain pending.		
26	EcBmBMGTManualStart <MODE> --metg --granules <G1>,<G2>		
27	S-4 Request the manual export of granule and collection metadata for a collection which is not enabled for export.		#comment
28	Manually export collection C3: EcBmBMGTManualStart <MODE> --metc --collections <C3>		
29	Manually export a granule belonging to collection C3: EcBmBMGTManualStart <MODE> --metg --granules <G3>		
30	S-5 Move the metadata file for a granule, such that it will not be found at the location indicated by its database record.		#comment
31	Change to granule G4's metadata directory in the small file archive: cd /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>		
32	Create a tmp directory below the granule's directory: mkdir tmp		
33	Move the granule's XML file into the tmp directory: mv <SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml tmp		
34	S-6 Move the descriptor file for a collection, such that it will not be found at the location indicated by its database record.		#comment
35	Change to the collection's descriptor directory: cd /stornext/smallfiles/<MODE>/descriptor/		
36	Create a tmp directory below the descriptor file's directory: mkdir tmp		

#	Action	Expected Result	Notes
37	Move the descriptor file into the tmp directory. mv *<SHORT_NAME>.<VERSION_ID>.desc tmp		
38	S-7 Perform a manual export of the granules and collections in S-5 and S-6.		#comment
39	Manually export collection C4: EcBmBMGTManualStart <MODE> --metc --collections <C4>		
40	Manually export granule G4: EcBmBMGTManualStart <MODE> --metg --granules <G4>		
41	S-8 Replace the files moved in S-5 and S-6.		#comment
42	Restore granule G4's XML file.		
43	Remove the tmp directory.		
44	Restore collection C4's descriptor file.		
45	Remove the tmp directory.		
46	Verification		#comment
47	V-1 Verify that the exports in S-3 encounter errors.		#comment
48	Verify the BMGT log indicates collections C1, C2 and granules G1, G2 encounter errors.		
49	V-2 Verify that the export requests for the items in S-1 and S-2 are marked as ‘failed’ in the GUI and that the associated errors can be viewed for each item.		#comment
50	Verify the BMGT GUI shows export attempts for collections C1, C2 and granules G1, G2.		
51	Verify in the BMGT GUI exports are maked SKIPPED for collection C1 and granule G1; BLOCKED for collection C2 and granule G2.		
52	Verify the BMGT GUI allows viewing of errors associated with exports for collections C1, C2 and granules G1, G2.		
53	V-3 Verify that an email is sent indicating that the items in S-1 failed due to a data related error and were skipped.		#comment
54	Verify an email is sent to the the address configured in the BMGT GUI.		
55	Verify the email says that collection C1 and granule G1 failed due to data related errors and were skipped.		
56	V-4 Verify that an email is sent indicating that the items in S-2 failed due to an ECHO ingest error. It is acceptable for this to be in the same email as the errors for S-1.		#comment
57	Verify the email says that collection C2 and granule G2 failed due to ECHO ingest errors.		

#	Action	Expected Result	Notes
58	V-5 Verify that the email messages above list the error messages and the number of items which encountered each error.		#comment
59	Verify the email lists the error messages from the BMGT GUI.		
60	Verify the email lists the number of items for each error.		
61	V-6 Verify that messages are printed to the log file for the errors in S-1 and S-2, including the item (collection or granule) ID, and the reason for the error.		#comment
62	Verify errors for collections C1, C2 and granules G1, G2 appear in the BMGT log file.		
63	Verify the logged errors include collection IDs for C1, C2.		
64	Verify the logged errors include granule IDs for G1, G2.		
65	Verify the logged errors include the reason for each error.		
66	V-7 Verify that in the GUI it is possible to list only those items which encountered data related errors.		#comment
67	Verify the BMGT GUI allows filtering on only data related errors.		
68	V-8 Verify that in the GUI it is possible to list only those items which encountered ECHO ingest errors.		#comment
69	Verify the BMGT GUI allows filtering on only ECHO ingest errors.		
70	V-9 Verify that no HTTP export request was made for the items in S-1, as their failure occurred prior to export and export was therefore ‘skipped’		#comment
71	Verify the mock ECHO records no requests for C1 or G1.		
72	V-10 Verify that for the export attempt in S-4, an error is printed to the log indicating that the requested items were not eligible for export.		#comment
73	Verify the BMGT log indicates C3 and G3 were not eligible for export.		
74	V-11 Verify that the items in S-4 are not added to the export request queue, and no export attempt is made.		#comment
75	Verify rows for C3 and G3 do not appear in the bg_export_request table.		
76	Verify the ECHO proxy logs no HTTP request for C3 or G3.		
77	V-10 Verify that the export of the granule and collection whose files were moved in S-5 and S-6 encounter an error, which is printed in the log.		#comment
78	Verify the BMGT log indicates that exporting C4 and G4 encountered errors.		
79	V-11 Verify that the export of the granule and collection in S-5 and S-6 is retried until the files are replaced, at which point the export succeeds.		#comment
80	Verify the BMGT log indicates G4 and C4 exports are retried until the metadata and descriptor files are restored.		
81	Verify the BMGT log indicates G4 and C4 exports succeed after the metadata		

#	Action	Expected Result	Notes
	and descriptor files are restored.		
82	V-12 Verify that no email is sent for the errors related to the granule and collection in S-5 and S-6.		#comment
83	Verify no email is sent to the configured email address for G4 or C4.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
				4 collections (C1, C2, C3, C4)				
				4 granules (G1, G2, G3, G4)				

EXPECTED RESULTS:

	V	260	1	Verify that the exports in S-3 encounter errors.		
	V	260	2	Verify that the export requests for the items in S-1 and S-2 are marked as ‘failed’ in the GUI and that the associated errors can be viewed for each item.		
	V	260	3	Verify that an email is sent indicating that the items in S-1 failed due to a data related error and were skipped.		
	V	260	4	Verify that an email is sent indicating that the items in S-2 failed due to an ECHO ingest error. It is acceptable for this to be in the same email as the errors for S-1.		
	V	260	5	Verify that the email messages above list the error messages and the number of items which encountered each error.		
	V	260	6	Verify that messages are printed to the log file for the errors in S-1 and S-2, including the item (collection or granule) ID, and the reason for the error.		
	V	260	7	Verify that in the GUI it is possible to list only those items which encountered data related errors.		

	V	260	8	Verify that in the GUI it is possible to list only those items which encountered ECHO ingest errors.		
	V	260	9	Verify that no HTTP export request was made for the items in S-1, as their failure occurred prior to export and export was therefore 'skipped'		
	V	260	10	Verify that for the export attempt in S-4, an error is printed to the log indicating that the requested items were not eligible for export.		
	V	260	11	Verify that the items in S-4 are not added to the export request queue, and no export attempt is made.		
	V	260	10	Verify that the export of the granule and collection whose files were moved in S-5 and S-6 encounter an error, which is printed in the log.		
	V	260	11	Verify that the export of the granule and collection in S-5 and S-6 is retried until the files are replaced, at which point the export succeeds.		
	V	260	12	Verify that no email is sent for the errors related to the granule and collection in S-5 and S-6.		

1.53 GUI (ECS-ECSTC-52)

DESCRIPTION:

	S	270	1	**[GUI]** All of the following steps shall be performed using each of the browsers and operating systems listed in "Technical Document 910-TDA-042, Browsers Baseline"		
	S	270	2	Navigate to the BMGT GUI. Enter as an operator, without administrative privileges.		
	S	270	3	Attempt to suspend BMGT processing via the GUI.		
	S	270	4	Navigate to the collection configuration page, and attempt to enable or disable a collection for export.		
	S	270	5	Navigate to the BMGT configuration page, and attempt to modify a configuration parameter.		
	S	270	6	Navigate to the current/history export page. Modify the number of items to display per page.		

	S	270	7	Navigate to the failed export page. Modify the number of items to display per page. Ensure that there are failed export requests in the system prior to doing this.		
	S	270	8	Log in to the BMGT GUI as an administrator.		
	S	270	9	On the main GUI page, suspend BMGT processing, both overall, and for one type of export (e.g. automatic).		
	S	270	10	Navigate to the collection configuration page. Enable a currently disabled collection for export. Disable a currently enabled collection.		
	S	270	11	Navigate to the BMGT configuration page. Modify and save the email address to which alerts are sent.		
	S	270	12	On the main GUI page, resume all BMGT processing.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
3	Ensure at there are at least 10 failed BMGT exports. E.g., in the small file archive modify a granule's metadata so that the RangeBeginningDate occurs after the RangeEndingDate; manually export the granule 10 times. Alternatively, configure mock ECHO to reject a specific granule, then manually export that granule at least 10 times.		
4	Ensure collections C1, C2 are installed.		
5	Ensure collection C1 is disabled for collection export.		
6	Ensure collection C2 is enabled for collection and granule export.		
7	Ensure granules G1, G2 are in AIM.		
8	Setup		#comment
9	S-1 All of the following steps shall be performed using each of the browsers and operating systems listed in “Technical Document 910-TDA-042, Browsers Baseline”		#comment
10	[FIXME] Identify browsers an operating systems in 910-TDA-042. Repeat the test for each combination.		
11	S-2 Navigate to the BMGT GUI. Enter as an operator, without		#comment

#	Action	Expected Result	Notes
	administrative privileges.		
12	Open the BMGT GUI in a browser. Do not log in with a password. The BMGT GUI does not use roles (such as operator or administrator). If a password is supplied, read-write access is granted. Without a password, access is limited to read-only.		
13	S-3 Attempt to suspend BMGT processing via the GUI.		#comment
14	In the system status tab, click the 'Pause' button next to 'Dispatcher'		
15	S-4 Navigate to the collection configuration page, and attempt to enable or disable a collection for export.		#comment
16	In the BMGT GUI, on the collection configuration page, attempt to enable or disable a collection for export by clicking the collection export check box next to it.		
17	S-5 Navigate to the BMGT configuration page, and attempt to modify a configuration parameter.		#comment
18	In the BMGT GUI, on the BMGT configuration page, attempt to modify a configuration parameter.		
19	S-6 Navigate to the current/history export page. Constrain the listed export requests to cover only a specific time period.		#comment
20	In the BMGT GUI, on the export request tab, modify the time range for which to display export requests.		
21	S-7 Navigate to the failed export page. Constrain the errors to cover only a specific time period. Ensure that there are failed export requests in the system prior to doing this.		#comment
22	In the BMGT GUI, on the export activity/error tab, filter for failed activities only and then modify the time range for which to display errors.		
23	S-8 Log in to the BMGT GUI as an administrator.		#comment
24	Log in to the BMGT GUI as an administrator.		
25	S-9 On the main GUI page, suspend BMGT processing, both overall, and for one type of export (e.g. automatic).		#comment
26	In the BMGT GUI, on the main page, suspend overall BMGT processing by pressing the 'Pause' button next to 'Dispatcher'.		
27	In the BMGT GUI, on the main page, suspend automatic BMGT processing by pressing the 'Pause' button next to 'EVENT'.		
28	Request a manual export of granule G1: EcBmBMGTManualStart --mode <MODE> --metg --granules <GRANULE_ID>		

#	Action	Expected Result	Notes
29	Update granule G2 by changing its DayNightFlag: update amgranule set daynightflag = 'Both' where granuleid = <GRANULE_ID>		This is not used by BMGT. BMGT uses the value in the native XML. though it may generate a GRUPDATE event... not sure 3/21/2013 -- Goff, Timothy
30	S-10 Navigate to the collection configuration page. Enable a currently disabled collection for export. Disable a currently enabled collection.		#comment
31	In the BMGT GUI, on the collection configuration page, enable collection C1 for export.		
32	In the BMGT GUI, on the collection configuration page, disable collection C2 for export.		
33	S-11 Navigate to the BMGT configuration page. Modify and save the email address to which alerts are sent.		#comment
34	In the BMGT GUI, on the BMGT configuration page, modify the email address to which alerts are sent.		
35	S-12 On the main GUI page, resume all BMGT processing.		#comment
36	In the BMGT GUI, on the main GUI page, resume all BMGT processing.		
37	Log out of the BMGT GUI.		
38	Clear the browser's history, including cache, cookies, and saved sessions.		
39	Restart the browser.		
40	Verification		#comment
41	V-1 Verify that all verification steps pass for each of the browsers and operating systems referred to in S-1.		#comment
42	V-2 Verify that the main GUI page in S-2 provides the number of completed metadata exports, number of pending metadata exports, and statistics about retries, warning, and errors.		#comment
43	Verify the BMGT GUI "Export Requests" page lists the number of completed metadata exports.		
44	Verify the BMGT GUI "Export Requests" page lists the number of pending metadata exports.		
45	Verify the BMGT GUI "Export Activity/Errors" page lists the number of errors.		
46	V-3 Verify that in S-3, it is not possible to suspend BMGT processing.		#comment

#	Action	Expected Result	Notes
47	Verify the BMGT GUI prevents read-only access from suspending BMGT processing.		
48	V-4 Verify that in S-4, all collections are displayed, grouped by their datapool group.		#comment
49	Verify the BMGT GUI collection configuration page displays all collections, grouped by datapool group.		
50	V-5 Verify that in S-4, it is not possible to enable or disable a collection.		#comment
51	Verify the BMGT GUI prevents read-only access from enabling or disabling a collection.		
52	V-6 Verify that in S-5, it is not possible to modify any configuration parameters. Verify however that the notification email address is visible on this page.		#comment
53	Verify the BMGT GUI prevents read-only access from changing configuration parameters.		
54	Verify the BMGT GUI BMGT configuration page displays the notification email address.		
55	V-7 Verify that in S-6, information is displayed for recent and pending export requests (it is allowable for recent and current request to be displayed on separate pages). Verify that each export request is clearly marked as to which type of export initiated it (e.g. automatic, manual, verification).		#comment
56	Verify the BMGT GUI "Export Requests" page displays recent and pending export requests.		
57	Verify the BMGT GUI "Export Requests" page marks each export as one of automatic, manual, or verification.		
58	V-8 Verify that in S-7, information is displayed for recent export failures and errors.		#comment
59	Verify the BMGT GUI "Export Activity/Errors" page displays recent export failures and errors.		
60	V-9 Verify that in S-6 and S-7, it is possible to constrain the items displayed on the page by time.		#comment
61	Verify the BMGT GUI current/history allows filtering to display only those export requests withing a certain time range.		
62	Verify the BMGT GUI failed export page allows filtering to display only those failed exports within a certain time range.		
63	V-10 Verify that after logging in as an administrator, suspending processing in S-9 results in a halting of BMGT processing.		#comment

#	Action	Expected Result	Notes
64	Verify that the manual export request of granule G1 does not get processed.		
65	Verify that granule G2's update does not get automatically processed.		
66	V-11 Verify that in S-10, it is possible to enable and disable export for collections, both for granule and collection metadata export.		#comment
67	Verify the BMGT GUI collection configuration page allows enabling collection C1 for export.		
68	Verify the BMGT GUI collection configuration page allows disabling collection C2 for export.		
69	V-12 Verify that in S-11, the change made to the notification email address is saved to the database.		#comment
70	Verify after restarting the browser, the BMGT GUI BMGT configuration page displays the new email address to which alerts are sent.		
71	V-13 Verify that after requesting resumption of BMGT processing in S-12, export requests resume being picked up and processed.		#comment
72	Verify after resuming BMGT processing, granule G1 is processed to completion as a manual export.		
73	Verify after resuming BMGT processing, granule G2 is processed to completion as an automatic export.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
				2 collections (C1, C2).				
				2 granules (G2, G2).				

EXPECTED RESULTS:

	V	270	1	Verify that all verification steps pass for each of the browsers and operating systems referred to in S-1.		
	V	270	2	Verify that the main GUI page in S-2 provides the number of completed metadata exports(over some period), number of pending metadata exports, and statistics about retries, warning, and errors.		

	V	270	3	Verify that in S-3, it is not possible to suspend BMGT processing.		
	V	270	4	Verify that in S-4, all collections are displayed, grouped by their datapool group.		
	V	270	5	Verify that in S-4, it is not possible to enable or disable a collection.		
	V	270	6	Verify that in S-5, it is not possible to modify any configuration parameters. Verify however that the notification email address is visible on this page.		
	V	270	7	Verify that in S-6, information is displayed for recent and pending export requests (it is allowable for recent and current request to be displayed on separate pages). Verify that each export request is clearly marked as to which type of export initiated it (e.g. automatic, manual, verification).		
	V	270	8	Verify that in S-7, information is displayed for recent export failures and errors.		
	V	270	9	Verify that in S-6 and S-7, it is possible to change the number of items displayed on the page.		
	V	270	10	Verify that after logging in as an administrator, suspending processing in S-9 results in a halting of BMGT processing.		
	V	270	11	Verify that in S-10, it is possible to enable and disable export for collections, both for granule and collection metadata export.		
	V	270	12	Verify that in S-11, the change made to the notification email address is saved to the database.		
	V	270	13	Verify that after requesting resumption of BMGT processing in S-12, export requests resume being picked up and processed.		

1.54 Configuring Collections For Export (ECS-ECSTC-53)

DESCRIPTION:

	S	280	1	**[Configuring Collections For Export]** Add a new ESDT to the ECS system and ingest some granules in that collection.		

	S	280	2	Go to the GUI collection configuration page.		
	S	280	3	Enable the collection for collection export.		
	S	280	4	Allow the export of the collection metadata to complete.		
	S	280	5	Enable the collection for granule export.		
	S	280	6	Find a collection which is not enabled for collection or granule export. Request the manual export of collection and granule metadata for this collection.		
	S	280	7	Find a collection which is enabled for collection, but not granule export. Request the manual export of collection and granule metadata for this collection.		
	S	280	8	Find a collection which is enabled for collection and granule export. Request the manual export of collection and granule metadata for this collection.		
	S	280	9	Find a collection which is not enabled for collection or granule export, but for which there is another ESDT with the same short name, but different version, which is. Request the manual export of collection and granule metadata for this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C1 is not installed.		
3	Ensure bg_collection_configuration has no row for collection C1.		
4	Ensure granules g1, g2, g3 are not in AIM.		
5	Ensure collections C2, C3, C4, C5, C6 are installed.		
6	Ensure granules g4 ... g18 are in AIM		
7	Ensure collection C2 is disabled for both collection and granule export.		
8	Ensure collection C3 is enabled for collection but not for granule export.		
9	Ensure collection C4 is enabled for both collection and granule export.		
10	Ensure collection C5 is disabled for collection or and granule export.		

#	Action	Expected Result	Notes
11	Ensure collection C6 is enabled for both collection and granule export.		
12	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
13	Setup and Verification		#comment
14	S-1 Add a new ESDT to the ECS system and ingest some granules in that collection.		#comment
15	Install collection C1.		
16	Add a row for collection C1 to the bg_collection_configuration table, setting both export flags to 'N'. See EcBgPopulateCollections.ksh for examples.		
17	Ingest granules g1, g2, g3.		
18	S-2 Go to the GUI collection configuration page.		#comment
19	Visit the BMGT GUI Collection Configuration page.		
20	V-1 On the collection configuration page in S-2, verify that all currently installed collections are listed with their current enabled/disabled status.		#comment
21	Verify the BMGT GUI's collection configuration page lists collections C1 ... C6.		
22	Verify the BMGT GUI indicates C1 is disabled for both collection and granule export.		
23	Verify the BMGT GUI indicates C2 is disabled for both collection and granule export.		
24	Verify the BMGT GUI indicates C3 is enabled collection export but disabled for granule export.		
25	Verify the BMGT GUI indicates C4 is enabled for both collection and granule export.		
26	Verify the BMGT GUI indicates C5 is disabled for both collection and granule export.		
27	Verify the BMGT GUI indicates C6 is enabled for both collection and granule export.		
28	V-2 On the collection configuration page in S-2, verify that the new collection is listed but not enabled for collection or granule export.		#comment
29	Verify collection C1 appears on the BMGT GUI's collection configuration page.		
30	Verify the BMGT GUI indicates C1 disabled for both collection and granule export.		
31	S-3 Enable the collection for collection export.		#comment
32	Enable C1 for collection export by clicking the collection export check box		

#	Action	Expected Result	Notes
	next to the collection.		
33	V-3 After enabling the collection for collection export in S-3, verify that the collection metadata for the collection is automatically exported.		#comment
34	Verify that after C1 is enabled for collection export, the BMGT GUI indicates C1's metadata is successfully exported.		
35	Verify that after C1 is enabled for collection export, a BMGT log records the export of C1's metadata.		
36	Verify that after C1 is enabled for collection export, the TCP proxy logs an HTTP PUT whose body contains C1's metadata.	The request should begin with a line such as PUT /catalog-rest/providers/${PROVIDERID}/datasets/${DATASETID} HTTP/1.1 where ${DATASETID} is the URL-encoded longname + ' V' + versionid, e.g., MODIS%2FAqua%20Gross%20Primary%20Productivity%208-Day%20L4%20Global%201km%20SIN%20Grid%20V005	
37	S-4 Allow the export of the collection metadata to complete.		#comment
38	Wait for collection C1's export request to be marked complete: select r.completiontime, r.status from bg_export_request r join amcollection c on c.colelctionid = r.collectionid where c.ShortName = '<SHORT_NAME>' and c.VersionId = <VERSION_ID> and itemtype = 'CL; Or, wait for the TCP proxy to log C1's HTTP PUT request.		
39	S-5 Enable the collection for granule export.		#comment
40	In the BMGT GUI, enable C1 for granule export by checking the granule export check box next to the collection in the collection configuration tab.		
41	V-4 After enabling the collection for granule export in S-5, verify that the granule metadata for every granule in the collection is automatically exported.		#comment
42	Verify that after C1 is enabled for granule export, the BMGT GUI indicates all metadata for all granules belonging to C1 is successfully exported (except logically deleted granules).		
43	Verify that after C1 is enabled for granule export, a BMGT log records the export of C1 granules' metadata (except logically deleted granules).		
44	Verify that after C1 is enabled for granule export, the TCP proxy logs an	Each granule request should begin	

#	Action	Expected Result	Notes
	HTTP PUT for each of C1's granules, containing granule metadata (except logically deleted granules).	with a line such as PUT /catalog-rest/providers/${PROVIDERID}/granules/${GRANULEUR} HTTP/1.1 where ${GRANULEUR} is a URL-encoded geoid, such as SC%3AMYD17A2.005%3A20062	
45	S-6 Find a collection which is not enabled for collection or granule export. Request the manual export of collection and granule metadata for this collection.		#comment
46	Request manual export of C2: EcBmBMGTManualStart --mode <MODE> --metg --metc --collections <C2_SHORT_NAME>.<C2_VERSION_ID>		
47	V-5 For the export attempt in S-6, verify that nothing is exported.		#comment
48	Verify the TCP proxy logs no request for C2's collection or granule metadata.		
49	Verify a BMGT log indicates no request is attempted for C2's collection or granule metadata.		
50	S-7 Find a collection which is enabled for collection, but not granule export. Request the manual export of collection and granule metadata for this collection.		#comment
51	Request manual export of C3: EcBmBMGTManualStart --mode <MODE> --metg --metc --collections <C3_SHORT_NAME>.<C3_VERSION_ID>		
52	V-6 For the export attempt in S-7, verify that only collection metadata is exported.		#comment
53	Verify the TCP proxy logs a single HTTP PUT request, containing C3's collection metadata. Get the request ID for the collection export from the GUI. Look in the BMGT manual log for pattern like "requestId":18629,"batchId":76		
54	Verify the TCP proxy logs no HTTP PUT requests for C3's granules.		
55	Verify a BMGT log records a single export request for C3's collection metadata.		
56	Verify a BMGT log records no export attempts for any C3 granule metadata.		
57	S-8 Find a collection which is enabled for collection and granule export. Request the manual export of collection and granule metadata for this collection.		#comment
58	Request manual export of C4: EcBmBMGTManualStart --mode		

#	Action	Expected Result	Notes
	<MODE> --metg --metc --collections <C4_SHORT_NAME>.<C4_VERSION_ID>		
59	V-7 For the export attempt in S-8, verify that both collection and granule metadata is exported.		#comment
60	Verify the TCP proxy logs a single HTTP PUT request, containing C4's collection metadata.		
61	Verify the TCP proxy logs a single HTTP PUT request for each of C4's granules, containing granule metadata (excepting any granules which are logically deleted).		
62	Verify a BMGT log records a single export request for C4's collection metadata.		
63	Verify a BMGT log records a single export request for each of C4's granules (excepting those which are logically deleted).		
64	S-9 Find a collection which is not enabled for collection or granule export, but for which there is another ESDT with the same short name, but different version, which is. Request the manual export of collection and granule metadata for this collection.		#comment
65	Request manual export of C5: EcBmBMGTManualStart --mode <MODE> --metg --metc --collections <C5_SHORT_NAME>.<C5_VERSION_ID>		
66	V-8 For the export attempt in S-9, verify that nothing is exported.		#comment
67	Verify the TCP proxy logs no request for C5's collection or granule metadata.		
68	Verify a BMGT log indicates no request is attempted for C5's collection or granule metadata.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
280				6 collections (C1 ... C6)			/sotestdata/DROP_802/BE_82_01/Criteria/280	
280				3 granules for each collection (g1 ... g18)			/sotestdata/DROP_802/BE_82_01/Criteria/280	

EXPECTED RESULTS:

	V	280	1	On the collection configuration page in S-2, verify that all currently installed collections are listed with their current enabled/disabled status.		
	V	280	2	On the collection configuration page in S-2, verify that the new collection is listed but not enabled for collection or granule export.		
	V	280	3	After enabling the collection for collection export in S-3, verify that the collection metadata for the collection is automatically exported.		
	V	280	4	After enabling the collection for granule export in S-5, verify that the granule metadata for every granule in the collection is automatically exported.		
	V	280	5	For the export attempt in S-6, verify that nothing is exported.		
	V	280	6	For the export attempt in S-7, verify that only collection metadata is exported.		
	V	280	7	For the export attempt in S-8, verify that both collection and granule metadata is exported.		
	V	280	8	For the export attempt in S-9, verify that nothing is exported.		

1.55 Re-Export Errors (ECS-ECSTC-54)

DESCRIPTION:

	S	290	1	**[Re-Export Errors]** Configure BMGT to perform re-exports without requiring operator interaction		
	S	290	2	Export a granule insert for a granule whose collection is enabled for export but does not exist in ECHO (e.g. manually remove the collection from ECHO), or perform the export to an ECHO stand in and artificially cause the response to indicate that the associated collection could not be found.		
	S	290	3	Configure BMGT to require operator interaction before processing re-exports.		
	S	290	4	Export a granule insert for a granule whose collection is enabled for export but does not exist in ECHO (e.g. manually		

				remove the collection from ECHO), or perform the export to an ECHO stand-in and artificially cause the response to indicate that the associated collection could not be found.		
	S	290	5	Perform a corrective re-export		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
4	Verify test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/290.		
5	Ensure test collections C1, C1 are installed. E.g., the DPL Ingest GUI shows the collections as configured data types.		
6	Ensure collections C1, C2 are enabled for collection and granule export. For each collection, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure granule g1, in collection C1, has been ingested: select * from amgranule where granuleid = ${g1_GRANULEID}		
8	Ensure ECHO has no metadata for collection C1: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO returns C1's metadata, delete it: curl -k -XDELETE -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-		

#	Action	Expected Result	Notes
	rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID}		
9	Ensure ECHO has metadata for collection C2: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Setup		#comment
12	S-1 Configure BMGT to perform re-exports without requiring operator interaction		#comment
13	Enable automatic corrective export in the BMGT GUI configuration tab by setting BMGT.ResponseHandler.BlockCorrectiveExports to false.		
14	S-2 Export a granule insert for a granule whose collection is enabled for export but does not exist in ECHO (e.g. manually remove the collection from ECHO), or perform the export to an ECHO stand in and artificially cause the response to indicate that the associated collection could not be found.		#comment
15	Update BMGT.ResponseHandler.MaxRetryCount to 0 and bounce the dispatcher.		
16	Manually export granule g1 ./EcBmBMGTManualStart ${MODE} --metg -g ${g1_GRANULEID}		
17	V-1 Verify that an export request for the collection associated with the granule in S-2 has been added to the queue.		#comment
18	Verify that the GUI and database show an export request for Collection C1.		
19	V-2 Verify that the original export request from S-2 has been re-queued.		#comment
20	Verify that the GUI and database show an export request for granule g1 into Collection C1.		
21	V-3 Verify that without any additional operator interaction, the collection is exported to ECHO or an ECHO stand-in, followed by the granule. The collection must be exported before the granule.		#comment
22	Verify that the TCP proxy shows an HTTP PUT request for Collection C1. (There may be more than one HTTP request, e.g., if there are network issues.)		
23	Verify that the TCP proxy shows a HTTP PUT request for granule g1 AFTER collection C1.		
24	S-3 Configure BMGT to require operator interaction before processing re-		#comment

#	Action	Expected Result	Notes
	exports.		
25	Disable automatic corrective export in the BMGT GUI configuration tab by setting BMGT.ResponseHandler.BlockCorrectiveExports to true.		
26	S-4 Export a granule insert for a granule whose collection is enabled for export but does not exist in ECHO (e.g. manually remove the collection from ECHO), or perform the export to an ECHO stand-in and artificially cause the response to indicate that the associated collection could not be found.		#comment
27	Enable Collection C2 for export.		
28	Manually remove it from ECHO (or simulate this in the mock ECHO).		
29	Manually export granule g2: ./EcBmBMGTManualStart <MODE> --metg -g <granule_g2>		
30	V-4 Verify that an export request for the collection associated with the granule in S-4 has been added to the queue with a status of ‘blocked’.		#comment
31	Verify that the GUI and database have an export request for Collection C2 with status 'blocked'.		
32	V-5 Verify that the original export request from S-4 has been re-queued and is not picked up for processing until after the newly added collection request is exported.		#comment
33	Verify that the GUI and database indicate an export request queued for granule g2 with status 'pending'.		
34	Verify that the export request for granule G2 remains in 'pending' until after the newly added collection export request for collection C2 is exported.		
35	V-6 Verify that the collection and granule ARE NOT automatically exported.		#comment
36	Verify that the TCP proxy does not show any exports for Collection C2 or granule g2.		
37	S-5 Perform a corrective re-export		#comment
38	./EcBmBMGTManualStart <MODE> --corrective		
39	V-8 Verify that the collection is exported to ECHO or an ECHO stand-in, followed by the granule. The collection must be exported before the granule.		#comment
40	Verify that the TCP proxy shows that an HTTP PUT request for Collection C2. (There may be more than one HTTP request, e.g., if there are network issues.)		
41	Verify that the TCP proxy shows an HTTP PUT request for the granule g2. (There may be more than one HTTP request, e.g., if there are network issues.)		

#	Action	Expected Result	Notes
42	V-9 Verify that an email is sent, within a reasonable time, to the configured notification email address, indicating that the exports in S-2 and S-4 resulted in an error which caused the requeue of the export request as well as a new export request.		#comment
43	Verify that an email is sent to the configured email address.		
44	Verify that the email is sent within a reasonable time.		
45	Verify that the email indicates the export error and re-export information for granules g1 and g2.		
46	V-10 Verify that the requests in S-2 and S-4 are marked as complete with warnings and that the BMGT GUI indicates that they have been resolved by a re-export.		#comment
47	Verify that the bmgt gui export request tab indicates that granules g1 and g2 have been re-exported.		
48	V-11 Verify that the BMGT GUI provides statistics on the number of errors which resulted in re-queueing and queueing of new exports.		#comment
49	In the BMGT GUI export request tab, verify that it is possible to determin the number of exports which resulted in re exports.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
290		2Collections C1 and C2 with a few science granules	g3acld.003 g3alsp.003				/sotestdata/DROP_802/BE_82_01/Criteria/290	

EXPECTED RESULTS:

	V	290	1	Verify that an export request for the collection associated with the granule in S-2 has been added to the queue.		
	V	290	2	Verify that the original export request from S-2 has been re-queued.		
	V	290	3	Verify that without any additional operator interaction, the collection is exported to ECHO or an ECHO stand-in, followed by the granule. The collection must be exported before the granule.		

	V	290	4	Verify that an export request for the collection associated with the granule in S-4 has been added to the queue with a status of 'blocked'.		
	V	290	5	Verify that the original export request from S-4 has been re-queued with a status of 'blocked'.		
	V	290	6	Verify that the collection and granule ARE NOT automatically exported.		
	V	290	7	Perform a corrective re-export via the GUI or command line.		
	V	290	8	Verify that the collection is exported to ECHO or an ECHO stand-in, followed by the granule. The collection must be exported before the granule.		
	V	290	9	Verify that an email[TR1][TG2] is sent, within a reasonable time[TR3][TG4], to the configured notification email address, indicating that the exports in S-2 and S-4 resulted in an error which caused the requeue of the export request as well as a new export request.		
	V	290	10	Verify that the requests in S-2 and S-4 are marked as complete with warnings and that the BMGT GUI indicates that they have been resolved by a re-export.		
	V	290	11	Verify that the BMGT GUI provides statistics[TR5][TG6] on the number of errors which resulted in re-queueing and queueing of new exports.		

1.56 Ignorable Errors (ECS-ECSTC-55)

DESCRIPTION:

	S	300	1	**[Ignorable Errors]** Export a granule delete for a granule which has never been exported, or perform the export and artificially cause the response to indicate that the granule could not be found. Ensure that the granule is indeed logically or physically deleted from AIM.		
	S	300	2	Export a collection delete for a collection which has never been exported, or perform the export and artificially cause the response to indicate that the collection could not be found. Ensure that the collection is indeed deleted from AIM.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;		
4	Verify test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/300.		
5	Ensure the test collections are installed. E.g., the DPL Ingest GUI shows them as a configured data types.		
6	Ensure the test collections are enabled for collection and granule export. For each collection, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each collection, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure collection C1 has at least one granule in AIM.		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 Export a granule delete for a granule which has never been exported, or perform the export and artificially cause the response to indicate that the granule could not be found. Ensure that the granule is indeed logically or physically deleted from AIM.		#comment
12	Disable automatic export		
13	Ingest a granule g1 in Collection C1.		

#	Action	Expected Result	Notes
14	Delete granule g1 and Verify the deletion from amGranule		
15	Manually export g1: EcBmBMGTManualStart <MODE> --deleteonly --metg -g <g1>		
16	S-2 Export a collection delete for a collection which has never been exported, or perform the export and artificially cause the response to indicate that the collection could not be found. Ensure that the collection is indeed deleted from AIM.		#comment
17	Disable automatic export		
18	Install Collection C2		
19	Delete Collection C2		
20	Manually export C2: EcBmBMGTManualStart <MODE> --deleteonly --metc -c <C2_shortname>.<C2_versionid>		
21	V-1 Verify that the exports in S-1 and S-2 are marked as complete, but that the GUI indicates that these requests received an ignored error.		#comment
22	Verify that the TCP proxy indicates 2 HTTP Deletes exported. (There may be more than two HTTP requests, e.g., if there are network issues.)		
23	Verify that the BMGT GUI indicates ignored error for both requests. Requests will appear complete, but will have an error associated with it on the Export Activity/Error tab for granules.		
24	V-2 Verify that no email is sent to the operator regarding the errors in S-1 and S-2.		#comment
25	Verify that no email is sent to the configured email address.		
26	V-3 Verify that the BMGT GUI provides statistics on the number of errors which were ignored.		#comment
27	Verify that the BMGT GUI export request tab shows the number of requests which encountered errors which were ignored. In BMGT GUI, -Select Date/Time Range covering the period Manual exports were done in S-1 and S-2 -Under Export Requests look for MAN under Export Queue -Under Export-Request Queue Summary locate MAN in Queue column and read values under Success and Warning columns. Compare the total of both columns to number of occurrances of MAN found under Export Queue.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
300		A collection with a few granules (C1)	g3assp.003				/sotestdata/DROP_802/BE_82_01/Criteria/300	
300		Another collection (C2)	g3at.003				/sotestdata/DROP_802/BE_82_01/Criteria/300	

EXPECTED RESULTS:

	V	300	1	Verify that the exports in S-1 and S-2 are marked as complete, but that the GUI indicates that these requests received an ignored error.		
	V	300	2	Verify that no email is sent to the operator regarding the errors in S-1 and S-2.		
	V	300	3	Verify that the BMGT GUI provides statistics on the number of errors which were ignored.		

1.57 Ingest Errors (ECS-ECSTC-56)

DESCRIPTION:

	S	310	1	**[Ingest Errors]** Perform an export of granule metadata which will fail ECHO ingest, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. EndingDateTime before BeginningDateTime).		
	S	310	2	Perform an export of collection metadata which will fail ECHO ingest, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. duplicate Additional Attributes names).		
	S	310	3	Resolve the issue which caused the failure in S-1. Using the GUI, request the re-export of the associated request.		
	S	310	4	Using the GUI, request the cancellation of the request in S-2.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
3	Ensure collection C1 is installed. E.g., ensure the DPL Ingest GUI shows C1 as a configured datatype.		
4	Ensure granule g1 belongs to a different collection than C1.		
5	Ensure granule g1 is in AIM: select * from amgranule where granuleid = <g1_GRANULEID> and deleteeffectivedate is null and deletefromarchive = 'N'		
6	Ensure BMGT is configured to send notification emails to a reachable address.		
7	Ensure BMGT.ResponseHandler.Monitor.MaxErrorCount is set to 2: update bg_configuration_property set propertyvalue = 2 where propertyname = 'BMGT.ResponseHandler.Monitor.MaxErrorCount'		
8	Setup		#comment
9	S-1 Configure BMGT to not automatically retry the type of error which will be used in this test.		#comment
10	update bg_echo_error_policy set bmgtresponse = 'BLOCK' where echoerrorcode = 'RECORD_INVALID'		
11	S-2 Perform an export of granule metadata which will fail ECHO ingest, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. EndingDateTime before BeginningDateTime).		#comment
12	Configure the mock ECHO to respond to granule g1's export with HTTP code 422 and an error message, such as <errors><error code="RECORD_INVALID">Description goes here.</error></errors> While Dispatcher is running and already got the tokens from ECHO, run proxy_stop then run proxy_start422. -OR- Modify the granule metadata in the small file archive to cause an error. E.g., swap the values of RangeBeginningDate and RangeEndingDate.		
13	Manually export granule g1: EcBmBMGTManualStart --mode <MODE> --metg --granules <GRANULE_ID>		
14	S-3 Perform an export of collection metadata which will fail ECHO ingest, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. duplicate Additional Attributes names).		#comment

#	Action	Expected Result	Notes
15	Configure the mock ECHO to respond to collection C1's export with HTTP code 422 and an error message, such as <errors><error code="RECORD_INVALID">Description goes here.</error></errors> Use the proxy 422 from an earlier step. -OR- Modify the collection metadata to cause an ECHO ingest error.		It's nontrivial to get ECHO to reject invalid collection metadata with a RECORD_INVALID error.
16	Manually export collection C1: EcBmBMGTManualStart --mode <MODE> --metc --collections <SHORT_NAME>.<VERSION_ID>		
17	V-1 Verify that the exports in S-2 and S-3 are marked as having encountered errors in the GUI.		#comment
18	In the BMGT GUI, verify the first export of granule g1 is marked as blocked in the Export Request tab.		
19	In the BMGT GUI, verify the export of collection C1 is marked as blocked in the Export Request tab.		
20	V-2 Verify that an email is delivered to the configured notification email address listing the errors in S-2 and S-3. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		#comment
21	Verify an email is sent from BMGT to the configured address.		
22	Verify the email lists the failed granule export.		
23	Verify the email identifies the granule.		
24	Verify the email includes the error message returned from ECHO (or the mock ECHO).		
25	Verify the email indicates 1 such error occurred.		
26	Verify the email lists the failed collection export.		
27	Verify the email identifies the collection.		
28	Verify the email includes the error message returned from ECHO (or the mock ECHO).		
29	Verify the email indicates 1 such error occurred.		
30	V-3 Verify that the BMGT GUI also lists the text of the error for the export requests in S-2 and S-3		#comment
31	Verify the BMGT GUI Export Activity/Error tab displays the error message returned from ECHO (or the mock ECHO) for the failed granule export. Click on the Activity to show the error.		
32	Verify the BMGT GUI Export Activity/Error tab displays the error message returned from ECHO (or the mock ECHO) for the failed collection export.<br		

#	Action	Expected Result	Notes
	/> Click on the Activity to show the error.		
33	S-4 Resolve the issue which caused the failure in S-2. Using the GUI, request the re-export of the associated request.		#comment
34	Configure the mock ECHO to return a success response for granule g1. Run proxy_stop422, then run proxy_start to resume the regular proxy. -OR- Repair the error introoduced to the metadata.		
35	In the BMGT GUI Export Request tab, request re-export of granule g1 by selecting the request and clicking the 'Release' button.		
36	V-4 Verify that after requesting re-export in S-4, the associated request is successfully re-exported.		#comment
37	Verify the reexport of granule g1 succeeds (i.e. has a state of "WARNING" in the GUI Export Request tab).		
38	Verify that a PUT request containing full granule metadata for g1 is seen in the TCP proxy log.		
39	S-5 Using the GUI, request the cancellation of the request in S-3.		#comment
40	In the BMGT GUI Export Export Request tab, request cancellation of the export of collection C1 by selecting the request and clicking the 'Cancel' button.		
41	V-5 Verify that after requesting cancellation in S-5, the request is no longer listed as an active request.		#comment
42	Verify after cancelling collection C1, the BMGT GUI no longer lists the request as active.		
43	Verify that the BMGT GUI Export Request tab lists the request for C1 as CANCELED.		
44	V-6 Verify that the BMGT GUI provides statistics on the number of errors which encountered. Verify that these statistics also indicate how many of these items were retried and how many were cancelled.		#comment
45	In the Export Activity/Error tab, set the filter to only show the time period of this test. Filter via start time. Make sure to include the times of your requests.		
46	Verify the BMGT GUI Export Activity/Error tab indicates there were 2 errors not automatically handled (i.e. two items in the "RETRY" state).		
47	In the Export Request tab, set the filter to only show the time period of this test. Filter via enqueuetime. Make sure to include your requests.		
48	Verify the BMGT GUI indicates the granule export was retried (i.e. is in the "WARNING" state).		

#	Action	Expected Result	Notes
49	Verify the BMGT GUI indicates the collection export was cancelled (i.e. is in the "CANCELED" state).		
50	Clean up		#comment
51	If the collection or granule metadata were damaged, restore them.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
				1 collection (C1)				
				1 granule (g1) not in C1				

EXPECTED RESULTS:

	V	310	1	Verify that the exports in S-1 and S-2 are marked as failed in the GUI.		
	V	310	2	Verify that an email is delivered to the configured notification email address listing the errors in S-1 and S-2. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		
	V	310	3	Verify that the BMGT GUI also lists the text of the error for the export requests in S-1 and S-2		
	V	310	4	Verify that after requesting re-export in S-3, the associated request is successfully re-exported.		
	V	310	5	Verify that after requesting cancellation in S-4, the request is no longer listed as an active request.		
	V	310	6	Verify that the BMGT GUI provides statistics on the number of errors which were not automatically handled and required manual intervention. Verify that these statistics also indicate how many of these items were retried after manual investigation and how many were cancelled.		

1.58 Retriable HTTP Level Errors (ECS-ECSTC-57)

DESCRIPTION:

	S	320	1	**[Retriable HTTP Level Errors]** In the BMGT GUI set the number of retries which will cause an alert to be thrown to a known (and reasonably small) number.		
	S	320	2	In the BMGT GUI set the export retry interval to a known (and reasonably small) number.		
	S	320	3	Perform an export of granule and collection metadata which will fail export due to a retriable error (e.g. configured endpoint does not exist). Ensure that these errors are not of a type (e.g. code 500) which will be immediately retried by the BMGT exporter.		
	S	320	4	Allow the exports to retry up to the configured number of times.		
	S	320	5	Fix underlying error and resume the queue.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/320 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	Ensure Collections C1 has been installed in the mode. (ESDT verification script)		
9	Verify Collection C1 is enabled for Collection and Granule Export.		
10	Configure BMGT.EmailLogger.To = <testEmail>		
11	S-1 In the BMGT GUI set the number of retries which will cause an alert		#comment

#	Action	Expected Result	Notes
	to be thrown to a known (and reasonably small) number.		
12	BMGT.Exporter.IngestClient.RetryRequestCount= 3		
13	S-2 In the BMGT GUI set the export retry interval to a known (and reasonably small) number.		#comment
14	BMGT.Exporter.IngestClient.RetryRequestWait = 60000		
15	V-1 Verify that it is possible to configure the values listed in S-1 and S-2.		#comment
16	Verify the GUI configuration. Check the database bg_configuration_property to make sure the configuration has been updated.		
17	Verify the GUI configuration. Check the database bg_configuration_property to make sure the configuration has been updated.		
18	S-3 Perform an export of granule and collection metadata which will fail export due to an HTTP error which does not result in the return of an HTTP status code (e.g. configured echo host does not exist).		#comment
19	NOTE: Some examples of java exceptions which fall into this category are: UnknownHostException (configured endpoint does not exist), SSLException, SocketException. SocketTimeOutException, which is thrown if ECHO is reachable but is taking too long to process a request, does not fall into this category, even though it will result in an exception rather than an HTTP response.		#comment
20	Bring the Mock ECHO down.		
21	Assume Collection C1 has one science granule g1		
22	EcBmBMGTManualStart <MODE> --metg --collections <C1_shortname>.<C1_versionid>		
23	S-4 Allow the exports to retry up to the configured number of times.		#comment
24	V-2 Verify that the exports in S-3 are marked as pending in the GUI.		#comment
25	Verify the status of the export request of granule g1 is "PENDING".		
26	V-3 Using the GUI or logs, verify that the retries occurred at the frequency configured in S-2.		#comment
27	Verify that the logs shows a retry request every minute.		
28	V-4 Verify that the GUI or logs indicate that the export was retried the number of times configured in S-1.		#comment
29	Verify that the export was retried 3 times.		
30	V-5 Verify that after the export is retried the configured number of times, BMGT processing is paused and no more exports are processed.		#comment

#	Action	Expected Result	Notes
31	V-6 Verify that the BMGT GUI displays an alert which contains the text of the error		#comment
32	Verify an alert is displayed on the GUI after the retries were paused.		
33	Verify that GUI shows, along with the alert, the text of the error. The error is displayed on the activity status.		
34	Verify that the configuration property BMGT.Dispatcher.ErrorMsg has the text of the error which caused Dispatcher to pause. Note: The ErrorMsg may get cleared when dispatcher does autoresume.		
35	V-7 Verify that an email is sent to the operator when BMGT processing is paused.		#comment
36	Verify that an email is sent to the email address <testEmail> after the Dispatcher is paused.		
37	V-8 Verify that the email lists the text of the error which caused processing to be paused.		#comment
38	Verify that the email lists the cause of the error which caused Dispatcher to pause.		
39	S-5 Fix underlying error and resume BMGT processing		#comment
40	Bring the Mock ECHO server back up		
41	Resume Dispatcher.		
42	V-9 Verify that the GUI allows the resumption of processing.		#comment
43	Verify that the GUI shows the requests being processed.		
44	V-10 Verify that after fixing the underlying error, and resuming BMGT processing in S-5, the exports in S-3 are picked up and processed to completion.		#comment
45	Verify that the TCP proxy shows an HTTP PUT request for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 Collection with 1 science granule					

EXPECTED RESULTS:

	V	320	1	Verify that it is possible to configure the values listed in S-1 and S-2.		
	V	320	2	Verify that the exports in S-3 are marked as having retriable errors in the GUI.		
	V	320	3	Using the GUI or logs, verify that the retries occurred at the frequency configured in S-2.		
	V	320	4	Verify that the GUI indicates that the export was retried the number of times configured in S-1.		
	V	320	5	Verify that the BMGT GUI also lists the text of the error for the export		
	V	320	6	Verify that the BMGT GUI provides statistics on the number of retriable export errors.		
	V	320	7	Verify that after the export is retried the configured number of times, BMGT processing is paused (at least for the queue associated with the exports in S-3) and no more exports are processed from that queue.		
	V	320	8	Verify that an email is sent to the operator when the queue is paused and also that there is an alert shown in the GUI. Verify that the email lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		
	V	320	9	Verify that the GUI allows the resumption of processing.		
	V	320	10	Verify that after fixing the underlying error, and resuming the queue in S-5, the exports in S-3 are picked up and processed to completion.		

1.59 Contact ECHO Errors (ECS-ECSTC-58)

DESCRIPTION:

	S	330	1	**[Contact ECHO Errors]** Perform an export of granule and collection metadata which will fail export due to an error which is classified as requiring ECHO Notification. For instance, an internal server error from ECHO or an ECHO stand-in		
	S	330	2	Resolve the underlying issue and use the GUI to retry the export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/330 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	Ensure Collection C1 has been installed in the mode. (ESDT verification script)		
9	Verify Collection C1 is enabled for Collection and Granule Export.		
10	S-1 Perform an export of granule and collection metadata which will fail export due to an error which is classified as requiring ECHO Notification. For instance, an internal server error from ECHO or an ECHO stand-in		#comment
11	With the dispatcher running, run proxy_stop, then proxy_start500 to start a proxy that returns HTTP 500 errors.		
12	./EcBmBMGTManualStart <MODE> --metc --metg -c <C1_shortname>.<C1_versionid>		
13	V-1 Verify that the exports in S-1 are marked as failed in the GUI.		#comment
14	Verify the BMGT GUI > Export Requests tab shows the request from S-1 as PENDING.		
15	Verify in the BMT GUI, the Activity associated with the request from the previous step is marked as ERROR.		
16	Verify the BMGT GUI > System Status tab shows the Dispatcher is paused.		
17	V-2 Verify that an email is delivered to the configured notification email address listing the errors in S-1. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		#comment

#	Action	Expected Result	Notes
18	Verify that an email is sent to the configured email address		
19	V-3 Verify that the email indicates that the operator should contact ECHO Operations staff regarding the error.		#comment
20	Verify that the email indicates the ECHO operator needs to be contacted		
21	V-4 Verify that the BMGT GUI also lists the text of the error for the export requests in S-1.		#comment
22	Verify that the bmgt GUI lists the export request error for granule g1 and Collection C1		
23	V-5 Verify that the BMGT GUI provides statistics on the number of errors which required interaction with ECHO.		#comment
24	Verify that the BMGT GUI lists the number of requests resulting in errors requiring contacting ECHO		
25	S-2 Resolve the underlying issue and use the GUI to retry the export.		#comment
26	Run proxy_stop500 then proxy_start to resume the regular proxy.		
27	Re-Export the requests from the GUI		
28	V-6 Once the underlying issue has been resolved, and the export retried in S-2, verify that the export is able to complete.		#comment
29	Verify that the TCP proxy shows an HTTP PUT request for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		
30	Verify that the TCP proxy shows a HTTP PUT request for Collection C1. (There may be more than one HTTP request, e.g., if there are network issues.)		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 Collection with a science granule					

EXPECTED RESULTS:

	V	330	1	Verify that the exports in S-1 are marked as failed in the GUI.		

	V	330	2	Verify that an email is delivered to the configured notification email address listing the errors in S-1. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		
	V	330	3	Verify that the email indicates that the operator should contact ECHO Operations staff regarding the error.		
	V	330	4	Verify that the BMGT GUI also lists the text of the error for the export requests in S-1.		
	V	330	5	Verify that the BMGT GUI provides statistics on the number of errors which required interaction with ECHO.		
	V	330	6	Once the underlying issue has been resolved, and the export retried in S-2, verify that the export is able to complete.		

1.60 Data Related Errors (ECS-ECSTC-59)

DESCRIPTION:

	S	340	1	**[Data Related Errors]** Modify the XML metadata for a granule and the ODL descriptor for a collection such that they are invalid and their export attempt will cause a 'data related error'. For instance, remove the start or end of an XML or ODL group so that the file fails basic validation.		
	S	340	2	Perform an update to the selected granule and collection such that they are picked up by the automatic export polling process. Alternatively, add a fake event to the AIM event history to cause the granule and collection to be picked up.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		

#	Action	Expected Result	Notes
5	Verify collections under /sotestdata/DROP_802/BE_82_01/340 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Setup		#comment
8	Ensure Collections C1 has been installed in the mode.		
9	Verify Collection C1 is enabled for Collection and Granule Export.		
10	Configure BMGT.EmailLogger.To = <testEmail>		
11	S-1 Modify the XML metadata for a granule and the ODL descriptor for a collection such that they are invalid and their export attempt will cause a 'data related error'. For instance, remove the start or end of an XML or ODL group so that the file fails basic validation.		#comment
12	Find granule G1's metadata file in the small file archive, make a back up copy, then remove the final closing tag. /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
13	Find collection C1's ODL descriptor file in the small file archive, make a backup copy, then remove a START_GROUP. /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
14	S-2 Perform an update to the selected granule and collection such that they are picked up by the automatic export polling process. Alternatively, add a fake event to the AIM event history to cause the granule and collection to be picked up.		#comment
15	Update the DsMdGREventHistory with a GRUPDATE and a CLUPDATE for granule g1 and Collection C1		
16	Verification		#comment
17	V-1 Verify that the export in S-1 encounters errors.		#comment
18	V-2 Verify that the export requests for the items in S-1 are marked as 'failed' in the GUI and that the associated errors can be viewed for each item.		#comment
19	Verify that the GUI shows errors for the requests for granule g1 and Collection C1		
20	V-3 Verify that an email is sent indicating that the items in S-1 failed due to data related errors and were skipped.		#comment
21	Verify that an email was sent to the configured email address		
22	V-4 Verify that the email message lists the error messages and the number		#comment

#	Action	Expected Result	Notes
	of items which encountered each error.		
23	Verify that the email lists the granule g1 and Collection C1 with the error messages		
24	V-5 Verify that messages are printed to the log file for the errors in S-1 , including the item (collection or granule) ID, the reason for the error, and indicate that the items were skipped.		#comment
25	Verify that the BMGT logs indicate granule g1 and Collection C1 as items skipped along with the reason for error		
26	V-6 Verify that in the GUI it is possible to list only those items which were skipped due to data related errors.		#comment
27	Verify that the GUI export request page allows filtering to view only those requests which were skipped due to data related errors.		
28	V-7 Verify that no HTTP export request was made for the items in S-1, as their failure occurred prior to export and export was therefore 'skipped'		#comment
29	Verify that the TCP proxy does not show a HTTP PUT request for granule g1.		
30	Verify that the TCP proxy does not show a HTTP PUT request for Collection C1.		
31	V-8 Verify that the BMGT GUI provides statistics on the number of items skipped due to data related errors.		#comment
32	Verify the bmgt GUI provides stats on the errors.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	340	1	Verify that the export in S-1 encounters errors.		
	V	340	2	Verify that the export requests for the items in S-1 are marked as 'failed' in the GUI and that the associated errors can be viewed for each item.		

	V	340	3	Verify that an email is sent indicating that the items in S-1 failed due to data related errors and were skipped.		
	V	340	4	Verify that the email message lists the error messages and the number of items which encountered each error.		
	V	340	5	Verify that messages are printed to the log file for the errors in S-1 , including the item (collection or granule) ID, the reason for the error, and indicate that the items were skipped.		
	V	340	6	Verify that in the GUI it is possible to list only those items which were skipped due to data related errors.		
	V	340	7	Verify that no HTTP export request was made for the items in S-1, as their failure occurred prior to export and export was therefore 'skipped'		
	V	340	8	Verify that the BMGT GUI provides statistics on the number of items skipped due to data related errors.		

1.61 Non Data Related Generation Errors (ECS-ECSTC-60)

DESCRIPTION:

	S	370	1	**[Non Data Related Generation Errors]** In the GUI, configure the number of retries and retry delay for non data related metadata generation failures. Ensure that the configuration is such that it will be possible to perform the rest of this criteria without the number of retires reaching the limit.		
	S	370	2	Move the metadata file for a granule, such that it will not be found at the location indicated by its database record.		
	S	370	3	Move the descriptor file for a collection, such that it will not be found at the location indicated by its database record.		
	S	370	4	Request the export of the granule and collection in S-2 and S-3		
	S	370	5	Replace the files moved in S-2 and S-3.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/370 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT requests.		
7	Setup		#comment
8	Ensure collections C1, C2 have been installed in the mode.		
9	Verify Collection C1 and C2 have been enabled for Collection and Granule Export.		
10	Configure BMGT.EmailLogger.To = <testEmail> in the BMGT GUI configuration tab		
11	S-1 In the GUI, configure the number of retries and retry delay for non data related metadata generation failures. Ensure that the configuration is such that it will be possible to perform the rest of this criteria without the number of retires reaching the limit.		#comment
12	Configure BMGT.Generator.IOError.RetryInterval = 60000 Configure BMGT.Generator.IOError.NumRetries = 10		
13	V-1 Verify that it was possible to modify the number of retries and retry interval in S-1.		#comment
14	Verify that the configuration changes were saved in the database in bg_configuration_property table		
15	S-2 Move the metadata file for a granule, such that it will not be found at the location indicated by its database record.		#comment
16	Locate granule g1's small file archive XML file /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml move it to /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.old		
17	S-3 Move the descriptor file for a collection, such that it will not be found		#comment

#	Action	Expected Result	Notes
	at the location indicated by its database record.		
18	Locate collection C1's ODL descriptor file /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc move it to /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.old		
19	S-4 Request the export of the granule and collection in S-2 and S-3		#comment
20	Assume Collection C1 has one science granule g1		
21	EcBmBMGTManualStart <MODE> --metc --metg --collections <C1_shortname>.<C1_versionid>		
22	Record start time to be t_start_time		
23	V-2 Inspect the log to verify that the errors encountered due to the missing files in S-2 and S-3 are listed there, and that the log entries indicate that BMGT tried to find these files repeatedly at the configured interval until it reached the maximum retry threshold.		#comment
24	Verify that the bmgt logs indicate the errors due to missing files		
25	V-3 Verify that the GUI indicates that the export of the items in S-2 and S-3 have failed due to a non-data related error and lists the error text.		#comment
26	Verify that the GUI indicates the error for granule 1 and Collection C1. request status should say BLOCKED activity status should say RETRY or ERROR		
27	S-5 Replace the files moved in S-2 and S-3.		#comment
28	Move /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.old to /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
29	Wait for 5 minutes after t_start_time		
30	Move /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.old to /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
31	V-4 Verify that once the error occurs, metadata generation is retried at the configured interval until the file is replaced.		#comment

#	Action	Expected Result	Notes
32	Verify that the retry occurs every 1 minute till the the granule file g1 was replaced		
33	V-5 Verify that the BMGT GUI provides statistics on the number of items which failed due to non data related generation errors.		#comment
34	Verify BMGT GUI allows filtering of requests to view only those which failed due to non data-related errors and provides a count of such errors.		
35	V-6 Verify that once the normal artifact cleanup time has passed the record of the skipped collection from S-3 is cleaned up, and no longer shows up in the GUI.		#comment
36	Verify that the Collection C1 does not show up on the GUI as an error or pending item after the configred artifact cleanup time.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 Collection and 1 granule					

EXPECTED RESULTS:

	V	370	1	Verify that it was possible to modify the number of retries and retry interval in S-1.		
	V	370	2	Inspect the log to verify that the errors encountered due to the missing files in S-2 and S-3 are listed there, and that the log entries indicate that BMGT tried to find these files repeatedly at the configured interval until it reached the maximum retry threshold.		
	V	370	3	Verify that the GUI indicates that the export of the items in S-2 and S-3 have failed due to a non-data related error and lists the error text.		
	V	370	4	Verify that once the error occurs, metadata generation is retried at the configured interval until the file is replaced.		
	V	370	5	Verify that the BMGT GUI provides statistics on the number of items which failed due to non data related generation errors.		

	V	370	6	Verify that once the normal artifact cleanup time has passed the record of the skipped collection from S-3 is cleaned up, and no longer shows up in the GUI.		

1.62 Alert After Numerous Retries (ECS-ECSTC-61)

DESCRIPTION:

	S	380	1	**[Alert After Numerous Retries]** Log in to the BMGT GUI as an operator. View and update the export retry interval and number of retries which will trigger an alert. Save the changes. Bounce BMGT server.		
	S	380	2	Cause an error situation wherein a retriable, unsuccessful HTTP code is returned to BMGT from ECHO (or an ECHO stand-in) on each export attempt. Retriable error codes include HTTP code 500.		
	S	380	3	Resolve the cause of the error such that successful (code 200 or 201) HTTP statuses are returned.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available, connected to the ecs database.		
4	Ensure test data under /sotestdata/DROP_802/BE_82_01/Criteria/380 is available.		
5	Ensure Collections C1, C2 have been installed in the mode. Look in the DPL Ingest GUI, or ask a lab lead.		
6	Ensure granules g1 .. g4 have been ingested and their granule IDs are saved to a text file granuleids.txt.		
7	Ensure collections C1 and C2 are enabled for Collection and Granule Export: select shortname, versionid, collectionexportflag, granuleexportflag from bg_collection_configuration where (shortname = <C1_SHORTNAME> and versionid = <C1_VERSIONID>) or (shortname = <C2_SHORTNAME> and versionid = <C2_VERSIONID>)		
8	Ensure BMGT.EmailLogger.To is set to a valid, reachable email address		

#	Action	Expected Result	Notes
	(such as labuser@f4eil01.edn.ecs.nasa.gov) via the BMGT GUI > BMGT Configuration tab or the database: update bg_configuration_property set propertyvalue = '<TEST_EMAIL>' where propertyname = 'BMGT.EmailLogger.To' set the log4j.logger.emailLogger=DEBUG,notifier,notifierLog in the log4j.properties file.		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 Log in to the BMGT GUI as an operator. View and update the export retry interval and number of retries which will trigger an alert, as well as an interval for retries after an alert. Save the changes. Bounce BMGT server.		#comment
12	Log in to the BMGT GUI, using the operator password.		
13	On the "BMGT Configuration" tab, set these values: BMGT.Exporter.IngestClient.RetryRequestCount = 3 BMGT.Exporter.IngestClient.RetryRequestWait = 10000 BMGT.Dispatcher.MaxAutoResumeRetries = 3 BMGT.Dispatcher.AutoResumeWait = 60000		
14	Restart the BMGT Server: EcBmBMGTDispatcherStop <MODE> EcBmBMGTDispatcherStart <MODE>		
15	V-1 Verify that in S-1 the GUI displays the export retry interval and the number of retries which will trigger an alert. Verify that it allows the operator to update the values.		#comment
16	Verify the BMGT GUI > BMGT Configuration tab displays values for the changed properties before the change.		
17	Verify the BMGT GUI > BMGT Configuration tab allows the operator to change the propery values.		
18	Log out of the BMGT GUI.		
19	Clear the browser cache.		
20	Log in to the BMGT GUI with the operator password.		
21	Verify the BMGT GUI > BMGT Configuration tab displays new values for the changed properties.		
22	S-2 Cause an error situation wherein a retriable, unsuccessful HTTP error is returned to BMGT from ECHO (or an ECHO stand-in) on each export attempt. HTTP errors include: 1. Cannot determine IP address of ECHO 2. ECHO not responding 3. Communication with		#comment

#	Action	Expected Result	Notes
	ECHO interrupted in the middle (e.g. by bringing down a proxy) 4. ECHO terminates a persistent connection from ECHO.		
23	Cause one of the following error scenarios: 1. Cannot determine IP address of ECHO 2. ECHO not responding 3. Communication with ECHO interrupted in the middle (e.g. by bringing down a proxy) 4. ECHO terminates a persistent connection from ECHO.		
24	Request export of collection C1 and its granules g1, g2: EcBmBMGTManualStart <MODE> --metc --metg --collections <C1_SHORTNAME>.<C1_VERSIONID>		
25	V-2 Verify that the amount of time between retries of the error in S-2 is equal to that defined in S-1.		#comment
26	Verify the mock ECHO log shows that metadata for collection C1 and granules g1, g2 is exported every 10 seconds after the initial attempt.		
27	V-3 Verify that once the retry threshold is reached, an alert is triggered, pausing the dispatcher.		#comment
28	Verify that after the 4th collection C1 export (3rd retry) the dispatcher is paused (about 30 seconds after the initial export attempt).		
29	V-3.1 Verify that the alert is visible in the GUI.		#comment
30	Verify that after the 4th collection C1 export (3rd retry) an alert appears in the GUI (about 30 seconds after the initial export attempt).		
31	V-3.2 Verify that an alert email is sent to the operator explaining that the dispatcher has been paused and giving a reason for the alert.		#comment
32	Verify that after the 4th collection C1 export (3rd retry) an email is sent to the configured email address (about 30 seconds after the initial export attempt).		
33	V-3.3 Verify that while the alert is in place, the dispatcher is unpaused at the configured interval allowing requests to be retried		#comment
34	Verify that while the alert is displayed in the BMGT GUI, the dispatcher is unpaused every minute, resulting in export attempts for queued export requests.		
35	S-3 Resolve the cause of the error such that successful (code 200 or 201) HTTP statuses are returned.	bring up proxy 200 or 201, the Dispatcher server won't clear the alert.	#comment
36	Configure the mock ECHO to respond to collection and granule exports with HTTP success codes, such as 200 or 201.	bring up proxy 200 or 201, the Dispatcher server won't clear the alert. bring up normal proxy, the dispatcher will clear the alert.	
37	V-4 Verify that once the cause of the error is resolved in S-3, the alert is		#comment

#	Action	Expected Result	Notes
	automatically cleared the next time the dispatcher is resumed and requests are able to successfully export, allowing all queued requests to once again start working off.		
38	Verify that after the mock ECHO begins returning HTTP success codes the alert no longer appears in the BMGT GUI.		
39	Verify that the mock ECHO log shows that after collection C1 is exported successfully, metadata is exported for granules g1 and g2.		
40	V-5 Cause a 422 HTTP status code to be returned to BMGT for one or more export requests, and verify that these requests are marked as having an ECHO ingest failure.		#comment
41	Configure the mock ECHO to respond to collection and granule exports with HTTP code 422 an dECHO Error code "RECORD_INVALID". (If using a TCP proxy, Duplicate the online access URL in the granule metadata file for granule g2 in Collection C2. (Online access urls Urls must be unique will cause 422 error)		
42	EcBmBMGTManualStart --mode <MODE> --metc --metg --collections <C2_shortname.C2_versionid>		
43	Verify that the GUI shows a "BLOCKED/PENDING" on the export request		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		Collection C1 with 2 granules g1, g2						
		Collection C2 with 2 granules g3, g4						

EXPECTED RESULTS:

	V	380	1	Verify that in S-1 the GUI displays the export retry interval and the number of retries which will trigger an alert. Verify that it allows the operator to update the values.		

	V	380	2	Verify that the amount of time between retries of the error in S-2 is equal to that defined in S-1.		
	V	380	3	Verify that once the retry threshold is reached, an alert is triggered. Verify that the alert is visible in the GUI. And also in an email sent to the operator. Verify that while the alert is in place, only a single export request will attempt retries, and all others will be paused.		
	V	380	4	Verify that once the cause of the error is resolved in S-3, the alert is automatically cleared, allowing all queued requests to once again start working off.		
	V	380	5	Cause a 422 HTTP status code to be returned to BMGT for one or more export requests, and verify that these requests are marked as having an ECHO ingest failure.		

1.63 Fault Recovery (ECS-ECSTC-62)

DESCRIPTION:

	S	390	1	**[Fault Recovery]** Set up an ECHO stand-in which will cause HTTP requests to hang for a long period of time before responding.		
	S	390	2	Initiate a manual export and trigger events such that there is a backlog of pending export requests for both manual and automatic export.		
	S	390	3	Once the export requests have started working off, and once some of the requests have reached the point where HTTP requests to the ECHO stand-in are hanging, forcibly kill the BMGT application.		
	S	390	4	Cause the ECHO stand-in to no longer hang on all requests.		
	S	390	5	Start the BMGT application.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a mock ECHO is capturing BMGT traffic.		

#	Action	Expected Result	Notes
3	Ensure granules G1 ... G20 are in AIM.		
4	Setup		#comment
5	S-1 Set up an ECHO stand-in which will cause HTTP requests to hang for a long period of time before responding.		#comment
6	Configure the mock ECHO to wait for 5 minutes before responding to each request.		
7	S-2 Initiate a manual export and trigger events such that there is a backlog of pending export requests for both manual and automatic export.		#comment
8	Manually export granules G1 ... G10: EcBmBMGTManualStart $MODE --metg --granules ${G1},${G2},...,${G10}		
9	Update granules G11 ... G20: update amgranule set daynightflag = 'Both' where granuleid in (${G11}, ${G12}, ..., ${G20})		
10	S-3 Once the export requests have started working off, and once some of the requests have reached the point where HTTP requests to the ECHO stand-in are hanging, forcibly kill the BMGT application.		#comment
11	Once the mock ECHO logs several requests, kill BMGT: ps aux | grep EcBmDispatcher | grep $MODE kill -9 ${BMGT_DISPATCHER_PID}		
12	S-4 Cause the ECHO stand-in to no longer hang on all requests.		#comment
13	Configure the mock ECHO to respond immediately to each request.		
14	S-5 Start the BMGT application.		#comment
15	Start the BMGT dispatcher: EcBmBMGTDispatcherStart $MODE		
16	Verification		#comment
17	V-1 Verify that when BMGT is restarted, all export requests which were in process, including those which were hanging during an HTTP request, are added back to the queue and worked off.		#comment
18	Verify that after restarting BMGT requests for granules G1 ... G20 complete.		
19	V-2 Verify that no export requests are failed.		#comment
20	Verify the BMGT GUI shows export requests for granules G1 ... G20 are successful.		
21	V-3 Verify that each of the export requests added in S-2 is worked off, and that none are lost due to the failure.		#comment
22	Verify that all automatic requests for granules G11 ... G20 are eventually exported.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
				20 granules (G1 ... G20).				

EXPECTED RESULTS:

	V	390	1	Verify that when BMGT is restarted, all export requests which were in process, including those which were hanging during an HTTP request, are added back to the queue and worked off.		
	V	390	2	Verify that no export requests are failed.		
	V	390	3	Verify that each of the export requests added in S-2 is worked off, and that none are lost due to the failure.		

1.64 Corrective Export (ECS-ECSTC-63)

DESCRIPTION:

	S	400	1	**[Corrective Export]** Configure BMGT to not automatically perform re-exports.		
	S	400	2	Export a granule insert for 2 granules whose collections have not been exported to ECHO, or perform the export and artificially cause the response to indicate that the associated collection could not be found.		
	S	400	3	Manually delete 2 granules and 2 collections from ECHO such that SDPS contains these items and ECHO does not.		
	S	400	4	Manually add to ECHO 2 collections and 2 granules (which belong to a valid collection) such that ECHO contains these items and SDPS does not.		
	S	400	5	Perform a collection short form verification		

	S	400	6	Perform a granule short form verification (for at least the collections containing the granules added/removed in S-3 and S-4).		
	S	400	7	View the re-export queue in the GUI.		
	S	400	8	Filter the re-export queue to include only those requests associated with a specific collection.		
	S	400	9	Filter the re-export queue to include only those requests associated with a specific collection group.		
	S	400	10	From the command line, request a re-export queue report.		
	S	400	11	From the command line, request a re-export queue report, filtered to include only those requests associated with a particular collection.		
	S	400	12	From the command line, request a re-export queue report, filtered to include only those requests associated with a particular collection group.		
	S	400	13	From the command line, request the removal from the re-export queue of an export request by granule ID. Then request a new re-export queue report.		
	S	400	14	From the command line, request the removal from the re-export queue of an export request by collection ID. Then request a new re-export queue report.		
	S	400	15	From the command line, request a re-export queue statistics report.		
	S	400	16	Perform a corrective re-export, requesting at the same time, the production of a re-export queue report.		
	S	400	17	Request the generation of a re-export queue report.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties		

#	Action	Expected Result	Notes
	files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/400.		
5	Ensure Collections C1, C4, C5, C6, C7, C8 have been installed in the mode (e.g., they are listed in the DPL Ingest GUI).		
6	Ensure Collection C1, C4, C5, C6, C7, C8 are enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where (shortname = '<SHORTNAME>' and versionid = <VERSIONID>) or (shortname = '<SHORTNAME>' and versionid = <VERSIONID>) ... If any collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Manually delete collections C4, C8 from ECHO, using curl (see below for examples).		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Examples for querying ECHO and removing items:		#comment
10	1) create a token request file, with contents similar to: <token> <username>bmgt_tester</username> <password>**********</password> <client_id>ESI Testing</client_id> <user_ip_address>127.0.0.1</user_ip_address> <provider>DEV08</provider> </token> (with the correct password of course)		#comment
11	2) request a token: curl -XPOST -H "Content-Type:application/xml" -d @./tokenRequest-EDF_DEV08 "http://testbed.echo.nasa.gov/echo-rest/tokens" <?xml version="1.0" encoding="UTF-8"?> <token> <id>E80548D8-D80B-712E-6981-4D38C5D0A807</id> <username>bmgt_tester</username> <client_id>ESI Testing</client_id> <user_ip_address>127.0.0.1</user_ip_address> <provider>DEV08</provider> </token> where ./tokenRequest-EDF_DEV08 is the path to the token request file. The value of <id> is your token		#comment
12	2) request a token: curl -XPOST -H "Content-Type:application/xml" -d @./tokenRequest-EDF_DEV08		#comment

#	Action	Expected Result	Notes
	"http://testbed.echo.nasa.gov/echo-rest/tokens" <?xml version="1.0" encoding="UTF-8"?> <token> <id>E80548D8-D80B-712E-6981-4D38C5D0A807</id> <username>bmgt_tester</username> <client_id>ESI Testing</client_id> <user_ip_address>127.0.0.1</user_ip_address> <provider>DEV08</provider> </token> where ./tokenRequest-EDF_DEV08 is the path to the token request file. The value of <id> is your token		
13	3) list all collections installed in ECHO for the provider: curl -XGET -H "Echo-Token:E80548D8-D80B-712E-6981-4D38C5D0A807" http://testbed.echo.nasa.gov/catalog-rest/providers/DEV08/datasets <references> ... <reference> <name>AMSR-E/Aqua Monthly L3 Global Snow Water Equivalent EASE-Grids V086</name> <id>C1000000325-DEV08</id> <location>https://testbed.echo.nasa.gov:443/catalog-rest/echo_catalog/datasets/C1000000325-DEV08</location> </reference> </references> OR curl -XGET -H "Echo-Token:E80548D8-D80B-712E-6981-4D38C5D0A807" http://testbed.echo.nasa.gov/catalog-rest/providers/DEV08/datasets/verify <DatasetsShortForm> ... <DatasetRef> <DatasetId>Text file output of ASTER L1B expedited PGE. V002</DatasetId> <LastUpdate>2013-04-02T15:46:04Z</LastUpdate> </DatasetRef> </DatasetReferences> </DatasetsShortForm>		#comment
14	4) delete a granule curl -XDELETE -H "Echo-Token:E80548D8-D80B-712E-6981-4D38C5D0A807" http://testbed.echo.nasa.gov/catalog-rest/providers/DEV08/granules/<GRANULEUR> e.g. curl -XDELETE -H "Echo-Token:E80548D8-D80B-712E-6981-4D38C5D0A807" http://testbed.echo.nasa.gov/catalog-rest/providers/DEV08/granules/SC%3AMIL3YAL.005%3A88021 granule URS are encoded versions of 'SC:<SHORTNAME>.<VERSION>:<GRANULE_ID>'. encoded, this looks like: 'SC%3A<shortname>.<version w/ padding zeroes>%3A<granuleid>'. you can look in the		#comment

#	Action	Expected Result	Notes
	EcBmBMGTExporter.log for examples, and to find the URL used to export a granule (to use later for deletion)		
15	5) delete a collection curl -XDELETE -H "Echo-Token:E80548D8-D80B-712E-6981-4D38C5D0A807" http://testbed.echo.nasa.gov/catalog-rest/providers/DEV08/datasets/<DATASETID> e.g. curl -XDELETE -H "Echo-Token:E80548D8-D80B-712E-6981-4D38C5D0A807" http://testbed.echo.nasa.gov/catalog-rest/providers/DEV08/datasets/Near-Real-Time%20SSMIS%20EASE-Grid%20Daily%20Global%20Ice%20Concentration%20and%20Snow%20Extent%20V004 Dataset IDs are encoded versions of '<LONG NAME> V<VERSION>' you can look in the EcBmBMGTExporter.log for examples, and to find the URL used to export a dataset (to use later for deletion) here is a simple method to get the properly encoded version of the dataset ID: echo "AMSR-E/Aqua Monthly L3 Global Snow Water Equivalent EASE-Grids V086" | perl -e 'use URI::Escape; my $s = <>; chomp($s); print uri_escape($s);' \| sed -e 's/$/\n/' AMSR-E%2FAqua%20Monthly%20L3%20Global%20Snow%20Water%20Equivalent%20EASE-Grids%20V086 replace the contents between the initial quotes with the unescaped datasetID		#comment
16	Setup		#comment
17	S-1 Configure BMGT to not automatically perform re-exports.		#comment
18	Disable automatic corrective export: Log in to the BMGT GUI. On the BMGT Configuration tab set BMGT.ResponseHandler.BlockCorrectiveExports = 'true' set BMGT.Manual.ShortVer.ReqStatus = 'BLOCKED'		
19	S-2 Export a granule insert for 2 granules whose collections have not been exported to ECHO, or perform the export and artificially cause the response to indicate that the associated collection could not be found.		#comment
20	Ingest a C4 granule g1.		
21	Ingest a C8 granule g2.		
22	S-3 Manually delete 2 granules and 2 collections from ECHO such that SDPS contains these items and ECHO does not.		#comment
23	Delete 2 C1 granules g3, g4 from ECHO, using the curl commands listed above.		
24	Delete collections C6 and C7 from ECHO, using the curl commands listed above.		

#	Action	Expected Result	Notes
25	S-4 Manually add to ECHO 2 collections and 2 granules (which belong to a valid collection) such that ECHO contains these items and SDPS does not.		#comment
26	Perform a manual export to ingest collections C2 and C3 into ECHO: EcBmBMGTManualStart <MODE> -metc -c C2,C3		
27	Perform a manual export to ingest C5 granules g5 and g6 into ECHO: EcBmBMGTManualStart <MODE> -metg -g g5,g6		
28	On the BMGT GUI "System Status" tab pause the EVENT and NEW queues.		
29	Logically delete granules g5 and g6, using EcDsBulkDelete.pl.		
30	Delete the collections C2 and C3, using the ESDT Maintenance GUI.		
31	Remove any BMGT requests triggered by these steps delete from bg_export_request where status = 'PENDING' and collectionid in (C2, C3) or granuleId in (g5, g6)		
32	Resume the BMGT EVENT and NEW queues.		
33	S-5 Perform a collection short form verification		#comment
34	./EcBmBMGTManualStart <MODE> --short --metc		
35	S-6 Perform a granule short form verification (for at least the collections containing the granules added/removed in S-3 and S-4).		#comment
36	./EcBmBMGTManualStart <MODE> --short --metg --c C1		
37	V-1 Verify that after all the queued requests in S-2 – S-6 are exported, the following requests have been enqueued (or re-queued): a) S-2: 2 collection and 2 granule exports. b) S-3: 2 collection and 2 granule exports. c) S-4: 2 collection and 2 granule exports. Verify that all the created requests are in the ‘blocked’ state (and any re-queued requests are in the ‘pending’ state but are not picked up for processing).		#comment
38	Verify that collection export requests queued for collection C4, C8 in the BLOCKED state.		
39	Verify that granule export requests are re-queued for granules g1 and g2 in the pending state, but not processed pending the collection exports for C4 and C8.		
40	Verify that collection export requests are queued for collections C6 and C7 in the BLOCKED state.		
41	Verify that granule export requests are queued for granules g3 and g4 in the BLOCKED state.		
42	Verify that collection delete requests queued for collections C2 and C3 are in the BLOCKED state		

#	Action	Expected Result	Notes
43	Verify that granule deletion requests queued for granules g5 and g6 are in the BLOCKED state.		
44	S-7 View the re-export queue in the GUI.		#comment
45	On the BMGT GUI "Export Requests" tab filter the export requests to view only those requests in the BLOCKED state.		
46	S-8 Filter the re-export queue to include only those requests associated with a specific collection.		#comment
47	Further filter to show only those requests in the BLOCKED state that are associated with a specific collection.		
48	V-4 Verify that the re-export queue in S-8 contains only re-queued requests associated with the specified collection.		#comment
49	Verify that the request list shows only those associated with the specified collection.		
50	S-9 Filter the re-export queue to include only those requests associated with a specific collection group.		#comment
51	Filter the export requests to show only those requests in the BLOCKED state that are associated with a specific collection group.		
52	V-5 Verify that the re-export queue in S-9 contains only re-queued requests associated with the specified collection group.		#comment
53	Verify that the request list shows only those associated with the specified collection group.		
54	S-10 From the command line, request a re-export queue report.		#comment
55	EcBmBMGTManualStart <MODE> -corrective -report		
56	V-6 Verify that the re-export queue in S-10 contains all re-queued requests.		#comment
57	Verify that the re-export queue in S-10 contains all re-queued requests.		
58	S-11 From the command line, request a re-export queue report, filtered to include only those requests associated with a particular collection.		#comment
59	EcBmBMGTManualStart <MODE> -corrective -report -c <ShortName>.<VersionID>		
60	V-7 Verify that the re-export queue in S-11 contains only re-queued requests associated with the specified collection.		#comment
61	Verify that the re-export queue in S-11 contains only re-queued requests associated with the specified collection.		
62	S-12 From the command line, request a re-export queue report, filtered to include only those requests associated with a particular collection group.		#comment

#	Action	Expected Result	Notes
63	EcBmBMGTManualStart <MODE> -corrective -report -p <GrpName>		
64	V-8 Verify that the re-export queue in S-12 contains only re-queued requests associated with the specified collection group.		#comment
65	Verify that the re-export queue in S-12 contains only re-queued requests associated with the specified collection group.		
66	S-13 From the command line, request the removal from the re-export queue of an export request by granule ID. Then request a new re-export queue report.		#comment
67	EcBmBMGTManualStart <MODE> -corrective -cancel -g <GranuleId>		
68	EcBmBMGTManualStart <MODE> -corrective -report		
69	V-9 Verify that the re-export report generated after the removal of requests in S-13 does not include any requests associated with the specified granule.		#comment
70	Verify that the re-export report generated after the removal of requests in S-13 does not include any requests associated with the specified granule.		
71	S-14 From the command line, request the removal from the re-export queue of an export request by collection ID. Then request a new re-export queue report.		#comment
72	EcBmBMGTManualStart <MODE> -corrective -cancel -c <ShortName>.<VersionID>		
73	EcBmBMGTManualStart <MODE> -corrective -report		
74	V-10 Verify that the re-export report generated after the removal of requests in S-14 does not include any requests associated with the specified collection.		#comment
75	Verify that the re-export report generated after the removal of requests in S-14 does not include any requests associated with the specified collection.		
76	S-15 From the command line, request a re-export queue statistics report.		#comment
77	EcBmBMGTManualStart <MODE> -corrective -statistics		
78	V-11 Verify that the re-export queue statistics report in S-15 contains for each collection, the number of corrective requests, as well as the time of report generation.		#comment
79	Verify that the re-export queue statistics report in S-15 contains for each collection, the number of corrective requests, as well as the time of report generation.		

#	Action	Expected Result	Notes
80	S-16 Perform a corrective re-export, requesting at the same time, the production of a re-export queue report.		#comment
81	EcBmBMGTManualStart <MODE> -corrective -cr		
82	V-12 Verify that a re-export queue is printed to the screen when the corrective re-export is started in S-16.		#comment
83	Verify that a re-export queue is printed to the screen when the corrective re-export is started in S-16.		
84	V-13 Verify that when the re-export is started in S-16, all of the enqueued re-export requests are placed in the 'pending' state and worked off by BMGT.		#comment
85	Verify that all the requests in the BLOCKED state have been moved to the PENDING state.		
86	S-17 Request the generation of a re-export queue report.		#comment
87	EcBmBMGTManualStart <MODE> -corrective -report		
88	V-16 Verify that the re-export queue report generated in S-17 is empty.		#comment
89	Verify that no requests are printed to the screen.		
90	V-2 Verify that the re-export queue reports obtained on the command line in S-7 – S-12 list all of the newly enqueued corrective requests, and lists, for each item: a) Item Type (collection/granule) b) Item ID c) Collection d) Group e) Link to the initiating export request for the corrective request, where appropriate. f) Reason for corrective export.		#comment
91	Verify that when filtering the export requests list in the BMGT GUI to show only BLOCKED requests, the following is visible for each item in the list : a) Item Type (collection/granule) b) Item ID c) Collection d) Group e) BatchID of the initiating export request, if applicable f) For items being re-exported (not on the CORR queue), the Error or reason for corrective export (click the activity link button to view asscoiated errors).		
92	V-3 Verify that the re-export queue in S-7 contains all re-queued requests.		#comment
93	Filter the export request queue to display only items with status of "BLOCKED". Verify that the list contains the following as a result of each setup step: a) S-2: 2 collection and 2 granule exports. b) S-3: 2 collection and 2 granule exports. c) S-4: 2 collection and 2 granule exports.		

#	Action	Expected Result	Notes
94	V-14 Verify that the corrective requests result in the export of: a) HTTP PUT containing full granule or collection metadata for each granule or collection which is in the database and not logically deleted. b) HTTP DELETE for each granule or collection which is logically or physically deleted.		#comment
95	Verify that the TCP proxy shows HTTP PUT requests for granules g1, g2, g3, g4 and collections C4, C8, C6, C7.		
96	Verify that the TCP proxy shows HTTP DELETE requests granules g5, g6 and collections C2 and C3.		
97	V-15 Verify that all of the corrective requests succeed and reach a successful terminal state.		#comment
98	Verify that all of the corrective requests enqueued in this test are successfully exported and reach a terminal state.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
400		8 collections (C1..C8) with couple of granules each	MI3DAEF.002 (C1) MI3DALF.002 (C2) MI3DCDF.002 (C3) MI3DLSF.002 (C4) MIL3DAE.004 (C5) MIL3DAL.006 (C6) MIL3DCLD.002 (C7) MIL3DLS.004 (C8)				/sotestdata/DROP_802/BE_82_01/Criteria/400	

EXPECTED RESULTS:

	V	400	1	Verify that after all the queued requests in S-2 – S-5 are exported, the following requests have been enqueued: a) S-2: 2 collection and 2 granule exports. b) S-3: 2 collection and 2 granule exports. c) S-4: 2 collection and 2 granule exports. Verify that all the created requests are in the 'blocked' state.		
	V	400	2	Verify that the re-export queue reports obtained on the command line and in the GUI in S-7 – S-12 list all of the newly enqueued corrective requests, and lists, for each item: a)Item Type (collection/granule) b)Item ID c)Collection d)Group e)Identifier of the initiating export request which caused the enqueuing of the corrective request. f)Error or reason for corrective export.		
	V	400	3	Verify that the re-export queue in S-7 contains all re-queued requests.		
	V	400	4	Verify that the re-export queue in S-8 contains only re-queued requests associated with the specified collection.		
	V	400	5	Verify that the re-export queue in S-9 contains only re-queued requests associated with the specified collection group.		
	V	400	6	Verify that the re-export queue in S-10 contains all re-queued requests.		
	V	400	7	Verify that the re-export queue in S-11 contains only re-queued requests associated with the specified collection.		
	V	400	8	Verify that the re-export queue in S-12 contains only re-queued requests associated with the specified collection group.		
	V	400	9	Verify that the re-export report generated after the removal of requests in S-13 does not include any requests associated with the specified granule.		
	V	400	10	Verify that the re-export report generated after the removal of requests in S-14 does not include any requests		

				associated with the specified collection.		
	V	400	11	Verify that the re-export queue statistics report in S-15 contains for each collection, the number of corrective requests, as well as the time of report generation.		
	V	400	12	Verify that a re-export queue is printed to the screen when the corrective re-export is started in S-16.		
	V	400	13	Verify that when the re-export is started in S-16, all of the enqueued re-export requests are placed in the 'pending' state and worked off by BMGT.		
	V	400	14	Verify that the corrective requests result in the export of: a)HTTP PUT containing full granule or collection metadata for each granule or collection which is in the database and not logically deleted. b)HTTP DELETE for each granule or collection which is logically or physically deleted.		
	V	400	15	Verify that all of the corrective requests succeed and reach a successful terminal state.		
	V	400	16	Verify that the re-export queue report generated in S-17 is empty.		

1.65 Long Form Verification - Granule (ECS-ECSTC-64)

DESCRIPTION:

	S	410	1	**[Long Form Verification - Granule]** Request the long form verification of granule metadata for a single granule.		
	S	410	2	Request the long form verification of granule metadata for all granules in specified collection, ensuring that the collection includes some granules in the following categories: a) Granules with browse links b) Public granules c) Granules with restriction flag set		
	S	410	3	For at least two granules in S-1 and S-2, request the manual export of granule metadata, and save off the exported XML for use in verification		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
4	Verify database connections to ecs connected to ecs db successfully in the configured mode		
5	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
6	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
7	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
8	Setup		#comment
9	Ensure collections C1, C2 have been installed in the mode.		
10	Ensure collections C1, C2 are enabled for Collection and Granule Export.		
11	Ensure collections C1, C2 have been exported to ECHO.		
12	S-1 Request the long form verification of granule metadata for a single granule.		#comment
13	Identify a granule g1 in Collection C1 that can be exported.		
14	EcBmBMGTManualStart <MODE> --long --metg -g <g1_granuleid>		
15	V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata of the requested granule and containing an HTTP query parameter which indicates that the request is for verification purposes.		#comment
16	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP put request for granule g1.o (There may be more than one HTTP request, e.g., if there are network issues.)		
17	Verify that the TCP proxy log shows full granule metadata of the requested granule.		
18	Verify that the TCP proxy log shows that the request contains a query		

#	Action	Expected Result	Notes
	parameter xml_diff=true, indicating it is for verification purposes.		
19	S-2 Request the long form verification of granule metadata for all granules in specified collection, ensuring that the collection includes some granules in the following categories:		#comment
20	a) Granules with browse links		#comment
21	Identify granules g2, g3 in Collection C2 that are linked to browse granules br2, br3: select g.granuleid, bx.browseid from amgranule g join ambrowsegranulexref bx on g.granuleid = bx.granuleid where g.shortname = <C2_SHORTNAME> and g.versionid = <C2_VERSIONID> limit 2;		
22	b) Public granules		#comment
23	Identify granules g4, g5 in Collection C2 which are public science granules: select granuleid from amgranule where IsOrderOnly is null and shortname = <C2_SHORTNAME> and versionid = <C2_VERSIONID> limit 2;		
24	c) Granules with restriction flag set		#comment
25	Identify granules g6, g7 in Collection C2 which can be used to set the restriction flag. dsmdrestrictionflag contains valid values for restriction_flag		
26	insert into dsmdrestrictionflag values (5, 'Testing Restriction Flag'); insert into dsmdrestrictionflag values (6, 'Testing Restriction Flag - 2'); insert into dsmdgranulerestriction values (<g6>, 5); insert into dsmdgranulerestriction values (<g7>, 6);		
27	S-2 Request the long form verification of granule metadata for all granules in specified collection, ensuring that the collection includes some granules in the following:		#comment
28	EcBmBMGTManualStart <MODE> --long --metg -c <C2_shortname>.<C2_versionid>		
29	V-2 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full granule metadata for each granule in the specified collection and containing an HTTP query parameter which indicates that the request is for verification purposes.		#comment
30	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP put request for each granule g2,g3,g4,g5,g6,g7. (There may be more than one HTTP request, e.g., if there are network issues.)		
31	Verify that the TCP proxy log shows full granule metadata for each of		

#	Action	Expected Result	Notes
	granules g2,g3,g4,g5,g6,g7.		
32	Verify that the TCP proxy shows that each of the requests contain a query parameter xml_diff=true, indicating it is for verification purposes.		
33	V-3 Verify that the verification metadata for the granules in S-2 contains the expected metadata including: a) URLs for Browse links b) Science, Metadata, and ancillary file (if appropriate) URLs for public granules. c) Restriction flag for restricted granules.		#comment
34	Verify that the TCP proxy shows that the metadata exported for the browse granule contains a browse linkage URL for granules g2 and g3 in Collection C2. xpath "/Granule/OnlineResources/OnlineResource[Type='BROWSE']/URL/text()" granule.xml should contain browse linkage URL		
35	Verify that the TCP proxy shows that the metadata contains the URLs for the public granules g4 and g5 in Collection C2. xpath "/Granule/OnlineAccessURLs/OnlineAccessURL/URL/text()" granule.xml should contain the science granule URL.		
36	xpath /Granule/RestrictionFlag g6.xml should = <restriction_flag_value>		
37	S-3 For at least two granules in S-1 and S-2, request the manual export of granule metadata, and save off the exported XML for use in verification		#comment
38	Manually Export granules in Collection C1 and C2 ./EcBmBMGTManualStart <MODE> --metg -g <g1_granuleid> ./EcBmBMGTManualStart <MODE> --metg -g <g2_granuleid>,<g3_granuleid>,<g4_granuleid>,<g5_granuleid>,<g6_granuleid>,<g7_granuleid>		
39	Save the exported XML to a file.		
40	V-4 For the granules whose manual exported metadata was obtained in S-3, verify that the verification metadata is identical to this manual exported metadata.		#comment
41	Verify the metadata files from S2 and S3 are identical for each granule exported.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
410		1 collection	1 science granule					
410		1 collection	4 science, 2 browse					

EXPECTED RESULTS:

	V	410	1	Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata of the requested granule and containing an HTTP query parameter which indicates that the request is for verification purposes.		
	V	410	2	Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full granule metadata for each granule in the specified collection and containing an HTTP query parameter which indicates that the request is for verification purposes.		
	V	410	3	Verify that the verification metadata for the granules in S-2 contains the expected metadata including: a) URLs for Browse links b) Science, Metadata, and ancillary file (if appropriate) URLs for public granules. c) Restriction flag for restricted granules.		
	V	410	4	For the granules whose manual exported metadata was obtained in S-3, verify that the verification metadata is identical to this manual exported metadata[TR1][TG2] .		

1.66 Long Form Verification - Collection (ECS-ECSTC-65)

DESCRIPTION:

	S	420	1	**[Long Form Verification – Collection]** Request the verification of metadata for a specified collection.		
	S	420	2	For at least two collections, request the manual export of collection metadata, and save off the exported XML for use in verification		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		#comment
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		#comment
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		#comment
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure Collection C1 has been installed in the mode.		
9	Verify Collections C1 is enabled for Collection and Granule Export.		
10	S-1 Request the verification of metadata for a specified collection.		#comment
11	EcBmBMGTManualStart <MODE> --long --metc -c <C1_SHORTNAME>.<C1_VERSIONID>		
12	V-1 Verify that the operation in S-1 results in the export of a single HTTP PUT containing the full metadata of the requested collection and containing an HTTP query parameter which indicates that the request is for verification purposes.		#comment
13	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP put request for Collection C1. (There may be more than one HTTP request, e.g., if there are network issues.)		
14	Verify that the TCP proxy log shows full collection metadata of the requested collection.		
15	Verify that the TCP proxy shows that the request contains a query parameter xml_diff=true, indicating it is for verification purposes.		
16	S-2 For at least two collections, request the manual export of collection metadata, and save off the exported XML for use in verification		#comment
17	EcBmBMGTManualStart <MODE> --metc -c <C1_SHORTNAME>.<C1_VERSIONID>		
18	Save the exported metadata to an XML file.		
19	V-2 For the collection whose manual exported metadata was obtained in		#comment

#	Action	Expected Result	Notes
	S-2, verify that the verification metadata is identical to this manual exported metadata.</i>		
20	Verify the metadata files from S1 and S2 are identical.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 Collection					

EXPECTED RESULTS:

	V	420	1	Verify that the operation in S-1 results in the export of a single HTTP PUT containing the full metadata of the requested collection and containing an HTTP query parameter which indicates that the request is for verification purposes.		
	V	420	2	For the collection whose manual exported metadata was obtained in S-2, verify that the verification metadata is identical to this manual exported metadata .		

1.67 Long Form Verification - Insert Time (ECS-ECSTC-66)

DESCRIPTION:

	S	430	1	**[Long Form Verification - Insert Time]** Identify two granules within a collection and their insert times. Ensure that there are other granules inserted in between those times within the same collection. Request the verification of granule metadata for all granules in the collection, specifying the colelcion, the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		
	S	430	2	Identify two collections and their insert times. Ensure that there are other collections inserted between those times. Request the verification of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		#comment
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		#comment
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		#comment
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure collections C1, C2, C3 has been installed in the mode.		
9	Ensure collections C1, C2, C3 are enabled for collection and granule export.		
10	Ensure the Collections C1, C2, C3 have a few ingested granules.		
11	S-1 Identify two granules within a collection and their insert times. Ensure that there are other granules inserted in between those times within the same collection. Request the verification of granule metadata for all granules in the collection, specifying the collection, the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		#comment
12	select granuleid, archivetime from AmGranule where shortname = <C1_SHORTNAME> and versionid = <C1_VERSIONID> order by archivetime limit 4; Assuming there are 4 granules returned g1,g2,g3,g4		
13	Let g1 be the first granule in the list. Let g4 be the last granule in the list.		
14	g1_insert_time = archivetime of g1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. g4_insert_time = archivetime of g4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
15	EcBmBMGTManualStart <MODE> --long --starttime <g1_insert_time> --endtime <g4_insert_time> --metg --collections <C1_SHORTNAME>.<C1_VERSIONID>		
16	V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata for each granule in the specified collection which was inserted during the specified time range (with		#comment

#	Action	Expected Result	Notes
	the exception noted in V-2). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		
17	V-2 Verify that the granule which was inserted at the start time of the specified range is exported, but that the granule inserted at the end of the range is not.		#comment
18	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request each for granules g1, g2, g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
19	Verify that the TCP proxy log shows that the export request contains the full granule metadata for each of the granules g1, g2, g3.		
20	Verify that the TCP proxy shows that the export request contains a HTTP query parameter xml_diff=true for each of the granules g1, g2, g3, indicating that the request is for verification purposes.		
21	S-2 Identify two collections and their insert times. Ensure that there are other collections inserted between those times. Request the verification of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified insert times and specifying that the range shall apply to insert times.		#comment
22	select collectionid, shortname, versionid, inserttime from AmCollection order by inserttime limit 4;		
23	Let C1 be the first Collection in the list. Let C4 be the last collection in the list		
24	C1_insert_time = inserttime of C1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. C4 = inserttime of C4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
25	EcBmBMGTManualStart <MODE> --long --starttime <C1_insert_time> --endtime <C4_insert_time> --metc		
26	V-3 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was inserted during the specified time range (with the exception noted in V-4). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		#comment
27	V-4 Verify that the collection which was inserted at the start time of the specified range is exported, but that the collection inserted at the end of the range is not.		#comment

#	Action	Expected Result	Notes
28	Verify that the TCP proxy log shows an HTTP PUT request each for Collection C1, C2, C3. (There may be more than one HTTP request, e.g., if there are network issues.)		
29	Verify that the TCP proxy log shows that the export request contains the full collection metadata for each of the Collections C1, C2, C3.		
30	Verify that the TCP proxy shows that the export request contains a HTTP query parameter xml_diff =true for each of the Collections C1, C2, C3, indicating that the request is for verification purposes.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			4 collections with granules					

EXPECTED RESULTS:

	V	430	1	Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata for each granule in the specified collection which was inserted during the specified time range (with the exception noted in V-2). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		
	V	430	2	Verify that the granule which was inserted at the start time of the specified range is exported, but that the granule inserted at the end of the range is not.		
	V	430	3	Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was inserted during the specified time range (with the exception noted in V-4). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		
	V	430	4	Verify that the collection which was inserted at the start time of the specified range is exported, but that the collection inserted at the end of the range is not.		

1.68 Long Form Verification - Last Update Time (ECS-ECSTC-67)

DESCRIPTION:

	S	440	1	**[Long Form Verification - Last Update Time]** Identify two granules within a collection and their last update times. Ensure that there are other granules updated in between those times within the same collection. Request the verification of granule metadata for all granules in the collection, specifying the datetime range defined by the two identified update times and specifying that the range shall apply to last update times.		
	S	440	2	Identify two collections and their last update times. Ensure that there are other collections updated between those times. Request the verification of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified update times and specifying that the range shall apply to last update times.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/10 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure collections C1, C2, C3, C4 has been installed in the mode.		
9	Ensure collections C1, C2, C3, C4 are enabled for collection and granule export.		
10	Ensure the Collections C1, C2, C3, C4 have a few ingested granules.		
11	S-1 Identify two granules within a collection and their last update times. Ensure that there are other granules updated in between those times within the same collection. Request the verification of granule metadata for all granules in the collection, specifying the datetime range defined by the two identified update times and specifying that the range shall		#comment

#	Action	Expected Result	Notes
	apply to last update times.		
12	select granuleid, lastupdate from AmGranule where shortname = <C1_SHORTNAME> and versionid = <C1_VERSIONID> order by lastupdate limit 4; Assuming there are 4 granules returned g1,g2,g3,g4.		
13	Let g1 be the first granule in the list. Let g4 be the last granule in the list		
14	g1_insert_time = lastupdate of g1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. g4_insert_time = lastupdate of g4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
15	EcBmBMGTManualStart <MODE> --long --starttime <g1_insert_time> --endtime <g4_insert_time> --lastupdate --metg --collections <C1_SHORTNAME>.<C1_VERSIONID>		
16	V-1 Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata for each granule in the specified collection which was updated during the specified time range (with the exception noted in V-2). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes (xml_diff=true).		#comment
17	V-2 Verify that the granule which was updated at the start time of the specified range is exported, but that the granule updated at the end of the range is not.		#comment
18	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request each for granules g1, g2, g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
19	Verify that the TCP proxy log shows that the export request contains the full granule metadata for each of the granules g1, g2, g3.		
20	Verify that the TCP proxy shows that the export request contains a HTTP query parameter xml_diff=true for each of the granules g1, g2, g3, indicating that the request is for verification purposes.		
21	S-2 Identify two collections and their last update times. Ensure that there are other collections updated between those times. Request the verification of collection metadata, without specifying a collection, but specifying the datetime range defined by the two identified update times and specifying that the range shall apply to last update times.		#comment
22	select collectionid, shortname, versionid, lastupdate from AmCollection order by lastupdate limit 4;		

#	Action	Expected Result	Notes
23	Let C1 be the first Collection in the list. Let C4 be the last collection in the list.		
24	C1_lastupdate_time = lastupdatetime of C1 in "YYYY-MM-DD HH:MM:SS" [quotes are required]. C4_lastupdate_time = lastupdatetime of C4 in "YYYY-MM-DD HH:MM:SS" [quotes are required].		
25	EcBmBMGTManualStart <MODE> --long --starttime <C1_lastupdate_time> --endtime <C4_lastupdate_time> --metc		
26	V-3 Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was updated during the specified time range (with the exception noted in V-4). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		#comment
27	V-4 Verify that the collection which was updated at the start time of the specified range is exported, but that the collection updated at the end of the range is not.		#comment
28	Verify that the TCP proxy log shows an HTTP PUT request each for Collection C1, C2, C3. (There may be more than one HTTP request, e.g., if there are network issues.)		
29	Verify that the TCP proxy log shows that the export request contains the full collection metadata for each of the Collections C1, C2, C3.		
30	Verify that the TCP proxy log shows that the export request contains a HTTP query parameter (xml_diff=true) for each of the Collections C1, C2, C3, indicating that the request is for verification purposes.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			4 collections with granules					

EXPECTED RESULTS:

	V	440	1	Verify that the export operation in S-1 results in the export of a single HTTP PUT containing the full granule metadata		

				for each granule in the specified collection which was updated during the specified time range (with the exception noted in V-2). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		
	V	440	2	Verify that the granule which was updated at the start time of the specified range is exported, but that the granule updated at the end of the range is not.		
	V	440	3	Verify that the export operation in S-2 results in the export of a single HTTP PUT containing the full collection metadata for each collection which was updated during the specified time range (with the exception noted in V-4). Verify that the requests contain an HTTP query parameter which indicates that the request is for verification purposes.		
	V	440	4	Verify that the collection which was updated at the start time of the specified range is exported, but that the collection updated at the end of the range is not.		

1.69 Long Form Verification - Granule Updates (ECS-ECSTC-68)

DESCRIPTION:

	S	450	1	**[Long Form Verification - Granule Updates]** Record the start time of the criteria. Identify the following public collections for use in this test: a. AMSR collection configured for QA and PH. b. Collection configured for HDF MAP.		
	S	450	2	Publish a hidden QA granule linked to a science granule in one of the selected collections.		
	S	450	3	Publish a hidden PH granule linked to a science granule in one of the selected collections.		
	S	450	4	Regenerate the HDF_MAP for a science granule in one of the selected collections.		
	S	450	5	Using the command line utility or GUI, request the verification of granule metadata for the chosen collection, specifying a start time equal to the beginning time of the test (leave end time blank), and specifying selection by update time.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
4	Verify database connections to ecs connected to ecs db successfully in the configured mode		
5	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
6	Verify test data exists under /sotestdata/DROP_802/BE_82_01/Criteria/450.		
7	Ensure AMSR collection C1 is installed.		
8	Ensure collection C1 is configured to be public on ingest.		
9	Ensure collection C2 is installed.		
10	Ensure collection C2 is configured to be public on ingest.		
11	Ensure collectio C2 is eligible for HDF MAP generation.		
12	Ensure collections C1, C2 are enabled for collection and granule export.		
13	Ensure granules g1, g2, belonging to AMSR collection C1, have been ingested.		
14	Ensure QA granule qa_1 is linked to granule g1.		
15	Ensure QA granule qa_1 is in the hidden data pool (unpublish if needed).		
16	Ensure PH granule ph_2 is linked to granule g2.		
17	Ensure PH granule ph_2 is in the hidden data pool (unpublish if needed).		
18	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
19	Setup		#comment
20	S-1 Record the start time of the criteria. Identify the following public collections for use in this test: a) AMSR collection configured for QA and PH. b) Collection configured for HDF MAP.		#comment
21	Record start time as t_start_time.		
22	S-2 Publish a hidden QA granule linked to a science granule in one of the selected collections.		#comment
23	Publish the hidden QA granule qa_1 EcDlPublishStart <MODE> -ecs -g <qa_1_granuleid>		
24	S-3 Publish a hidden PH granule linked to a science granule in one of the selected collections.		#comment

#	Action	Expected Result	Notes
25	Publish the hidden PH granule ph_2 EcDlPublishStart <MODE> -ecs -g <ph_1_granuleid>		
26	S-4 Regenerate the HDF_MAP for a science granule in one of the selected collections.		#comment
27	Generate HDF Map granule hdf_3 manually: EcAmInsertMapGenerationRequest.pl -mode <MODE> -g <g3>		
28	Ensure that the map generation is complete. Verify that the amhdfmapxref table has the new hdf map output granule hdf_3 linked to the science granule g3.		
29	S-5 Using the command line utility or GUI, request the verification of granule metadata for the chosen collection, specifying a start time equal to the beginning time of the test (leave end time blank), and specifying selection by update time.		#comment
30	EcBmBMGTManualStart <MODE> --long --starttime <t_start_time> --lastupdate --metg --collections <C1_SHORTNAME>.<C1_VERSIONID>,<C2_SHORTNAME>.<C2_VERSIONID>		
31	V-1 Verify that the granules used for S-2 - S-4 are exported by BMGT.		#comment
32	V-2 Verify that the exports of the granules in S-2 - S-4 contain the full granule metadata, including, as applicable, the QA, PH, and HDF map URLS (reflecting the correct URLs after the update).		#comment
33	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for granule g1. (There may be more than one HTTP request, e.g., if there are network issues.)		
34	Verify that the TCP proxy log shows that the HTTP PUT request contains the full granule metadata for g1.		
35	Verify that the TCP proxy log shows that the metadata exported for granule g1 contains the URL for QA granule qa_1. xpath /Granule/OnlineResources g1.xml should contain an OnlineResource element of type "Quality Assurance" and the URL of qa_1.		
36	Verify that the TCP proxy log shows that the export request contains an HTTP query parameter indicating that the request is for verification purposes (xml_diff=true).		
37	Verify that the TCP proxy log shows an HTTP PUT request for granule		

#	Action	Expected Result	Notes
	g2. (There may be more than one HTTP request, e.g., if there are network issues.)		
38	Verify that the TCP proxy log shows that the HTTP PUT request contains the full granule metadata for g2.		
39	Verify that the TCP proxy log shows that the metadata exported for granule g2 contains the URL for PH granule ph_2. xpath /Granule/OnlineResources g2.xml should contain an OnlineResource element of type "Production History" and the URL of ph2.		
40	Verify that the TCP proxy shows that the export request contains a HTTP query parameter indicating that the request is for verification purposes (xml_diff=true).		
41	Verify that the TCP proxy log shows an HTTP PUT request for granule g3. (There may be more than one HTTP request, e.g., if there are network issues.)		
42	Verify that the TCP proxy log shows that the HTTP PUT request contains the full granule metadata for g3.		
43	Verify that the TCP proxy log shows that the metadata exported for granule g3 contains the URL for HDF Map granule hdf_3. xpath /Granule/OnlineResources g3.xml should contain an OnlineResource element of type "HDF Map" and the URL of hdf_3.		
44	Verify that the TCP proxy shows that the export request contains a HTTP query parameter indicating that the request is for verification purposes (xml_diff=true).		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		Collection C1	1 QA with 1 science granule 1 PH with 1 Science granule					
		Collection C2	HDF Map generation capable granules					

EXPECTED RESULTS:

	V	450	1	Verify that the granules used for S-2 - S-4 are exported by BMGT.		
	V	450	2	Verify that the exports of the granules in S-2 - S-4 contain the full granule metadata, including, as applicable, the QA, PH, and HDF map URLS (reflecting the correct URLs after the update).		

1.70 Long Form Verification - Multiple Options/Select by Group (ECS-ECSTC-69)

DESCRIPTION:

	S	460	1	**[Long Form Verification - Multiple Options/Select by Group]** Request the verification of collection and granule metadata, specifying a Datapool Group and a time range and indicating that the range shall apply to update time.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/460 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure Collections C1, C2, C3, C4 have been installed in the mode in Collection Group G1		
9	Ensure Collections C1, C2, C3, C4 are enabled for Collection and Granule Export		

#	Action	Expected Result	Notes
10	S-1 Request the verification of collection and granule metadata, specifying a Datapool Group and a time range and indicating that the range shall apply to update time.		#comment
11	Ingest granules g1, g2 into Collection C1		
12	Record the start time t_start_time		
13	Ingest granules g3, g4 into Collection C2		
14	Ingest granules g5, g6 into Collection C3		
15	Logically Delete granule g5 ./EcDsBulkDelete.pl -physical -user <dbuser> -password <passwd> -geoidfile /path/to/geoids_g5.txt		
16	Record the end time t_end_time		
17	Ingest granules g7, g8 into Collection C4		
18	EcBmBMGTManualStart <MODE> --long --metc --metg --group <G1> --start_time <t_start_time> --end_time <t_end_time> --lastupdate		
19	V-1 Verify that the operation in S-1 results in the export of an HTTP PUT for each collection, containing its metadata - and an HTTP PUT for each granule in each collection - which belongs to the specified datapool collection group, was updated within the specified range, and is not logically deleted.		#comment
20	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for collection C4. (There may be more than one HTTP request, e.g., if there are network issues.)		
21	Verify that the TCP proxy log shows that the request contains the full collection metadata for collection C4.		
22	Verify that the TCP proxy log shows a HTTP PUT request for granules g3, g4 and g6		
23	Verify that the TCP proxy log shows that each request has complete granule metadata for granules g3, g4, g6.		
24	Verify that the TCP proxy shows that all the requests contain a header indicating verification (xml_diff=true).		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		4 collections in the same group with at least 2 granules each						

EXPECTED RESULTS:

	V	460	1	Verify that the operation in S-1 results in the export of an HTTP PUT for each collection, containing its metadata - and an HTTP PUT for each granule in each collection - which belongs to the specified datapool collection group, was updated within the specified range, and is not logically deleted.		

1.71 Long Form Verification - Size Limits (ECS-ECSTC-70)

DESCRIPTION:

	S	470	1	**[Long Form Verification - Size Limits]** In the BMGT GUI, configure the maximum number of granules for long verification operations.		
	S	470	2	Form the command line, initiate a long form granule verification which will result in the export of more granules than the configured maximum.		
	S	470	3	When prompted as to whether to continue, choose to exit.		
	S	470	4	From the GUI, initiate a long form granule verification which will result in the export of more granules than the configured maximum.		
	S	470	5	This time, when prompted, choose to continue.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/470 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure Collection C1, C2 have been installed in the mode. (ESDT verification script)		
9	Verify Collection C1, C2 are enabled for Collection and Granule Export.		
10	Ingest 30 granules into Collections C1 and C2.		
11	S-1 In the BMGT GUI, configure the maximum number of granules for long verification operations.		#comment
12	Configure the max granules for long verification in the GUI to 20. BMGT.Verification.MaxGranules = 20		
13	V-1 Verify that in S-1 it is possible to configure the maximum number for granules in a long verification operation.		#comment
14	Verify that the configuration parameter was changed successfully in the database: select propertyname, propertyvalue from bg_configuration_properties where propertyname = 'BMGT.Verification.MaxGranules' Verify propertyvalue = 20.		
15	S-2 Form the command line, initiate a long form granule verification which will result in the export of more granules than the configured maximum.		#comment
16	EcBmBMGTManualStart <MODE> --long --metg -c <C1_shortname>.<C1_versionid>		
17	V-2 Verify that the export in S-2 results in a prompt indicating that the export will exceed the configured maximum verification export size, and asking whether to continue anyway.		#comment
18	S-3 When prompted as to whether to continue, choose to exit.		#comment

#	Action	Expected Result	Notes
19	Exit at the prompt to exit to continue		
20	V-3 Verify that when selecting not to continue, the verification attempt is terminated, the reason listed in the log, and no requests are added to the export queue.		#comment
21	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) does not show any HTTP requests.		
22	Verify that under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log the verification attempt termination is logged with the reason for termination		
23	S-4 Initiate a long form granule verification which will result in the export of more granules than the configured maximum.		#comment
24	EcBmBMGTManualStart <MODE> --long --metg -c <C2_shortname>.<C2_versionid>		
25	V-4 Verify that the export in S-4 results in a prompt indicating that the export will exceed the configured maximum verification export size, and asking whether to continue anyway.		#comment
26	S-5 This time, when prompted, choose to continue.		#comment
27	Continue at the prompt to exit or continue		
28	V-5 Verify that when selecting to continue, the verification attempt continues, adding requests to the export queue, and exporting the associated metadata.		#comment
29	Verify that the TCP proxy log shows an HTTP put request for each of the 30 granules in Collection C2. (There may be more than one HTTP request per granule, e.g., if there are network issues.)		
30	Verify that the TCP proxy log shows full granule metadata of the 30 granules in Collection C2.		
31	Verify that the TCP proxy log shows that the requests contains the query parameter xml_diff=true.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		2 collections with 30 science						

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		granules each						

EXPECTED RESULTS:

	V	470	1	Verify that in S-1 it is possible to configure the maximum number for granules in a long verification operation.		
	V	470	2	Verify that the export in S-2 results in a prompt indicating that the export will exceed the configured maximum verification export size, and asking whether to continue anyway.		
	V	470	3	Verify that when selecting not to continue, the verification attempt is terminated, the reason listed in the log, and no requests are added to the export queue.		
	V	470	4	Verify that the export in S-4 results in a prompt indicating that the export will exceed the configured maximum verification export size, and asking whether to continue anyway.		
	V	470	5	Verify that when selecting to continue, the verification attempt continues, adding requests to the export queue, and exporting the associated metadata.		

1.72 Long Form Verification - Concurrent Operation (ECS-ECSTC-71)

DESCRIPTION:

	S	480	1	**[Long Form Verification - Concurrent Operation]** Initiate a manual export which will take a considerable amount of time to enqueue the associated requests (i.e. a large number of granules). Immediately afterwards initiate a long form verification operation from the command line.		
	S	480	2	When prompted as to whether to continue, choose to exit.		
	S	480	3	Repeat S-1, but initiating the operations from the GUI.		
	S	480	4	When prompted as to whether to continue, choose to continue.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/480 with ECS metadata		
6	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
7	Setup		#comment
8	Ensure collection C1, C2, C3, C4 have been installed in the mode.		
9	Verify collection C1, C2, C3, C4 are enabled for collection and granule export.		
10	Ingest 100 granules into Collections C1, C2, C3 and C4		
11	S-1 Initiate a manual export which will take a considerable amount of time to enqueue the associated requests (i.e. a large number of granules). Immediately afterwards initiate a long form verification operation from the command line.		#comment
12	EcBmBMGTManualStart <MODE> --metg -c <C1_shortname>.<C1_versionid>		
13	EcBmBMGTManualStart <MODE> --long --metg -c <C2_shortname>.<C2_versionid>		
14	V-1 Verify that the export in S-1 results in a prompt indicating that there is already an export running.		#comment
15	S-2 When prompted as to whether to continue, choose to exit.		#comment
16	Exit at the prompt to continue.		
17	V-2 Verify that when selecting not to continue, the verification attempt is terminated, the reason listed in the log, and no requests are added to the export queue for the terminated operation (but are for the other operation).		#comment

#	Action	Expected Result	Notes
18	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows 100 HTTP PUT requests for collection C1. (There may be more than one HTTP request per granule, e.g., if there are network issues.)		
19	Verify that the TCP proxy log shows full granule metadata of the 100 granules in Collection C1.		
20	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) does not show any HTTP requests for collection C2.		
21	Verify that under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log the verification attempt termination is logged with the reason for termination		
22	S-3 Repeat S-1.		#comment
23	EcBmBMGTManualStart --mode <MODE> --metg -c <C1_shortname>.<C1_versionid>		
24	EcBmBMGTManualStart --mode <MODE> --long --metg -c <C2_shortname>.<C2_versionid>		
25	V-3 Verify that the export in S-3 results in a prompt indicating that there is already an export running		#comment
26	S-4 When prompted as to whether to continue, choose to continue.		#comment
27	Continue at the prompt		
28	V-4 Verify that when selecting to continue, the verification attempt continues, adding requests to the export queue, and exporting the associated metadata (for both of the requested operations).		#comment
29	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP put request for each of the 100 granules in collection C3. (There may be more than one HTTP request per granule, e.g., if there are network issues.)		
30	Verify that the TCP proxy log shows full granule metadata of the 100 granules in Collection C3		
31	Verify that the TCP proxy log shows a HTTP put request for granule all 100 granules in Collection C4.		
32	Verify that the TCP proxy log shows full granule metadata of the 100 granules in Collection C4		
33	Verify that the TCP proxy log shows that the requests for granules in Collection C4 contains the query parameter xml_diff=true		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		testing concurrent exports	4 collections with 100 granules each					

EXPECTED RESULTS:

	V	480	1	Verify that the export in S-1 results in a prompt indicating that there is already a verification export running.		
	V	480	2	Verify that when selecting not to continue, the verification attempt is terminated, the reason listed in the log, and no requests are added to the export queue for the terminated operation (but are for the other operation).		
	V	480	3	Verify that the export in S-3 results in a prompt indicating that there is already a verification export running.		
	V	480	4	Verify that when selecting to continue, the verification attempt continues, adding requests to the export queue, and exporting the associated metadata (for both of the requested operations).		

1.73 Long Form Verification - Nominal (ECS-ECSTC-72)

DESCRIPTION:

	S	490	1	**[Long Form Verification – Nominal]** Via the BMGT GUI, suspend the processing of long form verification exports.		
	S	490	2	Use the command line or GUI interface to request a verification export of granule and/or collection metadata. Ensure that the ECS mode is specified either on the command line or in the GUI URL.		
	S	490	3	Via the BMGT GUI resume the processing of verification exports.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/490.		
5	Ensure the test collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
6	Ensure the test collection is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure the test granules are in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid in ('${LOCALGRANULEID}', ...) If needed, ingest the granules into the public data pool.	Should return 10 rows.	
9	Ensure ECHO has the test granules' metadata. For each test granule, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing a granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		

#	Action	Expected Result	Notes
11	Setup		#comment
12	S-1 Via the BMGT GUI, suspend the processing of long form verification exports.		#comment
13	Suspend the verification queue in the BMGT GUI.		
14	S-2 Use the command line or GUI interface to request a verification export of granule and/or collection metadata. Ensure that the ECS mode is specified either on the command line or in the GUI URL.		#comment
15	Note the current time as t0.		
16	EcBmBMGTManualStart <MODE> --long --metc -metg --collections <C1_shortname>.<C1_versionid>		
17	V-1 Verify that prior to resumption of processing in S-3, the requested operations are visible on the queue, but are not exported		#comment
18	Verify that the GUI displays export requests for the test collection and granules.		
19	Verify the TCP proxy log shows no PUT requests after time t0.		
20	V-2 Verify that each of the requested items (collection or granule) appears in the BMGT export request queue (viewed through the GUI), and indicates that it is a long form verification export.		#comment
21	Verify that the GUI indicates that the requests are long form verification exports		
22	S-3 Via the BMGT GUI resume the processing of verification exports.		#comment
23	Resume processing long form verification exports on the GUI		
24	V-3 Verify that each of the enqueued export requests results in exactly one export to ECHO (or an ECHO stand-in).		#comment
25	Verify that the TCP proxy log shows one HTTP request for each request shown queued in the GUI.		
26	V-4 Verify that when the verification process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which initiation of the verification export started and completed, as well as how many items were enqueued for export.		#comment
27	Verify that the BMGT log is under usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log		
28	Verify that the BMGT log contains a message that manual export process was started		

#	Action	Expected Result	Notes
29	Verify that the BMGT log contains a message that all requests have been added to the queue		
30	Verify that the BMGT log contains the time of start of the verification export		
31	Verify that the BMGT log contains the time of completion of the verification export		
32	Verify that the BMGT log contains a message with the number of items enqueued for export		
33	V-5 Verify that the options specified on the command line (or in the GUI) for the initiation of a verification export are printed to the log file.		#comment
34	Verify that the BMGT log contains a message showing the options used to start the verification export		
35	V-6 Verify that the database and the log files contain information on the process of each request through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		#comment
36	Verify that bg_export_request contains requests for each of the granules and the collection C1		
37	Verify that the bg_export_activity table shows the activities for each of the granules and the collection C1		
38	Verify that the bg_export_error table shows any errors in the export requests		
39	Verify that the EcBmBMGTGenerator.log contains a message showing when the metadata was generated for each request		
40	Verify that the BMGT EcBmBMGTExporter.log contains a message showing when the export was sent to ECHO for each request		
41	Verify that the EcBmBMGTExporter.log contains a message showing when the response was received from ECHO for each request		
42	V-7 Verify that the BMGT GUI displays the completed export events, indicating that they were the result of a long form verification export. Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or collection ID.		#comment
43	V-8 Verify that the BMGT GUI displays the following verification metrics overall, or for a particular time frame:		#comment
44	a) Number of collections/granules which were exported for verification.		#comment
45	Verify that the BMGT GUI indicates 1 collection was exported for verification		
46	Verify that the BMGT GUI indicates 10 granules were exported for		

#	Action	Expected Result	Notes
	verification		
47	b) Number of collections/granules which were successfully verified		#comment
48	Verify that the BMGT GUI indicates the number of collections which were successfully verified		
49	Verify that the BMGT GUI indicates the number of granules which were successfully verified. Note that a warning indicates success; the verification may have caused an update.		
50	c) Number of collections/granules which failed verification but were automatically repaired.		#comment
51	Verify that the BMGT GUI indicates the number of collections that failed verification and were automatically repaired. Collections can be filtered, using the query filter at the top of the GUI. Select ItemType = CL. Check "Export-Request Queue Summary" panel and apply the "Use the above Time-Range and Filters". An automatically repaired collection will show a "Warning".		
52	Verify that the BMGT GUI indicates the number of granules that failed verification and was automatically repaired. Granules can be filtered, using the query filter at the top of the GUI. Select ItemType = SC. Check "Export-Request Queue Summary" panel and apply the "Use the above Time-Range and Filters". An automatically repaired granule will show a "Warning".		
53	d) Number of collections/granules which failed verification and could not be automatically repaired.		#comment
54	Verify that the BMGT GUI indicates the number of collections that failed verification and were not automatically repaired. Requests that could not be automatically corrected should be listed as BLOCKED.		
55	Verify that the BMGT GUI indicates the number of granules that failed verification and were not automatically repaired. Requests that could not be automatically corrected should be listed as BLOCKED.		
56	e) Number of collections/granules skipped during export due to errors.		#comment
57	Verify that the BMGT GUI indicates the number of collections that skipped verification due to errors. Skipped means the request was not sent to ECHO due to errors, such as invalid metadata.		
58	Verify that the BMGT GUI indicates the number of granules that skipped verification due to errors. Skipped means the request was not sent to ECHO due to errors, such as invalid metadata.		

TEST DATA:

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
490		10 granules in the same collection	MOD44W.005				/sotestdata/DROP_802/BE_82_01/Criteria/490/MOD44W.005	

EXPECTED RESULTS:

	V	490	1	Verify that prior to resumption of processing in S-3, the requested operations are visible on the queue, but are not exported		
	V	490	2	Verify that each of the requested items (collection or granule) appears in the BMGT export request queue (viewed through the GUI), and indicates that it is a long form verification export.		
	V	490	3	Verify that each of the enqueued export requests results in exactly one export to ECHO (or an ECHO stand-in).		
	V	490	4	Verify that when the verification process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which initiation of the verification export started and completed, as well as how many items were enqueued for export.		
	V	490	5	Verify that the options specified on the command line (or in the GUI) for the initiation of a verification export are printed to the log file.		
	V	490	6	Verify that the database and the log files contain information on the process of each request through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		
	V	490	7	Verify that the BMGT GUI displays the completed export events, indicating that they were the result of a long form verification export. Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or collection ID.		
	V	490	8	Verify that the BMGT GUI displays the following verification metrics overall, or for a particular time frame: a) Number of collections/granules which were exported for verification. b) Number of collections/granules which were successfully verified c) Number of collections/granules which failed verification but were automatically repaired. d) Number of collections/granules which failed verification and could not be automatically repaired. e) Number of collections/granules skipped during export due to errors.		

1.74 Long Form Verification - Invocation via Command Line - a) Multiple granules (ECS-ECSTC-73)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure test collections C1, C2, C3 are enabled for collection and granule		

#	Action	Expected Result	Notes
	export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collections were newly enabled for export in this step, wait for them and their granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure the test granules g1_C1, g2_C1, g1_C2, g2_C2, g1_C3, g2_C3 are in AIM (2 granules per collection): select shortname, versionid, granuleid from amgranule where localgranuleid in ('${LOCALGRANULEID}', ...) If needed, ingest the granules into the public data pool.	Should return 6 rows.	
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 From the command line invoke verification export of metadata for the following: a) Multiple granules, specified on the command line.		#comment
12	Note the current time as t0.		
13	Request the test granules' manual export (there must be no spaces between commas and granule IDs): EcBmBMGTManualStart <MODE> --long --metg -g g1_C1,g2_C1,g1_C2,g2_C2,g1_C3,g2_C3		
14	V-1 Verify that the operation in S-1 results in the following exports: a) Single HTTP PUT.		#comment
15	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows, after time t0, a single HTTP PUT request for each granule g1_C1, g1_C2, g1_C2, g2_C2,g1_C3, g2_C3 listed on the command line.		
16	Verify that the TCP proxy log shows each granule's URL ends in '?xml_diff=true'. For example, PUT /catalog-rest/providers/DEV08/granules/SC%3AD5TES1.001%3A87900?xml_diff=true HTTP/1.1		

TEST DATA:

Test data is under /sotestdata/DROP_802/BE_82_01/Criteria/492

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
492 S-1a				3 Collections with 2 granules each			492_1_A	
492 S-1b				3 Collections with 2 granules each			492_1_B	
492 S-1c				1 collection with 3 granules			492_1_C	
492 S-1d				1 collection with 3 granules			492_1_D	
492 S-1e				3 collections			492_1_E	
492 S-1f				3 collections			492_1_F	
492 S-1g				2 datapool groups with 2 collections each with 2 granules in each collection			492_1_G	
492 S-1h				2 datapool groups with 2 collections each with 2 granules in each collection			492_1_H	

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: o. Single HTTP PUT. p. Single HTTP PUT. q. HTTP PUT for each granule in the collection. r. HTTP PUT for each granule in the collection. s. HTTP PUT for each specified collection. t. HTTP PUT for each specified collection. u. HTTP PUT for each granule and collection in		

				the specified group. v. HTTP PUT for each granule and collection in the specified group.		

1.75 Long Form Verification - Exclude Pending Exports (ECS-ECSTC-74)

DESCRIPTION:

	S	500	1	**[Long Form Verification - Exclude Pending Exports]** Pause the processing of automatic exports.		
	S	500	2	Perform an update on: a) One granule b) One collection		
	S	500	3	Perform a long form granule and collection verification, requesting the inclusion of the granule and collection updated in S-2 as well as at least one other granule and one other collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure the test collections are enabled for collection and granule export. For each of C1, C2, C3, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID}<br		

#	Action	Expected Result	Notes
	/> If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing a collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID} 		
8	Ensure granules g1, g2 in C1, g3, g4 in C2, g5, g6 in C3 are in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ('${LOCALGRANULEID}', ...) If needed, ingest the granules into the public data pool.	Should return 6 rows.	
9	Ensure ECHO has the test granule metadata. For each granule, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing granules, export them: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID},...		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Setup		#comment
12	S-1 Pause the processing of automatic exports.		#comment
13	From the GUI, Pause Event Queue. 		
14	S-2 Perform an update on: a) One granule		#comment
15	Identify granule g1 in collection C1 to update. declare l_daynightflag as varchar(5); select daynightflag from amgranule into l_daynightflag where granuleid = <g1>; update amgranule set daynightflag = 'TEST' where granuleid = <g1>; return l_daynightflag; -- save for resetting the flag back to original value		
16	b) One collection		#comment
17	Update the collection C3 ESDT type with a new descriptor file or update the DsGeESDTConfiguredType manually. To update the DsGeESDTConfiguredType manually perform both queries in sequence:<br		

#	Action	Expected Result	Notes
	/> update DsGeESDTConfiguredType set esdtstate = 'updating' where configuredname = <C3_ShortName> and versionid = <C3_VersionId>; update DsGeESDTConfiguredType set esdtstate = 'installed' where configuredname = <C3_ShortName> and versionid = <C3_VersionId>;		
18	S-3 Perform a long form granule and collection verification, requesting the inclusion of the granule and collection updated in S-2 as well as at least one other granule and one other collection.		#comment
19	(Note: There must be no spaces around commas.) EcBmBMGTManualStart <MODE> --long --metc --metg --c <C1_shortname>.<C1_versionid>,<C2_shortname>.<C2_versionid>,<C3_shortname>.<C3_versionid>		
20	Verification		#comment
21	V-1 Verify that the granules and collections listed in S-3 are queued and exported, with the exception of those listed in S-2, for which an automatic update is pending.		#comment
22	Verify that the TCP proxy shows an HTTP PUT request for each collection C1, C2. (There may be more than one HTTP request, e.g., if there are network issues.)		
23	Verify that the TCP proxy shows no HTTP PUT request for collection C3.		
24	Verify that the TCP proxy show an HTTP PUT request for each of the granules g2,g3,g4 g1 is not queued g5, g6 will be in pending state on the GUI until the collection C3 gets exported (There may be more than one HTTP request, e.g., if there are network issues.)		
25	Verify that the TCP proxy shows no HTTP PUT request for granule g1.		
26	Verify that the TCP log proxy shows each granule's URL ends in '?xml_diff=true'.		
27	Cleanup		#comment
28	reset_daynight_flag(a_granuleid, a_daynightflag): update amgranule set daynightflag = a_daynightflag where granuleid = a_granuleid Then from the GUI resume the Event queue		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			3 collections with 2 granules each					
			1 script to update the ESDT for Collection C3					

EXPECTED RESULTS:

	V	500	1	Verify that the granules and collections listed in S-3 are queued and exported, with the exception of those listed in S-2, for which an automatic update is pending.		

1.76 Long Form Verification - Errors (ECS-ECSTC-75)

DESCRIPTION:

	S	510	1	**[Long Form Verification - Errors]** Perform a long form granule verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO).		
	S	510	2	Perform a long form collection verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO).		
	S	510	3	Perform a long form granule verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO). Cause the export to also fail ingest, for instance, due to EndingDateTime before BeginningDateTime.		
	S	510	4	Perform a long form collection verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO). Cause the export to also fail ingest, for instance, due to duplicate Additional Attributes names.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3, C4 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure the test collections are enabled for collection and granule export. For each of C1, C2, C3, C4, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, C4, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing a collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure ECHO has granule metadata for all granules in C1, C2, C3, C4. Either query ECHO for each granule, or just export them. Query ECHO per granule: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} Export all granules in each collection: Save the collections' ${SHORTNAME}.${VERSIONID}, one per line, to a file collections.txt. EcBmBMGTManualStart ${MODE} --metg -cf /path/to/collections.txt		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		

#	Action	Expected Result	Notes
10	Ensure that the email threshold is configured in the BMGT GUI	BMGT.ResponseHandler.Monitor.EmailTimeOut is the threshold for email to be sent	
11	Setup		#comment
12	S1 Perform a long form granule verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO).		#comment
13	Find granule g1's metadata file in collection C1 in the small file archive: /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
14	Save off the file and change one of the metadata values.		
15	Perform a long form verification of granule g1: EcBmBMGTManualStart <MODE> --long --metg --granules ${g1_GRANULEID}		
16	V-1 Verify that the exports in S-1 and S-2 are marked as having encountered warnings.		#comment
17	Verify warnings in the bmgt log for the export		
18	Verify that the BMGT GUI shows that the export of g1 encountered a warning		
19	V-3 Verify that an email is delivered to the configured notification email address listing the warning in S-1 and S-2. Verify that it lists the collection or granule which caused the warning, as well as the text of the warning as received from ECHO, and number of occurrences of each warning. Verify that it also indicates that the discrepancies listed have been resolved by ECHO.		#comment
20	Verify that an email is sent to the configured email address		
21	Verify that the email lists granule g1 which caused the warning		
22	Verify that the email lists the text of the warning received from ECHO		
23	Verify that the email lists the number of occurrences of the warning		
24	Verify that the the email indicates if the discrepancy has been resolved by ECHO		
25	V-5 Verify that the BMGT GUI also lists the text of the warning for the export requests in S-1 and S-2		#comment
26	Verify that the GUI lists the warning message associated with the export of		

#	Action	Expected Result	Notes
	G1.		
27	S2 Perform a long form collection verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO).		#comment
28	Find collection 2's descriptor file: /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
29	Copy off the file and change one of the metadata values		
30	Perform a long form collection verification for C2: EcBmBMGTManualStart <MODE> --long --metc --collections <C2_shortname>.<C2_versionid>		
31	V-1 Verify that the exports in S-1 and S-2 are marked as having encountered warnings.		#comment
32	Verify warnings in the bmgt log for the export		
33	Verify that the BMGT GUI shows that the export of C2 encountered a warning		
34	V-3 Verify that an email is delivered to the configured notification email address listing the warning in S-1 and S-2. Verify that it lists the collection or granule which caused the warning, as well as the text of the warning as received from ECHO, and number of occurrences of each warning. Verify that it also indicates that the discrepancies listed have been resolved by ECHO.		#comment
35	Verify that an email is sent to the configured email address		
36	Verify that the email lists Collection C2 which caused the warning		
37	Verify that the email lists the text of the warning received from ECHO		
38	Verify that the email lists the number of occurrences of the warning		
39	Verify that the the email indicates if the discrepancy has been resolved by ECHO		
40	V-5 Verify that the BMGT GUI also lists the text of the warning for the export requests in S-1 and S-2		#comment
41	Verify that the GUI lists the warning message associated with the export of C2.		
42	S3 Perform a long form granule verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO). Cause the export to also fail ingest, for instance, due		#comment

#	Action	Expected Result	Notes
	to EndingDateTime before BeginningDateTime.		
43	Find granule g2's metadata file in the small file archive in Collection C3: /stornext/smallfiles/<MODE>/metadata/<SHORT_NAME>.<VERSION_ID>/<YEAR>.<MONTH>/<SHORT_NAME>.<VERSION_ID>.<GRANULE_ID>.xml		
44	Copy off the file and change the RangeEndingDate to be 1 year before the RangeBeginningDate.		
45	Perform a long granule verification for granule g2: EcBmBMGTManualStart <MODE> --long --metg --granules <G2.granuleId>		
46	V-2 Verify that the exports in S-3 and S-4 are marked as failed.		#comment
47	Verify the bmgt logs to indicate the verification export failed		
48	Verify that the BMGT GUI marks the export of granule g2 as failed.		
49	V-4 Verify that an email is delivered to the configured notification email address listing the errors in S-3 and S-4. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		#comment
50	Verify that an email is sent to the configured email address		
51	Verify that the email lists granule g2 which caused the error		
52	Verify that the email lists the text of the warning received from ECHO		
53	Verify that the email lists the number of occurrences of the error		
54	V-6 Verify that the BMGT GUI also lists the text of the error for the export requests in S-3 and S-4		#comment
55	Verify that the GUI lists the error message associated with the export of g2.		
56	S4 Perform a long form collection verification export which will result in a metadata mismatch, or perform the export and artificially cause the response to indicate an ECHO ingest error (e.g. mismatching metadata value between SDPS and ECHO). Cause the export to also fail ingest, for instance, due to duplicate Additional Attributes names.		#comment
57	Find collection C4's descriptor file: /stornext/smallfiles/<MODE>/descriptor/*<SHORT_NAME>.<VERSION_ID>.desc		
58	Copy off the file and then edit it to duplicate one of the Additonal Attributes Names		
59	Perform a collection long form verification for collection C4: EcBmBMGTManualStart <MODE> --long --metc --collections		

#	Action	Expected Result	Notes
	<C4_shortname>.<C4_versionid>		
60	V-2 Verify that the exports in S-3 and S-4 are marked as failed.		#comment
61	Verify the bmgt logs to indicate the verification export failed		
62	Verify that the BMGT GUI lists the export of C4 as failed.		
63	V-4 Verify that an email is delivered to the configured notification email address listing the errors in S-3 and S-4. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		#comment
64	Verify that an email is sent to the configured email address		
65	Verify that the email lists Collection C4 which caused the error		
66	Verify that the email lists the text of the warning received from ECHO		
67	Verify that the email lists the number of occurrences of the error		
68	V-6 Verify that the BMGT GUI also lists the text of the error for the export requests in S-3 and S-4		#comment
69	Verify that the GUI lists the error message associated with the export of G1.		
70	V-7 Verify that the BMGT GUI provides statistics on the number of verification exports which encountered discrepancies and the number which were handled automatically by ECHO		#comment
71	Verify the GUI shows the number of successful verification exports		
72	Verify the GUI shows the number of verification exports which encountered discrerpancies but which were fixed automatically (i.e. warnings)		
73	Verify the GUI shows the number of failed verification exports.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
510		4 Collections C1, C2, C3, C4 with a couple of science granules in each	MCD43B1.005 MCD43B2.005 MCD43B3.005 MCD43B4.005				/sotestdata/DROP_802/BE_82_01/Criteria/510	

EXPECTED RESULTS:

	V	510	1	Verify that the exports in S-1 and S-2 are marked as having encountered warnings.		
	V	510	2	Verify that the exports in S-3 and S-4 are marked as failed.		
	V	510	3	Verify that an email is delivered to the configured notification email address listing the warning in S-1 and S-2. Verify that it lists the collection or granule which caused the warning, as well as the text of the warning as received from ECHO, and number of occurrences of each warning. Verify that it also indicates that the discrepancies listed have been resolved by ECHO.		
	V	510	4	Verify that an email is delivered to the configured notification email address listing the errors in S-3 and S-4. Verify that it lists the collection or granule which caused the error, as well as the text of the error as received from ECHO, and number of occurrences of each error.		
	V	510	5	Verify that the BMGT GUI also lists the text of the warning for the export requests in S-1 and S-2		
	V	510	6	Verify that the BMGT GUI also lists the text of the error for the export requests in S-3 and S-4		
	V	510	7	Verify that the BMGT GUI provides statistics on the number of verification exports which encountered discrepancies and the number which were handled automatically by ECHO		

1.77 Incremental Verification - Nominal (ECS-ECSTC-76)

DESCRIPTION:

	S	520	1	**[Incremental Verification – Nominal]** Choose an ECS inventory that includes at least one hundred thousand (100,000) science granules that are eligible for ECHO export and covers at least three different collections. Configure a time period increment for automatic verification that is not in excess of one month and no less than one week. Configure the maximum incremental export operation size such that none of the incremental verifications performed during the test will encounter the export size limit, i.e., the number of inserted/updated granules within the time increment never exceeds the maximum export size. Ensure that all collections in the mode are completely unverified, resetting the verification status if necessary.		
	S	520	2	Initiate incremental verification repeatedly until there are no more granules to verify. Ensure that no other operations are going on in the system which would cause the update of any granules. Ensure that the incremental start utility requires the ECS mode as a command line option.		

				It is acceptable to use a cron job or script to automate the initiation of incremental verification		
	S	520	3	Once incremental verification has completed and verified the entire inventory, perform some granule inserts and updates. Pause automatic export before making these updates so that the events are not automatically exported. Ensure that at least some of the updated granules are each of the following: a) In the public datapool b) Have browse links c) Are restricted.		
	S	520	4	Initiate another incremental verification export.		
	S	520	5	Resume Automatic export and allow it to pick up and export the events in S-3.		
	S	520	6	Initiate another incremental verification export.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure no activity other than the test will occur in the mode for the duration of the test.		
3	Ensure BMGT configuration is current and correct (config files, properties files, database settings).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
5	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/520.		
6	Ensure collections C1..C10 are installed. E.g., ensure the DPL Ingest GUI shows C1..C10 as configured datatypes.		
7	Ensure collections C1..C10 are enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where (shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>) or (shortname = '<C2_SHORTNAME>' and versionid = <C2_VERSIONID>) ... or (shortname =		

#	Action	Expected Result	Notes
	'<C10_SHORTNAME>' and versionid = <C10_VERSIONID>)		
8	Ensure collections C1..C1 exist in ECHO.		
9	Ensure all test granules exist in ECHO.		
10	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
11	Setup		#comment
12	S-1 Choose an ECS inventory that includes at least one hundred thousand (100,000) science granules that are eligible for ECHO export and covers at least three different collections. Configure a time period increment for automatic verification that is not in excess of one month and no less than one week. Configure the maximum incremental export operation size such that none of the incremental verifications performed during the test will encounter the export size limit, i.e., the number of inserted/updated granules within the time increment never exceeds the maximum export size. Ensure that all collections in the mode are completely unverified, resetting the verification status if necessary.		#comment
13	Ensure that the collections C1..C10 have a large number of granules in AIM (1000 in the EDF; about 100 000 in the PVC)		
14	Configure BMGT.Incremental.Duration = 10 (the interval is in days).		
15	Configure BMGT.Verification.MaxGranules to 100 in the EDF or 10 000 in the PVC.		
16	In the BMGT GUI, reset verification status for all collections to be unverified or 0%.		
17	V-1 Inspect the verification report in the BMGT GUI before performing any incremental verification, and verify that the overall verification percentage as well as that for each collection and group, is equal to 0%.		#comment
18	In the BMGT GUI System Status tab, verify that the overall verification percentage is displayed as 0%		
19	In the BMGT GUI System Status tab, verify that verification percentage is displayed as 0% for each collection and group in the mode		
20	S-2 Initiate incremental verification repeatedly until there are no more granules to verify.		#comment
21	Ensure that no other operations are going on in the system which would cause the update of any granules.		
22	Ensure that the incremental start utility requires the ECS mode as a command line option. It is acceptable to use a cron job or script to automate the initiation of incremental verification.		
23	Ensure that there are no ingest, delete, or update operations ocurring in the		

#	Action	Expected Result	Notes
	mode		
24	Ensure the TCP proxy log will have only requests exported during this test: Stop the TCP proxy. Move the log to a new name. Start the proxy.		
25	./EcBmBMGTManualStart <MODE> --incremental		
26	V-2 After each of the first two incremental iterations, inspect the logs and export request queue to ensure that all granules whose last update falls within the time period covered by the iteration were added to the request queue.		#comment
27	Verify that for the first two incremantal iterations, any granule that falls between min_last_update_time and min_last_update_time + BMGT.Incremental.Duration (up to the maximum defined by BMGT.Verification.MaxGranules and bg_collection_configuration.maxgranulestoverify for each collection)		
28	Verify that after running multiple incremental iterations, each of the granules in each of the Collections C1..C10 in the mode <MODE> have been added to the export request queue - check bg_export_request table.		
29	V-3 Inspect the log file to verify that for each Incremental verification initiation, no more than the configured number of granules is enqueued.		#comment
30	Verify that for each incremental iteration, the total number of granules is less than or equal to BMGT.Verification.MaxGranules		
31	Verify that for each incremental iteration, the total number of granules for each collection is less than or equal to bg_collection_configuration.maxgranulestoverify for the given collection.		
32	V-4 Verify that when each verification process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which the verification export driver started and completed, as well as how many items were enqueued for export, how many were added per collection, and the time span represented by the update times of the added granules.		#comment
33	Verify that the bmgt log is written under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log		
34	Verify that the bmgt log indicates the time when the incremental verification was started.		
35	Verify that the bmgt log indicates the time when all request have been added to the queue		

#	Action	Expected Result	Notes
36	Verify that the bmgt log indicated the time when verification export was started.		
37	Verify that the bmgt log indicated the time when verification export was completed.		
38	Verify that the bmgt log indicates the time span of the verified granules in the iteration.		
39	V-5 Inspect the verification report in the BMGT GUI after the first two incremental verification iterations to verify that the overall verification percentage increases as well as that for the collections and groups which were selected for export.		#comment
40	Verify the overall verification percentage increases after the first two iterations		
41	Verify the verification percentage for collections C1 - C10 increases after the first two iterations		
42	Verify the overall verification percentage = 100% after running multiple incremental iterations		
43	Verify that the verification percentage for Collections C1..C10 = 100% after running multiple incremental iterations.		
44	V-7 Verify that when verification is complete, in S-2, across all incremental exports, every eligible granule has been exported exactly once. This can be done at a coarse level, ensuring that the number of exports per collection is as expected.		#comment
45	Save the TCP proxy log, and start a new one: Stop the TCP proxy. Move the log to a new name. Start the proxy.		
46	Save eligible granule counts to a file: select count(g.granuleid), esdt(g.shortname, g.versionid) from amgranule g join DsGeESDTConfiguredType t on (g.shortname = t.configuredname and g.versionid = t.versionid) join bg_collection_configuration bcc on g.collectionid = bcc.collectionid where g.deleteeffectivedate is null and g.deletefromarchive != 'Y' and t.esdtstate = 'installed' and bcc.collectionexportflag = 'Y' and bcc.granuleexportflag = 'Y' group by g.shortname, g.versionid order by g.shortname, g.versionid		
47	Save the exported granule counts to a file: sed -n 's/^PUT \/[^]*\/granules\/SC%3A\([^%]*\)%3A.*/\1/p' tcp.log | sort | uniq -c > exported_counts.txt		
48	Verify the number of granules eligible for export per collection matches the		

#	Action	Expected Result	Notes
	number found in the TCP proxy log: diff -w eligible_counts.txt exported_counts.txt		
49	V-11 Verify that each of the verification export requests results in exactly one export to ECHO (or an ECHO stand-in). Each Export shall take the form of a single HTTP PUT request containing the full granule metadata and an HTTP query parameter which indicates that the request is for verification purposes.		#comment
50	Verify that the TCP proxy shows that each verification export is a single HTTP PUT request.		
51	Verify that the TCP proxy shows that the request has the HTTP query parameter "xmldiff=true" indicating it is for verification purposes.		
52	V-12 Verify that each incremental interval also queues and exports for each collection which has granules included in the incremental interval, the export of the associated collection metadata. This export shall take the form of a single HTTP PUT request per collection, containing the full collection metadata and an HTTP query parameter indicating that the request is for verification purposes.		#comment
53	Verify that the TCP proxy shows a HTTP PUT request with Collection metadata for Collections C1..C10.		
54	Verify that the TCP proxy shows that the request is for verification purposes.		
55	S-3 Once incremental verification has completed and verified the entire inventory, perform some granule inserts and updates. Pause automatic export before making these updates so that the events are not automatically exported. Ensure that at least some of the updated granules are each of the following: a) In the public datapool b) Have browse links c) Are restricted.		#comment
56	Pause the EVENT queue via the BMGT GUI, so requests are queued and in PENDING state and will not be picked up by incremental verification.		
57	a) In the public datapool		#comment
58	Ingest Science granules g1..g10 into Collection C1 with default publishing on.		
59	b) Have browse links		#comment
60	Ingest Science granules g11..g20 into Collection C2 with default publishing on.		
61	c) Are restricted.		#comment
62	Choose or create a restriction flag: select * from		

#	Action	Expected Result	Notes
	dsmdrestrictionflag -OR- insert into dsmdrestrictionflag values(128, 'BE_82_01 Crit 520')		
63	Add the restriction flag to granules g21, g22: insert into dsmdgranulerestriction values(<GRANULE_ID>, 128)		
64	S-4 Initiate another incremental verification export.		#comment
65	./EcBmBMGTManualStart -mode <MODE> --incremental		
66	V-8 Verify that the export attempt in S-4 results in a warning message indicating that there are no granules eligible for incremental verification and that no granules are added to the queue.		#comment
67	Verify that there are no granules queued for verification export		
68	S-5 Resume Automatic export and allow it to pick up and export the events in S-3.		#comment
69	Resume the Event queue in the BMGT GUI.		
70	Verify that each of the granules in each of the g1..g10 in Collection C1 have been added to the export request queue. - check bg_export_request table or the bmgt GUI		
71	Verify that the bmgt logs also include the export of each of the granules in C1..C10 in Collection C1.		
72	Verify that each of the granules in each of the g11..g20 in Collection C2 have been added to the export request queue. - check bg_export_request table or the bmgt GUI		
73	Verify that the bmgt logs also include the export of each of the granules in g11..g20 in Collection C2.		
74	Verify that each of the granules in each of the g21..g22 in Collection C3 have been added to the export request queue. - check bg_export_request table or the bmgt GUI		
75	Verify that the bmgt logs also include the export of each of the granules in g21..g22 in Collection C3.		
76	S-6 Initiate another incremental verification export.		#comment
77	./EcBmBMGTManualStart <MODE> --incremental		
78	V-2 After each of the first two incremental iterations, inspect the logs and export request queue to ensure that all granules whose last update falls within the time period covered by the iteration were added to the request queue.		#comment
79	Verify that each of the granules in each of the g1..g10 in Collection C1 have been added to the export request queue. - check bg_export_request table or the bmgt GUI		

#	Action	Expected Result	Notes
80	Verify that the bmgt logs also include the export of each of the granules in C1..C10 in Collection C1.		
81	Verify that each of the granules in each of the g11..g20 in Collection C2 have been added to the export request queue. - check bg_export_request table or the bmgt GUI		
82	Verify that the bmgt logs also include the export of each of the granules in C11..C20 in Collection C2.		
83	Verify that each of the granules in each of the g21..g22 in Collection C3 have been added to the export request queue. - check bg_export_request table or the bmgt GUI		
84	Verify that the bmgt logs also include the export of each of the granules in g21,.g22 in Collection C3.		
85	V-3 Inspect the log file to verify that for each Incremental verification initiation, no more than the configured number of granules is enqueued.		#comment
86	Verify that for each incremental iteration, the total number of granules is less than or equal to BMGT.Verification.MaxGranules		
87	Verify that for each incremental iteration, the total number of granules for each collection is less than or equal to bg_collection_configuration.maxgranulestoverify for the given collection.		
88	V-4 Verify that when each verification process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which the verification export driver started and completed, as well as how many items were enqueued for export, how many were added per collection, and the time span represented by the update times of the added granules.		#comment
89	Verify that the bmgt log is written under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log		
90	Verify that the bmgt log indicates the time when the incremental verification was started.		
91	Verify that the bmgt log indicates the time when all request have been added to the queue		
92	Verify that the bmgt log indicated the time when verification export was started.		
93	Verify that the bmgt log indicated the time when verification export was completed.		

#	Action	Expected Result	Notes
94	Verify that the bmgt log indicates the time span of the verified granules in the cycle.		
95	V-5 Inspect the verification report in the BMGT GUI after the first two incremental verification iterations to verify that the overall verification percentage increases as well as that for the collections and groups which were selected for export.		#comment
96	Verify the overall verification percentage = 100%		
97	Verify that the verification percentage for Collections C1..C10 = 100%		
98	V-9 Verify that the export attempt in S-6 results in the queueing and export of the granules which were updated in S-3		#comment
99	V-10 Verify that the bodies of the verification exports in S-6 are exactly the same as the bodies of the automatic exports in S-5, including datapool URLs and restriction flags. Note that it is allowable for the verification to include additional exports not in the automatic export, e.g. the collection metadata for all collections for which there is a granule export.		#comment
100	Compare the granule metadata in the tcp log between the automatic export (S-5) and verification export (S-6) to verify that they are identical. NOTE: The verification exports will have collection exports for any granule exports in the collection.		
101	If there are extra collection exports in the verification, these can be ignored.		
102	V-11 Verify that each of the verification export requests results in exactly one export to ECHO (or an ECHO stand-in). Each Export shall take the form of a single HTTP PUT request containing the full granule metadata and an HTTP query parameter which indicates that the request is for verification purposes.		#comment
103	Verify that the TCP proxy shows that each verification export is a single HTTP PUT request.		
104	Verify that the TCP proxy shows that the request has the HTTP query parameter "xmldiff=true" indicating it is for verification purposes.		
105	V-12 Verify that each incremental interval also queues and exports for each collection which has granules included in the incremental interval, the export of the associated collection metadata. This export shall take the form of a single HTTP PUT request per collection, containing the full collection metadata and an HTTP query parameter indicating that the request is for verification purposes.		#comment
106	Verify that the TCP proxy shows a HTTP PUT request with Collection		

#	Action	Expected Result	Notes
	metadata for Collections C1...C10.		
107	Verify that the TCP proxy shows that the request is for verification purposeses (the url will contain a query parameter "xmldiff=true").		
108	V-13 Verify that the database and the log files contain information on the process of each request through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		#comment
109	Verify that the bmgt logs and database show when the request was generated for each verification request.		
110	Verify that the bmgt logs and database show when the metadata was generated for each verification request.		
111	Verify that the bmgt logs and database show when the export was sent to ECHO.		
112	Verify that the bmgt logs and database show when the response was received for each verification request.		
113	V-14 Verify that the BMGT GUI displays the completed export events, indicating that they were the result of an incremental verification export. Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or ID.		#comment
114	In the BMGT GUI export request tab, filter the request to veiw only those items on the INCR queue		
115	Verify that each request is listed in the SUCCESS state.		
116	Verify that each request has the associated granule or collection id listed.		
117	Verify that each request has its completion time listed.		
118	V-15 Verify that the BMGT GUI displays the following incremental verification metrics overall, or for a particular time frame: a) Number of collections/granules which were exported for verification. b) Number of collections/granules which were successfully verified c) Number of collections/granules which failed verification but were automatically repaired. d) Number of collections/granules which failed verification and could not be automatically repaired. e) Number of collections/granules skipped during export due to errors.		#comment
119	In the BMGT GUI Export Request tab, filter to view only requests on the INCR queue.		
120	select the "Batch Job Summary" sub tab		
121	Verify that the "Batch Job Summary" sub tab lists for each		

#	Action	Expected Result	Notes
	incremental iteration, or batch, the number of items exported for verification.		
122	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items successfully verified (suiccess column).		
123	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items which failed verification, but were automatically repaired by ECHO (warning colum).		
124	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items which failed verification, but were not automatically repaired by ECHO.		
125	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items which were skipped.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
520			10 collections (C1..C10) with at least a total of 1000 granules				/sotestdata/DROP_802/BE_82_01/Criteria/520/README.txt	

EXPECTED RESULTS:

	V	520	1	Inspect the verification report in the BMGT GUI before performing any incremental verification, and verify that the overall verification percentage as well as that for each collection and group, is equal to 0%.		
	V	520	2	After each of the first two incremental iterations, inspect the logs and export request queue to ensure that all granules whose last update falls within the time period covered by the iteration were added to the request queue.		
	V	520	3	Inspect the log file to verify that for each Incremental verification initiation, no more than the configured number of granules is enqueued.		

	V	520	4	Verify that when each verification process is started, a message is printed to the log, followed by another message when all requests have been added to the queue. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs) and indicate the time at which the verification export driver started and completed, as well as how many items were enqueued for export, how many were added per collection, and the time span represented by the update times of the added granules.		
	V	520	5	Inspect the verification report in the BMGT GUI after the first two incremental verification iterations to verify that the overall verification percentage increases as well as that for the collections and groups which were selected for export.		
	V	520	6	Inspect the verification report in the BMGT GUI after the final incremental verification iteration to verify that the overall verification percentage, as well as that of each enabled collection and group, is 100%.		
	V	520	7	Verify that when verification is complete, in S-2, across all incremental exports, every eligible granule has been exported exactly once. This can be done at a coarse level, ensuring that the number of exports per collection is as expected.		
	V	520	8	Verify that the export attempt in S-4 results in a warning message indicating that there are no granules eligible for incremental verification and that no granules are added to the queue.		
	V	520	9	Verify that the export attempt in S-6 results in the queueing and export of the granules which were updated in S-3		
	V	520	10	Verify that the bodies of the verification exports in S-6 are exactly the same as the bodies of the automatic exports in S-5, including datapool URLs and restriction flags.		
	V	520	11	Verify that each of the verification export requests results in exactly one export to ECHO (or an ECHO stand-in). Each Export shall take the form of a single HTTP PUT request containing the full granule metadata and an HTTP query parameter which indicates that the request is for verification purposes.		
	V	520	12	Verify that each incremental interval also queues and exports for each collection which has granules included in the incremental interval, the export of the associated collection metadata. This export shall take the form of a single HTTP PUT request per collection, containing the full collection metadata and an HTTP query parameter indicating that the request is for verification purposes.		
	V	520	13	Verify that the database and the log files contain information on the process of each request through the system such that it is possible to identify when the metadata was generated, when the export was sent to ECHO, and when the response was received, etc.		

	V	520	14	Verify that the BMGT GUI displays the completed export events, indicating that they were the result of an incremental verification export. Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or ID.		
	V	520	15	Verify that the BMGT GUI displays the following incremental verification metrics overall, or for a particular time frame: a) Number of collections/granules which were exported for verification. b) Number of collections/granules which were successfully verified c) Number of collections/granules which failed verification but were automatically repaired. d) Number of collections/granules which failed verification and could not be automatically repaired. e) Number of collections/granules skipped during export due to errors.		

1.78 Incremental Verification - Reset (ECS-ECSTC-77)

DESCRIPTION:

	S	530	1	**[Incremental Verification – Reset]** Ensure that incremental verification has completed and all collections are at 100% verification prior to the start of this test.		
	S	530	2	In the BMGT GUI, request the reset of verification for a single collection.		
	S	530	3	Continuously initiate incremental exports until there are no more granules to verify. It is acceptable to use a cron job or script to automate the initiation of incremental verification		
	S	530	4	In the BMGT GUI, request the reset of verification for the entire catalog.		
	S	530	5	Continuously initiate incremental exports until there are no more granules to verify. It is acceptable to use a cron job or script to automate the initiation of incremental verification		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		

#	Action	Expected Result	Notes
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/530.		
5	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
6	Setup		#comment
7	Assume Collections C1, C2, C3 have been installed in the mode. (ESDT verification script)		
8	Verify Collections C1, C2, C3 are enabled for Collection and Granule Export.		
9	Assume granules g1..g10 are installed on Collection C1. granules g11...g20 in Collection C2 and granules g21...g30 in Collection C3.		
10	ensure that the collections and granules referenced above have been previously exported to ECHO		
11	S-1 Ensure that incremental verification has completed and all collections are at 100% verification prior to the start of this test.		#comment
12	Ensure the BMGT GUI System Status tab indicates overall verification status at 100%.		
13	Ensure BMGT GUI System Status tab indicates the verification status for each of the collections C1, C2, C3 is 100%.		
14	S-2 In the BMGT GUI, request the reset of verification for a single collection.		#comment
15	Reset the verification status of Collection C1 from the BMGT GUI.		
16	V-1 Verify that after resetting verification for a single collection in S-2, the GUI displays a 0% verification progress for that collection, but that the rest of the collections remain at 100%		#comment
17	Verify that the BMGT GUI System Status tab displays verification status as 0% for collection C1.		
18	Verify that the BMGT GUI System Status Tab displays verification status as 100% for collections C2 and C3.		
19	S-3 Continuously initiate incremental exports until there are no more granules to verify. It is acceptable to use a cron job or script to automate the initiation of incremental verification		#comment
20	./EcBmBMGTManualStart -mode <MODE> --incremental		
21	V-2 Verify that when running incremental verification in S-3, only the reset collection is included in the verification exports		#comment
22	Verify that the TCP proxy log shows only granules from collection C1 being exported by the incremental verifications.		
23	S-4 In the BMGT GUI, request the reset of verification for the entire		#comment

#	Action	Expected Result	Notes
	catalog.		
24	Reset the verification of all three collections C1, C2, C3 in the mode.		
25	V-3 Verify that after resetting verification of the entire system, the overall progress, per group, and per collection verification progress is listed as 0% in the GUI.		#comment
26	Verify that the BMGT GUI System Status tab shows collection verification status as 0% for collections C1, C2 and C3, for each group, and for the system overall.		
27	S-5 Continuously initiate incremental exports until there are no more granules to verify. It is acceptable to use a cron job or script to automate the initiation of incremental verification		#comment
28	./EcBmBMGTManualStart -mode <MODE> --incremental		
29	V-4 Verify that after running incremental verification repeatedly in S-5, granules are queued and exported, and that eventually the collection verification status overall, and for each collection and group reaches 100%.		#comment
30	Verify that the TCP proxy log shows granules g1...g30 are exported.		
31	Verify that the BMGT GUI System Status tab shows collections C1, C2, C3 eventually reach 100%.		
32	Verify that the BMGT GUI System Status tab shows the overall verification status eventually reaches 100%.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
530		1 collection (C1) with 10 granules (g1 .. g10)	MCD45A1.005				/sotestdata/DROP_802/BE_82_01/Criteria/530	
530		1 collection (C2) with 10 granules (g11 .. g20)	MOD29P1N.005				/sotestdata/DROP_802/BE_82_01/Criteria/530	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
530		1 collection (C3) with 10 granules (g21 .. g30)	MYD29P1N.005				/sotestdata/DROP_802/BE_82_01/Criteria/530	

EXPECTED RESULTS:

	V	530	1	Verify that after resetting verification for a single collection in S-2, the GUI displays a 0% verification progress for that collection, but that the rest of the collections remain at 100%		
	V	530	2	Verify that when running incremental verification in S-3, only the reset collection is included in the verification exports.		
	V	530	3	Verify that after resetting verification of the entire system, the overall progress, per group, and per collection verification progress is listed as 0% in the GUI.		
	V	530	4	Verify that after running incremental verification repeatedly in S-5, granules are queued and exported, and that eventually the collection verification status overall, and for each collection and group reaches 100%.		

1.79 Incremental Verification - Max Granules Exceeded (ECS-ECSTC-78)

DESCRIPTION:

	S	540	1	**[Incremental Verification - Max Granules Exceeded]** Reset incremental verification.		
	S	540	2	In the BMGT GUI, increase the incremental coverage interval to be very large, and/or reduce the max number of granules, such that in order to fit the specified time interval into a single verification interval would necessarily exceed the maximum number of granules allowed (e.g. max granules = 100 and max interval = 2 years).		
	S	540	3	Initiate a number of incremental export operations.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available, connected to the ecs database.		
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/540.		
5	Ensure collection C1 is installed. E.g., ensure the DPL Ingest GUI shows C1 as a configured datatype.		
6	Ensure collection C1 is enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>		
7	Ensure collection C1 exists in ECHO.		
8	Ensure granules g1..g350 exist in AIM.		
9	Ensure granules g1..g350 exist in ECHO.		
10	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
11	Setup		#comment
12	S-1 Reset incremental verification.		#comment
13	From the BMGT GUI System Status tab, reset incremental verification for all collections.		
14	S-2 In the BMGT GUI, increase the incremental coverage interval to be very large, and/or reduce the max number of granules, such that in order to fit the specified time interval into a single verification interval would necessarily exceed the maximum number of granules allowed (e.g. max granules = 100 and max interval = 2 years).		#comment
15	Change BMGT.Incremental.Duration = 730 (2 years in Days).		
16	Change BMGT.Verification.MaxGranules = 100		
17	S-3 Initiate a number of incremental export operations.		#comment
18	./EcBmBMGTManualStart <MODE> --incremental		
19	V-1 Verify that the log indicates that the number of granules in each incremental export operation never exceeded the configured maximum.		#comment
20	Verify the BMGT log records the number of granules in incremental export is not greater than 100		
21	V-2 Verify that the log indicates that the population of each incremental operation was limited as the maximum number of granules was reached.		#comment

#	Action	Expected Result	Notes
22	Verify the BMGT log records the number of granules in incremental limited to 100 since max number of granules reached.		
23	./EcBmBMGTManualStart <MODE> --incremental		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
540		1 collection with 350 science granules	MOD14.005				/sotestdata/DROP_802/BE_82_01/Criteria/540	

EXPECTED RESULTS:

	V	540	1	Verify that the log indicates that the number of granules in each incremental export operation never exceeded the configured maximum.		
	V	540	2	Verify that the log indicates that the population of each incremental operation was limited as the maximum number of granules was reached.		

1.80 Incremental Verification - Concurrent Operation (ECS-ECSTC-79)

DESCRIPTION:

	S	550	1	**[Incremental Verification - Concurrent Operation]** Initiate a manual export which will take a considerable ammount of time to enqueue the associated requests (i.e. a large number of granules). Simultaneously initiate incremental verification operation from the command line.		
	S	550	2	When prompted as to whether to continue, choose to exit.		
	S	550	3	Repeat S-1		

	S	550	4	When prompted as to whether to continue, choose to continue.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/550.		
5	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	Ensure collection C1 has been installed in the mode. E.g., the DPL Ingest GUI shows it as a configured data type.		
8	Ensure collection C1 is enabled for Collection and Granule Export.		
9	Ensure granules g1..g500 are in AIM. Check the lastUpdateVerified date in bg_collection_status for the collection C1. select lastupdateverified from bg_collection_status where collectionid = <C1_collectionid> select propertyvalue from bg_collection_configuration where propertyname = 'BMGT.incremental.MaxDuration'; -- which is the max_duration in days select propertyvalue from bg_collection_configuration where propertyname = 'BMGT.incremental.MaxGranules'; -- which is the max_granules Calculate number of granules that were inserted between LastUpdateVerified date and (LastUpdateVerified date + max_duration . Let it be num_granules.		
10	S-1 Initiate a manual export which will take a considerable ammount of time to enqueue the associated requests (i.e. a large number of granules). Simultaneously initiate incremental verification operation from the command line.		#comment
11	Open 2 terminals, and cd to /usr/ecs/${MODE}/CUSTOM/utilities.		
12	In one terminal, ./EcBmBMGTManualStart <MODE> --metc --metg --c <C1_shortname>.<C1.versionid>		

#	Action	Expected Result	Notes
13	In a separate terminal, ./EcBmBMGTManualStart <MODE> --metc --incremental		
14	V-1 Verify that the operation in S-1 results in a prompt indicating that there is already an export running.		#comment
15	Verify a prompt indicating there is an export running.		
16	S-2 When prompted as to whether to continue, choose to exit		#comment
17	Exit at the prompt.		
18	V-2 Verify that when selecting not to continue, the verification attempt is terminated, and the reason listed in the log, but that the manual operation that was started first is allowed to run.		#comment
19	Verify that the verification is terminated.		
20	Verify that the BMGT log indicates that manual export was started first.		
21	S-3 Repeat S-1		#comment
22	./EcBmBMGTManualStart <MODE> --metc --metg --c <C1_shortname>.<C1.versionid>		
23	./EcBmBMGTManualStart <MODE> --incremental		
24	S-4 When prompted as to whether to continue, choose to continue.		#comment
25	Continue at prompt.		
26	V-4 Verify that when selecting to continue, the verification attempt continues, and both operations complete, resulting in the queueing and export of the expected granules and collections.		#comment
27	Verify that that the BMGT logs and GUI indicate that manual operation exports the 500 granules.		
28	Verify that the BMGT logs and GUI indicate that verification operation exports num_granules or max_granules, whichever is smaller.		
29	Verify that the lastUpdateVerified in bg_collection_status is updated to max_duration + original value of the lastupdateverified date.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
550		1 Collection with 500 science	MOD14.005				/sotestdata/DROP_802/BE_82_01/Criteria/550	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		granules						

EXPECTED RESULTS:

	V	550	1	Verify that the operation in S-1 results in a prompt indicating that there is already a verification export running.		
	V	550	2	Verify that when selecting not to continue, the verification attempt is terminated, and the reason listed in the log, but that the manual operation that was started first is allowed to run.		
	V	550	3	Verify that the operation in S-3 results in a prompt indicating that there is already a verification export running.		
	V	550	4	Verify that when selecting to continue, the verification attempt continues, and both operations complete, resulting in the queueing and export of the expected granules and collections.		

1.81 Incremental Verification - Errors (ECS-ECSTC-80)

DESCRIPTION:

	S	560	1	**[Incremental Verification - Errors]** Manually delete 1 granule and 1 collection from ECHO such that SDPS contains these items and ECHO does not.		
	S	560	2	Manually modify the metadata of one existing granule and one existing collection in ECHO such that ECHO and SDPS' metadata for these items are not in sync.		
	S	560	3	Perform an incremental verification which will cover the collection and granule modified in S-2 and S-3.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	ECHO REST API documentation:<br		#comment

#	Action	Expected Result	Notes
	/>https://testbed.echo.nasa.gov/catalog-rest/ingest-docs/ https://testbed.echo.nasa.gov/echo-rest/docs/index.html		
3	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
5	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/560.		
6	Ensure collections C1..C4 are installed. E.g., ensure the DPL Ingest GUI shows C1..C4 as configured datatypes.		
7	Ensure collection C1..C4 are enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where (shortname = '<C1_SHORTNAME>' and versionid = '<C1_VERSIONID>') or (shortname = '<C2_SHORTNAME>' and versionid = '<C2_VERSIONID>') or (shortname = '<C3_SHORTNAME>' and versionid = '<C3_VERSIONID>') or (shortname = '<C4_SHORTNAME>' and versionid = '<C4_VERSIONID>')		
8	Ensure collections C1..C4 exist in ECHO, using curl and the ECHO REST API.		
9	Ensure granules g1..g12 exist in AIM.		
10	Ensure granules g1..g12 exist in ECHO, using curl and the ECHO REST API.		
11	Ensure BMGT is configured to send email to a reachable address. If needed, update the value of BMGT.EmailLogger.To either in the BMGT GUI "BMGT Configuration" tab, or in the bg_configuration_property table in the ecs database.		
12	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
13	Setup		#comment
14	S-1 Manually delete 1 granule and 1 collection from ECHO such that SDPS contains these items and ECHO does not.		#comment
15	NOTE: can also cause an artificial error to be returned indicating that the colelction and granule are missing.		#comment
16	Delete collection C1 from ECHO, using curl and the ECHO REST API.		
17	Delete granule g2 from ECHO, using curl and the ECHO REST API.		
18	S-2 Manually modify the metadata of one existing granule and one existing collection in ECHO such that ECHO and SDPS’ metadata for these items are not in sync.		#comment

#	Action	Expected Result	Notes
19	Modify collection C3's descriptor file in the small file archive: Save a copy to restore later. Change, e.g., one or more ContactOrganizationAddress values.		
20	Modify granule g4's metadata file in the small file archive: Save a copy to restore later. Change, e.g., the value of a quality flag explanation or the value of ReprocessingPlanned.		
21	S-3 Perform an incremental verification which will cover the collection and granule modified in S-2 and S-3.		#comment
22	Reset verification status for all the 4 collections C1, C2, C3, C4.		
23	./EcBmBMGTManualStart -mode <MODE> --incremental		
24	 Verification 		
25	V-1 Verify that an email is sent to the configured notification email address listing the following discrepancies: a) 1 granule missing from ECHO b) 1 collection missing from ECHO c) 1 granule with mismatched metadata d) 1 collection with mismatched metadata Verify that the email contains the ECHO error messages for these discrepancies. Verify that it also indicates that the errors were all automatically repaired and there is no manual intervention needed.		#comment
26	Verify that an email is sent to the configured email address.		
27	Verify that the email contains notification that granule g1 is missing from ECHO.		
28	Verify that the email contains the notification that Collection C2 is missing from ECHO.		
29	Verify that the email contains the notification that granule g2 has mismatched metadata.		
30	Verify that the email contains the notification that the Collection C4 has mismatched metadata.		
31	Verify that the email indicates errors were automatically repaired.		
32	V-2 Verify that the incremental verification statistics for the affected collections is updated to indicate the number of discrepancies found and repaired.		#comment
33	Verify that the incremental status for the collection is updated to indicate the newly verified granules (increase in percentage verified).		
34	V-3 Verify that the BMGT GUI provides statistics on the number discrepancies found by each incremental verification attempt, and whether these discrepancies were automatically repaired.		#comment
35	Go to the BMGT GUI export request tab.		

#	Action	Expected Result	Notes
36	Filter the requests to view those on the INCR queue.		
37	Click the Batch Job Summary sub tab and find the row for the batch ID associated with the incremental verification performed above.		
38	Verify that the batch Job Summary indicates that this batch had 4 automatically repaired discrepancies (Warning Column).		
39	Clean up		#comment
40	Restore collection and granule metadata mangled during the test.		

TEST DATA:

Test data is under /sotestdata/DROP_802/BE_82_01/Criteria/560

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 collection (C1) with 1 granules (g1)				MOD11A1.005/	
			1 collection (C2) with 1 granule (g2)				MOD17A2.005/	
			1 collection (C3) with 1 granule (g3)				MYD11A1.005/	
			1 collection (C4) with 1 granule (g4)				MYD17A2.005/	

EXPECTED RESULTS:

	V	560	1	Verify that an email is sent to the configured notification email address listing the following discrepancies: a) 1 granule missing from ECHO b) 1 collection missing from ECHO c) 1 granule with mismatched metadata d) 1 collection with mismatched metadata Verify that the email contains the ECHO error messages for these discrepancies. Verify that it also indicates that the errors were all automatically repaired and there is no manual intervention needed		
	V	560	2	Verify that the incremental verification statistics for the affected collections is updated to indicate the number of		

				discrepancies found and repaired.		
	V	560	3	Verify that the BMGT GUI provides statistics on the number discrepancies found by each incremental verification attempt, and whether these discrepancies were automatically repaired.		

1.82 Short Form Verification - Granule (ECS-ECSTC-81)

DESCRIPTION:

	S	570	1	**[Short Form Verification – Granule]** Request the short form verification of granule metadata using each of the following methods, specifying one collection on the command line and one in an input.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/570 with ECS metadata		
6	Ensure collections C1,C2 have been installed in the mode.		
7	Verify collections C1, C2 are enabled for collection and granule export.		
8	Ensure granules g1..g100, belonging to collection C1, have been ingested.		
9	Ensure granules g101..g200, belonging to collection C2, have been ingested.		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Setup		#comment
12	S-1 Request the short form verification of granule metadata using each of the following methods, specifying one collection on the command line and one in an input.		#comment

#	Action	Expected Result	Notes
13	./EcBmBMGTManualStart <MODE> --short --metg --c <C1_shortname>.<C1_versionid>		
14	Copy the collection shortname and version <C2_shortname>.<C2_versionid> into a file collectionfile.txt ./EcBmBMGTManualStart <MODE> --short --metg --cf /path/to/collectionfile.txt		
15	V-1 Verify that for the operation in S-1, BMGT requests a listing of all granules in the referenced collections from ECHO, and compares this to the list of granules in those collections in AIM. Verify that the logs indicate the number of granules in ECHO and AIM for each collection.		#comment
16	Verify that the TCP proxy shows the HTTP verification query requests in the form: https://testbed.ECHO.nasa.gov/catalog-rest/providers/EDF_DEV02/datasets/<DATASET_ID>/verify for collections C1 and C2		
17	Verify that the BMGT logs indicate the comparison of the 100 granules in Collection C1 to the granule listing of C1 returned from ECHO.		
18	Verify that the BMGT logs indicate the compariaon of the 100 granules in Collection C2 to the granule listing of Collection C2 returned from ECHO. The EcBmBMGTManual.log has the comparisons. Comparisons start with ShortVerFilterer.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			Collection C1 and C2 with 100 science granules each					

EXPECTED RESULTS:

	V	570	1	Verify that for the operation in S-1, BMGT requests a listing of all granules in the referenced collections from ECHO, and compares this to the list of granules in those collections in AIM. Verify that the logs indicate the number of granules in ECHO and AIM for each		

				collection.		

1.83 Short Form Verification - Collection (ECS-ECSTC-82)

DESCRIPTION:

	S	580	1	**[Short Form Verification – Collection]** Request the short form verification of collection metadata.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/580 with ECS metadata		
6	Ensure there are some collections have been installed in the mode		
7	Ensure that these collections are enabled for Collection and Granule Export.		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Setup		#comment
10	S-1 Request the short form verification of collection metadata.		#comment
11	./EcBmBMGTManualStart <MODE> --short --metc		
12	V-1 Verify that for each operation in S-1, BMGT requests a listing of all collections from ECHO, and compares this to the list of collections in AIM. Verify that the logs indicate the number of collections in ECHO and AIM.		#comment
13	Verify that the TCP proxy log shows an HTTP verification query request in the form: GET /catalog-rest/providers/EDF_OPS/datasets/verify HTTP/1.1 with the path ending in "/verify".		
14	Verify that the BMGT manual driver log shows that the list of collections in		

#	Action	Expected Result	Notes
	AIM that are enabled for collection export are being compared to list of collections from ECHO.		
15	Verify the BMGT manual driver log indicates the number of collections in ECHO and in AIM.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			10 collections with granules					

EXPECTED RESULTS:

	V	580	1	Verify that for each operation in S-1, BMGT requests a listing of all collections from ECHO, and compares this to the list of collections in AIM. Verify that the logs indicate the number of collections in ECHO and AIM.		

1.84 Short Form Verification - Concurrent Operation (ECS-ECSTC-83)

DESCRIPTION:

	S	590	1	**[Short Form Verification - Concurrent Operation]** Initiate a manual export which will take a considerable ammount of time to enqueue the associated requests (i.e. a large number of granules). Simultaneously initiate a short form verification operation from the command line.		
	S	590	2	When prompted as to whether to continue, choose to exit.		
	S	590	3	Repeat S-1		
	S	590	4	When prompted as to whether to continue, choose to continue.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/590 with ECS metadata		
6	Ensure Collection C1 has been installed in the mode.		
7	Ensure Collection C1 is enabled for Collection and Granule Export.		
8	Ensure granules g1..g500, belonging to collection C1, have been ingested. Doesn't have to be 500 granules but a sufficient number to allow concurrent operation. 100 may be enough.		
9	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
10	Setup		#comment
11	S-1 Initiate a manual export which will take a considerable ammount of time to enqueue the associated requests (i.e. a large number of granules). Simultaneously initiate a short form verification operation from the command line.		#comment
12	Start a manual export in one terminal, followed immediately by a short form verification in another terminal: ./EcBmBMGTManualStart <MODE> --metg --c <C1_shortname>.<C1_versionid> ./EcBmBMGTManualStart <MODE> --metg --c <C1_shortname>.<C1_versionid> --short		
13	V-1 Verify that the operation in S-1 results in a prompt indicating that there is already a verification export running.		#comment
14	Verify a prompt is displayed indicating there is a manual export running		
15	S-2 When prompted as to whether to continue, choose to exit		#comment
16	Exit at the prompt		
17	V-2 Verify that when selecting not to continue, the verification attempt is		#comment

#	Action	Expected Result	Notes
	terminated, and the reason listed in the log, but that the manual operation that was started first is allowed to run.		
18	Verify that the verification is terminated.		
19	Verify that the bmgt log indicates that the verification was terminated due to a concurrent operation.		
20	S-3 Repeat S-1		#comment
21	./EcBmBMGTManualStart <MODE> --metg --c <C1_shortname>.<C1_versionid>		
22	./EcBmBMGTManualStart <MODE> --metg --c <C1_shortname>.<C1_versionid> --short		
23	S-4 When prompted as to whether to continue, choose to continue.		#comment
24	Continue at prompt		
25	V-4 Verify that when selecting to continue, the verification attempt continues, and both the short form and manual operations complete.		#comment
26	Verify that the BMGT logs and GUI indicate that manual operation exports the 500 granules		
27	Verify that the BMGT logs and GUI indicates that short verification operation completes successfully		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 Collection with 500 granules					

EXPECTED RESULTS:

	V	590	1	Verify that the operation in S-1 results in a prompt indicating that there is already a verification export running.		
	V	590	2	Verify that when selecting not to continue, the verification attempt is terminated, and the reason listed in the log, but that the other operation that was started first is allowed to run.		
	V	590	3	Verify that the operation in S-3 results in a prompt indicating that there is already a verification export running.		

	V	590	4	Verify that when selecting to continue, the verification attempt continues, and both the short form and manual operations complete.		

1.85 Short Form Verification - Nominal (ECS-ECSTC-84)

DESCRIPTION:

	S	600	1	**[Short Form Verification – Nominal]** Use the command line or GUI interface to request a short form verification of granules and/or collections Ensure that the ECS mode is specified either on the command line or in the GUI URL.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/600.		
5	Ensure collections C1, C2, C3 are installed. E.g., ensure the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensue collections C1, C2, C3 are enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where (shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>) or (shortname = '<C2_SHORTNAME>' and versionid = <C2_VERSIONID>) or (shortname = '<C3_SHORTNAME>' and versionid = <C3_VERSIONID>)		
7	Ensure a TCP proxy or mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 Use the command line or GUI interface to request a short form		#comment

#	Action	Expected Result	Notes
	verification of granules and/or collections. Ensure that the ECS mode is specified either on the command line or in the GUI URL.		
10	EcBmBMGTManualStart <MODE> --short --metc		
11	EcBmBMGTManualStart <MODE> --short --metg --collections <C1_SHORTNAME>.<C1_VERSIONID>		
12	Verification		#comment
13	V-1 Verify that when the verification process is started, a message is printed to the log, followed by another message when the verification is complete. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs).		#comment
14	Verify that the EcBmBMGTManualDriver.log is under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log		
15	Verify that the EcBmBMGTManualDriver.log contains a message that short form verification process was started		
16	Verify that the EcBmBMGTManualDriver.log contains the time of start of the verification operation		
17	Verify that the EcBmBMGTManualDriver.log contains the time of completion of the verification operation		
18	V-2 Verify that the options specified on the command line (or in the GUI) for the initiation of a verification export are printed to the log file.		#comment
19	Verify that the bmgt log contains a message showing the options used to start the verification export		
20	V-3 Verify that the BMGT GUI displays the status of each short form verification operation.		#comment
21	Go to the Export Request tab in the GUI		
22	Select the Batch Summary Statistics sub tab		
23	Find the batch associated with the short form verification performed above (get batch id from manual driver log).		
24	Verify that the status of the operation is shown, including the number of discrepancies found and number of successful verifications.		
25	V-4 Verify that the BMGT GUI displays the following metrics for each short form verification operation: a) Number of collections/granules included in the verification. b) Number of discrepancies found. c) Number of collections/granules skipped during export due to errors.		#comment
26	Verify that the summary for the short verification batch shows the total number of items (granules and collections) included in the operation		

#	Action	Expected Result	Notes
27	Verify that the summary for the short verification batch shows the number of discrepancies found in the short verification operation (under the warning column)		
28	Verify that the summary for the short verification batch shows the number of granules or collections skipped in the short verification operation (under the skipped column).		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			3 collections with granules					

EXPECTED RESULTS:

	V	600	1	Verify that when the verification process is started, a message is printed to the log, followed by another message when the verification is complete. Verify that the logs are written to the conventional ECS location (i.e. under /usr/ecs/<MODE>/CUSTOM/logs).		
	V	600	2	Verify that the options specified on the command line (or in the GUI) for the initiation of a verification export are printed to the log file.		
	V	600	3	Verify that the BMGT GUI displays the status of each short form verification operation..		
	V	600	4	Verify that the BMGT GUI displays the following metrics for each short form verification operation: a. Number of collections/granules included in the verification. b. Number of discrepancies found. c. Number of collections/granules skipped during export due to errors.		

1.86 Short Form Verification - Exclude Pending Exports (ECS-ECSTC-85)

DESCRIPTION:

	S	610	1	**[Short Form Verification - Exclude Pending Exports]** Pause the processing of automatic exports.		
	S	610	2	Insert: a) One granule b) One collection Delete: a) One granule b) One collection		
	S	610	3	Perform a short form collection verification.		
	S	610	4	Perform a short form verification for the collection which contains the granule updated in S-2.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/610 with ECS metadata		
6	Ensure collections C1, C2,C3 have been installed in the mode.		
7	Ensure collections C1, C2, C3 are enabled for collection and granule export.		
8	Ingest granules g1, g2 in C1, g3, g4 into C2, g5, g6 into C3		
9	Ensure that the granules and collections have been exported.		
10	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		

#	Action	Expected Result	Notes
11	Setup		#comment
12	S-1 Pause the processing of automatic exports.		#comment
13	From the GUI pause processing of automatic exports by pausing the 'NEW' and 'EVENT' queues on the System Status page of the BMGT GUI.		
14	S-2 Insert: a) One granule b) One collection Delete: a) One granule b) One collection		#comment
15	Ingest granule g7, belonging to collection C1.		
16	Install a new Collection C4 with granule g8, g9, g10		
17	Delete granule g3 from Collection C2		
18	Delete Collection C3		
19	S-3 Perform a short form collection verification.		#comment
20	EcBmBMGTManualStart <MODE> --short --metc		
21	S-4 Perform a short form verification for the collection which contains the granule updated in S-2.		#comment
22	EcBmBMGTManualStart <MODE> --short --metg -c <C1_shortname>.<C1_versionid>,<C2_shortname>.<C2_versionid>		
23	V-1 Verify that the collection verification in S-3 does not contain the collection inserted in S-2, as its export to ECHO has not yet occurred.		#comment
24	Verify the collection verification does not look for Collection C4 in the listing returned from ECHO. This can be verified by both log inspection and by verifying that no error is thrown or corrective action enqueued in response to the collection being missing from ECHO.		
25	V-2 Verify that the granule verification in S-4 does not contain the granule inserted in S-2, as its export to ECHO has not yet occurred.		#comment
26	Verify the granule verification does not look for granule g7 in the listing returned from ECHO. This can be verified by both log inspection and by verifying that no error is thrown or corrective action enqueued in response to the granule being missing from ECHO.		
27	V-3 Verify that the Collection verification in S-3 encounters a discrepancy due to the collection deleted in S-2, but that this does not result in an error due to the delete not yet having been exported. Verify that this is noted in the log.		#comment
28	Verify that the bmgt logs indicate the discrepancy about Collection C3 being deleted		
29	Verify that the log indicates that the discrepancy of collection C3 can be		

#	Action	Expected Result	Notes
	ignored.		
30	Verify that no corrective export is enqueued to repair the discrepancy of collection C3		
31	V-4 Verify that the Granule verification in S-3 encounters a discrepancy due to the granule deleted in S-2, but that this does not result in an error due to the delete not yet having been exported. Verify that this is noted in the log.		#comment
32	Verify that the bmgt logs indicate the discrepancy about granule g3 being deleted.		
33	Verify that the log indicates that the discrepancy of granuke g3 can be ignored.		
34	Verify that no corrective export is enqueued to repair the discrepancy of granule g3		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			4 collections with science granules					

EXPECTED RESULTS:

	V	610	1	Verify that the collection verification in S-3 does not contain the collection inserted in S-2, as its export to ECHO has not yet occurred.		
	V	610	2	Verify that the granule verification in S-4 does not contain the granule inserted in S-2, as its export to ECHO has not yet occurred.		
	V	610	3	Verify that the Collection verification in S-3 encounters a discrepancy due to the collection deleted in S-2, but that this does not result in an error due to the delete not yet having been exported. Verify that this is noted in the log.		
	V	610	4	Verify that the Granule verification in S-3 encounters a discrepancy due to the granule deleted in S-2, but that this does not result in an error due to the delete not yet having been exported. Verify that this is noted in the log.		

1.87 Short Form Verification - Errors (ECS-ECSTC-86)

DESCRIPTION:

	S	620	1	**[Short Form Verification - Errors]** Configure BMGT to not automatically export corrective actions		
	S	620	2	Manually delete 1 granule and 1 collection from ECHO such that SDPS contains these items and ECHO does not.		
	S	620	3	Manually add to ECHO 1 collection and 1 granule (which belongs to a valid collection) such that ECHO contains these items and SDPS does not.		
	S	620	4	Perform a collection short form verification		
	S	620	5	Perform a granule short form verification (for at least the collection containing the granules added/removed in S-2 and S-3).		
	S	620	6	Perform a corrective re-export.		
	S	620	7	Configure BMGT to automatically export corrective actions		
	S	620	8	Repeat steps S-2 through S-5		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	ECHO REST API documentation: https://testbed.echo.nasa.gov/catalog-rest/ingest-docs/ https://testbed.echo.nasa.gov/catalog-rest/catalog-docs/ https://testbed.echo.nasa.gov/echo-rest/docs/		#comment
3	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database.		

#	Action	Expected Result	Notes
5	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/620.		
6	Ensure collections C1..C4 are installed. E.g., ensure the DPL Ingest GUI shows the collections as configured datatypes.		
7	Ensure ECHO has granule g1.		
8	Ensure ECHO has collection C2.		
9	Ensure ECHO has granule g3.		
10	Ensure ECHO has collection C4.		
11	Ensure collections C1..C4 are enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where (shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>) or (shortname = '<C1 SHORTNAME>' and versionid = <C1_VERSIONID>) ... or (shortname = '<C4_SHORTNAME>' and versionid = <C4_VERSIONID>)		
12	Ensure BMGT is configured with a reachable email address. Confirm or set the value of BMGT.EmailLogger.To either in the database (bg_configuration_property) or in the BMGT GUI "BMGT Configuration" page.		
13	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
14	Setup		#comment
15	S-1 Configure BMGT to not automatically export corrective actions		#comment
16	In the BMGT GUI "BMGT Configuration" tab, set BMGT.Manual.ShortVer.ReqStatus = "BLOCKED"		
17	S-2 Manually delete 1 granule and 1 collection from ECHO such that SDPS contains these items and ECHO does not.		#comment
18	Use the ECHO REST API to granule g1 from ECHO: curl -k -XDELETE -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/granules/<GRANULE_UR>"		
19	Use the ECHO REST API to delete collection C2 from ECHO: curl -k -XDELETE -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/datasets/<DATASET_ID>" DATASET_ID is the urlencoded longname versionid.		

#	Action	Expected Result	Notes
20	Enable collection C2 in the bg_collection_configuration table. Deleting a collection also disables it in the table.		
21	S-3 Manually add to ECHO 1 collection and 1 granule (which belongs to a valid collection) such that ECHO contains these items and SDPS does not.		#comment
22	Use the ECHO REST API to get granule g3's metadata from ECHO: curl -k -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" -d @<METADATA_FILE_PATH> "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/granules/<GRANULE_UR>" > granule_g3.xml		
23	Physically delete granule g3 from SDPS: BulkDelete granule g3. Unpublish granule g3. Run DeletionCleanup, choosing the most destructive option at every opportunity.		
24	Use the ECHO REST API to add granule g3 to ECHO: curl -k -XPUT -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" -d @/path/to/granule_g3.xml "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/granules/<GRANULE_UR>"		
25	Use the ECHO REST API to get collection C4's metadata from ECHO: curl -k -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/datasets/<DATASET_ID>" > collection_C4.xml		
26	Use the ESDT maintenance GUI to delete collection C4.		
27	Use the ECHO REST API to add collection C4 to ECHO: curl -k -XPUT -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" -d @/path/to/collection_C4.xml "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/datasets/<DATASET_ID>"		
28	S-4 Perform a collection short form verification		#comment
29	./EcBmBMGTManualStart <MODE> --short --metc		
30	S-5 Perform a granule short form verification (for at least the collection containing the granules added/removed in S-2 and S-3).		#comment
31	./EcBmBMGTManualStart <MODE> --short --metg -c		

#	Action	Expected Result	Notes
	<C1_SHORTNAME>.<C1_VERSIONID>		
32	V-1 Verify that the short form verification log lists the following errors after the short verifications in S-4 and S-5 due to the actions in S-2 and S-3: a) 1 granule in ECHO and not in AIM b) 1 collection in ECHO and not in AIM c) 1 granule in AIM but not in ECHO d) 1 collection in AIM but not in ECHO.		#comment
33	Verify that the verification log lists granule g1 in AIM and not in ECHO		
34	Verify that the verification log lists Collection C2 in AIM not in ECHO		
35	Verify that the verification log lists granule g3 in ECHO and not in AIM		
36	Verify that the verification log lists Collection C4 in ECHO and not in AIM		
37	V-2 Verify that an email is sent to the configured notification email address listing the discrepancies noted in V-1. Verify that the email indicates what type and the number of discrepancies encountered, and that it indicates that a corrective export is needed to repair the discrepancies.		#comment
38	Verify that an email is sent to the configured email address.		
39	Verify that the email lists the four discrepancies.		
40	Verify that the email indicates a corrective export is needed.		
41	V-3 Verify that 4 export requests are added to the export request queue pertaining to the 2 granule and 2 collection discrepancies. Verify that the requests are designated as having originated from a short form verification. Verify that these export requests are in the ‘blocked’ state and are not automatically processed.		#comment
42	Verify via either the database or the BMGT GUI that 4 export requests were added to the export queue.		
43	Verify that the requests indicate they originated from short term verification. This is done via the batchId of the export requests, which matches the batch ID of a SHORT operation.		
44	Verify that the added requests are in the 'BLOCKED' state and will not be automatically processed.		
45	Verify that the added requests are in the CORR queue.		
46	S-6 Perform a corrective re-export.		#comment
47	EcBmBMGTManualStart <MODE> --corrective		
48	V-4 Verify that after performing the corrective re-export in S-6, the blocked requests are processed and complete successfully.		#comment
49	Verify that the TCP proxy log shows an HTTP PUT request for granule g1 was exported and that ECHO responded with a success code (200 or 201).		

#	Action	Expected Result	Notes
50	Verify that the TCP proxy log shows an HTTP PUT request for collection C2 was exported and that ECHO responded with a success code (200 or 201).		
51	Verify that the TCP proxy log shows an HTTP DELETE request for granule g3 was exported and that ECHO responded with a success code (204).		
52	Verify that the TCP proxy log shows an HTTP DELETE request for collection C4 was exported and that ECHO responded with a success code (204).		
53	S-7 Configure BMGT to automatically export corrective actions		#comment
54	In the BMGT GUI "BMGT Configuration" tab, set BMGT.Manual.ShortVer.ReqStatus = "PENDING"		
55	S-8 Repeat steps S-2 through S-5		#comment
56	S-2 Manually delete 1 granule and 1 collection from ECHO such that SDPS contains these items and ECHO does not.		#comment
57	Use the ECHO REST API to granule g1 from ECHO: curl -k -XDELETE -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/granules/<GRANULE_UR>"		
58	Use the ECHO REST API to delete collection C2 from ECHO: curl -k -XDELETE -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/datasets/<DATASET_ID>"		
59	S-3 Manually add to ECHO 1 collection and 1 granule (which belongs to a valid collection) such that ECHO contains these items and SDPS does not.		#comment
60	Use the ECHO REST API to add granule g3 to ECHO: curl -k -XPUT -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" -d @/path/to/granule_g3.xml "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/granules/<GRANULE_UR>"		
61	Use the ECHO REST API to add collection C4 to ECHO: curl -k -XPUT -H "Echo-Token:<TOKEN>" -H "Content-Type:application/xml" -d @/path/to/collection_C4.xml "https://testbed.echo.nasa.gov/catalog-rest/providers/<PROVIDER>/datasets/<DATASET_ID>"		
62	S-4 Perform a collection short form verification		#comment

#	Action	Expected Result	Notes
63	./EcBmBMGTManualStart <MODE> --short --metc		
64	S-5 Perform a granule short form verification (for at least the collection containing the granules added/removed in S-2 and S-3).		#comment
65	./EcBmBMGTManualStart <MODE> --short --metg -c <C1_SHORTNAME>.<C1_VERSIONID>		
66	V-5 After re-performing the verification in S-8, ,verify that the short form verification log lists the following errors after the short verifications in S-4 and S-5 due to the actions in S-2 and S-3: a) 1 granule in ECHO and not in AIM b) 1 collection in ECHO and not in AIM c) 1 granule in AIM but not in ECHO d) 1 collection in AIM but not in ECHO.		#comment
67	Verify that the verification log lists granule g1 in AIM and not in ECHO.		
68	Verify that the verification log lists collection C2 in AIM not in ECHO.		
69	Verify that the verification log lists granule g3 in ECHO and not in AIM.		
70	Verify that the verification log lists collection C4 in ECHO and not in AIM.		
71	V-6 After re-performing the verification in S-8, verify that an email is sent to the configured notification email address listing the discrepancies noted in V-1. Verify that the email indicates what type and the number of discrepancies encountered, and that it indicates that a corrective export is needed to repair the discrepancies.		#comment
72	Verify that an email is sent to the configured email address.		
73	Verify that the email lists the four discrepancies.		
74	Verify that the email indicates a corrective export is needed.		
75	V-7 Verify that without any manual intervention, the queued corrective requests are processed and exported to ECHO.		#comment
76	Verify that the TCP proxy log shows an HTTP PUT request for granule g1 was exported and that ECHO responded with a success code (200 or 201).		
77	Verify that the TCP proxy log shows an HTTP PUT request for collection C2 was exported and that ECHO responded with a success code (200 or 201).		
78	Verify that the TCP proxy log shows an HTTP DELETE request for granule g3 was exported and that ECHO responded with a success code (204).		
79	Verify that the TCP proxy log shows an HTTP DELETE request for collection C4 was exported and that ECHO responded with a success code (204).		
80	V-8 Verify that the BMGT GUI provides statistics on the number of discrepancies found by each short form verification attempt, and whether these discrepancies were automatically repaired.		#comment

#	Action	Expected Result	Notes
81	Visit the BMGT GUI "Export Requests" tab.		
82	Optionally apply a date range filter.		
83	Select the "Batch Job Summary" sub tab.		
84	Find the batch associated with the short form verifications performed above (get the batch ID from manual driver log).		
85	Verify that the status of the operation is shown, including the number of discrepancies found and number of successful verifications.		
86	Verify that the summary for the short verification batch shows the total number of items (granules and collections) included in the operation.		
87	Verify that the summary for the short verification batch shows the number of discrepancies found in the short verification operation (under the warning column). This should match the number of errors listed in the email above.		
88	Verify that the summary for the short verification batch shows the number of granules or collections skipped in the short verification operation (under the skipped column).		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 collection (C1) with 1 granule (g1)					
			1 collection (C2)					
			1 collection (C3) with 1 granule (g3)					
			1 collection (C4)					

EXPECTED RESULTS:

	V	620	1	Verify that the short form verification log lists the following errors after the short verifications in S-4 and S-5 due to the actions in S-2 and S-3: a) 1 granule in ECHO and not in AIM b) 1 collection in ECHO and not in AIM c) 1 granule in AIM but not in ECHO		

				d) 1 collection in AIM but not in ECHO.		
	V	620	2	Verify that an email is sent to the configured notification email address listing the discrepancies noted in V-1. Verify that the email indicates what type and the number of discrepancies encountered, and that it indicates that a corrective export is needed to repair the discrepancies.		
	V	620	3	Verify that 4 export requests are added to the export request queue pertaining to the 2 granule and 2 collection discrepancies. Verify that the requests are designated as having originated from a short form verification. Verify that these export requests are in the 'blocked' state and are not automatically processed.		
	V	620	4	Verify that after performing the corrective re-export in S-6, the blocked requests are processed and complete successfully.		
	V	620	5	After re-performing the verification in S-8, ,verify that the short form verification log lists the following errors after the short verifications in S-4 and S-5 due to the actions in S-2 and S-3: a) 1 granule in ECHO and not in AIM b) 1 collection in ECHO and not in AIM c) 1 granule in AIM but not in ECHO d) 1 collection in AIM but not in ECHO.		
	V	620	6	After re-performing the verification in S-8, verify that an email is sent to the configured notification email address listing the discrepancies noted in V-1. Verify that the email indicates what type and the number of discrepancies encountered, and that it indicates that a corrective export is needed to repair the discrepancies.		
	V	620	7	Verify that without any manual intervention, the queued corrective requests are processed and exported to ECHO.		
	V	620	8	Verify that the BMGT GUI provides statistics on the number of discrepancies found by each short form verification attempt, and whether these discrepancies were automatically repaired.		

1.88 Connecting to ECHO (ECS-ECSTC-87)

DESCRIPTION:

	S	630	1	**[Connecting to ECHO]** Go to the BMGT GUI without logging in as a privileged user. View and attempt to modify the ECHO connection information (ECHO URL, username, password, provider, etc.)		

	S	630	2	Log in to the BMGT GUI as a privileged user and update the ECHO connection information (ECHO URL, username, password, provider, etc.). Save the new configuration values. It may be useful to change to incorrect configuration, so it is easier to tell when the new values are being used.		
	S	630	3	Ensure that there are pending export requests for BMGT to work off.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Verify configurations - config files, properties files or database settings validate correctly for mode, host, and application		
3	Verify database connections to ecs connected to ecs db successfully in the configured mode		
4	Verify ECHO REST API service connections to ECHO connected to ECHO REST API successfully		
5	Verify collections under /sotestdata/DROP_802/BE_82_01/630 with ECS metadata		
6	Ensure collection C1 has been installed in the mode.		
7	Ensure collection C1 is enabled for Collection and Granule Export.		
8	Assume <user2> as a regular user without update privileges (default, read-only).		
9	Assume <user1> as a privileged user with configuration update privileges (password required; write access).		
10	Assume <EchoURL1> and <EchoURL2> are two Mock ECHO connections		
11	Ensure a mock ECHO or TCP proxy is configured to capture and log BMGT requests.		
12	Setup		#comment
13	S-1 Go to the BMGT GUI without logging in as a privileged user. View and attempt to modify the ECHO connection information (ECHO URL, username, password, provider, etc.)		#comment
14	Login to BMGT GUI as <user2>, without a password.		
15	On the BMGT Configuration tab, attempt to change the value of the property BMGT.Exporter.IngestClient.URL to a different URL, and save the		

#	Action	Expected Result	Notes
	configuration.		
16	V-1 Verify that in S-1, the GUI allows you to view the ECHO connection configuration, but prevents its update. Verify that the password is not shown in plain text.		#comment
17	Verify that user2 cannot update the ECHO connection configuration information		
18	Verify that when entering the ECHO password, the password is not displayed in plain text		
19	S-2 Log in to the BMGT GUI as a privileged user and update the ECHO connection information (ECHO URL, username, password, provider, etc.). Save the new configuration values. It may be useful to change to incorrect configuration, so it is easier to tell when the new values are being used.		#comment
20	Login to to the BMGT GUI with a privileged password.		
21	On the BMGT Configuration tab, attempt to change the value of the property BMGT.Exporter.IngestClient.URL to a different URL, and save the configuration.		
22	V-2 Verify that in S-2, the GUI allows you to both view and update the ECHO connection configuration. Verify that the password is never shown in plain text, even when it is being entered.		#comment
23	Verify that user1 can update the ECHO connection configuration information		
24	Verify that when entering the ECHO password, the password is not displayed in plain text		
25	S-3 Ensure that there are pending export requests for BMGT to work off.		#comment
26	Enable Automatic Export		
27	Ingest granules g1 and g2 into Collection C1 (or alternatively perform a manual export for 2 granules and a collection)		
28	Verify that the requests are shown pending on the GUI		
29	V-3 Verify that any exports which are picked up by BMGT and begin generation after the new values are saved (plus a reasonable lag time), are exported using the new configuration.		#comment
30	Verify that the TCP proxy on ECHO URL1 does not show any HTTP PUT requests for granule g1 and g2		
31	Verify that the TCP proxy on ECHO URL2 (if it exists) shows 2 HTTP PUT requests for granule g1 and g2		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			1 Collection with 2 science granules					

EXPECTED RESULTS:

	V	630	1	Verify that in S-1, the GUI allows you to view the ECHO connection configuration, but prevents its update. Verify that the password is not shown in plain text.		
	V	630	2	Verify that in S-2, the GUI allows you to both view and update the ECHO connection configuration. Verify that the password is never shown in plain text, even when it is being entered.		
	V	630	3	Verify that any exports which are picked up by BMGT and begin generation after the new values are saved (plus a reasonable lag time), are exported using the new configuration.		

1.89 Monitor System (ECS-ECSTC-88)

DESCRIPTION:

	S	640	1	**[Monitor System]** Bring down the BMGT server.		
	S	640	2	View the BMGT server status in Hyperic.		
	S	640	3	Pause one of the BMGT queues.		
	S	640	4	Bring up the BMGT server.		
	S	640	5	View the BMGT server status in Hyperic.		
	S	640	6	View the BMGT queue status in Hyperic.		

	S	640	7	Resume the BMGT queues.		
	S	640	8	View the BMGT queue status in Hyperic.		
	S	640	9	Throughout the remaining steps, continue to monitor BMGT status in Hyperic.		
	S	640	10	Allow BMGT to execute normally, enqueing automatic export requests on a regular interval. Cause events to be added to the AIM event queue which will subsequently be picked up. In essence, simulate normal operation of BMGT at a DAAC.		
	S	640	11	Perform a manual export.		
	S	640	12	Perform a long form verification export.		
	S	640	13	During normal BMGT processing as specified in S-10, cause some exports to result in errors and warnings.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a Hyperic agent is running on the BMGT server host. Log in to the Hyperic GUI as hqadmin. Click Resources (Tab on top of the page). Click on "Platforms(X)" in the table headers. In the platform list, the host should be "Available"; a green tick icon should be shown in the "Availability" column.		
3	Ensure BMGT resources are set up in the Hyperic. Click on "Resources" tab. Click on "Services(X)", where "X" is a number indicating the total number of services that the platform hosts. From the search menu (above "Platform(X)|Servers(X)|Services(X)... ", select "dataprovider_MODE" from the dropdown list, then click the green play button on the most right. Locate the service you need and click on its name. You will be directed to the service detail page.		
4	Setup and Verification		#comment
5	S-1 Bring down the BMGT server.		#comment

#	Action	Expected Result	Notes
6	Stop BMGT: EcBmBMGTAppStop $MODE		
7	S-2 View the BMGT server status in Hyperic.		#comment
8	Go to the service detail page (refer to Pre-Conditions).		
9	V-1 Verify that in S-2, Hyperic indicates that the BMGT server is down.		#comment
10	Verify that after 1-5 minutes, on the service detail page, the green dots on top of all charts go from green to red.		
11	S-3 Pause one of the BMGT queues.		#comment
12	Log in to the BMGT GUI to gain read-write access.		
13	On the "System Status" page, pause a queue.		
14	S-4 Bring up the BMGT server.		#comment
15	Start BMGT: EcBmBMGTAppStart $MODE		
16	S-5 View the BMGT server status in Hyperic.		#comment
17	Go to the service detail page (refer to Pre-Conditions).		
18	V-2 Verify that in S-5, Hyperic indicates that the BMGT server is up.		#comment
19	Verify that after 1-5 minutes, on the service detail page, the red dots turn green.		
20	S-6 View the BMGT queue status in Hyperic.		#comment
21	Go to the service detail page (refer to Pre-Conditions).		
22	V-3 Verify that in S-6, Hyperic indicates that one of the queues is pasued.		#comment
23	Verify that once the BMGT server is "available" in Hyperic, sizes of each queue are displayed on the charts below the dotted bar. The size of the queue that has been paused should keep increasing (rising), potentially.		
24	S-7 Resume the BMGT queues.		#comment
25	Log in to the BMGT GUI to gain read-write access.		
26	On the "System Status" page, resume the paused queue.		
27	S-8 View the BMGT queue status in Hyperic.		#comment
28	Go to the service detail page (refer to Pre-Conditions).		
29	V-4 Verify that in S-8, Hyperic indicates that all queues are active.		#comment
30	Verify the queue size drops after the queue is resumed. Note that, the size of the queue might increase or drop over the time.		
31	V-5 Verify that as events are processed by BMGT, the automatic export queue size is displayed in Hyperic.		#comment

#	Action	Expected Result	Notes
32	Verify that the service detail page displays a chart for the automatic export queue size.		
33	S-9 Throughout the remaining steps, continue to monitor BMGT status in Hyperic.		#comment
34	S-10 Allow BMGT to execute normally, enqueing automatic export requests on a regular interval. Cause events to be added to the AIM event queue which will subsequently be picked up. In essence, simulate normal operation of BMGT at a DAAC.		#comment
35	Ingest 10 granules into the public data pool, g1..g10.		
36	Unpublish granule g2: EcDlUnpublishStart.pl -mode $MODE -g $GRANULEID		
37	Logically delete granule g3: EcDsBulkDelete.pl -mode $MODE -physical -geoidfile /path/to/geoids.txt		
38	S-11 Perform a manual export.		#comment
39	Request manual export of granule g4: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		
40	V-6 Verify that after the manual export is initiated in S-11, the resulting requests are indicated in the manual queue size as displayed in Hyperic.		#comment
41	Verify that the service detail page displays a chart for the manual queue size.		
42	S-12 Perform a long form verification export.		#comment
43	Request a long form verification of granule g5: EcBmBMGTManualStart $MODE --long --metg -g $GRANULEID		
44	V-7 Verify that after the verification export is initiated in S-12, the resulting requests are indicated in the verification queue size as displayed in Hyperic.		#comment
45	Verify that on the service detail page the manule export queue chart indicates new requests for the long form verification.		
46	S-13 During normal BMGT processing as specified in S-10, cause some exports to result in errors and warnings.		#comment
47	Use the ECHO REST API to delete granule g6 from ECHO: curl -k -XDELETE -H "Echo-Token:${TOKEN}" -H "Content-Type:application/xml" "https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDER}/granules/${GRANULEUR}"		FIXME Explain how to cause errors and warnings. E.g., remove a granule from ECHO, then export a delete request; make a granule's metadata file malformed XML; mangle a granule's

#	Action	Expected Result	Notes
			metadata file, so it gets exported, but fails ECHO ingest.
48	Logically delete granule g6, and allow BMGT to export the deletion: EcDsBulkDelete.pl -mode $MODE -physical -geoidfile /path/to/geoids.txt		
49	Find granule g7's metadata file in the small file archive: select p.path || '/' || f.archivemetfilename from ammetadatafile f join dsmdxmlpath p on f.archivepathid = p.archivepathid where f.granuleid = $GRANULEID		
50	Move the file to a new name to restore later, then copy it back to the original name.		
51	Edit the file, and remove an opening XML tag.		
52	Find granule g8's metadata file in the small file archive: select p.path || '/' || f.archivemetfilename from ammetadatafile f join dsmdxmlpath p on f.archivepathid = p.archivepathid where f.granuleid = $GRANULEID		
53	Move the file to a new name to restore later, then copy it back to the original name.		
54	Edit the file, and change the RangeBeginningDate or RangeEndingDate to be outside the collection's date range (e.g., set year to 1900).		
55	Request manual export of granules g7 and g8: EcBmBMGTManualStart $MODE --metg -g $g7_GRANULEID,$g8_GRANULEID		
56	V-8 Verify that the error and warning counts displayed in Hyperic reflect the errors and warnings in S-13.		#comment
57	Verify that the service detail page displays charts for alerts/errors, open interventions, etc.		
58	Cleanup		#comment
59	Restore the g7 and g8 granule XML files from S-13.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

EXPECTED RESULTS:

	V	640	1	Verify that in S-2, Hyperic indicates that the BMGT server is down.		
	V	640	2	Verify that in S-5, Hyperic indicates that the BMGT server is up.		
	V	640	3	Verify that in S-6, Hyperic indicates that one of the queues is pasued.		
	V	640	4	Verify that in S-8, Hyperic indicates that all queues are active.		
	V	640	5	Verify that as events are processed by BMGT, the automatic export queue size is displayed in Hyperic.		
	V	640	6	Verify that after the manual export is initiated in S-11, the resulting requests are indicated in the manual queue size as displayed in Hyperic.		
	V	640	7	Verify that after the verification export is initiated in S-12, the resulting requests are indicated in the manual queue size as displayed in Hyperic.		
	V	640	8	Verify that the error and warning counts displayed in Hyperic reflect the errors and warnings in S-13.		

1.90 Artifact Cleanup (ECS-ECSTC-89)

DESCRIPTION:

	S	650	1	**[Artifact Cleanup]** Configure the cleanup interval for BMGT to a known value and bounce BMGT.		
	S	650	2	Cause a number of successful and failed exports to go through the BMGT system.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties		

#	Action	Expected Result	Notes
	files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/650.		
5	Ensure collections C1..C5 are installed. E.g., ensure the DPL Ingest GUI shows C1..C5 as configured datatypes.		
6	Ensure collections C1..C5 are enabled for collection and granule export: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where (shortname = '<C1_SHORTNAME>' and versionid = <C1_VERSIONID>) or (shortname = '<C2_SHORTNAME>' and versionid = <C2_VERSIONID>) or (shortname = '<C3_SHORTNAME>' and versionid = <C3_VERSIONID>) or (shortname = '<C4_SHORTNAME>' and versionid = <C4_VERSIONID>) or (shortname = '<C5_SHORTNAME>' and versionid = <C5_VERSIONID>)		
7	Ensure collections C1..C5 exist in ECHO.		
8	Ensure all test granules exist in ECHO.		
9	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 Configure the cleanup interval for BMGT to a known value and bounce BMGT.		#comment
12	Configure monitor interval to 5 minutes: update bg_configuration_property set propertyvalue = 300000 where propertyname = 'BMGT.Monitor.pollingFrequency'		
13	Configure cleanup age to 10 minutes: update bg_configuration_property set propertyvalue = 600000 where propertyname = 'BMGT.Monitor.purgeOlderThan'		
14	S-2 Cause a number of successful and failed exports to go through the BMGT system.		#comment
15	EcBmBMGTAppStart <MODE>		
16	Ingest 10 granules each g1..g40 from collections C1..C4.		
17	Ingest granules g41..g50 from collection C5.		
18	Save the granule IDs for granules g41..g50 to a text file, granuleids.txt.		
19	Modify the metadata in granules g41..g50 in the small file archive to have various errors, such as duplicate URLs, missing tags in the XML, and an EndTime less than the StartTime.		

#	Action	Expected Result	Notes
20	Manually export granules g41..g50: ./EcBmBMGTManualStart <MODE> --metg -gf /path/to/granuleids.txt		
21	Wait for granules g1..g40 to be exported.		
22	Wait for export attempts for granules g41..g50 to complete.		
23	After the exports have been processed, cancel some of the exports which have failed and been blocked.		
24	Verification		#comment
25	V-1 Verify that all artifacts (on filesystem and in database) for all successful and failed exports are removed after the configured time interval has passed.		#comment
26	Verify that all the exports for granules g1..g40 complete successfully.		
27	Verify that the manual exports for g41..g50 fail.		
28	Verify that after 10 minutes, the requests in terminal states (SUCCESS, CANCELLED) related to g1..g40 are removed from the database (bg_export_request, bg_export_activity, bg_export_error, and associated tables), and no longer show up in the GUI.		
29	Verify that after 10 minutes requests in the BLOCKED state remain in the database. BLOCKED is not a terminal state; such requests may be retried.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
			5 collections (C1..C5) with 10 granules each (g1..g50)				/sotestdata/DROP_802/BE_82_01/Criteria/650	

EXPECTED RESULTS:

	V	650	1	Verify that all artifacts (on filesystem and in database) for all successful and failed exports are removed after the configured time interval has passed.		

1.91 Export Takes Too Long (ECS-ECSTC-90)

DESCRIPTION:

	S	660	1	**[Export Takes Too Long]** In the BMGT GUI configure: a) Time limit between queueing an export request and beginning the export to ECHO. b) Timeout value for ingest requests to ECHO. c) Delay time for retrying automatic exports. d) Email to send alerts to. Also configure BMGT to automatically retry on errors.		
	S	660	2	With the BMGT Dispatcher down, but automatic driver up, perform an action (ingest, update, delete) which results in the queuing of a request for the export of either granule or collection metadata.		
	S	660	3	Cause ECHO, or an ECHO stand in, to take a long amount of time (greater than the Ingest timeout value) to process ingest requests.		
	S	660	4	Allow enough time to pass such that the configured time limit between queuing and beginning export is exceeded.		
	S	660	5	Start the BMGT DIspatcher.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is current and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database.		
4	Ensure test data is available under /sotestdata/DROP_802/BE_82_01/Criteria/660/NISE.004.		
5	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	Ensure collection C1 has been installed in the mode. E.g., the DPL Ingest GUI shows C1 as a configured data type.		
8	Ensure collection C1 is enabled for Collection and Granule Export.		

#	Action	Expected Result	Notes
9	S-1 In the BMGT GUI configure: a) Time limit between queueing an export request and beginning the export to ECHO. b) Timeout value for ingest requests to ECHO. c) Delay time for retrying automatic exports. d) Email to send alerts to. Also configure BMGT to automatically retry on errors.		#comment
10	On the BMGT GUI Configuration GUI		
11	BMGT.Monitor.staleAfter = 600000(5min)		
12	BMGT.Exporter.IngestClient.SoTimeout= 60000 (1 min)		
13	BMGT.Dispatcher.RetryRequestWait= 300000(5 min)		
14	Configure BMGT.EmailLogger.To = <testEmail>		
15	Enable automatic retry on errors - Set BMGT.ResponseHandler.MaxRetryCount > 0		
16	BMGT.ResponseHandler.Monitor.EmailTimeOut = 60000 (1min)		
17	BMGT.ResponseHandler.Monitor.MaxBlockCount = 0		
18	V-1 Ensure that it is possible to configure the items listed in S-1.		#comment
19	Ensure that the bg_configuration_property table shows the updated configuration values.		
20	S-2 With the BMGT Dispatcher down, but automatic driver up, perform an action (ingest, update, delete) which results in the queuing of a request for the export of either granule or collection metadata.		#comment
21	Run EcBmBMGTDispatcherStop <MODE>		
22	Start Automatic Driver: EcBmBMGTAutoStart <MODE>		
23	Start BMGT Monitor: EcBmBMGTMonitor <MODE>		
24	Ingest granule g1.		
25	Record the queue start time t_start_time.		
26	S-3 Cause ECHO, or an ECHO stand in, to take a long amount of time (greater than the Ingest timeout value) to process ingest requests.		#comment
27	Stop the TCP proxy, then start it with the 'pause' argument.		
28	S-4 Allow enough time to pass such that the configured time limit between queuing and beginning export is exceeded.		#comment
29	Wait for 5 minutes after the request has been queued.		
30	S-5 Start the BMGT DIspatcher.		#comment
31	Resume Dispatcher: EcBmBMGTDispatcherStart <MODE>		
32	V-2 After starting the Dispatcher in S-5, verify that a warning email is sent to the configured address informing that the export request was on the queue for longer than the configured limit.		#comment

#	Action	Expected Result	Notes
33	Verify that an email is set to <testEmail>.		
34	Verify that email indicates the the export request enqueued longer than the configured limit.		
35	V-3 Verify that the export is automatically retried once the configured retry interval has passed.		#comment
36	Verify that the export request is retried every 5 minutes.		
37	V-4 Verify that the export to ECHO or an ECHO stand-in is terminated and put in an error state when the retry count limit is exceeded.		#comment
38	Allow the export to retry BMGT.ResponseHandler.MaxRetryCount times.		
39	Verify that the request is put in the BLOCKED state		Email is not sent immediately after a single export is terminated. But is retried. If retry fails BMGT.ResponseHandler.MaxRetryCount many times, the request is blocked. And an email is sent about blocked requests if number of blocked requests exceed BMGT.ResponseHandler.Monitor.MaxBlockCount or if it has been longer than BMGT.ResponseHandler.Monitor.EmailTimeOut since last email is sent.
40	V-5 Verify that when the export is terminated an error email is sent.		#comment
41	Verify that an email is sent when the request is BLOCKED.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
660		1 Collection with a few granules	NISE.004				/sotestdata/DROP_802/BE_82_01/Criteria/660/NISE.004	

EXPECTED RESULTS:

	V	660	1	Ensure that it is possible to configure the items listed in S-1.		
	V	660	2	After starting the Dispatcher in S-5, verify that a warning email is sent to the configured address informing that the export request was on the queue for longer than the configured limit.		
	V	660	3	Verify that the export to ECHO or an ECHO stand-in is terminated and put in an error state when the export timeout is exceeded.		
	V	660	4	Verify that when the export is terminated an error email is sent.		
	V	660	5	Verify that the export is automatically retried once the configured retry interval has passed.		

1.92 ECS Collection Additional Metadata[S-02]: ECS GranuleSpatialRepresentation (ECS-ECSTC-99)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		

	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-2 Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than		#comment

#	Action	Expected Result	Notes
	“CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
8	Update the collection BMGT configuration table, setting the value of the S-2 collection's GranuleSpatialRepresentation to an invalid value. E.g., update bg_collection_configuration set GranuleSpatialRepresentation='NONEUCLIDEAN' where ShortName='<SHORT_NAME>' and VersionID=<VERSION_ID>		
9	Verification		#comment
10	V-2 Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		#comment
11	Request the export of the test collection: EcBmBMGTManualStart --mode $MODE --metc --collections ${SHORT_NAME}.${VERSION_ID}		
12	Verify the manual export fails.		
13	Verify the log file records the export failure and the reason for failure.		
14	Verify the log file records the export failure and the reason for failure.		
15	Verify the log file indicates the invalid GranuleSpatialRepresentation value caused the export failure.		
16	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
17	No collection metadata should have been exported.		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		

	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		

	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.93 ECS Collection Additional Metadata[S-03]: ISO CoordinateSystem (ECS-ECSTC-100)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		

	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-3 Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		#comment

#	Action	Expected Result	Notes
8	Set the test collection's CoordinateSystem to any string: update bg_collection_configuration set collectioncoordinatesystem = 'GEODETIC' where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
9	Request a manual export of the S-3 collection: EcBmBMGTManualStart --mode $MODE --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-3 Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		#comment
12	Verify the manual export fails.		
13	Verify the log file records the export failure.		
14	Verify the log file indicates the invalid CoordinateSystem value caused the export failure.		
15	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
16	No collection metadata should have been exported.		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the		

				failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC		

				• Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.94 ECS Collection Additional Metadata[S-04]: ISO GranuleSpatialRepresentation (ECS-ECSTC-101)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has		

				one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-4 Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		#comment
8	update the value of the S-4 collection's GranuleSpatialRepresentation to any string. E.g., update bg_collection_configuration set GranuleSpatialRepresentation='NO_SPATIAL' where ShortName = '$SHORTNAME' and VersionID = $VERSIONID ;		
9	Request a manual export of the S-4 collection: <br		

#	Action	Expected Result	Notes
	/>EcBmBMGTManualStart --mode $MODE --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-4 Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		#comment
12	Verify the manual export fails.		
13	Verify the log file records the export failure.		
14	Verify the log file indicates the invalid GranuleSpatialRepresentation value caused the export failure.		
15	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
16	No collection metadata should have been exported.		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log		

				indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		

	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.95 ECS Collection Additional Metadata[S-05]: backtrack (ECS-ECSTC-102)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the		

				manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the ECHO 10 collection schema is available to validate against (https://api.echo.nasa.gov/ingest/schema/Collection.xsd).		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-5 For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		#comment
8	Ensure each of the test collections has a valid backtrack type configured (one of: AMSR-A, AMSR-E, GLAS 14 Orbit, GLAS Quarter Orbit, GLAS Two Orbit, MISR) and that all valid backtrack types are represented.		
9	Create a text file with one test collection per line: $SHORT_NAME_1.$VERSION_ID_1 $SHORT_NAME_2.$VERSION_ID_2 ... $SHORT_NAME_5.$VERSION_ID_5		
10	Request manual export of all S-5 test collections: EcBmBMGTManualStart --mode $MODE --metc --collectionfile /path/to/S-5_collections.txt		
11	Verification		#comment

#	Action	Expected Result	Notes
12	V-5 Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		#comment
13	Verify the TCP proxy log shows XML metadata was exported for each collection.		
14	Verify a Spatial/OrbitParameters element exists for each collection's metadata, using an xpath utility: xpath '//Spatial/OrbitParameters' collection.xml		
15	Verify each collection's Spatial/OrbitParameters element is correct, according to BE_82_01_AdditionalMetadataDescription.doc. See the comments at the end of this test for the relevant section. Note that MISR granules may exclude the <StartCircularLatitude/> element if it would be empty.		
16	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
17	Verify each collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
18	V-15 For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		#comment
19	Verify, using an xpath utility, that each collection's exported metadata contains a CoordinateSystem element with a valid value (one of 'CARTESIAN', 'GEODETIC'): xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/CoordinateSystem' collection.xml		
20	Verify, using an xpath utility, that each collection's exported metadata contains a GranuleSpatialRepresentation element with a valid value (one of 'CARTESIAN', 'GEODETIC', 'ORBIT', or 'NO_SPATIAL'): xpath '/Collection/Spatial/GranuleSpatialRepresentation' collection.xml		
21	Backtrack orbit metadata (Granule & Collection) from BE_82_01_AdditionalMetadataDescription.doc		#comment
22	ECHO provides the ability to perform backtrack orbit searches, provided that the proper metadata is provided for granules. This metadata is mostly contained in ECS granule and collection metadata, but is not in		#comment

#	Action	Expected Result	Notes
	the proper format. It is either contained in the OrbitCalculatedSpatialDomain hierarchy, PSA values, or must be calculated from these values using standard rules and/or ancilliary files. A number of groups have been established, each with different rules for generating granule and collection backtrack metadata. Each collection may be associated with one of these groups via the Collection Configuration database table. Backtrack orbit metadata is not required.		
23	Valid values for BackTrack Orbit Group are (values in parenthesis refer to alternative names used in legacy code. We should use the more descriptive names below): • GLAS Quarter Orbit (AKA GLAS1) • GLAS Two Orbit (AKA GLAS2) • GLAS 14 Orbit (AKA GLAS3) • AMSR-E (AKA AMSR1) • AMSR-A (AKA AMSR2) • MISR (LARC) • LPDAAC (This was used at one point but from what I can tell is no longer used. Need to verify) The rules for each group are defined below, but first is a list of the variables that the rules reference, and where they are obtained from. Values obtained from OrbitCalculatedSpatialDomainContainer FIRST_ASC_CROSS = the EquatorCrossingLongitude value of the OrbitCalculatedSpatialDomainContainer enclosure with the earliest EquatorCrossingTime. SECOND_ASC_CROSS = the EquatorCrossingLongitude value of the OrbitCalculatedSpatialDomainContainer enclosure with the second earliest EquatorCrossingTime. Values obtained from PSAs TRACK_SEGMENT = The value of the 'Track_Segment' PSA. START_BLOCK = The value of the 'SP_AM_MISR_StartBlock' PSA. END_BLOCK = The value of the 'SP_AM_MISR_EndBlock' PSA. ASC_DESC_FLAG = The value of the ‘AscendingDescendingFlg’ PSA. Values obtained from separate MISRBlockLat.xml file Format: <properties> <property> <Block>180</Block> <FirstEdge>-66.695034</FirstEdge> <LastEdge>-65.502984</LastEdge> </property> … </properties> FIRST_EDGE = The FirstEdge value of the property with Block = START_BLOCK LAST_EDGE = The LastEdge value of the property with Block = END_BLOCK		#comment
24	Group Rules:		#comment
25	GLAS Quarter Orbit Collection: <OrbitParameters> 		#comment

#	Action	Expected Result	Notes
	<SwathWidth>2.0</SwathWidth> <Period>96.7</Period> <InclinationAngle>94.0</InclinationAngle> <NumberOfOrbits>0.25</NumberOfOrbits> <StartCircularLatitude>-50.0</StartCircularLatitude> </OrbitParameters> Granule: TRACK_SEGMENT = 1 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>D</EndDirection> </Orbit> TRACK_SEGMENT = 2 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>D</StartDirection> <EndLat>-50.0</EndLat> <EndDirection>D</EndDirection> </Orbit> TRACK_SEGMENT = 3 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-50.0</StartLat> <StartDirection>D</StartDirection> <EndLat>-50.0</EndLat> <EndDirection>A</EndDirection> </Orbit> TRACK_SEGMENT = 4 <Orbit> <AscendingCrossing>SECOND_ASC_CROSS</AscendingCrossing> <StartLat>-50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		
26	GLAS Single Orbit Collection: <OrbitParameters> <SwathWidth>2.0</SwathWidth> <Period>96.7</Period> <InclinationAngle>94.0</InclinationAngle> <NumberOfOrbits>2.0</NumberOfOrbits> <StartCircularLatitude>50.0</StartCircularLatitude> </OrbitParameters> Granule: <Orbit> 		#comment

#	Action	Expected Result	Notes
	<AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		
27	GLAS 14 Orbit Collection: <OrbitParameters> <SwathWidth>2.0</SwathWidth> <Period>96.7</Period> <InclinationAngle>94.0</InclinationAngle> <NumberOfOrbits>14.0</NumberOfOrbits> <StartCircularLatitude>50.0</StartCircularLatitude> </OrbitParameters> Granule: <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>50.0</StartLat> <StartDirection>A</StartDirection> <EndLat>50.0</EndLat> <EndDirection>A</EndDirection> </Orbit>		#comment
28	AMSR-E Collection: <OrbitParameters> <SwathWidth>1450.0</SwathWidth> <Period>98.88</Period> <InclinationAngle>98.15</InclinationAngle> <NumberOfOrbits>0.5</NumberOfOrbits> <StartCircularLatitude>-90.0</StartCircularLatitude> </OrbitParameters> Granule: ASC_DESC_FLAG = ‘Ascending’ or ‘ASCENDING’ <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-81.8</StartLat> <StartDirection>A</StartDirection> <EndLat>81.8</EndLat> <EndDirection>A</EndDirection> </Orbit> ASC_DESC_FLAG = ‘Descending’ <Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 167.64</AscendingCrossing> <StartLat>81.8</StartLat> <StartDirection>D</StartDirection> <EndLat>-81.8</EndLat> 		#comment

#	Action	Expected Result	Notes
	<EndDirection>D</EndDirection> </Orbit>		
29	AMSR-A Collection: <OrbitParameters> <SwathWidth>1600.0</SwathWidth> <Period>101.0</Period> <InclinationAngle>98.62</InclinationAngle> <NumberOfOrbits>0.5</NumberOfOrbits> <StartCircularLatitude>-90.0</StartCircularLatitude> </OrbitParameters> Granule: ASC_DESC_FLAG = ‘Ascending’ or ‘ASCENDING’ <Orbit> <AscendingCrossing>FIRST_ASC_CROSS</AscendingCrossing> <StartLat>-81.8</StartLat> <StartDirection>A</StartDirection> <EndLat>81.8</EndLat> <EndDirection>A</EndDirection> </Orbit> ASC_DESC_FLAG = ‘Descending’ or ‘DESCENDING’ <Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 167.375</AscendingCrossing> <StartLat>81.8</StartLat> <StartDirection>D</StartDirection> <EndLat>-81.8</EndLat> <EndDirection>D</EndDirection> </Orbit>		#comment
30	MISR Collection: <OrbitParameters> <SwathWidth>400.0</SwathWidth> <Period>98.88</Period> <InclinationAngle>98.3</InclinationAngle> <NumberOfOrbits>1.0</NumberOfOrbits> <StartCircularLatitude /> </OrbitParameters> Granule: EQ_CROSS_LONG > 347.65 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 527.65</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise </EndDirection> </Orbit> EQ_CROSS_LONG < -12.35 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS +		#comment

#	Action	Expected Result	Notes
	192.35</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise </EndDirection> </Orbit> EQ_CROSS_LONG between -12.35 and 347.65 <Orbit> <AscendingCrossing>FIRST_ASC_CROSS - 167.65</AscendingCrossing> <StartLat>FIRST_EDGE</StartLat> <StartDirection>D if START_BLOCK between 19 and 162. A otherwise</StartDirection> <EndLat>LAST_EDGE</EndLat> <EndDirection> D if END_BLOCK between 19 and 162. A otherwise </EndDirection> </Orbit>		
31	LPDAAC Collection: <OrbitParameters> <SwathWidth>1450.0</SwathWidth> <Period>98.88</Period> <InclinationAngle>98.15</InclinationAngle> <NumberOfOrbits>1.0</NumberOfOrbits> <StartCircularLatitude /> </OrbitParameters> Granule: NO ORBIT METADATA		#comment

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		

	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping:		

				• SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.96 ECS Collection Additional Metadata[S-06]: no backtrack (ECS-ECSTC-103)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		

	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-6 Request the export of metadata for a collection which is not configured for backtrack metadata.		#comment
8	Ensure collection C6 does not have a configured orbitGroup: update bg_collection_configuration set orbitGroup = NULL where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		

#	Action	Expected Result	Notes
9	Request manual export of the S-6 collection: EcBmBMGTManualStart <MODE> --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-6 Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		#comment
12	Verify, using an xpath utility, that the collection's exported metadata contains no Spatial/OrbitParameters element: xpath '//Spatial/OrbitParameters' collection.xml		
13	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
14	Verify S-6 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
15	V-15 For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		#comment
16	Verify for each collection that if it has a Spatial element, its exported metadata also contains a CoordinateSystem element with a valid value (one of 'CARTESIAN', 'GEODETIC'): xpath '/CollectionMetaDataFile/Collections/Collection/Spatial' xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/CoordinateSystem' collection.xml		
17	Verify for each collection that if it has a Spatial element, its exported metadata also contains a GranuleSpatialRepresentation element with a valid value (one of 'CARTESIAN', 'GEODETIC', 'ORBIT', or 'NO_SPATIAL'): xpath '/Collection/Spatial' collection.xml xpath '/Collection/Spatial/GranuleSpatialRepresentation' collection.xml		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it		

				is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		

	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.97 ECS Collection Additional Metadata[S-07]: TwoDCoordinateSystem (ECS-ECSTC-104)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		

	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		

#	Action	Expected Result	Notes
6	Setup		#comment
7	S-7 For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		#comment
8	Ensure that for each of the valid TwoDCoordinateSystem types in EcBmBmgtTwoDCoords.xml, an S-7 test collection is configured for that type select twodcoordinatesystem from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
9	Create a text file listing each of the S-7 collections, one per line: <SHORT_NAME_1>.<VERSION_ID_1> <SHORT_NAME_2>.<VERSION_ID_2> ... <SHORT_NAME_N>.<VERSION_ID_N>		
10	Request a manual export: EcBmBMGTManualStart <MODE> --metc --collectionfile /path/to/S-7_collections.txt		
11	Verification		#comment
12	V-7 Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		#comment
13	Verify XML metadata was exported once for each S-7 collection.		
14	Verify each collection's exported metadata has a TwoDCoordinateSystems/TwoDCoordinateSystem element: xpath '//TwoDCoordinateSystems/TwoDCoordinateSystem' collection.xml		
15	Get each collection's TwoDCoordinateSystem type: select twodcoordinatesystem from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
16	Use an xpath utility to get the TwoDCoordinateSystem block from each collection's exported XML: xpath '//TwoDCoordinateSystems/TwoDCoordinateSystem' collection.xml		
17	Verify each collection's TwoDCoordinateSystem block matches the block found in /usr/ecs/<MODE>/CUSTOM/data/BMGT/config/EcBmBMGTTwoDCoords.xml for the collection's TwoDCoordinateSystem type. E.g., <br		

#	Action	Expected Result	Notes
	/>MISR: <TwoDCoordinateSystem> <TwoDCoordinateSystemName>MISR</TwoDCoordinateSystemName> <Coordinate1> <MinimumValue>1</MinimumValue> <MaximumValue>233</MaximumValue> </Coordinate1> <Coordinate2> <MinimumValue>1</MinimumValue> <MaximumValue>180</MaximumValue> </Coordinate2> </TwoDCoordinateSystem> MODIS Tile: <TwoDCoordinateSystem> <TwoDCoordinateSystemName>MODIS Tile</TwoDCoordinateSystemName> <Coordinate1> <MinimumValue>0</MinimumValue> <MaximumValue>38</MaximumValue> </Coordinate1> <Coordinate2> <MinimumValue>0</MinimumValue> <MaximumValue>38</MaximumValue> </Coordinate2> </TwoDCoordinateSystem>		
18	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
19	Verify each S-7 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
20	V-15 For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		#comment
21	Verify for each collection that if it has a Spatial element, its exported metadata also contains a CoordinateSystem element with a valid value (one of 'CARTESIAN', 'GEODETIC'): xpath '/Collection/Spatial' collection.xml xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/CoordinateSystem' collection.xml		
22	Verify for each collection that if it has a Spatial element, its exported metadata also contains a GranuleSpatialRepresentation element with a valid value (one of 'CARTESIAN', 'GEODETIC', 'ORBIT', or 'NO_SPATIAL'): xpath '/Collection/Spatial' collection.xml xpath '/Collection/Spatial/GranuleSpatialRepresentation' collection.xml		

TEST DATA:
See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the		

				Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.98 ECS Collection Additional Metadata[S-08]: no TwoDCoordinateSystem (ECS-ECSTC-105)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		

	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID><br		

#	Action	Expected Result	Notes
	/> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-8 Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		#comment
8	Ensure the S-8 test collection has no TwoDCoordinateSystem configured. select twodcoordinatesystem from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
9	Request manual export of the S-8 collection: EcBmBMGTManualStart <MODE> --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-8 Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		#comment
12	Verify XML collection metadata was exported for the S-8 collection.		
13	Verify the collection's exported metadata contains no TwoDCoordinateSystems/TwoDCoordinateSystem element: xpath '//TwoDCoordinateSystems/TwoDCoordinateSystem' collection.xml		
14	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
15	Verify the S-8 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
16	V-15 For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		#comment
17	Use an xpath utility, such as /tools/common/test/BE_82_01/bin/xpath, to find XML elements.		#comment
18	Verify for each collection that if it has a Spatial element, its exported metadata also contains a CoordinateSystem element with a valid value (one		

#	Action	Expected Result	Notes
	of 'CARTESIAN', 'GEODETIC'): xpath '/Collection/Spatial' collection.xml xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/CoordinateSystem' collection.xml		
19	Verify for each collection that if it has a Spatial element, its exported metadata also contains a GranuleSpatialRepresentation element with a valid value (one of 'CARTESIAN', 'GEODETIC', 'ORBIT', or 'NO_SPATIAL'): xpath '/Collection/Spatial' collection.xml xpath '/Collection/Spatial/GranuleSpatialRepresentation' collection.xml		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		

	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.99 ECS Collection Additional Metadata[S-09]: Product Specific Attributes (PSAs) (ECS-ECSTC-106)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-9 Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		#comment
8	Create a text file of all PSA metadata from the original collection ODL metadata.		
9	Request manual export of the S-9 test collection: EcBmBMGTManualStart --mode $MODE --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-9 Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		#comment
12	Verify that XML collection metadata was exported once for the S-9 collection.		
13	Verify the exported collection metadata contains the same number AdditionalAttribute elements as there are PSAs the ECS collection ODL file: xpath '//AdditionalAttributes/AdditionalAttribute' collection.xml		
14	Verify each AdditionalAttribute element includes all the information in the		

#	Action	Expected Result	Notes
	corresponding original PSA by comparing it to the PSA text file created earlier.		
15	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
16	Verify the S-9 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
17	V-15 For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		#comment
18	Use an xpath utility, such as /tools/common/test/BE_82_01/bin/xpath, to find XML elements.		#comment
19	Verify, using an xpath utility, that the collection's exported metadata contains a CoordinateSystem element with a valid value (one of 'CARTESIAN', 'GEODETIC'): xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/CoordinateSystem' collection.xml		
20	Verify, using an xpath utility, that the collection's exported metadata conains a GranuleSpatialRepresentation element with a valid value (one of 'CARTESIAN', 'GEODETIC', 'ORBIT', or 'NO_SPATIAL'): xpath '/Collection/Spatial/GranuleSpatialRepresentation' collection.xml		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		

	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping:		

				• SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.100 ECS Collection Additional Metadata[S-10]: DIF ID (ECS-ECSTC-107)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		

	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-10 Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		#comment
8	In the BMGT GUI collection configuration tab, modify the DIF ID of the collection that has one.		
9	Log out of the BMGT GUI.		
10	Clear the browser's cache.		

#	Action	Expected Result	Notes
11	S-11 For the two collections identified in the previous step, request the manual export of collection metadata		#comment
12	Request manual export of both S-10 collections: EcBmBMGTManualStart --mode $MODE --metc --collections $SHORT_NAME_DIF_ID.$VERSION_ID_DIF_ID,$SHORT_NAME_NO_DIF_ID.$VERSION_ID_NO_DIF_ID		
13	Verification		#comment
14	V-10 Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		#comment
15	Log in to the BMGT GUI.		
16	Verify that the modified DIF ID has the new value.		
17	V-11 Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		#comment
18	Verify XML collection metadata was exported once for each S-10 collection.		
19	Use an xpath utility, such as /tools/common/test/BE_82_01/bin/xpath, to find XML elements.		#comment
20	Verify the exported metadata contains a DIF/EntryId element for the collection that has a DIF ID,and that it matches the configured value in the GUI. xpath '//DIF/EntryId' collection.xml		
21	Verify the exported metadata contains no DIF/EntryId element for the collection that does not have one: xpath '//DIF/EntryId' collection.xml		
22	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
23	Verify each S-10 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and		

				contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.101 ECS Collection Additional Metadata[S-12]: null ECS CoordinateSystem (ECS-ECSTC-108)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		

	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here:<br		

#	Action	Expected Result	Notes
	/>https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-12 Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		#comment
8	update bg_collection_configuration set CollectionCoordinateSystem =NULL where shortname = $SHORT_NAME and versionid = $VERSION_ID;		
9	Request a manual export of the S-12 collection: EcBmBMGTManualStart --mode $MODE --metc --collections $SHORT_NAME.$VERSION_ID		
10	Verification		#comment
11	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
12	Verify the S-12 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
13	V-13 Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		#comment
14	Query the BMGT configuration table to find the default CoordinateSystem value (property named 'BMGT.Common.CoordinateSystemDefault').		
15	Verify, using an xpath utility, the collection's exported metadata uses the default CoordinateSystem value: xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/CoordinateSystem/text()' collection.xml		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		

	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		

	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.102 ECS Collection Additional Metadata[S-13]: null ECS GranuleSpatialRepresentation (ECS-ECSTC-109)

DESCRIPTION:

	S	20	1	**[ECS Collection Additional Metadata]** Attempt to configure the CoordinateSystem in the database or configuration file for an ECS collection to a value other than “CARTESIAN” or “GEODETIC”.		
	S	20	2	Attempt to configure the GranuleSpatialRepresentation in the database or configuration file for an ECS collection to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL”.		
	S	20	3	Attempt to configure the CoordinateSystem for an ISO collection to a non null value.		
	S	20	4	Attempt to configure the GranuleSpatialRepresentation for an ISO collection to a non null value.		
	S	20	5	For each of the valid backtrack types, request the export of collection metadata for a collection which is configured for that type.		
	S	20	6	Request the export of metadata for a collection which is not configured for backtrack metadata.		
	S	20	7	For each of the valid TwoDCoordinateSystem types, request the export of collection metadata for a collection which is configured for that type.		

	S	20	8	Request the export of metadata for a collection which is not configured for a TwoDCoordinateSystem.		
	S	20	9	Request the export of metadata for a collection whose native metadata contains Product Specific Attributes (PSAs).		
	S	20	10	Find one collection with a DIF ID defined and one without. Attempt to modify the DIF ID for the collection which has one defined.		
	S	20	11	For the two collections identified in the previous step, request the manual export of collection metadata		
	S	20	12	Attempt to configure the CoordinateSystem for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		
	S	20	13	Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual export of this collection.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure all test collections are installed in the mode.		
3	Ensure all test collections are configured for collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = <SHORT_NAME> and versionid = <VERSION_ID> update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
4	Ensure a local copy of the collection schema is available to validate against. If needed, download from here: https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
5	Ensure a TCP proxy is capturing BMGT traffic.		
6	Setup		#comment
7	S-13 Attempt to configure the GranuleSpatialRepresentation for an ECS collection to null (i.e. automatically populate). Request the manual		#comment

#	Action	Expected Result	Notes
	export of this collection.		
8	update bg_collection_configuration set GranuleSpatialRepresentation = NULL where ShortName = $SHORT_NAME and VersionId = $VERSION_ID ;		
9	Request a manual export of the S-13 collection: EcBmBMGTManualStart --mode $MODE --metc --collections ${SHORT_NAME}.${VERSION_ID}		
10	Verification		#comment
11	V-12 Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		#comment
12	Verify S-13 collection's exported XML validates against the schema: xmllint --noout --schema Collection.xsd collection.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata c collection.xml		
13	V-14 Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: SpatialSearchType => GranuleSpatialRepresentation GPolygon => GEODETIC Orbit => ORBIT Point => GEODETIC Rectangle => CARTESIAN NotSupported => NO_SPATIAL		#comment
14	Query AmCollection for the S-13 collection's SpatialSearchType: select SpatialSearchType from AmCollection where ShortName = $SHORT_NAME and VersionId = $VERSION_ID		
15	Verify, using an xpath utility, that the S-13 collection's exported metadata has a GranuleSpatialRepresentation value derived from the collection's SpatialSearchType, using the V-14 mapping: xpath '/Collection/Spatial/HorizontalSpatialDomain/Geometry/GranuleSpatialRepresentation' collection.xml		

TEST DATA:

See Test Case 595.

EXPECTED RESULTS:

	V	20	1	Verify that in S-1, it is not possible to set CoordinateSystem to a value other than “CARTESIAN” or “GEODETIC” or that if it		

				is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	2	Verify that in S-2, it is not possible to set the GranuelSpatialRepresentation to a value other than “CARTESIAN”, “GEODETIC”, “ORBIT”, or “NO_SPATIAL” or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	3	Verify that in S-3, it is not possible to populate the CoordinateSystem for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Coordinate System value (and that the log indicates the reason for the failure).		
	V	20	4	Verify that in S-4, it is not possible to populate the GranuleSpatialRepresentation for an ISO collection or that if it is possible, subsequent metadata export for this collection fails due to the invalid Granule Spatial Representation value (and that the log indicates the reason for the failure).		
	V	20	5	Verify that the collection metadata generated in S-5 contains a Spatial/OrbitParameters element which is populated according to the rules for the backtrack metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	20	6	Verify that the collection metadata generated in S-6 contains no Spatial/OrbitParameters element.		
	V	20	7	Verify that the collection metadata generated in S-7 contains a TwoDCoordinateSystems/TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	20	8	Verify that the collection metadata generated in S-8 contains no TwoDCoordinateSystems/TwoDCoordinateSystem element.		
	V	20	9	Verify that the collection metadata generated in S-9 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original collection metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	20	10	Verify that in S-10, it is possible to modify the DIF ID for a collection via the GUI.		
	V	20	11	Verify that the metadata exported in S-11 contains the specified DIF ID for the collection for which one was provided, and contains no DIF ID for the other collection.		

	V	20	12	Verify that all Collection metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Collection.xsd		
	V	20	13	Verify that the collection metadata generated in S-12 contains the default CoordinateSystem value.		
	V	20	14	Verify that the collection metadata generated in S-13 contains the proper GranuleSpatialMetadata value derived from AmCollection.SpatialSearchType using the standard mapping: • SpatialSearchType=>GranuleSpatialRepresentation • GPolygon => GEODETIC • Orbit => ORBIT • Point => GEODETIC • Rectangle => CARTESIAN • NotSupported => NOSPATIAL		
	V	20	15	For the exports in S-5 – S-9, verify that the configured CoordinateSystem and GranuleSpatialRepresentation values are included in the metadata.		

1.103 ECS Granule Additional Metadata[S-2]: No Backtrack Orbit Metadata (ECS-ECSTC-127)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover		

				from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure the S-2 granule's collection is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the DPL Ingest GUI shows the S-2 granule's collection is configured to be public on ingest.		
4	Ensure the S-2 granule's collection is enabled for collection and granule export: update bg_collection_configuration set granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID};		
5	Ensure the S-2 granule's collection's orbitgroup is NULL, so backtrack metadata will not be exported: select orbitgroup from bg_collection_configuration where shortname = ${SHORT_NAME} and versionid = ${VERSIONID} If the orbit group is not null, set it: update bg_collection_configuration set orbitgroup = null where shortname = ${SHORT_NAME} and versionid = ${VERSIONID}		
6	Ensure ECHO has the S-2 granule's collection metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		

#	Action	Expected Result	Notes
7	Ensure the S-2 granule is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
8	Ensure ECHO has the S-2 granule's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granule, export it: EcBmBMGTManualStart ${MODE} --metg --granules ${GRANULEID}		
9	Ensure a local copy of the ECHO 10 schema files is available for schema validation (e.g., under /tools/common/test/BE_82_01/echo10/schema). If needed, download *.xsd files from https://api.echo.nasa.gov/ingest/schema/.		
10	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
11	Ensure the BMGT Dispatcher is running. On the BMGT host: ps auxww | grep $MODE | grep EcBmDispatcher If needed, start the dispatcher: ./EcBmBMGTDispatcherStart $MODE		
12	Setup		#comment
13	S-2 Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		#comment
14	Note the current time as t0.		
15	Request the manual export of the S-2 granule: EcBmBMGTManualStart ${MODE} --metg -g ${GRANULEID}		
16	Verification		#comment
17	V-2 Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		#comment
18	From the TCP proxy log, save the body of the S-2 granule's PUT request after time t0 to an XML file.		
19	Use the Ruby script /tools/common/test/BE_82_01/bin/xpath to search in XML files.		#comment
20	Verify the the S-2 granule's exported metadata contains no Spatial/HorizontalSpatialDomain/Orbit element: xpath '//Spatial/HorizontalSpatialDomain/Orbit' granule.xml		
21	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
22	Verify the S-2 granule's exported metadata validates against the ECHO 10 schema: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g		

#	Action	Expected Result	Notes
	g1.xml		

TEST DATA:
See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		
	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		

	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.104 ECS Granule Additional Metadata[S-3]: TwoDCoordinateSystem (ECS-ECSTC-128)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Ensure the test collections exist in AIM. E.g., the DPL Ingest GUI shows them as configured datatypes.		
3	Ensure the test collections are configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
4	Ensure collections have been configured so that the test data covers each of the TwoDCoordinateSystem types, 'MODIS Tile SIN', 'MODIS Tile EASE', and 'MISR': select shortname, versionid, TwoDCoordinateSystem from bg_collection_configuration where (shortname = 'MOD10A2' and versionid = 5 and TwoDCoordinateSystem = 'MODIS Tile SIN') or (shortname = 'MOD29P1D' and versionid = 5 and TwoDCoordinateSystem = 'MODIS Tile EASE') or (shortname = 'MB2LMT' and versionid = 2 and TwoDCoordinateSystem = 'MISR') If needed, run the populate collection script: EcBgPopulateCollections.ksh $MODE		
5	Ensure test granules gS, gE, gM have been ingested.		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available. (E.g., in /tools/common/test/BE_82_01/echo10/schema/)		
8	Setup		#comment
9	S-3 For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		#comment
10	Write the TwoDCoordinateSystem granule IDs to a text file: s3_granule_ids.txt		
11	Supply the granule IDs to the BMGT manual export script: EcBmBMGTManualStart $MODE --metg --granulefile /path/to/s3_granule_ids.txt		
12	V-3 Verify that the granule metadata generated in S-3 contains a		#comment

#	Action	Expected Result	Notes
	TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
13	Use /tools/common/test/BE_82_01/bin/xpath to find XML elements.		#comment
14	From the TCP proxy (or mock ECHO) log, save the body of each BMGT request to a separate XML file.		
15	Using an XPath utility, verify the exported metadata for each of granules gS, gE, gM has a TwoDCoordinateSystem element: xpath '//TwoDCoordinateSystem' granule.xml		
16	MODIS Tile SIN		#comment
17	Find the horizontal tile number in granule gS's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="HORIZONTALTILENUMBER"]/Values/Value[1]/text()' granule-MODIS_Tile_SIN.xml		
18	Verify the numeric value of StartCoordinate1 in granule gS's exported metadata matches the numeric value of its horizontal tile number: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MODIS Tile SIN"]/StartCoordinate1/text()' granule-MODIS_Tile_SIN.xml		
19	Find vertical tile number in granule gS's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="VERTICALTILENUMBER"]/Values/Value[1]/text()' granule-MODIS_Tile_SIN.xml		
20	Verify the numeric value of StartCoordinate2 in granule gS's exported metadata matches the numeric value of its vertical tile number: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MODIS Tile SIN"]/StartCoordinate2/text()' granule-MODIS_Tile_SIN.xml		
21	MODIS Tile EASE		#comment
22	Find the horizontal tile number in granule gS's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="HORIZONTALTILENUMBER"]/Values/Value[1]/text()' granule-MODIS_Tile_EASE.xml		

#	Action	Expected Result	Notes
23	Verify the numeric value of StartCoordinate1 in granule gS's exported metadata matches the numeric value of its horizontal tile number: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MODIS Tile EASE"]/StartCoordinate1/text()' granule-MODIS_Tile_EASE.xml		
24	Find vertical tile number in granule gS's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="VERTICALTILENUMBER"]/Values/Value[1]/text()' granule-MODIS_Tile_EASE.xml		
25	Verify the numeric value of StartCoordinate2 in granule gS's exported metadata matches the numeric value of its vertical tile number: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MODIS Tile EASE"]/StartCoordinate2/text()' granule-MODIS_Tile_EASE.xml		
26	MISR		#comment
27	Find path number in granule gM's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="SP_AM_PATH_NO"]/Values/Value[1]/text()' granule-MISR.xml		
28	Verify the numeric value of StartCoordinate1 in granule gM's exported metadata matches the numeric value of its path number: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MISR"]/StartCoordinate1/text()' granule-MISR.xml		
29	Find MISR start block in granule gM's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="SP_AM_MISR_StartBlock"]/Values/Value[1]/text()' granule-MISR.xml		
30	Verify the numeric value of StartCoordinate2 in granule gM's exported metadata matches the numeric value of its MISR start block: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MISR"]/StartCoordinate2/text()' granule-MISR.xml		
31	Find MISR end block in granule gM's exported metadata: xpath '/Granule/AdditionalAttributes/AdditionalAttribute[Name="SP_AM_MISR_EndBlock"]/Values/Value[1]/text()' granule-MISR.xml		
32	Verify the numeric value of EndCoordinate2 in granule gM's exported metadata matches the numeric value of its MISR end block: xpath '/Granule/TwoDCoordinateSystem[TwoDCoordinateSystemName="MI		

#	Action	Expected Result	Notes
	SR"]/EndCoordinate2/text()' granule-MISR.xml		
33	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
34	Verify each granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		

TEST DATA:
See Test Case 601

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		

	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.105 ECS Granule Additional Metadata[S-4]: No TwoDCoordinateSystem (ECS-ECSTC-129)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Ensure the test granule's collection exists in AIM. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
3	Ensure the test collection is configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both export flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it to be exported.		
4	Ensure the test granule has been ingested: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
5	Ensure the test granule's collection is configured with no TwoDCoordinateSystem: select collectionid from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} and TwoDCoordinateSystem is null If needed, remove the TwoDCoordinateSystem: update bg_collection_configuration set TwoDCoordinateSystem = null where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
6	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
7	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available, e.g, under /tools/common/test/BE_82_01/echo10/schema.		
8	Setup		#comment
9	S-4 Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		#comment
10	Note the current time as t0.		
11	Request a manual export of the S-4 granule: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		

#	Action	Expected Result	Notes
12	From the TCP proxy (or mock ECHO) log, save the body of each HTTP PUT request after time t0 to a separate XML file.		
13	V-4 Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		#comment
14	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
15	Using an XPath utility, verify the S-4 granule's exported metadata has no TwoDCoordinateSystem element: xpath '//TwoDCoordinateSystem' granule.xml		
16	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
17	Verify the granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		

TEST DATA:

See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each		

				Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		
	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.106 ECS Granule Additional Metadata[S-5]: Product Specific Attributes (PSAs) (ECS-ECSTC-130)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		

	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure the S-5 granule's collection exists in AIM. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
4	Ensure the S-5 granule's collection is configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both export flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it and its granules to be exported.		
5	Ensure the S-5 granule exists in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = '${LOCALGRANULEID}'		
6	Ensure the S-5 granule's native XML has PSAs, and save them for later verification. Locate the granule's metadata file in the small file archive: select p.path || '/' || m.archivemetfilename from ammetadatafile m join dsmdxmlpath p on m.archivepathid = p.archivepathid where m.granuleid = ${GRANULEID} Find the PSAs in the metadata file: xpath '//PSAs'		

#	Action	Expected Result	Notes
	/path/to/file.xml For example, <PSAs> <PSA> <PSAName>QAPERCENTGOODQUALITY</PSAName> <PSAValue>0</PSAValue> </PSA> <PSA> <PSAName>QAPERCENTOTHERQUALITY</PSAName> <PSAValue>100</PSAValue> </PSA> <PSA> <PSAName>HORIZONTALTILENUMBER</PSAName> <PSAValue>03</PSAValue> </PSA> <PSA> <PSAName>VERTICALTILENUMBER</PSAName> <PSAValue>08</PSAValue> </PSA> <PSA> <PSAName>TileID</PSAName> <PSAValue>31003008</PSAValue> </PSA> <PSA> <PSAName>SEAICEPERCENT</PSAName> <PSAValue>0</PSAValue> </PSA> </PSAs>		
7	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available, e.g., under /tools/common/test/BE_82_01/echo10/schema.		
8	Setup		#comment
9	S-5 Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		#comment
10	Note the current time as t0.		
11	Request the S-5 granule's manual export: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		
12	From the TCP proxy (or mock ECHO) log, save the body of each HTTP PUT request after time t0 to a separate XML file.		
13	V-5 Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		#comment
14	Extract the AdditionalAttributes/AdditionalAttribute elements from the S-5 granule's exported metadata: xpath '//AdditionalAttributes/AdditionalAttribute' exported_granule_metadata.xml		
15	Verify there is an AdditionalAttributes/AdditionalAttribute element for each		

#	Action	Expected Result	Notes
	PSA in the original granule metadata.		
16	For each AdditionalAttributes/AdditionalAttribute, verify it includes all the inormation on the corresponding PSA from the original granule metadata.		
17	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
18	Verify the granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		

TEST DATA:

See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		

	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		
	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.107 ECS Granule Additional Metadata[S-6]: Cloud Cover from Core Metadata (ECS-ECSTC-131)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		

	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure the S-6 granule's collection exists in AIM. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
4	Ensure the S-6 granule's collection is configured to get its cloud cover from core metadata (dlcloudcoversource.sourcetype = 'C'): select s.sourcetype from amcollection c join dlcloudcoversource s on c.cloudcoversourceid = s.cloudsourceid where c.shortname = ${SHORTNAME} and c.versionid = $VERSIONID}		
5	Ensure the S-6 granule's collection is configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enable for export in this step, wait for it and its granules to be exported.		
6	Ensure the S-7 granule exists in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available, e.g., under /tools/common/test/BE_82_01/echo10/schema.		
9	Setup		#comment
10	S-6 Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		#comment
11	Note the curent time as t0.		

#	Action	Expected Result	Notes
12	Request the S-6 granule's manual export: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		
13	From the TCP proxy (or mock ECHO) log, save the body of each HTTP PUT request after time t0 to a separate XML file.		
14	V-6 Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		#comment
15	Locate the granule's metadata file in the small file archive: select p.path || '/' || m.archivemetfilename from ammetadatafile m join dsmdxmlpath p on m.archivepathid = p.archivepathid where m.granuleid = ${GRANULEID}		
16	Extract the cloud cover from the granule's ECS metadata: xpath '//MeasuredParameter/MeasuredParameterContainer/QAStats/QAPercentCloudCover' /stornext/.../granule.xml		
17	Extract the cloud cover from the granule's exported metadata: xpath '//Granule/CloudCover' granule.xml		
18	Extract the cloud cover from the granule's exported metadata: xpath '//Granule/CloudCover' granule.xml		
19	Verify the CloudCover value matches the first QAPercentCloudCover value.		
20	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
21	Verify the granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		

TEST DATA:

See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		

	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		
	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.108 ECS Granule Additional Metadata[S-7]: Cloud Cover from PSA (ECS-ECSTC-132)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a		

				collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure the S-7 granule's collection exists in AIM. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
4	Ensure the S-7 granule's collection is configured to get its cloud cover from PSAs (dlcloudcoversource.sourcetype = 'P'): select s.sourcetype from AmCollection c join DlCloudCoverSource s on c.cloudcoversourceid = s.cloudsourceid where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
5	Ensure the S-7 granule's collection is configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from		

#	Action	Expected Result	Notes
	bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it and its granules to be exported.		
6	Ensure the S-7 granule exists in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available, e.g., under /tools/common/test/BE_82_01/echo10/schema.		
9	Setup		#comment
10	S-7 Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		#comment
11	Note the current time as t0.		
12	Request the S-7 granule's manual export: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		
13	From the TCP proxy (or mock ECHO) log, save the body of each HTTP PUT request after time t0 to a separate XML file.		
14	V-7 Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		#comment
15	Find the cloud cover PSAs source name: select s.sourcename from amcollection c join dlcloudcoversource s on c.cloudcoversourceid = s.cloudsourceid where c.shortname = ${SHORTNAME} and c.versionid = ${VERSIONID}		
16	Locate the granule's metadata file in the small file archive: select p.path || '/' || m.archivemetfilename from ammetadatafile m join dsmdxmlpath p on m.archivepathid = p.archivepathid where m.granuleid = ${GRANULEID}		
17	Extract the ECS granule's PSA cloud cover: xpath "//PSAs/PSA[PSAName='${SOURCE_NAME}']/PSAValue" /stornext/.../granule.xml		
18	Extract the exported metadata's cloud cover: xpath '//Granule/CloudCover' granule.xml		

#	Action	Expected Result	Notes
19	Verify the CloudCover value matches the PSAValue.		
20	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
21	Verify the granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		

TEST DATA:
See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		

	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.109 ECS Granule Additional Metadata[S-8]: No CloudCover (ECS-ECSTC-133)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		
	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure the S-8 granule's collection exists in AIM. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
4	Ensure the S-8 granule's collection is configured to have no cloud cover (dlcloudcoversource.sourcetype is NULL): select s.sourcetype from AmCollection c left join DlCloudCoverSource s on c.cloudcoversourceid = s.cloudsourceid where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
5	Ensure the S-8 granule's collection is configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it and its granules to be exported.		
6	Ensure the S-8 granule exists in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available, e.g., under /tools/common/test/BE_82_01/echo10/schema.		
9	Setup		#comment
10	S-8 Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		#comment
11	Note the current time as t0.		
12	Request the S-8 granule's manual export: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		
13	From the TCP proxy (or mock ECHO) log, save the body of each HTTP PUT		

#	Action	Expected Result	Notes
	request after time t0 to a separate XML file.		
14	V-8 Verify that the granule metadata generated in S-8 contains no CloudCover element.		#comment
15	Verify the S-8 granule's exported metadata contains no CloudCover element: xpath '//CloudCover' granule.xml		
16	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
17	Verify the granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		

TEST DATA:
See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified		

				core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		
	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		
	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.110 ECS Granule Additional Metadata[S-9]: Invalid Cloud Cover Location (ECS-ECSTC-134)

DESCRIPTION:

	S	50	1	**[ECS Granule Additional Metadata]** For each of the valid backtrack types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	2	Request the export of granule metadata for a granule belonging to a collection which is not configured for backtrack metadata.		
	S	50	3	For each of the valid TwoDCoordinateSystem types, request the export of granule metadata for a granule belonging to a collection which is configured for that type.		
	S	50	4	Request the export of granule metadata for a granule belonging to a collection which is not configured for a TwoDCoordinateSystem.		
	S	50	5	Request the export of granule metadata for a granule whose native metadata contains Product Specific Attributes (PSAs).		
	S	50	6	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from core metadata.		
	S	50	7	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a PSA.		

	S	50	8	Request the export of granule metadata for a granule belonging to a collection which is configured to have no cloud cover source.		
	S	50	9	Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure the S-9 granule's collection exists in AIM. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
4	Ensure the S-9 granule's collection is configured to get its cloud cover from PSAs (dlcloudcoversource.sourcetype = 'P'): select s.sourcetype from AmCollection c join DlCloudCoverSource s on c.cloudcoversourceid = s.cloudsourceid where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
5	Ensure the S-9 granule's collection is configured for BMGT collection and granule metadata export (collectionexportflag = 'Y' and granuleexportflag = 'Y'): select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it and its granules to be exported.		
6	Ensure the S-9 granule exists in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Ensure a local copy of https://api.echo.nasa.gov/ingest/schema/Granule.xsd is available, e.g., under /tools/common/test/BE_82_01/echo10/schema.		
9	Setup		#comment

#	Action	Expected Result	Notes
10	S-9 Request the export of granule metadata for a granule belonging to a collection which is configured to get its cloud cover from a location (Core Metadata or PSA) which is not valid.		#comment
11	Record the test collection's cloudcoversourceid to restore later: select cloudcoversourceid from amcollection where shortname = ${SHORTNAME} and versionid = ${VERSIONID}		
12	Add a dummy row to dlcloudcoversource, so that the sourcename will match no PSAs: insert into dlcloudcoversource values (99, 'XXX', 'Dummy cloud cover source', 'P')		
13	Update the test collection's cloudcoversourceid to use the dummy cloud cover source: update amcollection set cloudcoversourceid = 99		
14	Note the current time as t0.		
15	Request the S-9 granule's manual export: EcBmBMGTManualStart $MODE --metg -g $GRANULEID		
16	From the TCP proxy (or mock ECHO) log, save the body of each HTTP PUT request after time t0 to a separate XML file.		
17	V-9 Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		#comment
18	Verify the S-9 granule's exported metadata contains no CloudCover element: xpath '//CloudCover' granule.xml		
19	Verify the BMGT dispatcher or generator log records an error obtaining cloud cover metadata: grep ${GRANULEID} EcBmBMGT{Dispatcher,Generator}.log	The log should have a message such as "CloudCover configuration is incorrect, no CloudCover will be inserted."	
20	V-10 Verify that all granule metadata in the previous steps validates against https://api.echo.nasa.gov/ingest/schema/Granule.xsd		#comment
21	Verify the granule's exported metadata validates against the Granule.xsd schema file: xmllint --noout --schema Granule.xsd granule.xml or /tools/common/test/BE_82_01/bin/echo10_validate_metadata g granule.xml		
22	Cleanup		#comment
23	Restore the saved cloudcoversourceid: update amcollection set cloudcoversourceid = ${CLOUDCOVERSOURCEID}		
24	Remove the dummy cloud cover source row: delete		

#	Action	Expected Result	Notes
	dlcloudcoversource where cloudsourceid = 99		

TEST DATA:
See Test Case 601.

EXPECTED RESULTS:

	V	50	1	Verify that the granule metadata generated in S-1 contains a Spatial/HorizontalSpatialDomain/Orbit element which is populated according to the rules for the back track metadata type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of the backtrack metadata.		
	V	50	2	Verify that the granule metadata generated in S-2 does not contain a Spatial/HorizontalSpatialDomain/Orbit element.		
	V	50	3	Verify that the granule metadata generated in S-3 contains a TwoDCoordinateSystem element which is populated according to the rules for the TwoDCoordinateSystem type associated with the collection. See the document BE_82_01_AdditionalMetadataDescription.doc for a detailed outline of TwoDCoordinateSystem metadata.		
	V	50	4	Verify that the granule metadata generated in S-4 does not contain a TwoDCoordinateSystem element.		
	V	50	5	Verify that the granule metadata generated in S-5 contains an AdditionalAttributes/AdditionalAttribute element for each Product Specific Attribute (PSA) in the original granule metadata, and that all information in the PSA is also conveyed in the Additional Attribute.		
	V	50	6	Verify that the granule metadata generated in S-6 contains a CloudCover element populated with the value from the specified core metadata element of the native metadata.		
	V	50	7	Verify that the granule metadata generated in S-7 contains a CloudCover element populated with the value from the specified PSA element of the native metadata.		
	V	50	8	Verify that the granule metadata generated in S-8 contains no CloudCover element.		
	V	50	9	Verify that the granule metadata generated in S-9 contains no CloudCover element, and that an error message is printed in the BMGT log indicating that there was an error obtaining cloud cover metadata for the granule.		

	V	50	10	Verify that all granule metadata in the previous steps validates againsthttps://api.echo.nasa.gov/ingest/schema/Granule.xsd		

1.111 ISO Nominal Granule Export: Granule Ingest (ECS-ECSTC-135)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure the BMGT automatic driver is running.		
5	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
6	Setup		#comment
7	S-2 For one of the collections in S-1 (or another collection which has ISO		#comment

#	Action	Expected Result	Notes
	metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		
8	S-2a		#comment
9	Ingest an ISO granule.		
10	Verification		#comment
11	Verify that the operations in S-2, excepting subclauses b-d and k each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
12	Verify the TCP proxy shows a single PUT for granule S-2a.		
13	Save the request body to an XML file.		
14	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
15	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
16	Verify that each granule's exported metadata validates against TBD ISO schema: /tools/libxml2-2.9.1/bin/xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema/1.0/ISO19115-2_EOS.xsd METADATA_FILE		
17	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
18	Query amgranule for the exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_IDS)		
19	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:		

#	Action	Expected Result	Notes
	citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		
20	Verify the granule's exported metadata InsertTime and LastUpdate matches its amgranule archivetime and lastupdate.		
21	V-8 Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		#comment
22	Verify the granule's exported metadata contains no RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		

	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as 'hidden' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as 'restricted' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as 'hidden' or 'restricted' in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.112 ISO Nominal Granule Export: Logical Delete (ECS-ECSTC-136)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export):		

				a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2b has been ingested.		
5	Ensure the BMGT automatic driver is running. (EcBmBMGTAutoStart <MODE>)		
6	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
7	Setup		#comment
8	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment
9	S-2b		#comment
10	Create a geoid file for granule S-2b: SC:<SHORT_NAME>.<VERSION_ID>:<GRANULE_ID&		

#	Action	Expected Result	Notes
	gt;		
11	Logically delete granule S-2b: EcDsBulkDelete.pl -mode <MODE> -physical -geoidfile /path/to/geoid_file		
12	Verification		#comment
13	V-3 Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		#comment
14	Verify the TCP proxy log shows a single DELETE for granule S-2b.		
15	Verify the TCP proxy log shows the request is sent to a URL containing the GranuleId.		
16	Verify the TCP proxy log shows the request body is empty.		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		

	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.113 ISO Nominal Granule Export: Physical Delete (ECS-ECSTC-137)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule.		

				h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2c has been ingested.		
5	Ensure the BMGT automatic driver is not running (EcBmBMGTAutoStop <MODE>).		
6	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
7	Setup		#comment
8	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment
9	S-2c		#comment
10	Create a geoid file for granule S-2c: SC:<SHORT_NAME>.<VERSION_ID>:<GRANULE_ID>		
11	Logically delete granule S-2c: EcDsBulkDelete.pl -mode <MODE> -physical -geoidfile /path/to/geoid_file		
12	Unpublish granule S-2c: EcDlUnpublishStart.pl -mode <MODE> -g <GRANULE_ID>		
13	Physically delete granule S-2c: EcDsDeletionCleanup.pl -mode		

#	Action	Expected Result	Notes
	<MODE> Follow the prompts, always choosing the most destructive option.		
14	Start the automatic driver EcBmBMGTAutoStart <MODE>		
15	Verification		#comment
16	V-3 Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		#comment
17	Verify the TCP proxy log shows a single DELETE for granule S-2c.		
18	Verify the TCP proxy log shows the request is sent to a URL containing the GranuleId.		
19	Verify the TCP proxy log shows the request body is empty.		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		

	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.114 ISO Nominal Granule Export: DFA (ECS-ECSTC-138)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule.		

				h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2d has been ingested.		
5	Ensure the BMGT automatic driver is running (EcBmBMGTAutoStart <MODE>).		
6	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
7	Setup		#comment
8	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment
9	S-2d		#comment
10	Create a geoid file for granule S-2d: SC:<SHORT_NAME>.<VERSION_ID>:<GRANULE_ID>		
11	DFA granule S-2d: EcDsBulkDelete.pl -mode <MODE> -dfa -geoidfile /path/to/geoid_file		
12	Verification		#comment
13	V-3 Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but		#comment

#	Action	Expected Result	Notes
	not containing any granule metadata in the request body.</i>		
14	Verify the TCP proxy log shows a single DELETE for granule S-2d.		
15	Verify the TCP proxy log shows the request is sent to a URL containing the GranuleId.		
16	Verify the TCP proxy log shows the request body is empty.		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance		

				to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.115 ISO Nominal Granule Export: Hide (ECS-ECSTC-139)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2e has been ingested.		
5	Ensure the BMGT automatic driver is running (EcBmBMGTAutoStart <MODE>).		
6	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
7	Setup		#comment
8	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment
9	S-2e		#comment
10	Set granule S-2e's DeleteFromArchive column to 'H': update amgranule set deletefromarchive = 'H' where granuleid = <GRANULE_ID>		
11	Verification		#comment
12	Verify that the operations in S-2, excepting subclauses b-d and k each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
13	Verify the TCP proxy shows a single PUT for granule S-2e.		
14	Save the request body to an XML file.		
15	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
16	V-4 Verify that the metadata exported in S-1 and S-2, except S-2		#comment

#	Action	Expected Result	Notes
	subclauses b-d, validates against TBD ISO Schema.		
17	Verify that each granule's exported metadata validates against ISO schema: xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		This currently gives an error, but i think it is a problem with the schema itself. 4/26/2013 -- Goff, Timothy
18	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
19	Query amgranule for the exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_IDS)		
20	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		
21	Verify the granule's exported metadata InsertTime and LastUpdate matches its amgranule archivetime and lastupdate.		
22	V-6 Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		#comment
23	Verify the granule's exported metadata contains a RestrictionFlag element with a value of 255: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		

TEST DATA:
See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		

	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.116 ISO Nominal Granule Export: Restrict (ECS-ECSTC-140)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2f has been ingested.		
5	Ensure the BMGT automatic driver is running (EcBmBMGTAutoStart <MODE>).		

#	Action	Expected Result	Notes
6	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
7	Setup		#comment
8	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment
9	S-2f		#comment
10	Choose or create a restriction flag: select * from dsmdrestrictionflag insert into dsmdrestrictionflag values(128, 'BE_82_01 Crit 100 S-2f')		
11	Add the restriction flag to granule S-2f: insert into dsmdgranulerestriction values(<GRANULE_ID>, 128)		
12	Verification		#comment
13	Verify that the operations in S-2, excepting subclauses b-d and k each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
14	Verify the TCP proxy shows a single PUT for granule S-2f.		
15	Save the request body to an XML file.		
16	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
17	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
18	Verify that each granule's exported metadata validates against ISO schema: xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		This currently gives an error, but i think it is a problem with the schema itself. 4/26/2013 -- Goff, Timothy
19	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert		#comment

#	Action	Expected Result	Notes
	time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
20	Query amgranule for the exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_IDS)		
21	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		
22	Verify the granule's InsertTime and LastUpdate matches its archivetime and lastupdate from amgranule.		
23	V-7 Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		#comment
24	Verify the granule's exported metadata contains a RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		
25	Verify the exported RestrictionFlag value matches the restrictionflag value in AIM: select restrictionflag from dsmdgranulerestriction where granuleid = <GRANULE_ID>		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		

	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.117 ISO Nominal Granule Export: Unrestrict (ECS-ECSTC-141)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2g has been ingested.		
5	Ensure granule S-2g has a restriction flag set.		
6	Ensure the BMGT automatic driver is running (EcBmBMGTAutoStart <MODE>).		
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Setup		#comment
9	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest		#comment

#	Action	Expected Result	Notes
	a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		
10	S-2g		#comment
11	Remove the restriction flag from granule S-2g: delete dsmdgranulerestriction where granuleid = <GRANULE_ID>		
12	Verification		#comment
13	Verify that the operations in S-2, excepting subclauses b-d and k each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
14	Verify the TCP proxy shows a single PUT for granule S-2g.		
15	Save the request body to an XML file.		
16	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
17	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
18	Verify that each granule's exported metadata validates against ISO schema: xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		This currently gives an error, but i think it is a problem with the schema itself. 4/26/2013 -- Goff, Timothy
19	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
20	Query amgranule for the exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_ID)		
21	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath		

#	Action	Expected Result	Notes
	'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		
22	Verify the granule's exported metadata InsertTime and LastUpdate matches its amgranule archivetime and lastupdate.		
23	V-8 Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		#comment
24	Verify the granule's exported metadata contains no RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		

	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as 'hidden' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as 'restricted' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as 'hidden' or 'restricted' in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.118 ISO Nominal Granule Export: Publish (ECS-ECSTC-142)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export):		

				a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2h has been ingested.		
5	Ensure granule S-2h is in the hidden data pool.		
6	Ensure the BMGT automatic driver is running (EcBmBMGTAutoStart <MODE>).		
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Setup		#comment
9	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment
10	S-2h		#comment
11	Publish granule S-2g: EcDlPublishUtilityStart <MODE> -		

#	Action	Expected Result	Notes
	ecs -g <GRANULE_ID>		
12	Verification		#comment
13	Verify that the operations in S-2, excepting subclauses b-d and k each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
14	Verify the TCP proxy shows a single PUT for granule S-2h.		
15	Save the request body to an XML file.		
16	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
17	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
18	Verify that each granule's exported metadata validates against ISO schema: xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		This currently gives an error, but i think it is a problem with the schema itself. 4/26/2013 -- Goff, Timothy
19	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
20	Query amgranule for the exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_ID)		
21	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		
22	Verify the granule's exported metadata InsertTime and LastUpdate matches its amgranule archivetime and lastupdate.		
23	V-8 Verify that the metadata for the granule in S-2 subclause g (and any		#comment

#	Action	Expected Result	Notes
	other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
24	Verify the granule's exported metadata contains no RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		
25	V-9 Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		#comment
26	Verify granule S-2h's exported metadata contains an OnlineAccess URL for the science file and an OnlineResource URL for the metadata file: xpath 'gmi:MI_Metadata/gmd:distributionInfo/gmd:MD_Distribution/gmd:distributor/gmd:MD_Distributor/gmd:distributorTransferOptions/gmd:MD_DigitalTransferOptions/gmd:onLine/gmd:CI_OnlineResource' granule.xml Metadata file will have name/CharacterString == 'METADATA'. Science file will have no gmd:name/gco:CharacterString		
27	Verify, using either ftp, http, or the data pool filesystem, that the URLs point to the correct file locations.		
28	Verify that for each URL, gmd:applicationProfile/gco:CharacterString contains the correct mime type.		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		

	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as 'hidden' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as 'restricted' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as 'hidden' or 'restricted' in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.119 ISO Nominal Granule Export: Unpublish (ECS-ECSTC-143)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name,		

				but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure granule S-2i has been ingested.		
5	Ensure granule S-2i is in the public data pool (EcBmBMGTAutoStart <MODE>).		
6	Ensure the BMGT automatic driver is running.		
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Setup		#comment
9	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a		#comment

#	Action	Expected Result	Notes
	granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		
10	S-2i		#comment
11	Unpublish granule S-2i: EcDlUnpublishStart.pl -mode <MODE> -g <GRANULE_ID>		
12	Verification		#comment
13	Verify that the operations in S-2, excepting subclauses b-d and k each result in the export of a single HTTP PUT containing the full granule metadata.		#comment
14	Verify the TCP proxy shows a single PUT for granule S-2i.		
15	Save the request body to an XML file.		
16	Verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
17	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
18	Verify that each granule's exported metadata validates against ISO schema: xmllint --noout --schema /ecs/formal/ESDT/Sm/schema/ISO-SMAP/schema.xsd METADATA_FILE		This currently gives an error, but i think it is a problem with the schema itself. 4/26/2013 -- Goff, Timothy
19	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
20	Query amgranule for the exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where granuleid in ($GRANULE_ID)		
21	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:		

#	Action	Expected Result	Notes
	citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		
22	Verify the granule's exported metadata InsertTime and LastUpdate matches its amgranule archivetime and lastupdate.		
23	V-8 Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		#comment
24	Verify the granule's exported metadata contains no RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		
25	V-10 Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		#comment
26	Verify the granule's exported metadata contains no OnlineAccessURL or OnlineResource elements: xpath 'gmi:MI_Metadata/gmd:distributionInfo/gmd:MD_Distribution/gmd:distributor/gmd:MD_Distributor/gmd:distributorTransferOptions/gmd:MD_DigitalTransferOptions/gmd:onLine/gmd:CI_OnlineResource' granule.xml		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the		

				granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as 'hidden' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as 'restricted' in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as 'hidden' or 'restricted' in accordance to the ISO schema andSDPS to ECHO ICD.		
	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.120 ISO Nominal Granule Export: Move Collection (ECS-ECSTC-144)

DESCRIPTION:

	S	100	1	**[ISO Nominal Granule Export]** Find two collections with ISO metadata and which share the same short name, but have different version IDs. Ensure that both collections are enabled for collection and granule export. Request		

				the manual export of granule metadata for all granules in one of these collections.		
	S	100	2	For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a)Ingest a new granule into the ECS inventory. b)Logically delete a granule from the ECS inventory. c)Physically delete a granule from the ECS inventory. d)DFA a granule. e)Hide a granule. f)Restrict a granule. g)Unrestrict a granule. h)Publish a granule in the datapool. i)Unpublish a granule in the datapool. j)Change the collection which a granule belongs to. k)Move a collection. l)Perform XML replacement on a granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure test collection C2 is installed.		
3	Ensure collection C2 is configured to be public on ingest.		
4	Ensure the BMGT automatic driver is running (EcBmBMGTAutoStart <MODE>).		
5	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
6	Setup		#comment
7	S-2 For one of the collections in S-1 (or another collection which has ISO metadata, and is enabled for collection and granule export): a) Ingest a new granule into the ECS inventory. b) Logically delete a granule from the ECS inventory. c) Physically delete a granule from the ECS inventory. d) DFA a granule. e) Hide a granule. f) Restrict a granule. g) Unrestrict a granule. h) Publish a granule in the datapool. i) Unpublish a granule in the datapool. j) Change the collection which a granule belongs to. k) Move a collection. l) Perform XML replacement on a granule.		#comment

#	Action	Expected Result	Notes
8	S-2k		#comment
9	Save granule IDs of all non-deleted granules in collection C2 for later verification: select granuleid from amgranule where shortname = <SHORT_NAME> and versionid = <VERSION_ID> and deleteeffectivedate is NULL		
10	Find collection C2's current filesystem, either via the Data Pool Maintenance GUI or by querying the database: select filesystemlabel from amcollection where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
11	Move collection C2 to a different filesystem (e.g,. from FS1 to FS2): EcDlMoveCollection.pl <MODE> -shortname <SHORT_NAME> -versionid <VERSION_ID> -sourcefs FS1 -targetfs FS2 -verbose		
12	Verification		#comment
13	V-4 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		#comment
14	perform a spot check of at least two of the exported granules to verify that each granule's exported metadata is complete and correct when compared with its local metadata. To do this, find the path to the native metadata (printed in the generator log with the prefix 'MetadataFilePath') and diff this file against the exported metadata. Verify that the exported metadata is a superset of the native metadata.		
15	V-5 Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		#comment
16	Query amgranule for each exported granule's InsertTime and LastUpdate: select granuleid, archivetime, lastupdate from amgranule where shortname = <SHORT_NAME> and versionid = <VERSION_ID>		
17	Use an XPath utility to extract the InsertTime and LastUpdate from each granule's exported metadata: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='UpdateTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:title/gco:CharacterString='InsertTime']/gmd:date/gmd:CI_Date/gmd:date/gco:DateTime' granule.xml		

#	Action	Expected Result	Notes
18	Verify each granule's exported metadata InsertTime and LastUpdate matches its amgranule archivetime and lastupdate.		
19	V-8 Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		#comment
20	Determine which non-deleted granules in collection C2 have restriction flags: select r.granuleid from dsmdgranulerestriction r join amgranule g on r.granuleid = g.granuleid where g.shortname = <SHORT_NAME> and g.versionid = <VERSION_ID> and deleteeffectivedate is NULL		
21	Verify the exported metadata for each granule with an AIM restriction flag contains a RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		
22	Determine which non-deleted granules in collection C2 have restriction flags: select r.granuleid from dsmdgranulerestriction r join amgranule g on r.granuleid = g.granuleid where g.shortname = <SHORT_NAME> and g.versionid = <VERSION_ID> and deletefromarchive = 'H' Verify each granule's exported metadata contains a RestrictionFlag element with a value of 255: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		
23	Verify exported metadata for any granule without an AIM restriction flag contains no RestrictionFlag element: xpath 'gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification[gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString='RestrictionFlag']/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:otherConstraints/gco:CharacterString' granule.xml		
24	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected (excepting those which are logically deleted).		#comment
25	Verify the TCP proxy shows a single PUT for each non-deleted granule in		

#	Action	Expected Result	Notes
	collection C2.		
26	Save each request body to a separate XML file.		
27	Verify each public granule's granuleid appears in the URL of a PUT request.		
28	Verify there is exactly one XML body for each public ganule's granuleid.		

TEST DATA:

See Test Case 624.

EXPECTED RESULTS:

	V	100	1	Verify that the manual export in S-1 results in multiple HTTP PUT requests containing the full granule metadata for each granule in the requested collection (but not any granules belonging to other collections sharing a short name but with a different version ID).		
	V	100	2	Verify that the operations in S-2, excepting subclauses b-d each result in the export of a single HTTP PUT containing the full granule metadata.		
	V	100	3	Verify that the operations in S-2 subclauses b-d each result in the export of a single HTTP DELETE, with the ID of the granule in the URL, but not containing any granule metadata in the request body.		
	V	100	4	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d, validates against TBD ISO Schema.		
	V	100	5	Verify that the metadata exported in S-1 and S-2, except S-2 subclauses b-d has the following elements: a) InsertTime = The insert time of the granule recorded in the AIM database. b) LastUpdate = The last update time of the granule recorded in the AIM database		
	V	100	6	Verify that the metadata for the granule in S-2 subclause e includes elements marking the granule as ‘hidden’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	7	Verify that the metadata for the granule in S-2 subclause f includes elements marking the granule as ‘restricted’ in accordance to the ISO schema and SDPS to ECHO ICD.		
	V	100	8	Verify that the metadata for the granule in S-2 subclause g (and any other subclauses for which the affected granule did not have a restriction flag set) do not contain any elements marking the granules as ‘hidden’ or ‘restricted’ in accordance to the ISO schema andSDPS to ECHO ICD.		

	V	100	9	Verify that the metadata generated by the action in S-2 subclause h contains URLs for the science and metadata files.		
	V	100	10	Verify that the metadata generated by the action in S-2 subclause i contains no datapool URLs.		
	V	100	11	Verify that the metadata generated by the action in S-2 subclause j contains a reference to the newly assigned collection.		
	V	100	12	Verify that the action performed in S-2 subclause k results in the export of full granule metadata for every granule in the collection affected.		
	V	100	13	Verify that the metadata generated by the action in S-2 subclause l contains XML reflecting the replacement which was performed.		

1.121 Manual Export â€“ Invocation via Command Line[b]: granules w/ granule file (ECS-ECSTC-145)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test granules' collection is installed. (ESDT Verification script)		
4	Ensure the test granules' collection is configured for collection and granule export.		
5	Ensure the test granules G7 ... G12 are ingested.		
6	Setup		#comment
7	S-1 From the command line invoke manual export of metadata for b) Granule metadata for multiple granules, specified in an input file.		#comment
8	Prepare an input file, granules.txt, with granule IDs for granules G7, G9, and a geoid for G8, one per line: <G7_GRANULE_ID> SC:<SHORT_NAME>.<VERSION_ID>:<G8_GRANULE_ID> <G9_GRANULE_ID>		
9	Run BMGT manual export, passing the input file to the --granulefile option: EcBmBMGTManualStart <MODE> --metg --		

#	Action	Expected Result	Notes
	granulefile /path/to/granules.txt		
10	Verification		#comment
11	V-1 Verify that the operation in S-1 results in b) Single HTTP PUT for each granule.		#comment
12	Verify a single HTTP request is sent for each granule G7, G8, G9 from the tcp.log. (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
13	Repeat the test, replacing option --granulefile with the short form: --gf		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.122 Manual Export â€“ Invocation via Command Line[c]: granules w/ collection arg (ECS-ECSTC-146)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection C1 is installed.(ESDT Verification script)		
4	Ensure the test collection C1 enabled for collection and granule export.		
5	Ensure the test granules G7, G8, G9 are ingested.		
6	Setup		#comment
7	S-1 From the command line invoke manual export of metadata for c) Granule metadata for all granules in a collection, specified on the command line.		#comment
8	Run BMGT manual export on granules in collection C1: EcBmBMGTManualStart <MODE> --metg --collections		

#	Action	Expected Result	Notes
	<C1_SHORT_NAME>.<C1_VERSION_ID>		
9	Verification		#comment
10	V-1 Verify that the operation in S-1 results in c) HTTP PUT for each granule in the collection.		#comment
11	Verify a single HTTP request is sent for each granule G7, G8, G9 in collection C1 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
12	Repeat the test, replacing option --collections with the short form: -c		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.123 Manual Export â€“ Invocation via Command Line[d]: granules w/ collection file (ECS-ECSTC-147)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection C2 is installed.(ESDT Verification script)		
4	Ensure the test collection C2 is enabled for collection and granule export.		
5	Ensure the test granules G10, G11, G12 are ingested.		
6	Setup		#comment
7	S-1 From the command line invoke manual export of metadata for d) Granule metadata for all granules in a collection, specified in an input file.		#comment
8	Prepare an input file, collections.txt, listing test collection C2: <br		

#	Action	Expected Result	Notes
	/><SHORTNAME>.<VERSION_ID>		
9	Run BMGT manual export on granules in collections specified in the input file: EcBmBMGTManualStart <MODE> --metg --collectionfile /path/to/collections.txt		
10	Verification		#comment
11	V-1 Verify that the operation in S-1 results in d) HTTP PUT for each granule in the collection.		#comment
12	Verify a single HTTP request is sent for each granule G10, G11, G12 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
13	Repeat the test, replacing option --collectionfile with the short form: --cf		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.124 Manual Export â€“ Invocation via Command Line[e]: collections w/ collection args (ECS-ECSTC-148)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure test collections C3, C4, C5 are installed.(ESDT Verification script)		
4	Ensure test collections C3, C4, C5 are enabled for collection export.		
5	Setup		#comment
6	S-1 From the command line invoke manual export of metadata for e) Collection metadata for multiple collections, specified on the command line.		#comment

#	Action	Expected Result	Notes
7	Run BMGT manual export on collections C3, C4, C5 (ensure no spaces between commas or collection names): EcBmBMGTManualStart <MODE> --metc --collections <C3>,<C4>,<C5>		
8	Verification		#comment
9	V-1 Verify that the operation in S-1 results in e) HTTP PUT for each specified collection.		#comment
10	Verify a single HTTP PUT is sent for each collection C3, C4, C5 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
11	Repeat the test, replacing option --collections with the short form: -c		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.125 Manual Export â€“ Invocation via Command Line[f]: collections w/ collection file (ECS-ECSTC-149)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure test collections C6, C7, C8 are installed.(ESDT Verification script)		
4	Ensure test collections C6, C7, C8 are enabled for collection export.		
5	Setup		#comment
6	S-1 From the command line invoke manual export of metadata for f) Collection metadata for multiple collections, specified in an input file.		#comment
7	Prepare an input file, collections.txt, with collections C6, C7, C8, one per line: <C6_SHORT_NAME>.<C6_VERSION_ID><br		

#	Action	Expected Result	Notes
	/><C7_SHORT_NAME>.<C7_VERSION_ID> <C8_SHORT_NAME>.<C8_VERSION_ID>		
8	Run BMGT manual export for collections specified in the input file: EcBmBMGTManualStart <MODE> --metc --collectionfile /path/to/collections.txt		
9	Verification		#comment
10	V-1 Verify that the operation in S-1 results in f) HTTP PUT for each specified collection.		#comment
11	Verify a single HTTP PUT is sent for each collection C6, C7, C8 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
12	Repeat the test, replacing option --collectionfile with the short form: --cf		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.126 Manual Export â€“ Invocation via Command Line[g]: metadata w/ group arg (ECS-ECSTC-150)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure test collections C9, C10 are installed.(ESDT Verification script)		
4	Ensure test granules G13 ... G18 are ingested.		
5	Ensure test collections are enabled for collection and granule export.		
6	Ensure test collections C9, C10 belong to a Data Pool group.		
7	Ensure all other collections belonging to the same Data Pool group are		

#	Action	Expected Result	Notes
	disabled for collection or granule export. For each collection other than C9 or C10, update bg_collection_configuration set collectionexportflag = 'N', granuleexportflag = 'N' where shortname = '${SHORTNAME}' and versionid = '${VERSIONID}'		
8	Setup		#comment
9	S-1 From the command line invoke manual export of metadata for g) Granule and collection metadata for all collections in a Datapool Group, specified on the command line.		#comment
10	Run BMGT manual export on the data pool group: EcBmBMGTManualStart $MODE --metc --metg --groups ${DPL_GROUP}		
11	Verification		#comment
12	V-1 Verify that the operation in S-1 results in g) Granule and collection metadata for all collections in a Datapool Group, specified on the command line.		#comment
13	Verify a single HTTP PUT is sent for each collection C9, C10 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
14	Verify each collection's request includes the collection's metadata from tcp.log		
15	Verify a single HTTP PUT is sent for each granule G13 ... G18 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log)		
16	Verify each granule's request includes the granule's metadata from the tcp.log		
17	Repeat the steps, replacing the option --group with the short form: -p		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.127 Manual Export â€“ Invocation via Command Line[h]: metadata w/ group file (ECS-ECSTC-151)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure test collection C11 is installed.(ESDT Verification script)		
4	Ensure test granules G19, G20, G21 are ingested.		
5	Ensure test collection C11 is enabled for collection and granule export.		
6	Ensure test collection C11 belongs to a Data Pool group.		
7	Ensure all other collections belonging to the same Data Pool group are disabled for collection or granule export. For each collection other than C11, update bg_collection_configuration set collectionexportflag = 'N', granuleexportflag = 'N' where shortname = '${SHORTNAME}' and versionid = '${VERSIONID}'		
8	Setup		#comment
9	S-1 From the command line invoke manual export of metadata for h) Granule and collection metadata for all collections in a Datapool Group, specified in an input file.		#comment
10	Prepare an input file, groups.txt, listing the test Data Pool group.		
11	Run BMGT manual export on the group in the input file: EcBmBMGTManualStart <MODE> --metc --metg --groupfile /path/to/groups.txt		
12	Verification		#comment
13	V-1 Verify that the operation in S-1 results in h) HTTP PUT for each granule and collection in the specified group.		#comment
14	Verify an HTTP PUT is sent for collection C11 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log). (More than one PUT may be sent, e.g., if there are network issues.)		
15	Verify an HTTP PUT is sent for each granule G19 ... G21 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log). (More than one PUT may be sent, e.g., if there are network issues.)		
16	Repeat the test, replacing option --groupfile with the short form: -pf		

TEST DATA:

See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.128 Manual Export â€“ Invocation via Command Line[i]: granule deletions w/ collection arg (ECS-ECSTC-152)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection C12 is installed.(ESDT Verification script)		
4	Ensure the test collection C12 is enabled for collection and granule export.		
5	Ensure the test granules G22 ... G27 are ingested.		
6	Ensure the test granules G22, G23 are logically deleted.		
7	Ensure the test granules G24, G25 are DFAed.		
8	Ensure the test granules G26, G27 are neither logically deleted nor DFAed.		
9	Setup		#comment
10	S-1 From the command line invoke manual export of metadata for i) Granule deletions for all logically deleted granules in a collection, specified on the command line.		#comment
11	Run BMGT manual export on deleted granules in collection C12: EcBmBMGTManualStart <MODE> --metg --deleteOnly --collections <SHORT_NAME>.<VERSION_ID>		
12	Verification		#comment
13	V-1 Verify that the operation in S-1 results in i) HTTP DELETE for each logically deleted granule in the specified collection.		#comment
14	Verify an HTTP DELETE is sent for each granule G22, G23, G24, G25 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) (More than one DELETE may be sent, e.g., if there are network issues.)		
15	Verify no HTTP requests are sent for granules G26, G27 from the tcp.log		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.129 Manual Export â€“ Invocation via Command Line[j]: granule deletions w/ collection file (ECS-ECSTC-153)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection C13 is installed.(ESDT Verification script)		
4	Ensure the test collection C13 is configured for collection and granule export.		
5	Ensure the test granules G28 ... G33 are ingested.		
6	Ensure the test granules are G28, G29 are logically deleted.		
7	Ensure the test granules are G30, G31 are DFAed.		
8	Ensure the test granules are G32, G33 are neither logically deleted nor DFAed.		
9	Setup		#comment
10	S-1 From the command line invoke manual export of metadata for j) Granule deletions for all logically deleted granules in a collection, specified in an input file.		#comment
11	Prepare an input file, collections.txt, listing the test collection: <SHORT_NAME>.<VERSION_ID>		
12	Run BMGT manual export on deleted granule in collections specified in the input file: EcBmBMGTManualStart <MODE> --metg --deleteOnly --collectionfile /path/to/collections.txt		
13	Verification		#comment

#	Action	Expected Result	Notes
14	V-1 Verify that the operation in S-1 results in j) HTTP DELETE for each logically deleted granule in the specified collection.		#comment
15	Verify an HTTP DELETE is sent for each granule G28, G29, G30, G31 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) (More than one DELETE may be sent, e.g., if there are network issues.)		
16	Verify no HTTP request is sent for granules G32, G33 from the tcp.log		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.130 Manual Export â€“ Invocation via Command Line[k]: non-deleted granules w/ collection arg (ECS-ECSTC-154)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection C14 is installed.(ESDT Verification script)		
4	Ensure the test collection C14 is configured for collection and granule export.		
5	Ensure the test granules G34 ... G29 are ingested.		
6	Ensure granules G34, G35 are logically deleted.		
7	Ensure granules G36, G37 are DFAed.		
8	Ensure granules G38, G39 are neither logically deleted nor DFAed.		
9	Setup		#comment
10	S-1 From the command line invoke manual export of metadata for k) Granule metadata for all non-deleted granules in a collection, specified on		#comment

#	Action	Expected Result	Notes
	the command line.		
11	Run BMGT manual export on non-deleted granules, specified by collection: EcBmBMGTManualStart <MODE> --insertonly --metg --collections <SHORT_NAME>.<VERSION_ID>		
12	Verification		#comment
13	V-1 Verify that the operation in S-1 results in k) HTTP PUT for each non-deleted granule in the specified collection.		#comment
14	Verify an HTTP PUT is sent for granules G38, G39 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) (More than one PUT may be sent, e.g., if there are network issues.)		
15	Verify no HTTP DELETEs are sent from the tcp.log		
16	Verify no HTTP requests are sent for granules G34, G35, G36, G37 from the tcp.log		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.131 Manual Export â€“ Invocation via Command Line[l]: non-deleted granules w/ collection file (ECS-ECSTC-155)

DESCRIPTION:
See Test Case 644.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure a TCP proxy or mock ECHO is capturing BMGT HTTP traffic.		
3	Ensure the test collection C15 is installed.(ESDT Verification script)		
4	Ensure the test collection C15 is configured for collection and granule export.		
5	Ensure the test granules G40 ... G45 are ingested.		

#	Action	Expected Result	Notes
6	Ensure granules G40, G41 are logically deleted.		
7	Ensure granules G42, G43 are DFAed.		
8	Ensure granules G44, G45 are neither logically deleted nor DFAed.		
9	Setup		#comment
10	S-1 From the command line invoke manual export of metadata for l) Granule metadata for all non-deleted granules in a collection, specified in an input file.		#comment
11	Prepare an input file, collections.txt, that lists test collection C23: <SHORT_NAME>.<VERSION_ID>		
12	Run BMGT manual export on non-deleted granules specified by the intput file: EcBmBMGTManualStart <MODE> --insertonly --metg --collectionfile /path/to/collections.txt		
13	Verification		#comment
14	V-1 Verify that the operation in S-1 results in l) HTTP PUT for each non-deleted granule in the specified collection.		#comment
15	Verify an HTTP PUT is sent for granules G44, G45 from the tcp.log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) (More than one PUT may be sent, e.g., if there are network issues.)		
16	Verify no HTTP DELETEs are sent from the tcp.log		
17	Verify no HTTP requests are sent for granules G40 ... G43 from the tcp.log		

TEST DATA:
See Test Case 644.

EXPECTED RESULTS:
See Test Case 644.

1.132 Long Form Verification â€“ Invocation via Command Line - b) Granules in Granule File (ECS-ECSTC-156)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command		

				line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure test collections C1, C2, C3 are enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collections were newly enabled for export in this step, wait for them and their		

#	Action	Expected Result	Notes
	granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure the test granules g1_C1, g2_C1, g1_C2, g2_C2, g1_C3, g2_C3 are in AIM (2 granules per collection): select shortname, versionid, granuleid from amgranule where localgranuleid in ('${LOCALGRANULEID}', ...) If needed, ingest the granules into the public data pool.	Should return 6 rows.	
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 From the command line invoke verification export of metadata for the following: b) Multiple granules, specified in an input file.		#comment
12	Write the test granule IDs to a text file, granules.txt.		
13	Note the current time as t0.		
14	EcBmBMGTManualStart <MODE> --long --metg -gf /path/to/granules.txt		
15	V-1 Verify that the operation in S-1 results in the following exports: b) Single HTTP PUT.		#comment
16	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows, after time t0, a single HTTP PUT request for each granule g1_C1, g1_C2, g1_C2, g2_C2,g1_C3, g2_C3 listed in granules.txt.		
17	Verify that the TCP proxy log shows each granule's URL ends in '?xml_diff=true'. For example, PUT /catalog-rest/providers/DEV08/granules/SC%3AD5TES1.001%3A87900?xml_diff=true HTTP/1.1		

TEST DATA:

refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: w. Single HTTP PUT. x. Single HTTP PUT. y. HTTP PUT for each granule in the collection. z. HTTP PUT for each granule in the collection. aa. HTTP PUT for each specified collection. bb. HTTP PUT for each specified collection. cc. HTTP PUT for each granule and collection in the specified group. dd. HTTP PUT for each granule and collection in the specified group.		

1.133 Long Form Verification â€“ Invocation via Command Line - c) Granules in Collection (ECS-ECSTC-157)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
4	Ensure a test collection C1 is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
5	Ensure collection C1 is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for it and its granules to be exported.		
6	Ensure ECHO has collection C1's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 From the command line invoke verification export of metadata for the following: c) HTTP PUT for each granule in the collection.		#comment
10	Find the number of granules belonging to C1 that are eligible for export (i.e., neither logically deleted nor DFAed): select count(1) from amgranule where shortname = '${SHORTNAME}' and versionid = ${VERSIONID} and deleteeffectivedate is null and deletefromarchive != 'Y'		
11	Note the current time as t0.		
12	EcBmBMGTManualStart <MODE> --long --metg --c <C1_SHORTNAME>.<C1_VERSIONID>		

#	Action	Expected Result	Notes
13	Verification		#comment
14	V-1 Verify that the operation in S-1 results in the following exports: c) HTTP PUT for each granule in the collection.		#comment
15	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows, after time t0, a single HTTP PUT request for each granule in collection C1.		
16	Verify that the TCP proxy log shows each granule's URL ends in '?xml_diff=true'. For example, PUT /catalog-rest/providers/DEV08/granules/SC%3AD5TES1.001%3A87900?xml_diff=true HTTP/1.1		

TEST DATA:
refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: ee. Single HTTP PUT. ff. Single HTTP PUT. gg. HTTP PUT for each granule in the collection. hh. HTTP PUT for each granule in the collection. ii. HTTP PUT for each specified collection. jj. HTTP PUT for each specified collection. kk. HTTP PUT for each granule and collection in the specified group. ll. HTTP PUT for each granule and collection in the specified group.		

1.134 Long Form Verification â€“ Invocation via Command Line - d) Granules in Collection File (ECS-ECSTC-158)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification		

				export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
3	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
4	Ensure a test collection C1 is installed. E.g., the DPL Ingest GUI shows the collection as a configured datatype.		
5	Ensure collection C1 is enabled for collection and granule export: select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for it and its granules to be exported.		

#	Action	Expected Result	Notes
6	Ensure ECHO has collection C1's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
7	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
8	Setup		#comment
9	S-1 From the command line invoke verification export of metadata for the following: d) All granules in a collection, specified in an input file.		#comment
10	Find the number of granules belonging to C1 that are eligible for export (i.e., neither logically deleted nor DFAed): select count(1) from amgranule where shortname = '${SHORTNAME}' and versionid = ${VERSIONID} and deleteeffectivedate is null and deletefromarchive != 'Y'		
11	Write the collection shortname and version <C1_SHORTNAME>.<C1_VERSIONID> into a file collectionfile.txt		
12	Note the current time as t0.		
13	EcBmBMGTManualStart <MODE> --long --metg --cf /path/to/collectionfile.txt		
14	Verification		#comment
15	V-1 Verify that the operation in S-1 results in the following exports: d) HTTP PUT for each granule in the collection.		#comment
16	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows, after time t0, a single HTTP PUT request for each granule in collection C1 (listed in the input file).		
17	Verify that the TCP proxy log shows each granule's URL ends in '?xml_diff=true'. For example, PUT /catalog-rest/providers/DEV08/granules/SC%3AD5TES1.001%3A87900?xml_diff=true HTTP/1.1		

TEST DATA:

refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: mm. Single HTTP PUT. nn. Single HTTP PUT. oo. HTTP PUT for each granule in the collection. pp. HTTP PUT for each granule in the collection. qq. HTTP PUT for each specified collection. rr. HTTP PUT for each specified collection. ss. HTTP PUT for each granule and collection in the specified group. tt. HTTP PUT for each granule and collection in the specified group.		

1.135 Long Form Verification â€“ Invocation via Command Line - e) Multiple Collections (ECS-ECSTC-159)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure the test collections are enabled for collection and granule export. For each of C1, C2, C3, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing a collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID} 		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Setup		#comment
10	S-1 From the command line invoke verification export of metadata for the following: e) Multiple collections, specified on the command line.		#comment
11	(Note: There must be no spaces between commas and collections.) EcBmBMGTManualStart <MODE> --long --metc --c		

#	Action	Expected Result	Notes
	<C1_shortname>.<C1_versionid>,<C2_shortname>.<C2_versionid>,<C3_shortname>.<C3_versionid>		
12	V-1 Verify that the operation in S-1 results in the following exports: e) HTTP PUT for each specified collection.		#comment
13	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for each collection C1, C2, C3 listed in the command line. (There may be more than one put, e.g., if there are network issues.)		
14	Verify that the TCP log proxy shows each request URL ends in '?xml_diff=true'.		

TEST DATA:
refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: uu. Single HTTP PUT. vv. Single HTTP PUT. ww. HTTP PUT for each granule in the collection. xx. HTTP PUT for each granule in the collection. yy. HTTP PUT for each specified collection. zz. HTTP PUT for each specified collection. aaa. HTTP PUT for each granule and collection in the specified group. bbb. HTTP PUT for each granule and collection in the specified group.		

1.136 Long Form Verification â€“ Invocation via Command Line - f) Collections in Collection File (ECS-ECSTC-160)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command**		

				Line] From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure the test collections are enabled for collection and granule export. For each of C1, C2, C3, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update		

#	Action	Expected Result	Notes
	bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing a collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID} 		
8	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
9	Setup		#comment
10	S-1 From the command line invoke verification export of metadata for the following: f) Multiple collections, specified in an input file.		#comment
11	Write the shortname and versionids for C1, C2, C3 into a file collectionfile.txt: <C1_SHORTNAME>.<C1_VERSIONID> <C2_SHORTNAME>.<C2_VERSIONID> <C3_SHORTNAME>.<C3_VERSIONID>		
12	EcBmBMGTManualStart <MODE> --long --metc --cf /path/to/collectionfile.txt		
13	Verification		#comment
14	V-1 Verify that the operation in S-1 results in the following exports: f) HTTP PUT for each specified collection.		#comment
15	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for each collection C1, C2, C3 listed in the input file. (There may be more than one HTTP request, e.g., if there are network issues.)		
16	Verify that the TCP log proxy shows each request URL ends in '?xml_diff=true'.		

TEST DATA:

refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: ccc. Single HTTP PUT. ddd. Single HTTP PUT. eee. HTTP PUT for each granule in the collection. fff. HTTP PUT for each granule in the collection. ggg. HTTP PUT for each specified collection. hhh. HTTP PUT for each specified collection. iii. HTTP PUT for each granule and collection in the specified group. jjj. HTTP PUT for each granule and collection in the specified group.		

1.137 Long Form Verification â€“ Invocation via Command Line - g) Datapool Group (ECS-ECSTC-161)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3, C4 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure collections C1, C2 belong to one data pool group (CG1) and C3, C4 belong to a different data pool group (CG2).		
7	Ensure the test collections are enabled for collection and granule export. For each of C1, C2, C3, C4, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
8	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, C4, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing a collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID} 		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 From the command line invoke verification export of metadata for the following: g) All collections and granules in a Datapool Group, specified on the command line.		#comment
12	(Note; There must be no spaces between commas and groups.) EcBmBMGTManualStart --mode <MODE> --long --metg --metc --		

#	Action	Expected Result	Notes
	group <CG1>,<CG2>		
13	V-1 Verify that the operation in S-1 results in the following exports: g) HTTP PUT for each granule and collection in the specified group.		#comment
14	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for each collection C1, C2, C3 and C4 in the datapool groups CG1, CG2. (There may be more than one HTTP request, e.g., if there are network issues.)		
15	Verify that the TCP proxy log shows an HTTP PUT request for each granule that is neither logically deleted nor DFAed, in each collection C1, C2, C3, C4 in the datapool groups CG1 and CG2. (There may be more than one HTTP request, e.g., if there are network issues.)		
16	*Verify that the TCP log proxy shows each granule's URL ends in '?xml_diff=true'.		

TEST DATA:

refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: kkk. Single HTTP PUT. lll. Single HTTP PUT. mmm. HTTP PUT for each granule in the collection. nnn. HTTP PUT for each granule in the collection. ooo. HTTP PUT for each specified collection. ppp. HTTP PUT for each specified collection. qqq. HTTP PUT for each granule and collection in the specified group. rrr. HTTP PUT for each granule and collection in the specified group.		

1.138 Long Form Verification â€“ Invocation via Command Line - h) Group File (ECS-ECSTC-162)

DESCRIPTION:

	S	492	1	**[Long Form Verification – Invocation via Command Line]** From the command line invoke verification export of metadata for the following: a. Multiple granules, specified on the command line. b. Multiple granules, specified in an input file. c. All granules in a collection, specified on the command line. d. All granules in a collection, specified in an input file. e. Multiple collections, specified on the command line. f. Multiple collections, specified in an input file. g. All collections and granules in a Datapool Group, specified on the command line. h. All collections and granules in a Datapool Group, specified in an input file.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure test collections C1, C2, C3, C4 are installed. E.g., the DPL Ingest GUI shows the collections as configured datatypes.		
6	Ensure collections C1, C2 belong to one data pool group (CG1) and C3, C4		

#	Action	Expected Result	Notes
	belong to a different data pool group (CG2).		
7	Ensure the test collections are enabled for collection and granule export. For each of C1, C2, C3, C4, select granuleexportflag , collectionexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If either export flag is not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If a collection was newly enabled for export in this step, wait for it and its granules to be exported.		
8	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, C4, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing a collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID} 		
9	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
10	Setup		#comment
11	S-1 From the command line invoke verification export of metadata for the following: Copy the name of the collection h) All collections and granules in a Datapool Group, specified in an input file.		#comment
12	Write the group name for datapool groups CG1 and CG2, one per line, into a text file, groupfile.txt.		
13	EcBmBMGTManualStart <MODE> --long --metg --metc --groupfile /path/to/groupfile.txt		
14	Verification		#comment
15	V-1 Verify that the operation in S-1 results in the following exports: h) HTTP PUT for each granule and collection in the specified group.		#comment
16	Verify that the TCP proxy log (/tools/common/test/BE_82_01/proxy/<MODE>/tcp.log) shows an HTTP PUT request for each Collection C1, C2, C3 and C4 in the datapool groups CG1, CG2. (There may be more than one HTTP request, e.g., if there are network issues.)		
17	Verify that the TCP proxy log shows an HTTP PUT request for each granule that is neither logically deleted nor DFAed, in each Collection C1, C2, C3, C4 in the datapool groups CG1 and CG2 in the input file. (There		

#	Action	Expected Result	Notes
	may be more than one HTTP request, e.g., if there are network issues.)		
18	Verify that the TCP log proxy shows each granule's URL ends in '?xml_diff=true'.		

TEST DATA:
refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: sss. Single HTTP PUT. ttt. Single HTTP PUT. uuu. HTTP PUT for each granule in the collection. vvv. HTTP PUT for each granule in the collection. www. HTTP PUT for each specified collection. xxx. HTTP PUT for each specified collection. yyy. HTTP PUT for each granule and collection in the specified group. zzz. HTTP PUT for each granule and collection in the specified group.		

1.139 Datapool URLs[S-6] EcDlCleanupGranules (ECS-ECSTC-348)

DESCRIPTION:

	S	60	1	**[Datapool URLs]** Find a collection which is enabled for collection and granule export.		
	S	60	2	Stop the BMGT servers.		
	S	60	3	For the collection identified in S-1: a. Publish a granule, and then delete it. b. Publish another granule, and then unpublish it.		

	S	60	4	Bring up BMGT.		
	S	60	5	For the collection identified in S-1: aaaa. Publish a granule. bbbb. Unpublish another granule.		
	S	60	6	Perform the following Online Archive repair functions: f. Execute EcDlCleanupGranules on a public granule. g. Execute EcDlCleanupGranules on a browse granule linked to a public granule. h. Execute EcDlCleanupGranules on a QA granule linked to a public granule. i. Execute EcDlCleanupGranules on a PH granule linked to a public granule. j. Execute EcDlCleanupGranules on a HDF MAP granule linked to a public granule.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-conditions		#comment
2	Use /tools/common/test/BE_82_01/bin/xpath to search for XML elements.		#comment
3	Ensure BMGT configuration is complete and correct (config files, properties files, database settings, etc.).		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database: /tools/postgres/current32/bin/psql -U ${USERNAME} -h f4dbl03 -d ecs select public.set_search_path('AIM', '${MODE}'); show search_path;	Should list the schemas installed in the mode. Should include 'aim_${MODE}'.	
5	Ensure the test collection is installed. E.g., the DPL Ingest GUI shows it as configured datatypes.		
6	Ensure the test collection is enabled for collection and granule export: select collectionexportflag, granuleexportflag from bg_collection_configuration where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If both flags are not 'Y', set them: update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = ${SHORTNAME} and versionid = ${VERSIONID} If the collection was enabled for export in this step, wait for it and its granules to be exported.		

#	Action	Expected Result	Notes
7	Ensure ECHO has the test collection's metadata: curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/${URL_ENCODED_DATASETID} If ECHO is missing the collection, export it: EcBmBMGTManualStart ${MODE} --metc --collections ${SHORTNAME}.${VERSIONID}		
8	Ensure granules gA, gC, gD, and gE are in the public data pool (publishtime is non-null): select shortname, versionid, granuleid, publishtime from amgranule where granuleid in (${gA_GRANULEID}, ${gC_GRANULEID}, ${gD_GRANULEID}, ${gE_GRANULEID}) If needed, publish the granules: EcDlPublishStart $MODE -ecs -g ${gA_GRANULEID},${gC_GRANULEID},${gE_GRANULEID},${gE_GRANULEID}		
9	Ensure granule gC is linked to an associated QA granule: select qaid from amqagranulexref where scienceid = ${gC_GRANULEID} Note the qaid as gC_QAID.		
10	Ensure granule gD is linked to an associated PH granule: select phid from amphgranulexref where scienceid = ${gD_GRANULEID} Note the phid as gD_PHID.		
11	Ensure granule gE is linked to an associated HDF_MAP granule: select hdfmapid from amhdfmapgranulexref where scienceid = ${gE_GRANULEID} Note the hdfmapid as gE_HDFMAPID.		
12	Ensure ECHO has the test granules' metadata. For each of gA, gC, gD, gE, curl -k -H Echo-Token:${TOKEN} https://testbed.echo.nasa.gov/catalog-rest/providers/${PROVIDERID}/granules/${URL_ENCODED_GEOID} If ECHO is missing the granules, export them: EcBmBMGTManualStart ${MODE} --metg --granules ${gA_GRANULEID},${gC_GRANULEID},${gD_GRANULEID},${gE_GRANULEID}		
13	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic.		
14	Setup		#comment
15	S-1 Find a collection which is enabled for collection and granule export.		#comment
16	Done in preconditions.		#comment
17	S-4 Bring up BMGT.		#comment

#	Action	Expected Result	Notes
18	Start BMGT: EcBmBMGTAppStart $MODE		
19	S-6 Perform the following Online Archive repair functions: a. Execute EcDlCleanupGranules on a public granule. b. DELETED c. Execute EcDlCleanupGranules on a QA granule linked to a public granule. d. Execute EcDlCleanupGranules on a PH granule linked to a public granule. e. Execute EcDlCleanupGranules on a HDF MAP granule linked to a public granule.		#comment
20	Create text file with IDs of the granules to clean up: gA_GRANULEID gC_QAID gD_PHID gE_HDFMAPID		
21	Note the current time as t0.		
22	Execute the Data Pool Cleanup Granules utility on the granule IDs file: EcDlCleanupGranules.pl $MODE -cleanuptype ecs -file /path/to/granuleids		
23	Run EcDlRestoreOlaFromTapeStart to trigger granule exports: EcDlRestoreOlaFromTapeStart $MODE -contents granuleids -file /path/to/granuleids		
24	Publish the granules to create events: EcDlPublishUtilityStart ${MODE} -ecs -file /path/to/granuleids		
25	Verification		#comment
26	V-5 Verify that each of the operations in S-6 results in the export of granule metadata for the associated science granule containing updated URLs reflecting the modifications made by EcDlCleanupGranules.		#comment
27	Verify the TCP proxy log indicates a PUT was sent for granules gA, gC, gD, gE after time t0.		
28	Verify each S-6 granule's PUT has an XML body.		
29	Save each of the S-6 granules' exported metadata to a separate XML file.		
30	Verify each granule's exported metadata contains data pool URLs to the science granules: xpath '//OnlineAccessURL/URL' granule.xml		
31	Verify the exported metadata for each of gC, gD, and gE contains data pool URLs to the associated granules (gC_QA, gD_PH, gE_HDFMAP): xpath '//OnlineResource' granule.xml		
32	Verify each data pool URL refers to the current granule location by either using ftp or the data pool filesystem.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
60 S-3a,b S-5a,b			MOP03N.005	4 granules belonging to a collection not configured to be public on ingest (g1,g2,g3,g4).			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6a			AE_DySno.002	1 granule belonging to a collection configured to be public on ingest (gA).			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6c			AE_DySno.002	1 public granule with an associated QA granule (gC, qC)			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6d			AE_DySno.002	1 public granule with an associated PH granule (gD, pD)			/sotestdata/DROP_802/BE_82_01/Criteria/060	
60 S-6e			AE_DySno.002	1 public granule with an associated HDF MAP granule (gE, mE)			/sotestdata/DROP_802/BE_82_01/Criteria/060	

EXPECTED RESULTS:

	V	60	1	Verify that for S-3, subclause a, an HTTP DELETE is exported.		
	V	60	2	Verify that for S-3, subclause b, granule metadata is exported, but contains no online access or online resource URLs.		
	V	60	3	Verify that for S-5, subclause a, full granule metadata is exported, including online access and online resource URLs.		
	V	60	4	Verify that for S-5, subclause b, granule metadata is exported, but contains no online access or online resource URLs.		

	V	60	5	Verify that each of the operations in S-6 results in the export of granule metadata for the associated science granule containing updated URLs reflecting the modifications made by EcDlCleanupGranules.		

2 SD_82_01

2.1 Invalid Collection/Series Level Schema Referenced (ECS-ECSTC-117)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	80	1	**Invalid Collection/Series Level Schema Referenced** Preparean ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting.	FC	S-AIM-00757
S	80	2	Prepare a schema file that will cause a validation failure for valid SMAP collection metadata.	FC	S-AIM-00757
S	80	3	Prepare a modified SMAP ISO-19115 metadata file that will cause a validation failure for valid SMAP metadata schema.	FC	S-AIM-00757
S	80	4	Replace the ESDT Maintenance GUI jar file schema with the prepared modified schema.	FC	S-AIM-00757
S	80	5	Using the ESDT Maintenance GUI, install the ESDT with valid metadata.	FC	S-AIM-00757
S	80	6	Replace the ESDT Maintenance GUI jar file schema with the correct SMAP collection metadata schema.	FC	S-AIM-00757
S	80	7	Using the ESDT Maintenance GUI, install the ESDT with invalid metadata.	FC	S-AIM-00757
S	80	8	Using the ESDT Maintenance GUI, install the ESDT with valid metadata.	FC	S-AIM-00757

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Note that this is 3 independent tests: 1. Attempt to install a valid ISO/SMAP collection against an invalid schema file. 2. Attempt to install an invalid ISO/SMAP collection against a valid schema file. 3. Install a valid ISO/SMAP collection against a valid schema file.		#comment
3	Ensure test collection under /sotestdata/DROP_802/SD_82_01/Criteria/080/080_1 is not installed.		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting.		#comment
6	Use the files under /sotestdata/DROP_802/SD_82_01/Criteria/080/080_1. sed -n '/DataModelType/,/DataModelType/{;s/^ *Value *= *//p;}' DsESDTSmSPL1AP.003.desc		
7	S-2 Prepare a schema file that will cause a validation failure for valid SMAP collection metadata.		#comment
8	In /sotestdata/DROP_802/SD_82_01/Criteria/080/080_2/EcDsAmIsoSchemas_bad.jar the file /Sm/schema/ISO-SMAP/gmd/metadataApplication.xsd has been modified to cause a validation failure for valid SMAP collection metadata.		
9	S-3 Prepare a modified SMAP ISO-19115 metadata file that will cause a validation failure for valid SMAP metadata schema.		#comment
10	Use the files under /sotestdata/DROP_802/SD_82_01/Criteria/080/080_3.		
11	S-4 Replace the ESDT Maintenance GUI jar file schema with the prepared modified schema.		#comment
12	Stop the Tomcat instance of the ESDT Maintenance GUI.		
13	On the ESDT Maintenance GUI host (e.g., f5dpl01v), as cm${mode}, move /usr/ecs/OPS/CUSTOM/WWW/DSS/ESDTMaint/WEB-INF/lib/EcDsAmIsoSchemas.jar to a temporary location to restore later.		
14	On the ESDT Maintenance GUI host, as cm${mode}, copy /sotestdata/DROP_802/SD_82_01/Criteria/080/080_2/EcDsAmIsoSchemas_bad.jar to /usr/ecs/OPS/CUSTOM/WWW/DSS/ESDTMaint/WEB-INF/lib/EcDsAmIsoSchemas.jar		

#	Action	Expected Result	Notes
15	Start the Tomcat instance of the ESDT Maintenance GUI.		
16	S-5 Using the ESDT Maintenance GUI, install the ESDT with valid metadata.		#comment
17	On the ESDT Maintenance GUI host, as cmshared, copy the ISO/SMAP collection files from the 080_1 test data directory to the ESDT Maintenance GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/080/080_1/* /usr/ecs/${MODE}/CUSTOM/data/ESS		
18	Log in to the ESDT Maintenance GUI.		
19	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
20	Find collection SPL1AP.003 in the list of collections, and enable its check box.		
21	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
22	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.	
23	V-1 Verify that the insert operation fails due to failing to validate the series level metadata file due to the modified schema.		#comment
24	Verify the ESDT Maintenance GUI indicates the collection failed installation because the series-level metadata file failed validation.		
25	V-2 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
26	Verify no files from the test collection were copied to the small file archive descriptor or MCF directories (within the last 5 minutes): find /stornext/smallfiles/${MODE}/{descriptor,mcf} -mmin -5 -name '*SPL1AP.003*'		
27	S-6 Replace the ESDT Maintenance GUI jar file schema with the correct SMAP collection metadata schema.		#comment
28	Stop the Tomcat instance of the ESDT Maintenance GUI.		
29	On the ESDT Maintenance GUI host, restore the original EcDsAmIsoSchemas.jar file.		
30	Start the Tomcat instance of the ESDT Maintenance GUI.		
31	S-7 Using the ESDT Maintenance GUI, install the ESDT with invalid metadata.		#comment
32	As cmshared, copy the ISO/SMAP collection files from the 080_3 test data directory to the ESDT Maintenance GUI directory: cp		

#	Action	Expected Result	Notes
	/sotestdata/DROP_802/SD_82_01/Criteria/080/080_3/* /usr/ecs/${MODE}/CUSTOM/data/ESS		
33	Log in to the ESDT Maintenance GUI.		
34	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
35	Find collection SPL1AP.003 in the list of collections, and enable its check box.		
36	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
37	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.	
38	V-3 Verify that the insert operation fails due to failing to validate the series level metadata file due to the modified schema.		#comment
39	Verify the ESDT Maintenance GUI indicates the collection failed installation because the series-level metadata file failed validation.		
40	V-4 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
41	Verify no files from the test collection were copied to the small file archive descriptor or MCF directories (within the last 5 minutes): find /stornext/smallfiles/${MODE}/{descriptor,mcf} -mmin -5 -name '*SPL1AP.003*'		
42	S-8 Using the ESDT Maintenance GUI, install the ESDT with valid metadata.		#comment
43	As cmshared, copy the ISO/SMAP collection files from the 080_1 test data directory to the ESDT Maintenance GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/080/080_1/* /usr/ecs/${MODE}/CUSTOM/data/ESS		
44	Log in to the ESDT Maintenance GUI.		
45	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
46	Find collection SPL1AP.003 in the list of collections, and enable its check box.		
47	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
48	Click on the 'Ok' button.		
49	V-5 Verify that the install is successful.		#comment
50	Verify the ESDT Maintenance GUI displays a page indicating that the ESDT		

#	Action	Expected Result	Notes
	installation succeeded.		
51	Verify test collection files were copied to the small file archive (within the last 5 minutes): find /stornext/smallfiles/${MODE}/{descriptor,mcf} -mmin -5 -name '*SPL1AP.003*'		

TEST DATA:

Crit id	Critccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
80		A valid ISO/SMAP collection	SPL1AP.003	A new SMAP ISO-19115 collection with a valid ISO-compliant series level metadata file.			/sotestdata/DROP_802/SD_82_01/Criteria/080/080_1	
80		An ISO/SMAP collection schema that will cause a valid ISO/SMAP collection to fail validation.					/sotestdata/DROP_802/SD_82_01/Criteria/080/080_2	
80		An ISO/SMAP collection with an XML file that will fail validation.	SPL1AP.003				/sotestdata/DROP_802/SD_82_01/Criteria/080/080_3	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	80	1	Verify that the insert operation fails due to failing to validate the series level metadata file due to the modified schema.	FC	S-AIM-00757
V	80	2	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.	FC	S-AIM-00757
V	80	3	Verify that the insert operation fails due to failing to validate the series level metadata file due to the modified schema.	FC	S-AIM-00757
V	80	4	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.	FC	S-AIM-00757
V	80	5	Verify that the install is successful.	FC	S-AIM-00757

2.2 Invalid Collection/Series Metadata Attribute Value (ECS-ECSTC-118)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	90	1	**Invalid Collection/Series Metadata Attribute Value** Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Edit the ISO-compliant series level metadata file and change an attribute value to a value that is invalid according to the schema specified in the metadata file.	EC	S-AIM-00140
S	90	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	EC	S-AIM-00140
S	90	3	Using the ESDT Maintenance GUI, install the ESDT.	EC	S-AIM-00140

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C9 is not installed (it should not be possible to install it).		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Edit the ISO-compliant series level metadata file and change an attribute value to a value that is invalid according to the schema specified in the metadata file.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/090.		
7	Ensure that in collection C9's descriptor file (*.desc), the DataModelType attribute has the value 'ISO-SMAP'.		
8	Ensure collection C9's series-level metadata file (*.series.xml) violates the specified schema (see the README.txt file in the test data directory for specifics).		
9	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
10	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
11	As cmshared, copy collection C9's descriptor and dataset XPath files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/090/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
12	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
13	Note the current time as t0.		
14	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
15	Find collection C9 in the list of collections, and enable its check box.		
16	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	

#	Action	Expected Result	Notes
17	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.	
18	Verification		#comment
19	V-1 Verify that the insert operation fails due to failing to validate the series level metadata file due to failing the schema validation because of the invalid metadata value.		#comment
20	Verify the ESDT Maintenance GUI indicates collection C9's installation failed because the series-level metadata failed validation becuase of the invalid metadata value.		
21	V-2 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
22	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		
23	Verify no files associated with collection C9 is present: ls -l *<C9_ESDT>*		
24	Verify no MCF file was created for C9: ls -l /stornext/smallfiles/<MODE>/mcf/*<C9_ESDT>*		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
90		Collection C9		A new SMAP ISO-19115 collection with an ISO-compliant series level metadata file that has an invalid attribute value, according to the schema specified in the metadata file.			/sotestdata/DROP_802/SD_82_01/Criteria/090	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	90	1	Verify that the insert operation fails due to failing to validate the series level metadata file due to failing the schema validation because of the invalid metadata value.		
V	90	2	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		

2.3 XSD Generation Verification-ISO (ECS-ECSTC-119)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	100	1	**XSD Generation Verification- ISO** Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting.	FC	S-AIM-00150
S	100	2	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00150

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C10 is not installed.		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment

#	Action	Expected Result	Notes
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/100.		
7	Ensure that in collection C10's descriptor file (*.desc), the DataModelType element has the value 'ISO-SMAP'.		
8	S-2 Using the ESDT Maintenance GUI, install the ESDT.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C10's descriptor and dataset XPath files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/100/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	Note the current time		
12	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation.	
13	Find collection C10 in the list of collections, and enable its check box.		
14	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
15	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.	
16	Verification		#comment
17	V-1 Verify by inspection that the XSD file was not generated for the ESDT.		#comment
18	Verify no schema file was created for collection C10 in the small file archive collection directory: ls -l /stornext/smallfiles/<MODE>/descriptor/*<C10_ESDT>*.xsd		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
100		Collection C10.		A new SMAP ISO-19115 collection.			/sotestdata/DROP_802/SD_82_01/Criteria/100	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	100	1	Verify by inspection that the XSD file was not generated for the ESDT.		

2.4 XSD Generation Verification - ECS (ECS-ECSTC-120)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	110	1	**XSD Generation Verification -ECS** Prepare an ESDT descriptor for a new collection with the data model type attribute set to an ECS Metadata Model setting.	FC	S-AIM-00150
S	110	2	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00150

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C11 is not installed.		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to an ECS Metadata Model setting.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/110.		
7	Ensure that in collection C11's descriptor file (*.desc), the DataModelType element has the value 'ECS'.		

#	Action	Expected Result	Notes
8	S-2 Using the ESDT Maintenance GUI, install the ESDT.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C11's descriptor and dataset XPath files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/110/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	Note the current time as t0.		
12	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation.	
13	Find collection C11 in the list of collections, and enable its check box.		
14	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
15	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.=	
16	Verification		#comment
17	V-1 Verify by inspection that the XSD file was generated for the ESDT and resides in the ESDT descriptor target directory.		#comment
18	Verify a schema file was created for collection C11 in the small file archive collection directory: ls -l /stornext/smallfiles/<MODE>/descriptor/*<C11_ESDT>*.xsd		
19	Verify collection C11's schema file's timestamp is on or after time t0.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
110		Collection C11		A new ECS collection.			/sotestdata/DROP_802/SD_82_01/Criteria/100	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	110	1	Verify by inspection that the XSD file was generated for the		

			ESDT and resides in the ESDT descriptor target directory.		

2.5 Data Model Type Attribute Missing (ECS-ECSTC-113)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	40	1	**Data Model Type Attribute Missing** Prepare an ESDT descriptor for a new collection without a data model type attribute.	FC	S-AIM-00110, S-AIM-00160
S	40	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	FC	S-AIM-00110, S-AIM-00160
S	40	3	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00110, S-AIM-00160

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C4 is not installed. If it is, use the ESDT maintenance GUI to remove it as follows: On the ESDT Maintenance GUI main page, select collection C4. Select the 'Delete selected ESDTs' button. After a few seconds, the GUI should display a message indicating the ESDT was successfully 'processed'.		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h		

#	Action	Expected Result	Notes
	f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection without a data model type attribute.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/040.		
7	Ensure collection C4's descriptor file has no DataModelType element.		
8	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C4's files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/040/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	Note the current time.		
13	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
14	Find collection C4 in the list of collections, and enable its check box.		
15	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
16	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT has been installed successfully.	
17	Verification		#comment
18	V-1 Verify by inspecting the AIM database that the collection was correctly installed and that the database attribute is set to an ECS Metadata Model setting.		#comment
19	Query the AIM database for collection C4: select * from amcollection where shortname = <C4_SHORTNAME> and versionid = <C4_VERSIONID>		
20	Verify the datamodeltype is 'ECS'.		
21	Verify the other values are correct, using the descriptor file as a guide.		
22	V-2 Verify that the files for the given ESDT were copied into the ESDT descriptor and MCF target directories.		#comment
23	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		

#	Action	Expected Result	Notes
24	Verify collection C4's descriptor file is in the collection directory: ls -l *<C4_ESDT>*.desc		
25	Verify C4's descriptor file from previous step has the same contents as the file in the test data directory, using diff: diff *<C4_ESDT>*.desc /sotestdata/DROP_802/SD_82_01/Criteria/010/*<C4_ESDT>*.desc		
26	Verify an XML schema file was created for C4: ls -l *<C4_ESDT>*.xsd		
27	Verify an MCF file was created for C4: ls -l /stornext/smallfiles/<MODE>/mcf/<C4_ESDT>*.mcf		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
40		Collection C4		A new collection without a data model type attribute.			/stornext/smallfiles/<MODE>/descriptor	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	40	1	Verify by inspecting the AIM database that the collection was correctly installed and that the database attribute is set to an ECS Metadata Model setting.		
V	40	2	Verify that the files for the given ESDT were copied into the ESDT descriptor and MCF target directories.		

2.6 Collection/Series Level XPath File Missing Verification (ECS-ECSTC-114)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	50	1	**Collection/Series Level XPATH File Missing Verification** Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Ensure that the collection level XPATH file is not available.	FC	S-AIM-00120, S-AIM-00130
S	50	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	FC	S-AIM-00120, S-AIM-00130
S	50	3	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00120, S-AIM-00130

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C5 is not installed (it should not be possible to install it).		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Ensure that the collection level XPATH file is not available.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/050.		
7	Ensure collection C5 has no collection-level XPath file (*.series.xpath).		

#	Action	Expected Result	Notes
8	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C5's descriptor, XML, and dataset (granule) XPath files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/050/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	Note the current time.		
13	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
14	Find collection C5 in the list of collections, and enable its check box.		
15	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
16	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.	
17	Verification		#comment
18	V-1 Verify that the ESDT install operation fails due to the missing collection level XPATH file.		#comment
19	Verify the ESDT Maintenance GUI indicates collection C5's installation failed due to the missing collection-level XPath file.		
20	V-2 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
21	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		
22	Verify no files associated with collection C5 are present: ls -l *<C5_ESDT>*		
23	Verify no MCF file was created for C5: ls -l /stornext/smallfiles/<MODE>/mcf/*<C5_ESDT>*		

TEST DATA:

Crit id	Critccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
50		Collection C5		A new SMAP ISO-19115 collection without a collection level XPath file.			/sotestdata/DROP_802/SD_82_01/Criteria/050	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	50	1	Verify that the ESDT install operation fails due to the missing collection level XPATH file.		
V	50	2	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		

2.7 Granule/Dataset Level XPath File Missing (ECS-ECSTC-115)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	60	1	**Granule/Dataset Level XPATH File Missing** Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Ensure that the granule level XPATH file is not available.	FC	S-AIM-00120, S-AIM-00130
S	60	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	FC	S-AIM-00120, S-AIM-00130

S	60	3	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00120, S-AIM-00130

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C6 is not installed (it should not be possible to install it).		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Ensure that the granule level XPath file is not available.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/060.		
7	Ensure collection C6 has no granule-level XPath file (*.dataset.xml).		
8	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C6's descriptor, XML, and dataset (granule) XPath files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/060/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	Note the current time		
13	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation.	
14	Find collection C6 in the list of collections, and enable its check box.		
15	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
16	Click on the 'Ok' button.	After a second or two, the GUI should	

#	Action	Expected Result	Notes
		display a page indicating that the ESDT installation failed.	
17	Verification		#comment
18	V-1 Verify that the ESDT install operation fails due to the missing granule level XPath file.		#comment
19	Verify the ESDT Maintenance GUI indicates collection C6's installation failed due to the missing granule-level XPath file.		
20	V-2 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
21	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		
22	Verify no files associated with collection C6 is present: ls -l *<C6_ESDT>*		
23	Verify no MCF file was created for C6: ls -l /stornext/smallfiles/<MODE>/mcf/*<C6_ESDT>*		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
60		Collection C6.		A new SMAP ISO-19115 collection without a granule level XPath file.				

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	60	1	Verify that the ESDT install operation fails due to the missing granule level XPATH file.		
V	60	2	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		

2.8 Collection/Series Level Metadata File Missing (ECS-ECSTC-116)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	70	1	**Collection/Series Level Metadata File Missing** Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Ensure that the ISO-compliant series (collection) level metadata file is not available.	EC	S-AIM-00120, S-AIM-00130
S	70	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	FC	S-AIM-00120, S-AIM-00130
S	70	3	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00120, S-AIM-00130

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C7 is not installed (it should not be possible to install it).		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Ensure that the ISO-compliant series (collection) level metadata file is not available.		#comment

#	Action	Expected Result	Notes
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/070.		
7	Ensure collection C7 has no series-level metadata file (*.series.xml).		
8	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C7's descriptor and dataset XPath files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/070/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	Note the current time		
13	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation.	
14	Find collection C7 in the list of collections, and enable its check box.		
15	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
16	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.=	
17	Verification		#comment
18	V-1 Verify that the ESDT install operation fails due to the missing series (collection) level metadata file.		#comment
19	Verify the ESDT Maintenance GUI indicates collection C7's installation failed due to the missing series-level XML file.		
20	V-2 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
21	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		
22	Verify no files associated with collection C7 are present: ls -l *<C7_ESDT>*		
23	Verify no MCF file was created for C7: ls -l /stornext/smallfiles/<MODE>/mcf/*<C7_ESDT>*		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
70		Collection C7		A new SMAP ISO-19115 collection without an ISO-compliant series (collection) level metadata file.			/sotestdata/DROP_802/SD_82_01/Criteria/070	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	70	1	Verify that the ESDT install operation fails due to the missing series (collection) level metadata file.		
V	70	2	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		

2.9 XML metadata files for ISO Distribution (ECS-ECSTC-125)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	160	1	**XML metadata files for ISO Distribution** Order a granule for a collection whose metadata model type is ISO, specifying a PULL distribution. Ensure that the user's email address is configured to be a .met-file recipient.	FC	S-OMS-00100

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Determine the following parameters: <OM Host> - the host on which the Order Manager is executing		The granule must be available before it can be ordered.
3	Setup		#comment
4	S-1 Order a granule for a collection whose metadata model type is ISO, specifying a PULL distribution. Ensure that the user's email address is configured to be a .met-file recipient.		#comment
5	Configure the OMS to offer packaging options for HTTP Pull. Using a cmshared command prompt on <OM Host>, edit the OrderManager configuration file after first making a copy: > cd /usr/ecs/<MODE>/CUSTOM/cfg > cp EcOmOrderManager.cfg EcOmOrderManager.cfg.<ddmmYYYY> > vi EcOmOrderManager.cfg Set the following configuration values: DownloadType = HTTP FTP_PULL_OPTIONS = GZIP UNIX TAR ZIP 		I usually move the config file to a new name, then copy it back to its original name. That preserves the permissions on the original and makes cmshared the owner of the originally-named copy. Remember to include a restore step at the end.
6	On the OM GUI main page, expand the 'OM Configuration' menu option list and select 'Media'. 	This should display a list of media configuration options.	
7	If the 'Pull Gran Dpl Time (days) is not set to 1, make a note of its current value, then set it to 1 and click the 'Apply' button at the bottom of the page		
8	On the OM GUI, expand the 'OM Configuration' menu option list and select Odl Metadata Users		
9	If labuser@f4eil01.edn.ecs.nasa.gov is not present in list, enter labuser@f4eil01.edn.ecs.nasa.gov in the text box and click the 'Add New User' button at the bottom of the page		
10	Configure an OMSCLI acquire script parameter file: ECSUSERPROFILE = ECSGuest PRIORITY = NORMAL DDISTMEDIATYPE = FtpPull DDISTMEDIAFMT = FILEFORMAT USERSTRING = smap_test DDISTNOTIFYTYPE = MAIL NOTIFY = labuser@f4eil01.edn.ecs.nasa.gov		
11	Configure an OMSCLI acquire script geoid file: SC:<SHORTNAME>.<VERSIONID>:<GRANULEID>		
12	Using a cmshared command prompt on <acquire Host>, run the acquire		The <random tag>

#	Action	Expected Result	Notes
	script > cd /usr/ecs/<MODE>/CUSTOM/utilities > ./acquire <MODE> -p <path to parameter file> -f <path to geoidfile> -t <random tag>		argument is not optional and must be unique for each run of the acquire script. It can be any string of characters.
13	Verification		#comment
14	V-1 When the DN is received, ensure that the metadata file for the granule is in XML format.		#comment
15	Determine the FTPDIR from the DN. > ssh f4eil01 > vi /var/spool/mail/labuser		
16	Navigate to the FTPDIR and verify metadata file is in XML format > cd /datapool/<MODE>/user/<FTPDIR> > vi <metadata file>		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
160				An ISO granule.			/sotestdata/DROP_802/SD_82_01/Criteria/160	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	160	1	When the DN is received, ensure that the metadata file for the granule is in XML format.		

2.10 QA Update Failure Verification (ECS-ECSTC-126)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	170	1	**QA Update Failure Verification** Using the QA Update Utility, perform a QA Update operation	EC	S-AIM-00190

			on a granule for a collection whose metadata model type is ISO.		

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure the test SMAP granule has been ingested into the public data pool.		#comment
3	Ensure an ingest provider is configured to ingest SMAP granules: Provider Type: Polling with DR Preprocessing Type: SIPS Max Active Data Volume: 1000.00 Max Active Granules: 100 Transfer Type: Local Notification Method: Email Only E-Mail address: labuser@f4eil01.edn.ecs.nasa.gov		
4	Ensure the test collection's data type has been installed.		
5	Ensure the test collection's data type is configured in the DPL Ingest GUI > Configuration > Data Types to be public on ingest.		
6	Copy the test granule's PDR file from /sotestdata/DROP_802/SD_82_01/Criteria/170 to the test provider's polling directory.		
7	Wait for the DPL Ingest GUI to show the granule has completed successfully.		
8	Get the granule's granuleid (adjust interval as needed): select granuleid from amgranule where shortname = <SHORTNAME> and versionid = <VERSIONID> and archivetime > now() - interval '5 minutes';		
9	Ensure a PostgreSQL client is available: From a command prompt, connect to the database: $ /tools/postgres/current32/bin/psql -U <user> -d ecs -h f4dbl03 At the Postgres client prompt, set the search path to access the appropriate mode database: > select public.set_search_path('aim', '<MODE>');		
10	Find the granule's metadata XML file in the small file archive (<METADATA_ARCHIVE_PATHNAME>): select x.path || '/' || m.archivemetfilename from amgranule g join ammetadatafile m on g.granuleid = m.granuleid join dsmdxmlpath x on m.archivepathid = x.archivepathid where g.granuleid = <GRANULEID>		

#	Action	Expected Result	Notes
11	Make a local copy of the granule's metadata file to compare against later: mkdir -p /tools/common/test/SD_82_01/criteria/170 cd !$ cp -p <METADATA_ARCHIVE_PATHNAME> .		
12	Setup		#comment
13	S-1 Using the QA Update Utility, perform a QA Update operation on a granule for a collection whose metadata model type is ISO.		#comment
14	Move the QAUU properties file to a new name, and make a working copy: cd /usr/ecs/<MODE>/CUSTOM/cfg mv EcDsAmQaUpdateUtility.properties EcDsAmQaUpdateUtility.properties.SD_82_01_C170 cp EcDsAmQaUpdateUtility.properties.SD_82_01_C170 EcDsAmQaUpdateUtility.properties		
15	Append lines to the QAUU properties file to allow a new site to run QAUU: SD8201C170_EMAIL_FROM_ADDRESSES=labuser@edn.ecs.nasa.gov SD8201C170_EMAIL_REPLY_ADDRESS=labuser@edn.ecs.nasa.gov SD8201C170_NOTIFICATION_ON_SUCCESS=N		
16	Add a row to the dsqamutesdtsite table: insert into dsqamutesdtsite values(<SHORTNAME>, 'SD8201C170');		
17	Create a QAUU request file to update the test granule.		#comment
18	Name the file according to following pattern: <MODE>_<Site>_QAUPDATE<description>.<YY><MM><DD><hh><mm><ss> E.g., DEV08_SD8201C170_QAUPDATE_SD_82_01_C170.130411144600		
19	Ensure the file contents are as follows, with <TAB> replaced by an actual TAB character: From SITE begin QAMetadataUpdate Science GranuleUR SHORTNAME<TAB>VERSIONID<TAB>GRANULEUR<TAB>PARAMETER_NAME<TAB>SCENCE_FLAG<TAB>COMMENT end QAMetadataUpdate For example, using granule SC:SPL1CS0.003:123456, From SD8201C170 begin QAMetadataUpdate Science GranuleUR SPL1CS0<TAB>3<TAB>SC:SPL1CS0.003:123456<TAB>Surface Soil Moisture<TAB>Passed<TAB>Updated for SD_82_01 C170 end QAMetadataUpdate		
20	Copy the request file into the QAUU request file directory (the value of QA_REQUEST_DIR in EcDsAmQaUpdateUtility.properties): <br		

#	Action	Expected Result	Notes
	/>/usr/ecs/<MODE>/CUSTOM/data/DSS/QAUU/QAUURequest		
21	Run QAUU (if the request file is the only one in the request directory, the filename may be omitted): cd /usr/ecs/<MODE>/CUSTOM/utilities ./EcDsAmQAUUStart <MODE> -file <REQUEST_FILENAME> -noprompt		
22	Verification		#comment
23	V-1 Verify that the QA update utility displays an appropriate message indicating failure.		#comment
24	Verify EcDsAmQauu.ops0.log reports that the update request failed.		The QAUU 609 implies that the message will only appear in the log: 609-EED-001, Rev 01 4.8.9.3.4 Outputs Output of update events and errors will be always appended to a single log file. If specified as an option, a confirmation prompt will be displayed to the screen.
25	V-2 Verify that the XML metadata file for the granule was not modified by the operation.		#comment
26	Verify the timestamp on the granule's metadata file in the small file archive has not changed: ls -l /tools/common/test/SD_82_01/criteria/170/<METADATA_FILE> ls -l <METADATA_ARCHIVE_PATHNAME>		
27	Verify the granule's metadata contents have not changed: cd /tools/common/test/SD_82_01/criteria/170 diff <METADATA_FILE> <METADATA_ARCHIVE_PATHNAME>		
28	Cleanup		#comment
29	Replace the altered properties file with the original: mv EcDsAmQaUpdateUtility.properties.SD_82_01_C170 EcDsAmQaUpdateUtility.properties		
30	Remove the row added to the ESDT site table: delete from dsqamutesdtsite where site = 'SD8201C170';		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
170				An ISO granule.			/sotestdata/DROP_802/SD_82_01/Criteria/170	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	170	1	Verify that the QA update utility displays an appropriate message indicating failure.		
V	170	2	Verify that the XML metadata file for the granule was not modified by the operation.		

2.11 Ingest XPath Validation (ECS-ECSTC-121)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	140	1	**Ingest XPATH Validation** Prepare a SMAP granule for ingest, including the associated granule level metadata file.	FC	S-DPL-00120
S	140	2	Ingest the SMAP granule	FC	S-DPL-00120

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure an ingest provider is configured to ingest SMAP granules: <br		

#	Action	Expected Result	Notes
	/>Provider Type: Polling with DR Preprocessing Type: SIPS Max Active Data Volume: 1000.00 Max Active Granules: 100 Transfer Type: Local Notification Method: Email Only E-Mail address: labuser@f4eil01.edn.ecs.nasa.gov		
3	Ensure the test collection's data type has been installed.		
4	Ensure the test collection's data type is configured in the DPL Ingest GUI > Configuration > Data Types to be public on ingest.		
5	Ensure a PostgreSQL client is available: From a command prompt, connect to the database: $ /tools/postgres/current32/bin/psql -U <user> -d ecs -h f4dbl03 At the Postgres client prompt, set the search path to access the appropriate mode database: > select public.set_search_path('aim', '<MODE>');		
6	Setup		#comment
7	S-1 Prepare a SMAP granule for ingest, including the associated granule level metadata file.		#comment
8	Use the provided test granules under />/sotestdata/DROP_802/SD_82_01/Criteria/140		
9	S-2 Ingest the SMAP granule.		#comment
10	Copy the granule's PDR file into the test provider's polling location.		
11	Wait for the DPL Ingest GUI to show the granule has completed successfully.		
12	Get the granule's granuleid (adjust interval as needed): select granuleid from amgranule where shortname = <SHORTNAME> and versionid = <VERSIONID> and archivetime > now() - interval '5 minutes';		
13	Verification		#comment
14	V-1 Verify by inspecting the AIM database that the proper metadata values from the granule level metadata file were inserted. [Note that this verifies that the XPath extraction worked correctly.]		#comment
15	Retrieve the ingested test granule details: select * from amgranule where granuleid = <GRANULEID>;		
16	Using the ISO dataset XPath file as a guide, verify the ingested granule metadata in the AIM database matches the metadata in the ISO metadata XML file.	The values should match within the limits of type conversion.	
17	V-2 Verify by inspecting the DataPool and StorNext directories that the SMAP science granule and metadata files were ingested into their proper locations.		#comment
18	Verify that the ingested ISO/SMAP granule metadata file has been copied to		

#	Action	Expected Result	Notes
	the small file archive: diff <SMAP_GRANULE>.xml /stornext/smallfiles/<MODE>/metadata/<ESDT>/<YYYY>.<MM>/<ESDT>.<GRANULEID>.xml		
19	Find the granule metadata file's data pool pathname: select f.absolutefilesystempath || c.groupid || m.onlinemetdirectorypath || m.onlinemetfilename from amgranule g join amcollection c on g.collectionid = c.collectionid join ammetadatafile m on g.granuleid = m.granuleid join dlfilesystems f on c.filesystemlabel = f.filesystemlabel where g.granuleid = <GRANULEID>		
20	Verify that the ingested ISO/SMAP granule metadata file has been copied to the data pool: diff <SMAP_GRANULE_XML> <DATAPOOL_PATHNAME>		
21	Find the granule's data file data pool pathname: select f.absolutefilesystempath || c.groupid || d.directorypath || d.onlinefilename from amgranule g join amcollection c on g.collectionid = c.collectionid join amdatafile d on g.granuleid = d.granuleid join dlfilesystems f on c.filesystemlabel = f.filesystemlabel where g.granuleid = <GRANULEID>		
22	Verify the SMAP granule's data file has been copied to the data pool: diff <SMAP_GRANULE_DATA> <DATAPOOL_PATHNAME>		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
140			SPL1CS0.030	SMAP granule, including the associated granule level metadata file.			/sotestdata/DROP_802/SD_82_01/Criteria/140	
		ISO Dataset (granule) XPath file	SPL1CS0.030				/sotestdata/DROP_802/SD_82_01/Criteria/010	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	140	1	Verify by inspecting the AIM database that the proper metadata values from the granule level metadata file were inserted. [Note that this verifies that the XPATH extraction worked correctly.]		
V	140	2	Verify by inspecting the DataPool & StorNext directories that the SMAP science granule and metadata files were ingested into their proper locations		

2.12 ISO Ingest Attribute Type Conversion (ECS-ECSTC-122)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	145	1	**ISO Ingest Attribute Type Conversion** Prepare a SMAP granule for ingest, including the associated granule level metadata file. In the metadata file, note the values for all of the attributes that will undergo attribute type conversion during Ingest of the granule. [Note that the full list of attributes that will undergo attribute type conversion will be produced by DDR.]	FC	S-DPL-00130
S	145	2	Ingest the SMAP granule	FC	S-DPL-00130

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure an ingest provider is configured to ingest SMAP granules: <br		

#	Action	Expected Result	Notes
	/>Provider Type: Polling with DR Preprocessing Type: SIPS Max Active Data Volume: 1000.00 Max Active Granules: 100 Transfer Type: Local Notification Method: Email Only E-Mail address: labuser@f4eil01.edn.ecs.nasa.gov		
3	Ensure the test collection's data type has been installed.		
4	Ensure the test collection's data type is configured in the DPL Ingest GUI > Configuration > Data Types to be public on ingest.		
5	Ensure a PostgreSQL client is available: From a command prompt, connect to the database: $ /tools/postgres/current32/bin/psql -U <user> -d ecs -h f4dbl03 At the Postgres client prompt, set the search path to access the appropriate mode database: > select public.set_search_path('aim', '<MODE>');		
6	Setup		#comment
7	S-1 Prepare a SMAP granule for ingest, including the associated granule level metadata file. In the metadata file, note the values for all of the attributes that will undergo attribute type conversion during Ingest of the granule. [Note that the full list of attributes that will undergo attribute type conversion will be produced by DDR.]		#comment
8	Use the provided test data under />/sotestdata/DROP_802/SD_82_01/Criteria/145		
9	Using the dataset-level XPath file as a guide, note all the granule metadata values that will be converted on ingest (as of 2013-04-11; see the dataset-level XPath for the current list): 1) gmd:edition => VersionID 2) gml:beginPosition => RangeBeginningDate 3) gml:beginPosition => RangeBeginningTime 4) gml:endPosition => RangeEndingDate 5) gml:endPosition => RangeEndingTime		
10	S-2 Ingest the SMAP granule.		#comment
11	Copy the granule's PDR file into the test provider's polling location.		
12	Wait for the DPL Ingest GUI to show the granule has completed successfully.		
13	Get the granule's granuleid (adjust interval as needed): select granuleid from amgranule where shortname = <SHORTNAME> and versionid = <VERSIONID> and archivetime > now() - interval '5 minutes';		
14	Verification		#comment
15	V-1 For each of the attributes identified in setup step 1, verify by inspecting the AIM database that the properly converted metadata values from the granule level metadata file were inserted.		#comment

#	Action	Expected Result	Notes
16	Retrieve the converted granule metadata values from the AIM database: select VersionID, RangeBeginningDate, RangeBeginningTime, RangeEndingDate, RangeEndingTime from amgranule where granuleid = <GRANULEID>;		
17	To use the "/tools/common/test/BE_82_01/bin/xpath" utility to extract metadata from granule XML files, start a bash shell as cmshared.		#comment
18	1) Verify VersionID is obtained by extracting the digit from the 3rd character of the gmd:edition field value. xpath '//gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation[gmd:identifier/gmd:MD_Identifier/gmd:description/gco:CharacterString="The ECS Short Name"]/gmd:edition/gco:CharacterString/text()' <SMAP_GRANULE_METADATA_XML> | sed 's/^..\(.\).*/\1/'		
19	2) Verify RangeBeginningDate is obtained by extracting the portion of the gml:beginPosition value that precedes the "T". xpath '//gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/gml:TimePeriod/gml:beginPosition/text()' <SMAP_GRANULE_METADATA_XML> | sed 's/\([^T]*\)T.*/\1/'		
20	3) Verify RangeBeginningTime is obtained by extracting the portion of the gml:beginPosition value that follows the "T". xpath '//gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/gml:TimePeriod/gml:beginPosition/text()' <SMAP_GRANULE_METADATA_XML> | sed 's/^[^T]*T\(.*\)/\1/'		
21	4) Verify RangeEndingDate is obtained by extracting the portion of the gml:endPosition value that precedes the "T". xpath '//gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/gml:TimePeriod/gml:endPosition/text()' SMAP_L1C_S0_HIRES_00016_A_20150530T160100_D03999_001.h5.iso.xml | sed 's/\([^T]*\)T.*/\1/'		
22	5) Verify RangeEndingTime is obtained by extracting the portion of the gml:endPosition value that follows the "T". xpath '//gmi:MI_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/gmd:temporalElement/gmd:EX_TemporalExtent/gm		

#	Action	Expected Result	Notes
	d:extent/gml:TimePeriod/gml:endPosition/text()' SMAP_L1C_S0_HIRES_00016_A_20150530T160100_D03999_001.h5.iso.xml | sed 's/^[^T]*T\(.*\)/\1/'		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
145			SPL1CS0.030	A SMAP granule, including the associated granule level metadata file, with values representing all the types that require conversion during ingest.			/sotestdata/DROP_802/SD_82_01/Criteria/145	
		ISO series and dataset level XPath files					/sotestdata/DROP_802/SD_82_01/Criteria/010	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	145	1	For each of the attributes identified in setup step 1, verify by inspecting the AIM database that the properly converted metadata values from the granule level metadata file were inserted.		

2.13 Invalid Granule/Dataset Metadata Attribute Value (ECS-ECSTC-123)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	130	1	**Invalid Granule/Dataset Metadata Attribute Value** Prepare a SMAP granule for ingest, editing the granule metadata file so that an attribute will be invalid according to the schema configured for the collection.	EC	S-DPL-00110
S	130	2	Initiate ingest of the SMAP Granule.	EC	S-DPL-00110

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure an ingest provider is configured to ingest SMAP granules: Provider Type: Polling with DR Preprocessing Type: SIPS Max Active Data Volume: 1000.00 Max Active Granules: 100 Transfer Type: Local Notification Method: Email Only E-Mail address: labuser@f4eil01.edn.ecs.nasa.gov		
3	Ensure the test collection's data type has been installed.		
4	Ensure the test collection's data type is configured in the DPL Ingest GUI > Configuration > Data Types to be public on ingest.		
5	Ensure a PostgreSQL client is available: From a command prompt, connect to the database: $ /tools/postgres/current32/bin/psql -U <user> -d ecs -h f4dbl03 At the Postgres client prompt, set the search path to access the appropriate mode database: > select public.set_search_path('aim', '<MODE>');		
6	Setup		#comment
7	S-1 Prepare a SMAP granule for ingest, editing the granule metadata file so that an attribute will be invalid according to the schema configured for the collection.		#comment
8	The prepared test data has an invalid value (90.322525) for the gmd:northBoundLatitude attribute. (See the README.txt file in the test		

#	Action	Expected Result	Notes
	data directory.)		
9	S-2 Initiate ingest of the SMAP Granule.		#comment
10	Copy the granule's PDR file into the test provider's polling location.		
11	Wait for the DPL Ingest GUI to show the granule has arrived at a terminal state.		
12	Verification		#comment
13	V-1 Verify that the ingest failed due to failure to validate the metadata attribute with the schema.		#comment
14	Verify the DPL Ingest GUI shows that the granule failed ingest.		
15	Verify the request details show that the granule failed because the metadata did not conform to the schema.		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
130			SPL1CS0.039	A SMAP granule whose granule metadata file has an invalid attribute, according to the schema configured for the collection.			/sotestdata/DROP_802/SD_82_01/Criteria/130	
		ISO series and dataset level XPath files					/sotestdata/DROP_802/SD_82_01/Criteria/010	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	130	1	Verify that the ingest failed due to failure to validate the metadata attribute with the schema.		

2.14 ODL metadata files for ECS Distribution (ECS-ECSTC-124)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	150	1	**ODL metadata files for ECS Distribution** Order a granule for a collection whose metadata model type is ECS, specifying a PULL distribution. Ensure that the user's email address is configured to be a .met-file recipient.	FC	S-OMS-00100

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure an ingest provider is configured to ingest ECS granules: Provider Type: Polling with DR Preprocessing Type: SIPS Max Active Data Volume: 1000.00 Max Active Granules: 100 Transfer Type: Local Notification Method: Email Only E-Mail address: labuser@f4eil01.edn.ecs.nasa.gov		
3	Ensure the test collection's data type has been installed.		
4	Ensure the test collection's data type is configured in the DPL Ingest GUI > Configuration > Data Types to be public on ingest.		
5	Ensure a PostgreSQL client is available: From a command prompt, connect to the database: $ /tools/postgres/current32/bin/psql -U <user> -d ecs -h f4dbl03 At the Postgres client prompt, set the search path to access the appropriate mode database: > select public.set_search_path('aim', '<MODE>');		
6	Ensure an ECS granule is available to order.		#comment
7	Copy the test granule's PDR file into the test provider's polling location.		
8	Wait for the DPL Ingest GUI to show the granule has completed successfully.		
9	Get the granule's granuleid (adjust interval as needed): select granuleid from amgranule where shortname = <SHORTNAME> and versionid = <VERSIONID> and archivetime > now() - interval '5 minutes';		

#	Action	Expected Result	Notes
10	Determine the following parameters: <OM Host> - the host on which the Order Manager is executing <acquire Host> - the host on which the acquire script is executed		
11	Setup		#comment
12	S-1 Order a granule for a collection whose metadata model type is ECS, specifying a PULL distribution. Ensure that the user's email address is configured to be a .met-file recipient.		#comment
13	Configure the OMS to offer packaging options for HTTP Pull. Using a cmshared command prompt on <OM Host>, move the OrderManager configuration file to a new name, then make a copy: > cd /usr/ecs/<MODE>/CUSTOM/cfg > mv EcOmOrderManager.cfg EcOmOrderManager.cfg.SD8201C150 > cp EcOmOrderManager.cfg.SD8201C150 EcOmOrderManager.cfg > vi EcOmOrderManager.cfg Set the following configuration values: DownloadType = HTTP FTP_PULL_OPTIONS = GZIP UNIX TAR ZIP		
14	On the OM GUI main page, expand the 'OM Configuration' menu option list and select 'Media'.		
15	If the 'Pull Gran Dpl Time (days) is not set to 1, make a note of its current value, then set it to 1 and click the 'Apply' button at the bottom of the page		
16	On the OM GUI, expand the 'OM Configuration' menu option list and select Odl Metadata Users		
17	If labuser@f4eil01.edn.ecs.nasa.gov is not present in list, enter labuser@f4eil01.edn.ecs.nasa.gov in the text box and click the 'Add New User' button at the bottom of the page		
18	Configure an OMSCLI acquire script parameter file: ECSUSERPROFILE = ECSGuest PRIORITY = NORMAL DDISTMEDIATYPE = FtpPull DDISTMEDIAFMT = FILEFORMAT USERSTRING = smap_test DDISTNOTIFYTYPE = MAIL NOTIFY = labuser@f4eil01.edn.ecs.nasa.gov		
19	Configure an OMSCLI acquire script geoid file: SC:<SHORTNAME>.<VERSIONID>:<GRANULEID>		
20	Using a cmshared command prompt on <acquire Host>, run the acquire script > cd /usr/ecs/<MODE>/CUSTOM/utilities > ./acquire <MODE> -p <path to parameter file> -f <path to geoidfile> -t <random tag>		The <random tag> argument is not optional and must be unique for each run of the acquire script. It can be any string of characters.

#	Action	Expected Result	Notes
21	Verification		#comment
22	V-1 When the DN is received, ensure that the metadata file for the granule is in ODL format.		#comment
23	Determine the FTPDIR from the DN. > ssh f4eil01 > vi /var/spool/mail/labuser		
24	Navigate to the FTPDIR and verify metadata file is in ODL format > cd /datapool/<MODE>/user/<FTPDIR> > vi <metadata file>		
25	Cleanup		#comment
26	Using a cmshared command prompt on <OM Host>, replace the altered OrderManager configuration file with the original: > cd /usr/ecs/<MODE>/CUSTOM/cfg > mv EcOmOrderManager.cfg.SD8201C150 EcOmOrderManager.cfg		
27			

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
150				An ECS granule.			/sotestdata/DROP_802/SD_82_01/Criteria/150	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	150	1	When the DN is received, ensure that the metadata file for the granule is in ODL format.		

2.15 Invalid Model Type (ECS-ECSTC-112)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	30	1	**Invalid Model Type** Prepare an ESDT descriptor for a new collection with the data model type attribute set to an invalid setting.	EC	S-AIM-00100
S	30	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	EC	S-AIM-00100
S	30	3	Using the ESDT Maintenance GUI, install the ESDT.	EC	S-AIM-00100

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C3 is not installed (C3 is invalid, so this should be trivial).		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to an invalid setting.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/030.		
7	Ensure collection C3's descriptor file has a DataModelType element with Value neither "ISO-SMAP" nor "ECS" (e.g., "ISO-ECS").		
8	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		

#	Action	Expected Result	Notes
10	As cmshared, copy collection C3's descriptor file from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/030/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
13	Find the collection C3 in the list of collections, and enable its check box.		
14	Click on the 'Proceed with installation/update' button.		
15	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT installation failed.	
16	Verification		#comment
17	V-1 Verify that the Maintenance GUI Install ESDT operation fails with error messages sufficient to identify the problem with the descriptor file.		#comment
18	Perform criterion text.		
19	V-2 Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.		#comment
20	Verify collection C3's descriptor file is not in the collection directory: ls -l /stornext/smallfiles/<MODE>/descriptor/*<C3_ESDT>*		
21	Verify no MCF file was generated for collection C3: ls -l /stornext/smallfiles/<MODE>/mcf/*<C3_SHORTNAME>*<C3_VERSIONID>*		
22	V-3 Verify that no inserts were made into the AIM database for the ESDT.		#comment
23	Query the AIM database for collection C3: select * from amcollection where shortname = <C3_SHORTNAME> and versionid = <C3_VERSIONID>		
24	Verify no rows are returned.		

TEST DATA:

Crit id	Critccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
30		Collection C3		A new collection with the data model type attribute set to an invalid setting.			/sotestdata/DROP_802/SD_82_01/Criteria/030	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	30	1	Verify that the Maintenance GUI Install ESDT operation fails with error messages sufficient to identify the problem with the descriptor file.	EC	
V	30	2	Verify that no files for the given ESDT were copied into the ESDT descriptor or MCF target directories.	EC	
V	30	3	Verify that no inserts were made into the AIM database for the ESDT.	EC	

2.16 Install ECS ESDT (ECS-ECSTC-111)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	20	1	**Install ECS ESDT** Prepare an ESDT descriptor for a new collection with the data model type attribute set to an ECS Metadata Model setting.	FC	S-AIM-00100, S-AIM-0016
S	20	2	Copy the prepared file into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	FC	S-AIM-00100, S-AIM-0016

S	20	3	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00100, S-AIM-0016,

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C2 is not installed. If it is, use the ESDT maintenance GUI to remove it as follows: On the ESDT Maintenance GUI main page, select collection C2. Select the 'Delete selected ESDTs' button. After a few seconds, the GUI should display a message indicating the ESDT was successfully 'processed'.		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to an ECS Metadata Model setting.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/020.		
7	Ensure collection C2's descriptor file either has no DataModelType element, or if it does, the Value is 'ECS'.		
8	S-2 Copy the prepared file into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C2's descriptor file from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/020/*.desc /usr/ecs/<MODE>/CUSTOM/data/ESS		
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.		
13	Find collection C2 in the list of collections, and enable its check box.		

#	Action	Expected Result	Notes
14	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
15	Click on the 'Ok' button.	After a second or two, the GUI shuold display a page indicating that the ESDT has been installed successfully.	
16	Verification		#comment
17	V-1 Verify by inspecting the AIM database that the collection was correctly installed and that the database attribute is set to an ECS Metadata Model setting.		#comment
18	Query the AIM database for collection C2: select * from amcollection where shortname = <C2_SHORTNAME> and versionid = <C2_VERSIONID>		
19	Verify the datamodeltype is 'ECS'.		
20	Verify the other values are correct, using the descriptor file as a guide.		
21	V-2 Verify that an ESDT-specific XML schema file was created in the ESDT descriptor target directory and that an MCF file was created in the MCF target directory.		#comment
22	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		
23	Verify collection C2's descriptor file is in the collection directory: ls -l *<C2_ESDT>*.desc		
24	Verify C2's descriptor file from previous step has the same contents as the file in the test data directory, using diff: diff *<C2_ESDT>*.desc /sotestdata/DROP_802/SD_82_01/Criteria/010/*<C2_ESDT>*.desc		
25	Verify an XML schema file was created for C2: ls -l *<C2_ESDT>*.xsd		
26	Verify an MCF file was created for C2: ls -l /stornext/smallfiles/<MODE>/mcf/<C2_ESDT>*.mcf		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
20		Collection	GLAS_ANC.001	An ECS			/sotestdata/DROP_802/SD_82_01/Criteria/020	

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
		C2		collection.				

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
V	20	1	Verify by inspecting the AIM database that the collection was correctly installed and that the database attribute is set to an ECS Metadata Model setting.		
V	20	2	Verify that an ESDT-specific XML schema file was created in the ESDT descriptor target directory and that an MCF file was created in the MCF target directory.		

2.17 Install ISO ESDT (ECS-ECSTC-110)

DESCRIPTION:

Setup Flag	Criteria ID	Clause ID	Criteria Text	Type	L4 ID
S	10	1	**Install ISO ESDT** Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Prepare a collection level XPATH file, granule level XPATH file, and an ISO-compliant series (collection) level metadata file.	FC	S-AIM-00100, S-AIM-00160, S-AIM-00170, S-AIM-00180
S	10	2	Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.	FC	S-AIM-00100, S-AIM-00160, S-AIM-00170, S-AIM-00180

S	10	3	Using the ESDT Maintenance GUI, install the ESDT.	FC	S-AIM-00100, S-AIM-00160, S-AIM-00170, S-AIM-00180

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure collection C1 is not installed. If it is, use the ESDT maintenance GUI to remove it as follows: On the ESDT Maintenance GUI main page, select collection C1. Select the 'Delete selected ESDTs' button. After a few seconds, the GUI should display a message indicating the ESDT was successfully 'processed'.		
3	Ensure a PostgreSQL prompt is available, with the search path set correctly for the mode. E.g., From a command prompt, connect to the database: /tools/postgres/current32/bin/psql -U <USER> -d ecs -h f4dbl03 At the psql prompt, set the search path: select public.set_search_path('aim', '<MODE>');		
4	Setup		#comment
5	S-1 Prepare an ESDT descriptor for a new collection with the data model type attribute set to a SMAP ISO-19115 Metadata Model setting. Prepare a collection level XPATH file, granule level XPATH file, and an ISO-compliant series (collection) level metadata file.		#comment
6	Use the prepared test data in /sotestdata/DROP_802/SD_82_01/Criteria/010.		
7	Ensure collection C1's descriptor file has a DataModelType element with Value = "ISO-SMAP".		
8	S-2 Copy the prepared files into the source directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
9	ssh to the ESDT Maintenance GUI host (e.g., f4dpl01 or f4hel01).		
10	As cmshared, copy collection C1's files from the test data directory to the ESDT GUI directory: cp /sotestdata/DROP_802/SD_82_01/Criteria/010/* /usr/ecs/<MODE>/CUSTOM/data/ESS		

#	Action	Expected Result	Notes
11	S-3 Using the ESDT Maintenance GUI, install the ESDT.		#comment
12	Note the current time as t0.		
13	On the ESDT Maintenance GUI main screen, select the 'Install new ESDTs/Update existing ESDTs' button.	The GUI should display a list of collections available for installation	
14	Find collection C1 in the list of collections, and enable its check box.		
15	Click on the 'Proceed with installation/update' button.	A dialog box should pop up.	
16	Click on the 'Ok' button.	After a second or two, the GUI should display a page indicating that the ESDT has been installed successfully.	
17	Verification		#comment
18	V-1 Verify by inspecting the AIM database that the collection was correctly installed and that the database attribute is set to a SMAP ISO-19115 Metadata Model setting.		#comment
19	Query the AIM database for collection C1: select * from amcollection where shortname = <C1_SHORTNAME> and versionid = <C1_VERSIONID>		
20	Verify the datamodeltype is 'ISO-SMAP'.		
21	Verify the other values are correct, using the descriptor file and series XPath file as guides.		
22	V-2 Verify that the ESDT descriptor file, the two xpath files, and the ISO series XML file have been copied to the target directory configured in the ESDT maintenance GUI for ESDT descriptor files.		#comment
23	Change to the small file archive collection directory: cd /stornext/smallfiles/<MODE>/descriptor		
24	Verify collection C1's descriptor file, 2 XPath files, and ISO series XML file are in the collection directory: ls -l *<C1_ESDT>*		
25	Verify each file from previous step has the same contents as the file in the test data directory, using diff: diff *<C1_ESDT>*.desc /sotestdata/DROP_802/SD_82_01/Criteria/010/*<C1_ESDT>*.desc diff *<C1_ESDT>*.dataset.xpath /sotestdata/DROP_802/SD_82_01/Criteria/010/*<C1_ESDT>*.dataset.xpath diff *<C1_ESDT>*.series.xpath /sotestdata/DROP_802/SD_82_01/Criteria/010/*<C1_ESDT>*.series.xpath diff *<C1_ESDT>*.series.xml /sotestdata/DROP_802/SD_82_01/Criteria/010/*<C1_ESDT>*.series.xml		
26	V-3 Verify by inspection that the XPATH files and the series level		#comment

#	Action	Expected Result	Notes
	metadata file were correctly associated with the collection.		
27	Verify (from step 16) that all of collection C1's files have timestamps within a few seconds of each other and after time t0.		
28	Verify collection C1's series-level metadata file can be viewed in the ESDT Maintenance GUI.		
29	V-4 Verify by inspection that the MCF file was not generated.		#comment
30	Verify no MCF file was generated for collection C1: ls -l /stornext/smallfiles/<MODE>/mcf/*<C1_SHORTNAME>*<C1_VERSIONID>*	The file should not exist.	

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
10				A new SMAP ISO-19115 collection with a collection level XPATH file, granule level XPATH file, and an ISO-compliant series (collection) level metadata file.			/sotestdata/DROP_802/SD_82_01/Criteria/010	

EXPECTED RESULTS:

Setup Flag	Criteria ID	Clause ID	Crit Text	Type	L4 ID
V	10	1	Verify by inspecting the AIM database that the collection was correctly installed and that the database attribute is set to a SMAP ISO-19115 Metadata Model setting.		
V	10	2	Verify that the ESDT descriptor file, the two xpath files, and the ISO series XML file have been copied to the target		

			directory configured in the ESDT maintenance GUI for ESDT descriptor files.		
V	10	3	Verify by inspection that the XPATH files and the series level metadata file were correctly associated with the collection.		
V	10	4	Verify by inspection that the MCF file was not generated.		S-AIM-00170

3 FULL RESOLUTION BROWSE

3.1 Install AST_L1T ESDT (ECS-ECSTC-589)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the mode is up and that BMGT is configured to export either to a fake ECHO endpoint, or to a proxy which will allow logging of all exports.		
2	Obtain the Descriptor file for AST_L1T and as cmshared copy the AST_L1T descriptor file to /usr/ecs//CUSTOM/data/ESS	The descriptor should have CSDTDescription/Implementation = "HDF_EOS". It should have PSAs for FullResolutionVisibleBrowseAvailable and FullResolutionThermalBrowseAvailable as well as a PSA for a Digital Object Indentifier for the granule in addition to all the other elements required by the ESDT.	
3	Use the AIM ESDT Maintenance GUI to install the AST_L1T ESDT/descriptor file	The aim schema wlll contain entries in dsgeesdtconfiguredtype and amcollection for the new ESDT.	
4	Use the DataPool Maintenance GUI to update the AST_L1T ESDT with the data pool specific requirements. The ESDT should go into the existing		

#	Action	Expected Result	Notes
	ASTER collection group if present. If no ASTER collection group is present one should be added.		
5	Use the Ingest GUI to configure the AST_L1T volume groups and other Ingest specific configuration items (PublishByDefault should be false) Provider Type: Polling with DR Preprocessing Type: SIPS Max Active Data Volume: 1000.00 Max Active Granules: 100 Transfer Type: Local Notification Method: Email Only E-Mail address: labuser@f4eil01.edn.ecs.nasa.gov	For the EDF this can go into an existing policy class. For the PVC we might need to determine how many files we expect to store before choosing a Volume Group / policy class.	
6			

TEST DATA:

EXPECTED RESULTS:

3.2 Ingest AST_L1T granules into hidden online archive (ECS-ECSTC-590)

DESCRIPTION:

PRECONDITIONS:
This test case assumes the following AST_L1T test cases have been run:
Install AST_L1T ESDT

STEPS:

#	Action	Expected Result	Notes
1	Obtain 3 PDRs for AST_L1T. Each PDR should contain an HDF datafile, 2 Full Resolution Browse Images (tif files), a metadata file, and an HDF browse file. The AST_L1T metadata file contains LocalGranuleID.	The file_type for the data file and two FBR files should be set to HDF in the PDR	
2	Copy 2 of the 3 AST_L1T PDRs into the polling directory configured the ESDT	The 2 AST_L1T granules should be ingested into hidden DPL.	

TEST DATA:

EXPECTED RESULTS:
Verify the 6 files in the AST_L1T volume group were named with the structure:
:sc:<short_name>.<versionid>:<granuleid>_<sequence_number>

and the extension of the files should match the extensions from AST_L1T descriptor CSDTDescription->Implementation, in our case it's HDF-EOS.

Verify the metadata file is stored in the correct METADATA directory.

Verify that the Online Archive contains the 3 data files and one metadata fille for each granule in the correct "date" directory. The science file and two FRB files for each granule should have file names that match those provided in the PDR but with _versionnumber added before the extension.

3.3 Export AST_L1T to ECHO (ECS-ECSTC-591)

DESCRIPTION:

PRECONDITIONS:

This test case assumes the following AST_L1T test cases have been run:
Install AST_L1T ESDT
Ingest AST_L1T granules into the hidden online archive

STEPS:

#	Action	Expected Result	Notes
1	Obtain a copy of the ECHO collection metadata schema (https://api.echo.nasa.gov/ingest/schema/Collection.xsd) and the ECHO granule schema https://api.echo.nasa.gov/ingest/schema/Granule.xsd		
2	Use the DPL Maintenance GUI to update the AST_L1T collection to allow publication into the Online Archive	the amcolleciton(allowpublishflag) will be 'Y'	
3	Publish all the AST_L1T granules using the EcDlPublishUtility with the -collection option	The publish utility should return success and the hidden granules should be moved to the public online archive	
4	Ensure the "mock" ECHO is running for the mode		
5	Ensure a row exists in bg_collection_configuration for collection AST_L1T. If required execute the following query by providing the collectionid (use amcollection table to get collectionid) insert into bg_collection_configuration(collectionid, shortname, versionid, longname, granuleexportflag, collectionexportflag, difid, maxgranulestoverify) select collectionid, shortname, versionid, longname,'N', 'N', shortname, 5000 from amcollection where collectionId =	The BMGT GUI should show an entry for the collection.	
6	Using the BMGT GUI configure the AST_L1T collection for exporting to ECHO.	select collectionexportflag from bg_collection_configuration where shortname = 'AST_L1T' should return	

#	Action	Expected Result	Notes
		'Y'	
7	Using the BMGT GUI enable granule export	The "mock" ECHO should receive a copy of the AST_L1T collection metadata via a PUT command	
8	Validate the collection metadata entry received at the "mock" ECHO; xmllint --noout --schema Collection.xsd collection.xml	No errors	
9	Do a manual export EcBmBMGTManualStart --metg --collections AST_L1T.VERSIONID		
10	Validate at least one of the granule metadata entries received in the "mock" ECHO using the ECHO schema (https://api.echo.nasa.gov/ingest/schema/Granule.xsd)	The granule validation should have no errors and the metadata should contain values for the two FRB PSAs with values conforming to the domain described in the descriptor file.	
11	Verify at least one of Granule metadata entries in the mock ECHO for OnlineAccessURL/URL with file extension .tif has mimetype application/x-geotiff	http://f4ftl01//FS1/ASTT/AST_L1T.001/2014.04.30/AST_L1T_00104302014161831_20140501105036_18638_T.tif application/x-geotiff http://f4ftl01//FS1/ASTT/AST_L1T.001/2014.04.30/AST_L1T_00104302014161831_20140501105036_18638_V.tif application/x-geotiff	
12	From the Granule metadata entries in the mock ECHO from above, if the metadata contains OnlineAccessURL/URL file ending with _T.tif, verify that the granule metadata contains an FullResolutionThermalBrowseAvailable	FullResolutionThermalBrowseAvailable Y Should be present	
13	From the Granule metadata entries in the mock ECHO from above, if the metadata contains OnlineAccessURL/URL file ending with _V.tif, verify that the granule metadata contains an FullResolutionVisibleBrowseAvailable	FullResolutionVisibleBrowseAvailable Y	

#	Action	Expected Result	Notes
		should be present.	

TEST DATA:

EXPECTED RESULTS:
1) The two AST_L1T granules (along with any other AST_L1T granules) should be moved to the "public" Online Archive and the AmGranule.IsOrderOnly column should be changed from "H" to NULL.

2) Verify that the ECHO AST_L1T collection metadata has the following elements:
a) Visible = true
b) Orderable = false
c) InsertTime = The insert time of the collection recorded in the AIM database.
d) LastUpdate = The last update time of the collection recorded in the AIM database.

3.4 Add AST_L1T subscription with ftp push action (ECS-ECSTC-592)

DESCRIPTION:

PRECONDITIONS:
This test case assumes the following AST_L1T test cases have been run:
Install AST_L1T ESDT
Ingest AST_L1T granules into the hidden online archive
Export AST_L1T to ECHO

STEPS:

#	Action	Expected Result	Notes
1	Using the Spatial Subscritption GUI add a subscription for AST_L1T for the user "labuser" on insert events with an ftp push action (you don't need to add any qualifications but are free to do so)	no errors	
2	Ingest AST_L1T granule into public DPL, and verify that this granule is ordered and shipped to user based on the subscription information.		

TEST DATA:

EXPECTED RESULTS:
select * from ecnbsubscription where esdt_id like 'AST_L1T%'
should show the new subscription

select * from ecnbactiondefinition d
join ecnborderaction o on o.actionid = d.actionid
where d.subscriptionid in (select subscriptionid from ecnbsubscription where esdt_id like 'AST_L1T%')
should show the ftp push action

3.5 Ingest a public AST_L1T granule (ECS-ECSTC-593)

DESCRIPTION:

PRECONDITIONS:
This test case assumes the following AST_L1T test cases have been run:
Install AST_L1T ESDT
Ingest AST_L1T granules into the hidden online archive
Export AST_L1T to ECHO
Add AST_L1T subscription with ftp push action

STEPS:

#	Action	Expected Result	Notes
1	Configure AST_L1T data type to be public. Place another AST_L1T PDR in the configured polling directory	The granule should be ingested into the public online archive	
2	Verify the subscription on AST_L1T fired and placed an order to ftp push the granule		
3	Verify that OMS pushed the granule to the configured user (labuser)		
4	Verify the distributed file names should be the same as the ones in the online archive and PDR.		

TEST DATA:

EXPECTED RESULTS:
The granule should be ingested into the public online archive into the associated AST_L1T date directory
The granule directory should contain the 3 data files (1 hdf-eos and 2 tif (frb files))
The granule directory should contain 1 metadata file

The granule directory should contain one or multiple symbolic links to the public jpeg browse images
The file names in the online archive should be their original filenames in the PDR.

The 3 granule data files should be ingested into the current volume group directory for AST_L1T
The directory should contain 1 hdf-eos file and 2 tiff files, the names should be based on the
naming convention :sc:shortname.versionid:granuleid_sequencenumber.extension
where the extension is from AST_L1T descriptor file CSDTDescription->Implementation.

select * from aim_<mode>.amdatafile where granuleid = <new granule id>
verify the file names in the database match the names in the file systems

Verify the granule was pushed to the configured push destination and the file
names match those in the online archive.

3.6 Run Rollup and EMS for AST_L1T (ECS-ECSTC-594)

DESCRIPTION:

PRECONDITIONS:
This test case assumes the following AST_L1T test cases have been run:
Install AST_L1T ESDT
Ingest AST_L1T granules into the hidden online archive
Export AST_L1T to ECHO
Add AST_L1T subscription with ftp push action
Ingest a public AST_L1T granule with ftp push action

STEPS:

#	Action	Expected Result	Notes
1	Obtain the "Interface Control Document between the ESDIS Metrics System (EMS) and the Data Providers" revision B dated December 2013		The document will be used to verify reports that are generated for EMS
2	Ensure there are at least 3 AST_L1T granules ingested.		
3	Download the 1st public AST_L1T granules from the public online archive using anonymous ftp to the f5eil01v.edn.ecs.nasa.gov host	no errors	
4	Download the 2nd public AST_L1T granules using HTTP	no errors	
5	Download the FRBV file for the 3rd AST_L1T using anonymous ftp	no errors	
6	Download the FRBT file for the 3rdAST_L1T granules using HTTP	no errors	
7	Run the full set of Rollup Scripts for mode	no errors	
8	Order 1 granule for each media type via OMS (FtpPush, FtpPull, SCP)	Requests ship	
9	Run the EcDbEMSdataExtractor.pl script with only the mode option so that it	no errors	

#	Action	Expected Result	Notes
	will process all EMS reports		
10	Verify the new ESDT report (Meta) contains an entry for the new AST_L1T collection	There should be a single line with entries matching the the ICD table 4.2.8-7	
11	Verify the daily Ingest inventory report (Arch) has: Entries for each of the 3 AST_L1T hdf files ingested with Product = AST_L1T and number of files = 1 		
12	Verify the ingest metrics report (Ing) contains an entries for each of the 3 AST_L1T hdf files with product equal to AST_L1T and conforming to ICD table 4.2.8-1		
13	Verify that the ftp distribution report DistFTP...DataPool, Dist contains entries for the 1st AST_L1T granule 1 entry for each of the following AST_L1T hdf-eos file with product = AST_L1T, FRBV tif file with product = AST_FRBV, and FRBT tif file with product = AST_FRBT. Entries should conform to ICD table		
14	Verify that the ftp distribution report DistFTP...DataPool contains entry for the 3rd AST_L1T granule The entry should be for one FRBV tif file with product = AST_FRBV		

TEST DATA:

EXPECTED RESULTS:

3.7 Ingest a replacement AST_L1T granule (ECS-ECSTC-595)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Follow the steps in test case "Ingest a public AST_L1T granule" using the same granule but a new PDR	The granule should be replaced in the online archive. The new granule is in public DPL now, and old one is moved to hidden DPL.	

TEST DATA:

EXPECTED RESULTS:

The granule should be ingested into the public online archive into the associated AST_L1T date directory
The granule directory should contain the 3 data files (1 hdf-eos and 2 tif (frb files))
The granule directory should contain 1 metadata file
The granule directory should contain one or multiple symbolic links to the public jpeg browse images
The online file names should be the original file names from the PDR.

The previous version of the granule should be moved to the hidden online archive

run the sql:
select * from aim_<mode>.amgranulereplacement where duplicategranid = <old granule id>
verify the new granule is recorded as a replacement for the old one

The 3 granule data files should be ingested into the current volume group directory for AST_L1T
The directory should contain 1 hdf-eos file and 2 tiff files, the names should be based on the
naming convention :sc:shortname.versionid:granuleid_sequencenumber.extension
where extension is from AST_L1T descriptor file CSDTDescription->Implementation.
The files for the previous version of granule should remain unchanged

run the sql:
select * from aim_<mode>.amdatafile where granuleid = <new granule id>;
select * from aim_<mode>.ammetadatafile where granuleid = <new granule id>;
verify the file names in the database match the names in the file systems

select versionnumber from aim_<mode>.amgranule where granuleid = <new granule id>
verify the version number is now version 1

Verify the granule was pushed to the configured push destination and the file
names match those in the online archive.

4 8.3.2 RELEASE

4.1 NCR8051409 - Provide KML Outputs (ECS-ECSTC-661)

DESCRIPTION:

Derived from the original DP_81_02_TP001 - Test Case ID 10, and from HEG Performance testing conducted originally in 2011 and in various subsequent evolutions.

A standard set of requests are submitted to HEG via HegService, including Parameter-Subsetting (Data Objects extraction), Spatial-Subsetting, Reprojection with format to KML output formats.

PRECONDITIONS:

Mode is configured for HegService to run.

STEPS:

#	Action	Expected Result	Notes
1	Copy ruby scripts from clearcase: /ecs/formal/DPL/DataAccess/adapters/HegService/test/Reformat_KML/*.rb to a working directory - on a machine with access to /sotestdata and to /workingdata. You will need all ruby scripts in the same working directory - they call each other and assume the home directory of the currently executing script is where the other scripts will be found. Also copy the /ecs/formal/DPL/DataAccess/adapters/HegService/test/Reformat_KML/summarize.csh file to the same directory	ruby scripts available in working directory	
2	Edit the scripts to make any adjustments for working host, port, mode, output-directory, etc. By default the scripts will write to /workingdata/TestResults/RelB/DataAccess/GLAS/ESDT_Baseline/. If there is data already there, check if it needs to be moved to establish baseline-results for a previous release or test run.	Scripts updated for specific test conditions.	
3	Run NCR8051409_Test.rb (no parameters)	Console will show Curl command executions. Outputs will be saved to TestResults directory, along with .time and .xml summary files per test case.	
4	Run summarize.csh	Console will show summary outputs including unique counts of status messages, output file listings, etc. Inspect for valid outputs.	

TEST DATA:
Similar to TP001_10...

Several Granules from the following collections:

MOD10CM.005 - MODIS Grid, whole earth

MYD10A1.005, MOD10A2.005, MOD10A1.005, - MODIS Sinusoidal Grid

AE_SI6.005, AE_SI12.002, AE_SI25.002 - AMSR_E Swath

AE_L2A.002, - AMSR_E Swath

AE_DySno.002 - AMSR_E Polar Grid

EXPECTED RESULTS:

4.2 NCR8051408 - Provide netCDF Outputs (ECS-ECSTC-662)

DESCRIPTION:
Derived from the original DP_81_02_TP001 - Test Case ID 10, and from HEG Performance testing conducted originally in 2011 and in various subsequent evolutions.

A standard set of requests are submitted to HEG via HegService, including Parameter-Subsetting (Data Objects extraction), Spatial-Subsetting, Reprojection with format to KML output formats.

PRECONDITIONS:
Mode is configured for HegService to run.

STEPS:

#	Action	Expected Result	Notes
1	Copy ruby scripts from clearcase:	ruby scripts available in working	

#	Action	Expected Result	Notes
	/ecs/formal/DPL/DataAccess/adapters/HegService/test/Reformat_NetCDF/*.rb to a working directory - on a machine with access to /sotestdata and to /workingdata. You will need all ruby scripts in the same working directory - they call each other and assume the home directory of the currently executing script is where the other scripts will be found. Also copy the /ecs/formal/DPL/DataAccess/adapters/HegService/test/Reformat_NetCDF/summarize.csh file to the same directory	directory	
2	Edit the scripts to make any adjustments for working host, port, mode, output-directory, etc. By default the scripts will write to /workingdata/TestResults/RelB/DataAccess/GLAS/ESDT_Baseline/. If there is data already there, check if it needs to be moved to establish baseline-results for a previous release or test run.	Scripts updated for specific test conditions.	
3	Run NCR8051408_Test.rb (no parameters)	Console will show Curl command executions. Outputs will be saved to TestResults directory, along with .time and .xml summary files per test case.	
4	Run summarize.csh	Console will show summary outputs including unique counts of status messages, output file listings, etc. Inspect for valid outputs.	

TEST DATA:

Similar to TP001_10...

Several Granules from the following collections:

MOD10CM.005 - MODIS Grid, whole earth

MYD10A1.005, MOD10A2.005, MOD10A1.005, - MODIS Sinusoidal Grid

AE_SI6.005, AE_SI12.002, AE_SI25.002 - AMSR_E Swath

AE_L2A.002, - AMSR_E Swath

AE_DySno.002 - AMSR_E Polar Grid

EXPECTED RESULTS:

4.3 NCR8051422 - Provide ASCII Outputs (ECS-ECSTC-664)

DESCRIPTION:

Derived from the original DP_81_02_TP001 - Test Case ID 10, and from HEG Performance testing conducted originally in 2011 and in various subsequent evolutions.

A standard set of requests are submitted to HEG via HegService, including Parameter-Subsetting (Data Objects extraction), Spatial-Subsetting, Reprojection with format to KML output formats.

PRECONDITIONS:

Mode is configured for HegService to run.

STEPS:

#	Action	Expected Result	Notes
1	Copy ruby scripts from clearcase: /ecs/formal/DPL/DataAccess/adapters/HegService/test/Reformat_Ascii/*.rb to a working directory - on a machine with access to /sotestdata and to /workingdata. You will need all ruby scripts in the same working directory - they call each other and assume the home directory of the currently executing script is where the other scripts will be found. Also copy the /ecs/formal/DPL/DataAccess/adapters/HegService/test/Reformat_Ascii/summarize.csh file to the same directory	ruby scripts available in working directory	
2	Edit the scripts to make any adjustments for working host, port, mode, output-directory, etc. By default the scripts will write to /workingdata/TestResults/RelB/DataAccess/GLAS/ESDT_Baseline/. If there is data already there, check if it needs to be moved to establish baseline-results for a previous release or test run.	Scripts updated for specific test conditions.	
3	Run NCR8051422_Test.rb (no parameters)	Console will show Curl command executions. Outputs will be saved to TestResults directory, along with .time and .xml summary files per test case.	
4	Run summarize.csh	Console will show summary outputs	

#	Action	Expected Result	Notes
		including unique counts of status messages, output file listings, etc. Inspect for valid outputs.	

TEST DATA:
Similar to TP001_10...

Several Granules from the following collections:

MOD10CM.005 - MODIS Grid, whole earth

MYD10A1.005, MOD10A2.005, MOD10A1.005, - MODIS Sinusoidal Grid

AE_SI6.005, AE_SI12.002, AE_SI25.002 - AMSR_E Swath

AE_L2A.002, - AMSR_E Swath

AE_DySno.002 - AMSR_E Polar Grid

EXPECTED RESULTS:

5 SMAP HDF5 SUBSETTER

6 GLAS HDF5 SUBSETTER

6.1 End to End GLAS HDF5 subsetting - multiple granules, multiple collections (ECS-ECSTC-573)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test GLAS collections are installed in the mode and exported to ECHO. e.g. GLAH05.033 snd GLAH07.033		
2	Ensure that the GLAS subsetter service is installed in the mode enabled for the test collections		
3	Ensure that the spatial and temporal subsetting is enabled for the collections for the GLAS subsetting service		
4	Navigate to the Reverb testbed GUI. http://testbed.echo.nasa.gov/reverb		
5	Search for granules for the test GLAS collections.		
6	Select at least a couple of granules from each test collection and View cart.		
7	Click on the Granule Detail icon (i) and download the OnlineAccess URL for the granules.		
8	Run /tools/common/test/DataAccess/glasdump.rb test_granule.H5 to create dumps for the latitude, longitude and utc times or manually run the h5dump as follows... To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_INPUT_GRANULE.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_INPUT_GRANULE.H5 To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		
9	Record the spatial extent for the test granules from lat1hz.dat and lon1hz.dat from the glasdump tool or h5dump or from the info and map view of the granules in reverb gui		
10	Record the temporal range for the test granules from the raw_time1hz from the glasdump tool or from the info and temporal extent displayed in reverb		
11	Click on Perform Service and Verify that the form options are displayed correctly.		
12	Check the box for 'Use these values for all applicable items'		
13	Enter a valid email address		
14	Select GLAS processing tool		
15	Spatial Subsetting: Enter bounding box options that are within the spatial extent of each granule using the spatial extent latitudes and longitudes that were downloaded with the glasdump or h5dump earlier or graphically from		

#	Action	Expected Result	Notes
	the mapview in reverb.		
16	Verify that there are multiple bounding boxes		
17	Verify that there is a maximum limit of bounding boxes (5) . Enter options in all five bounding boxes		
18	Submit Service Request and verify that the requests succeeded.		
19	Verify in the EGI/ESIR logs that stitching has been called for each of the collections with multiple granules.		
20	Verify that all the stitch links are listed under "Download the files stitched together by Product:"		
21	Verify that there is a url for enabling stitching displayed and there is one link for each collection that has multiple granules.		
22	Click on stitched output link for each test collection to download the stitched ouput.		
23	Verify that an email is sent to the requested email address that the service request is complete with links to download the subsetted files		
24	Download the files from the output links and verify that the subsetting matches the spatial subset options selected in the request.You can verify it by running /tools/common/test/DataAccess/validate_GlasTest.rb --bbox 'W,S,E,N' subsetted_stitiched_granule.H5. You can also verify it by doing a h5dump of the subsetted_stitched output and checking the longitude and laitude values in the output as follows. To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o raw_lat1hz.dat subsetted_stitched_granule.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o raw_lon1hz.dat subsetted_stitched_granule.H5		
25	Verify that the stitched output is in temporal order. Sort raw_time1hz.dat > raw.sorted. diff raw raw.sorted should not return anything		
26	Verify that you get a stitching complete email for each collection in the request.		
27	Repeat step 11 to 23 to do temporal subsetting instead of spatial subsetting. Enter Start and End times within the temporal extent of the granule.		
28	Verify that the values in raw_time1hz.dat is within the time temporal subsetting options used. Use /tools/common/test/DataAccess/glastime.rb if you need to convert the start and end time used to utc seconds. You can verify the temporal subsetting by running		

#	Action	Expected Result	Notes
	/tools/common/test/DataAccess/validate_GlasTest.rb e.g. validate_GlasTest.rb --startTime 2004-03-02T02:51:00 --endTime 2005-02-17T21:46:00 204_GLAH12_033.H5 . You can also verify using the h5dump directly as follows and checking the values in the raw output. To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat stitched_granule.H5		
29	Repeat step 24 to verify the stitched output is in temporal order		
30	Repeat step 9 to 23 to do both temporal and spatial subsetting together.		
31	Verify that the stitched output is within the spatial and temporal settings in the request. /tools/common/test/DataAccess/validate_GlasTest.rb e.g. validate_GlasTest.rb --bbox 'W,S,E,N' --startTime YYYY-MM-DDTHH:MM:SS --endTime YYYY-MM-DDTHH:MM:SS stitched_subsetted_GLAH.H5 or do a manual h5dump to get the latitudes, longitudes and utctimes of the stitched output as follows: To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_INPUT_GRANULE.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_INPUT_GRANULE.H5 To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		
32	Repeat step 24 to Verify that the output is in temporal order.		

TEST DATA:

EXPECTED RESULTS:

6.2 End to End GLAS HDF5 subsetting - single granule, single collection (ECS-ECSTC-575)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test GLAS collection is installed in the mode and exported to ECHO.		
2	Ensure that the GLAS subsetter service is installed in the mode enabled for the test collection		
3	Ensure that the spatial, temporal and band subsetting is enabled for the collection for the GLAS subsetting service		
4	Navigate to the Reverb testbed GUI. http://testbed.echo.nasa.gov/reverb		
5	Search for granules for the test GLAS collection.e.g. GLAH05.033 EDF_DEV06		
6	Select a single test granule to the cart and View cart.		
7	Click on the Granule Detail icon (i) and download the OnlineAccess URL for the granules		
8	Run /tools/common/test/DataAccess/glasdump.rb test_granule.H5 to create dumps for the latitude, longitude and utc times or do a h5dump manually to get the longitude, latitude and utctimes as follows.. To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_INPUT_GRANULE.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_INPUT_GRANULE.H5 To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		
9	Record the spatial extent for the test granule from lat1hz.dat and lon1hz.dat from the glasdump or h5dump tool or from the info and map view of the granule in reverb		
10	Record the temporal range for the test granule from the raw_time1hz from the glasdump or h5dump tool or from the info and temporal extent displayed in reverb		
11	Record if there is any browse group data in the input granule. h5ls -r GLAH_INPUT_GRANULE.H5 > input_groups.dat		
12	Record the original attributes in the input granule. h5dump -A GLAH_INPUT_GRANULE.H5 > input_attributes.dat		
13	Click on Perform Service. Verify that the service options for the collection are displayed correctly.		

#	Action	Expected Result	Notes
14	Enter a valid email address		
15	Select GLAS processing tool		
16	Spatial Subsetting: Enter bounding box options that are within the spatial extent of each granule using the spatial extent latitudes and longitudes that were downloaded with the glasdump or h5dump earlier or graphically from the mapview in reverb.		
17	Submit Service Request. Verify that the service request suceeded.		
18	Verify that the subsetting request, constraints, statistics, outputs are logged.	Some of the logging is in EcDlDaRqs.debug0.log and Egi logs.	
19	Verify that an email is sent to the requested email address that the service request is complete with links to download the subsetted files		
20	Verify that no stitching url link options are provided.		
21	Download the subsetted file.		
22	Compare the attributes in the input file and the subsetted output file. h5dump -A input_file versus h5dump -A subsetted_output_file . Verify that there is a warning attribute in the subsetted output file for the root and metadata groups		
23	Verify that there is a link to download the request summary		
24	Verify that the file naming convention in the subsetted output file preserves the file naming convention used in the input files		
25	Verify that the subsetted output files have the same internal structure as the input data files. h5ls input_granule.H5 and h5ls subsetted_granule.H5 should be the same except for the BROWSE group that is excluded by default		
26	Verify that the subsetted output does not have any BROWSE or Image data. h5dump -w 20 -y -g 'BROWSE' -o raw_subsetted_browse subsetted_granule.H5 should throw an error or have no output	h5dump error: unable to open group "BROWSE" is the expected result.	
27	Verify that the subsettedoutput file has latitude and longitude values that are within the spatial options in the subset request. Run /tools/common/test/DataAccess/validate_GlasTest.rb --bbox 'W, S, E, N' subsetted_granule.H5 or use h5dump manually to get the values of longitude, latitude as follows... To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_subsetted_output.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_subsetted_output.H5		

#	Action	Expected Result	Notes
28	Repeat the steps 12-26 performing temporal subsetting. Temporal Subsetting: Enter Start and End times within the temporal extent of the granule.		
29	Verify that the subsetted output file have time values within the temporal subset range in the request. run /tools/common/test/DataAccess/validate_GlasTest.rb e.g. validate_GlasTest.rb --startTime YYYY-MM-DD200THH:MM:SS --endTime YYYY-MM-DDTHH:MM:SS subsetted_granule.H5 . For manual verification with a h5dump, To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_subsetted_output.H5		
30	Repeat the steps 12-26 with both spatial and temporal subsetting.		
31	Verify that the subsetted output meets both the spatial and temporal constraints. /tools/common/test/DataAccess/validate_GlasTest.rb e.g. validate_GlasTest.rb -bbox 'W, S, E, N' --startTime YYYY-MM-DD200THH:MM:SS --endTime YYYY-MM-DDTHH:MM:SS subsetted_granule.H5 or do manual verification with a h5dump as follows..To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_INPUT_GRANULE.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_INPUT_GRANULE.H5 To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		

TEST DATA:

EXPECTED RESULTS:

6.3 End to End GLAS HDF5 subsetting - multiple granules single collection (ECS-ECSTC-576)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test GLAS collection is installed in the mode and exported to ECHO		
2	Ensure that the GLAS subsetter service is installed in the mode enabled for the test collection		
3	Ensure that the spatial and temporal subsetting is enabled for the collection for the GLAS subsetting service		
4	Navigate to the Reverb testbed GUI. http://testbed.echo.nasa.gov/reverb		
5	Search for granules for the test GLAS collection. e.g. GLAH12.033 EDF_OPS		
6	Select at least two test granules in the GLAS collection to the cart and View cart.		
7	Click on the Granule Detail icon (i) and download the OnlineAccess URL for the granules.		
8	Run /tools/common/test/DataAccess/glasdump.rb test_granule.H5 to create dumps for the latitude, longitude and utc times or do h5dump manually to get the longitude, latitude and utctimes for the input granule as follows... To get the list of latitudes h5dump -w 20 -y -m '%.0f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_INPUT_GRANULE.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.0f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_INPUT_GRANULE.H5 To get the utc times h5dump -w 20 -y -m '%.0f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		
9	Record the spatial extent for the test granules from lat1hz.dat and lon1hz.dat from the glasdump or h5dump tool or from the info and map view of the granule in reverb		
10	Record the temporal range for the test granules from the raw_time1hz from the glasdump or h5dump tool or from the info and temporal extent displayed		

#	Action	Expected Result	Notes
	in reverb		
11	Click on Perform Service. Verify that the correct service options are for the collection are displayed.		
12	Enter a valid email address		
13	Select GLAS processing tool		
14	Spatial Subsetting: Enter bounding box options that are within the spatial extent of each granule using the spatial extent latitudes and longitudes that were downloaded with the glasdump or h5dump earlier or graphically from the mapview in reverb.		
15	Submit Service Request and Verify that the request succeded.		
16	Check The EGI and ESIR logs and verify that Stitching has been called for the collection.		
17	Verify that an email is sent to the requested email address that the service request is complete with links to download the subsetted files.		
18	Verify that an email is sent to the requested email address that the stitching is complete.		
19	Verify that a link is displayed to stitch all the subsetted input granules		
20	Verify that the stitch link is listed under "Download the files stitched together by Product:"		
21	Click on the Stitch granules link to perform granule stitching. Verify that a single stitched output is provided		
22	Click on the stitched output to download the files.		
23	Verify that the stitched output values are within the spatial constraints in the test. verify it by running /tools/common/test/DataAccess/validate_GlasTest.rb --bbox 'W,S,E,N' subsetted_stitched_granule.H5 or verify manually doing a h5dump as follows: To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat subsetted_stitched_granule.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat subsetted_stitched_granule.H5 To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat subsetted_stitched_granule.H5		
24	Verify that the subsetted data is in temporal order in the output file. The raw_time1hz.dat should be in ascending order.. sort raw_time1hz.dat > raw.sorted. diff raw raw.sorted should not return anything		

#	Action	Expected Result	Notes
25	Repeat 9-22 with temporal subsetting. Temporal Subsetting: Enter Start and End times within the temporal extent of each granule.		
26	Verify that the values in raw_time1hz.dat is within the time temporal subsetting options used. Use /tools/common/test/DataAccess/glastime.rb to convert the start and end time used to utc seconds. Verify the temporal subsetting by running /tools/common/test/DataAccess/validate_GlasTest.rb e.g. validate_GlasTest.rb --startTime 2004-03-02T02:51:00 --endTime 2005-02-17T21:46:00 204_GLAH12_033.H5 or do manual verification with h5dump as follows... To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		
27	Repeat step 22		
28	Repeat 9-20 with both spatial and temporal constraints. and verify with steps 21,24 and 22		

TEST DATA:

EXPECTED RESULTS:

6.4 EGI GLAS requests (ECS-ECSTC-578)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Make async request with FILE_IDS		
2	Run the following command from a psql interface with the MODE you are working in to select some granule ids: select granuleid from AmGranule where ShortName like 'GLAH%' and versionid =33 and IsOrderOnly is null;	list of granule ids given that you can request subsetting for	
3	From command line run: curl "http://f5eil01v:22500/egi/request?SUBAGENT_ID=GLAS&FILE_IDS=[comma seperated numbers from above query]"	asynchronous ESI response with an order id	

#	Action	Expected Result	Notes
4	From command line run: curl "http://f5eil01v:22500/esir/request/[order id returned in asynchronous ESI response"	gives status of request. Should eventually give a "complete" status	this did not work
5	Make a sync request for a single granule		
6	From command line run: curl "http://f5eil01v:22500/egi/request?SUBAGENT_ID=GLAS&REQUEST_MODE=sync&FILE_IDS=[single granuleid from above]"	download urls are give in ESI output	
7	From command line run: curl "[first download url from step 6(esir link)]"	ESIR response with href links to subsetted granule	
8	From command line run: curl "[href link from step 7 on "For Input Granule" line]" > outputfile.H5	outputfile.H5 is populated with subsetted granule	
9	verify downloaded granule using standard hdf5 tools like h5dump located in /tools/HDF5/bin/ and make sure no errors show up when run. i. e. /tools/HDF5/bin/h5dump outputfile.H5	No errors shown on command line from running the tools.	

TEST DATA:

EXPECTED RESULTS:

6.5 ESIR Test Cases for GLAS (ECS-ECSTC-580)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	GLAS ESIR Test Case(s) 1) Output file page groups results by collection. 2) Zip file output matches constituent files. 3) Stitched output files do not exceed configured size limit.* 4) Concurrent zip downloads do not yield partial zips. 5) Concurrent stitch downloads do not yield partial output. 6) Stitch		

#	Action	Expected Result	Notes
	output is pre-generated. *Associated with requirements list. **Not currently implemented/fixed. 		
2	Setup: - x5eil01> cat /usr/ecs/OPS/CUSTOM/cfg/EcDlDaEsir.properties | grep STITCH - Note STITCH.MAX.FILESIZE; it represents gigabyte values (GB) - Record the value of STITCH.DIRECTORY. 	- Expected result (Integer values only need to be set, not match): STITCH.DIRECTORY=/datapool//user/FS1/temp/stitch ## Should be similar to http://:/rqs_<MODE>/GLASSTITCH STITCH.URL=http://f5dpl01v.edn.ecs.nasa.gov:22500/rqs_DEV06/GLASSTITCH STITCH.MAX.FILESIZE=1 STITCH.TIMEOUT=300 STITCH.CLEANUP.INTERVAL=3600 STITCH.CLEANUP.AGE=43200 	
3	- In Reverb (http://testbed.echo.nasa.gov/reverb), perform a service on two or more granules from at least two distinct collections. (Minimum of four granules should be selected total.) To test the stitched output restrictions, it is wisest to choose a collection with large input files, like GLAH07.033. Add all available granules for each collection to the request. - Use at least two distinct bounding boxes. 		
4	- When the request is fulfilled (complete), view the provided *.html page. - Calculate the sum of the file sizes for each collection -> FILE_SIZE_SUM. - If FILE_SIZE_SUM for at least one collection does not exceed (x 1024 MB), then start over and choose different collections. 		
5	Verify: 6) List the contents of the stitch output directory (STITCH.DIRECTORY). Among the files therein should be a file that		

#	Action	Expected Result	Notes
	was created around the same time as the request was completed (check email timestamp.) 		
6	Verify: 1) Listed granule IDs are grouped by collection. 		
7	2) Zip integrity: *> wget *> jar xvf If this command fails, use a different method to extract files from the zip. Compare the size of the extracted files to that of the individual files listed on the *.html page. 	2) *> jar xvf If this command fails, use a different method to extract files from the zip. Extracted file sizes equal those of individual files listed. 	
8	3) Stitched output limits: Click the stitch link for the collection whose total file size exceeds GB. 	3) The number of links available should equal ((FILE_SIZE_SUM/STITCH.MAX.FILESIZE) + 1) 	
9	4) Partial zip files cannot be downloaded: Click the zip file link several times in a short amount of time. (Double/Triple/Quadruple-click) Download all of the zip files as prompted. 	4) Each downloaded zip (when unzipped) should be the same size--somewhat equal to the sum of all file sizes listed on the page. 	
10	5) Partial output cannot be obtained from stitch download links. Click the stitch file link multiple times. Download the stitched files as prompted.	4) Each downloaded stitch file should be the same size--somewhat equal to the sum of all file sizes listed on the page. 	

TEST DATA:

EXPECTED RESULTS:

6.6 EMS reporting for GLAH requests (ECS-ECSTC-579)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Make an EGI request multiple granules for multiple collections		requests need to go through EGI, ESI requests are not reported to EMS
2	Download all output products, zips and stitched collections from ESIR	Apache logs downloads to /usr/ecs//CUSTOM/data/DPL	may need to verify that apache's httpd.conf has the following setting for your mode: SetEnvIf Request_URI "/esir/" esirfile_ops SetEnvIf Request_URI "/esir/.*\.zip$" esirzip_ops !esirfile_ops SetEnvIf Request_URI "/esir/.*\/stitched\/.*GLAH.*" esirstitch_ops !esirfile_ops !esirzip_ops SetEnvIf Content-Type "text/html" !esirfile_ops !esirzip_ops !esirstitch_ops CustomLog /usr/ecs/OPS/CUSTOM/data/DPL/EcDlDataAccessApache.log dataaccess env=esirfile_ops CustomLog /usr/ecs/OPS/CUSTOM/data/DPL/EcDlDataAccessApache.log dataaccesszip env=esirzip_ops CustomLog /usr/ecs/OPS/CUSTOM/data/DPL/EcDlDataAccessApache.log dataaccessstitch env=esirstitch_ops
3	ssh f5eil01v		

#	Action	Expected Result	Notes
4	Verify that the downloads show up in the Apache log: /usr/ecs/CUSTOM//data/DPL		
5	Individual files that were downloaded should show up as single insert statements. Zip files that were downloaded should include a join against AmDaJob. Stitched file that were downloaded should also include a join against AmDaJob.		
6	cat /usr/ecs/OPS/CUSTOM/data/DPL/EcDlDataAccessApache.log	Paste basic output here	
7	Run the Rollup of the Apache log data into the database		
8	/usr/ecs/OPS/CUSTOM/utilities/EcDlDaRollupApacheLogs.ksh OPS	output of script is output of postgres for sql in Apache log: INSERT 0 6 indicates 0 for success and 6 for the number of rows (the number of input granules that were included in your download)	
9	verify that the rollup worked		
10	select * from ecemsextracttypedaterange	If you don't see the time range you want you must put it as specified in the two steps below	this table is in the OMS schema
11	update ecemsextracttypedaterange set startdate = [time of first download you want to see]		
12	update ecemsextracttypedaterange set enddate = getdate()	The time range you are interested in is in ecemsextracttypedaterange	the end date has to be at least 24 hours greater than the start date.. update ecemsextracttypedaterange set enddate= 'YYYYMMDD HH:MM:SS' for the end time to be set
13	select * from EMSDistHTTPTransfer_View	should list all the transfers you made	
14	verify that the number of rows in the table is equal to the number of input files that were used in all the downloads		
15	ssh to f5dpl01v		
16	Run the EMS data extractor to create a flat file that can be exported to EMS		
17	/usr/ecs//CUSTOM/utilities/EcDbEMSdataExtractor.pl -mode OPS -extracttype DistHTTP -startdate YYYYMMDD -enddate YYYYMMDD		If start date is yesterdays date and enddate is todays date.
18	check the EMS log for errors	no errors found	
19	get the directory where the flat file was sent to		

#	Action	Expected Result	Notes
20	grep scp EcDbEMSdataExtractor.log		
21	open the file and verify the data matches what is in EMSDistHTTPTransfer_View		
22	compare : cat flat_file_name | wc select count(*) from EMSDistHTTPTransfer_View	numbers should be equal	

TEST DATA:

EXPECTED RESULTS:

6.7 Configuring Limits - requests, output file size (ECS-ECSTC-582)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure max granules per request in the Data Access GUI . Set MAX_GRANS_SINGLE_ASYNC_REQUEST to 2		
2	Configure max requests in the Data Access GUI. Set MAX_GRANS_QUEUE_ASYNC_REQUESTS to 5		
3	Submit concurrent requests for GLAS subsetting for different granules and collections. Submit subset requests for 2 granules.		
4	Verify that the requests complete successfully		
5	Submit requests exceeding the configured request limit. Submit subset requests for 10 granules.		
6	Verify that not more than the max configured requests are queued and an error response is returned...		
7	Select a really large spatial area so that the output file size is larger than the configured limit.		

TEST DATA:

EXPECTED RESULTS:

6.8 GLAS ESDT Baseline Capability (ECS-ECSTC-585)

DESCRIPTION:

This test provides a baseline, nominal behavior check on the GLAS-Subsetter and GLAS-Tool-Adapter, for each of the ESDTs in the current set of GLAS ESDTs (GLAH01, GLAH05-GLAH15).

For each granule (one or two per ESDT) one run is made with a bounding-box (overlapping data contents), a second is run with a temporal range constraint (again overlapping data contents) and a third run is made with both constraints (with the constraints overlapping each other).

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Copy ruby scripts from clearcase: /ecs/formal/DPL/DataAccess/adapters/Glas/test*.rb to a working directory - on a machine with access to /sotestdata and to /workingdata. You will need all ruby scripts in the same working directory - they call each other and assume the home directory of the currently executing script is where the other scripts will be found		
2	Edit the scripts to make any adjustments for working host, port, mode, output-directory, etc. By default the scripts will write to /workingdata/TestResults/RelB/DataAccess/GLAS/ESDT_Baseline_Test/. If there is data already there, check if it needs to be moved to establish baseline-results for a previous release or test run.		
3	Run GlasEsdtBaselineTest.rb (no parameters)		
4	Run validate_GlasEsdtBaselineTest.rb to verify the output files	Summary of failures should be empty. Listing of Time-Jumps should be as expected.	
5	If possible, do an h5diff between the current subsetted h5 outputs with the baselines h5 outputs in /workingdata/TestResults	Verify that the subsetted files match the baseline h5 output files in the /workingdata directory	
6	Consider if test results should be moved to appropriate folder location under /workingdata/testresults/		

TEST DATA:
The test draws on the following .h5 files under /sotestdata...

/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH05.033/GLAH05_633_2107_002_0110_3_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH05.033/GLAH05_633_2107_002_0111_1_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH06.033/GLAH06_633_2107_002_0110_3_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH06.033/GLAH06_633_2107_002_0113_1_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH07.033/GLAH07_633_2107_002_0111_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH07.033/GLAH07_633_2107_002_0113_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH08.033/GLAH08_633_2107_002_0113_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH09.033/GLAH09_633_2107_002_0113_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH10.033/GLAH10_633_2107_002_0113_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH11.033/GLAH11_633_2107_002_0113_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH12.033/GLAH12_633_2107_002_0113_0_02_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH13.033/GLAH13_633_2107_002_0113_0_01_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH14.033/GLAH14_633_2107_002_0113_0_02_0001.H5
/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH15.033/GLAH15_633_2107_002_0113_0_01_0001.H5
EXPECTED RESULTS:

6.9 ESIR timeout of stitching (ECS-ECSTC-596)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Create request with lots of large granules (GLAH07)		
2	Download stitched product that will take longer than the timeout configured in ESIR for stitching		
3	Verify stitching is being performed by logging on to box that does stitching and running ps -ef | grep stitch_glas		
4	After request has timed out in your browser verify that sttich_glas is no longer running		
5	ps -ef | grep stitch_glas	only the grep command matches	This may not currently work. We may want to make the call to stitch_glas from ESIR thread so it could

#	Action	Expected Result	Notes
			kill the process when timeout occurs.

TEST DATA:

EXPECTED RESULTS:

6.10 Cleanup by GLAS tool adapters on error condition (ECS-ECSTC-586)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Log in as cmshared on f5dpl01v		
2	Submit successful RQS request to subset and stitch a granule.		This should create an output director or output file that will be cleaned up by the direct invocation of the tool adapters below.
3	Submit RQS request for subset_glas and stitch_glas that will fail processing but generate output files.	As of now we don't have a way to generate an output file and error out in subset_glas at the same time, so we will just verify that the previous successful directory was cleaned up	If you look in the RQS log you may be able to find tool adapter invocations that failed. If you can only find successful requests you can use one of those and change the input file to be a non-existent file.
4	grep "ConvertResource.executeCLI running" *Rqs* to get a command to execute		example: /usr/ecs/DEV06/CUSTOM/utilities/GlasStitcherToolAdapter --MAX_FILESIZE '1' --FILE_URLS '/datapool/DEV06/user/FS1/

#	Action	Expected Result	Notes
			temp/glas/178:344/subsetted_GLAH07_633_1102_001_0071_0_01_0001.H5,/datapool/DEV06/user/FS1/temp/glas/178:343/subsetted_GLAH07_633_2107_002_0111_0_01_0001.H5' --OUTPUT_DIR '/datapool/DEV06/user/FS1/temp/stitch/178_GLAH07_033.H5'
5	edit command by changing the name of the FILE_URLS to a file that does not exist and the OUTPUT_DIR to be something unique		example: /usr/ecs/DEV06/CUSTOM/utilities/GlasStitcherToolAdapter --MAX_FILESIZE '1' --FILE_URLS '/datapool/DEV06/user/FS1/temp/glas/178:344/subsetted_GLAH07_633_1102_001_0071_0_01_0001.H5,/datapool/DEV06/user/FS1/temp/glas/178:343/subsetted_GLAH07_633_2107_002_0111_0_01_0001.H51' --OUTPUT_DIR '/datapool/DEV06/user/FS1/temp/stitch/test178_GLAH07_033.H5'
6	verify the output directory does not exist. For GlasToolAdapter this is OUTPUT_DIR/IDENTIFIER, For GlasStitcherToolAdapater this is just OUTPUT_DIR		example: ls /datapool/DEV06/user/FS1/temp/stitch/test178_GLAH07_033.H5
7	Repeat above for both GlasToolAdapter and GlasStitcherToolAdapater		

TEST DATA:

EXPECTED RESULTS:

6.11 Test Cases for subset_glas (ECS-ECSTC-587)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	1) Temporal Subsetting in the beginning of the granule. Using the select statement to get the granule Date/time range in AmGranule table for the test. select esdt(shortname, versionid) ||':'|| granuleid, RangeBeginningDate, RangeBeginningTime, RangeEndingDate, RangeEndingTime, LocalGranuleID from AmGranule where LocalGranuleId = 'GLAH06_633_2107_002_0111_1_01_0001.H5';		
2			
3	Original granule Date/time range from the query result from step 3 above RangeBeginningDate = 2004-03-01 RangeBeginningTime = 01:05:53 RangeEndingDate = 2004-03-01 RangeEndingTime = 01:27:14 run subset_glas -f /sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH06.033/GLAH06_633_2107_002_0111_1_01_0001.H5 -s 2004-03-01T01:05:53 -e 2004-03-01T01:27:14 -o GLAH06_633_2107_002_0111_1_01_0001.H5 --excludegroup BROWSE	output product GLAH06_633_2107_002_0111_1_01_0001.H5 is produced in local directory	
4	run /h5dump -w 20 -y -m '%.0f' -d 'Data_1HZ/DS_UTCTime_1' -o Test1_GLAH06_633_utc GLAH06_633_2107_002_0111_1_01_0001.H5	Test1_GLAH06_633_utc contains data out from GLAH06_633_2107_002_0111_1_01_0001.H5	
5	open the Test1_GLAH06_633_utc file in step 5 to get 1st UTC 131375155 and Last UTC 131376434. 131375155, 131375156, 131375157, ...		

#	Action	Expected Result	Notes
	..... 131376432, 131376433, 131376434		
6	to make sure that all the values in the DS_UTC… data field are within your time range in step4 above. run GlasTime.rb 131375155 2004-03-01T01:05:55 run GlasTime.rb 131376434 2004-03-01T01:27:14	1st UTC time 131375155 = 2004-03-01T01:05:55 last UTC time 131376434 = 2004-03-01T01:27:14	
7	2) Temporal Subsetting within the granule.		
8	run subset_glas -f /sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH06.033/GLAH06_633_2107_002_0111_1_01_0001.H5 -s 2004-03-01T01:21:53 -e 2004-03-01T01:33:14 -o Test2_GLAH06_633_2107_002_0111_1_01_0001.H5 --excludegroup BROWSE	output product Test2_GLAH06_633_2107_002_0111_1_01_0001.H5 is produced in local directory	
9	run the commandline h5dump for outputing times below. run /h5dump -w 20 -y -m '%.0f' -d 'Data_1HZ/DS_UTCTime_1' -o test2_raw_GLAH06_633_utc Test2_GLAH06_633_2107_002_0111_1_01_0001.H5	Test2_GLAH06_633_utc contains data out from GLAH06_633_2107_002_0111_1_01_0001.H5	
10	when the h5dump command line finished, opening the test2_GLAH06_633_utc step 10 to get 1st UTC 131376115 and Last UTC 131376435. 131376115, 131376116, 131376117, ... 131376433, 131376434, 131376435		
11	to make sure that all the values in the DS_UTC… data field are within your time range in step9 above. Using the 1st and last utc times step 11 run GlasTime.rb 131376115 2004-03-01T01:21:55 run GlasTime.rb 131376435	1st UTC time 131376115 = 2004-03-01T01:21:55 last UTC time 131376435 = 2004-03-01T01:27:15 1st UTC time 131376115 = 2004-03-01T01:21:55 should match to starttime	

#	Action	Expected Result	Notes
	2004-03-01T01:27:15	in step 9 last UTC time 131376435 = 2004-03-01T01:27:15 should match to endtime in step 9	
12	3) Spatial Subsetting		
13	run subset_glas --bbox '359,-90,1,90' -f /sotestdata/DROP_82_HDF5_subsetter/GLAS/GLAH01.033/GLAH01_033_1102_001_0071_1_01_0001.H5 -o subsetted_glas.H5		
14	run /tools/hdf5/bin/h5dump subsetted_glas.H5 > out.h5.dmp		
15	Open out.h5.dmp. Search for DATASET "d1_pred_lat" . Verify that there are just two numbers (0.702106 and 359.823)		

TEST DATA:

EXPECTED RESULTS:

6.12 Verify Spatial Search Capability for GLAH01.033 (ECS-ECSTC-588)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use HDFView to find the spatial extent for each granule in the collection: f5dpl01v> /net/d5drl01v/devdata1/RCCCO/HDFView/bin/hdfview.sh `ls /sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH01.033/GLAH01_033*.H5` - Data_1HZ > Geolocation > d1_pred_(lat|lon)	Granule> South, East, North, West \<ol\> 50.215855, 95.92374799999999, 86.022775, -79.827765 -50.069131999999996, -79.842457, 50.185072, -96.12877 -86.023322, -96.14337, -50.130638, 88.04826899999999 -50.146969999999996, 88.03363499999999, 50.102391,	

#	Action	Expected Result	Notes
		71.747761 -86.015098, -156.702474, -50.043881999999996, 27.510407999999998 50.119558999999995, 11.175714, 86.014512, -164.558112	
2	In Reverb, enter the coordinates for a given granule into the spatial search feature. Choose the dataset(s) from which granules should be returned. (GLAH01.033, in this case.) Click [Search for granules]. (The search may take an extended period of time to complete.) Verify that the shown granules all occupy a coordinate(s) in the searched area. Repeat for each granule in the collection. 		

TEST DATA:

EXPECTED RESULTS:

6.13 End to End Performance Test (ECS-ECSTC-597)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login to the Data Access GUI for the test mode. Ensure that the MAX_GRANS_QUEUE_ASYNC_REQUESTS is set to a value greater than 2000		
2	Ensure that MAX_GRANS_SINGLE_ASYNC_REQUEST is greater than 2000		
3	On the reverb GUI, search and add at least 2000 GLAH06 granules to the cart.	request does not time out	

#	Action	Expected Result	Notes
4	Select temporal or spatial subsetting or parameter options and perform service request.		
5	Verify that the request completes successfully without errors or timeouts		
6	Verify that the stitched output can be downloaded sucessfully without any errors		
7	Verify that the zip file outputs can be downloaded successfully		
8	Attempt to order 1750 GLAH06 granules and 250 GLAH08 granules in a single request. Repeat steps 4-7	stitched product can be successfully downloaded.	
9			

TEST DATA:

EXPECTED RESULTS:

6.14 ESI GLAS Error Response Handling (ECS-ECSTC-604)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Run the following commands and make sure you get the appropriate error messages. Make sure to modify the urls based on which MODE you are using.		
2	Valid RQS invocation; if it doesn't work, the test should stop here. curl -# -XGET 'http://f5dpl01v:22500/rqs/GLAS?FILE_URLS=/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH01.033/GLAH01_033_1102_001_0071_1_01_0001.H5&BBOX=280,50,95,86&START=2003-02-20T20:38:48&END=2003-02-20T21:00:07&IDENTIFIER=GLAS_42&SUBAGENT_ID=GLAS'		
3	InvalidParameterValue (Can't open input file): curl -# -XGET 'http://f5dpl01v:22500/rqs/GLAS?FILE_URLS=/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH01.033/GL01_033_1102_001_0071_1_01_0001.H		

#	Action	Expected Result	Notes
	5&BBOX=280,50,95,86&START=2003-02-20T20:38:48&END=2003-02-20T21:00:07&IDENTIFIER=GLAS_42&SUBAGENT_ID=GLAS'		
4	InvalidParameterValue (Date-time): curl -# -XGET 'http://f5dpl01v:22500/rqs/GLAS?FILE_URLS=/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH01.033/GLAH01_033_1102_001_0071_1_01_0001.H5&START=2004-03-01T04:30:0&END=2004-03-01T04:30:00&IDENTIFIER=GLAS_42&SUBAGENT_ID=GLAS'		
5	InvalidParameterValue (Bounding box): curl -# -XGET 'http://f5dpl01v:22500/rqs/GLAS?FILE_URLS=/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH01.033/GLAH01_033_1102_001_0071_1_01_0001.H5&BBOX=280,50,95,&IDENTIFIER=GLAS_42&SUBAGENT_ID=GLAS'		

TEST DATA:

EXPECTED RESULTS:

6.15 GLAS Parameter Subsetting Test (ECS-ECSTC-633)

DESCRIPTION:

This test provides a functional check on the GLAS-Subsetter and GLAS-Tool-Adapter, for parameter subsetting.

For each granule (one each from three ESDTs) one run is made with just an appropriate subset_data_layers specification. A second is run with additional temporal and spatial constraints, with the constraints overlapping each other. Validation verifies the resultant datasets in the output files and the temporal and spatial subsetting performed.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Copy ruby scripts from clearcase: /ecs/formal/DPL/DataAccess/adapters/Glas/test*.rb to a working directory - on a machine with access to /sotestdata and to /workingdata. You will need all ruby scripts in the same working directory - they call each other and assume the home directory of the currently executing script is where the other		

#	Action	Expected Result	Notes
	scripts will be found		
2	Edit the scripts to make any adjustments for working host, port, mode, output-directory, etc. - in particular the SubmitGlasRequest.rb script. By default the scripts will write to /workingdata/TestResults/RelB/DataAccess/GLAS/ParamSubsetTest/. If there is data already there, check if it needs to be moved to establish baseline-results for a previous release or test run.		
3	Execute: ruby ./GlasParamSubsetTest.rb (no parameters)	Results summary at end of test should be empty (no faults or errors)	
4	Consider if test results should be moved to an appropriate folder location under /workingdata/testresults/		

TEST DATA:

The test draws on the following .h5 files under /sotestdata...

/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH05.033/GLAH05_633_2107_002_0110_3_01_0001.H5

/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH06.033/GLAH06_633_2107_002_0110_3_01_0001.H5

/sotestdata/DROP_802/HDF5_subsetter/GLAS/GLAH07.033/GLAH07_633_2107_002_0111_0_01_0001.H5

EXPECTED RESULTS:

6.16 GLAS Subsetter Unmatched Subset Constraints (ECS-ECSTC-634)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	For any two available GLAH granules, find a bounding box that only has coordinates in one of the two granules or a temporal range that only coincides with one of the two granules.	Bounding box: ullat=1, ullon=1, lrlat=0, lrlon=0 GLAH01 - No data. GLAH09 - Has	You can also see temporal ranges on the Reverb GUI when searching for granules.

#	Action	Expected Result	Notes
		data.	
2	In Reverb, perform a service (GLAS) on the two granules, using the bounding box or temporal range determined in Step 1. Submit the request.		Can be done via EGI as well if reverb is not available.
3	Ensure that the request completes. Ensure there is a message that states: 1 or more of your requested files did not meet the subset constraints	On the Reverb GUI ensure that the request reaches a status of 'Complete' Ensure that there is a message on the GUI that states: 1 or more of your requested files did not meet the subset constraints.	Can check the status of the EGI request if Reverb is not available.
4	Repeat steps 1-3 but this time only submit the granule that did not match the bounding box/temporal constraints. Specify the same bounding Box/Temporal range. Ensure the request completes but this time with a message that states: The files you selected contained no data that satisfy your subset constraints.	On the Reverb GUI ensure that the request reaches a status of 'Complete' Ensure that there is a message on the GUI that states: The files you selected contained no data that satisfy your subset constraints.	

TEST DATA:

EXPECTED RESULTS:

6.17 EGI Allocation of SubsetDataLayers (ECS-ECSTC-635)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Identify three granules in test mode, one each for GLAH05, GLAH06, GLAH07. This can be done by bringing up the ESI Inventory page for the collection in your web browser: http://f5dpl01v:22500/esi_TS2/inventory/GLAH07.33	GLAH07.033:81612:[Capabilities][Order Form][ECHO Form XML][Granule Info][Data Objects] GLAH07.033:81635:[Capabilities][Order Form][ECHO Form XML][Granule Info][Data Objects] The GranuleIds are the integer	Substitute host, port, mode as necessary

#	Action	Expected Result	Notes
		values following the short-name-version-id: e.g. 81612	
2	Submit the following URL for running EGI, substituting the host, port, mode and granule Ids as necessary: f5eil01v:22500/egi_TS2/request?FILE_IDS=81606,81607,81612&SUBAGENT_ID=GLAS&REQUEST_MODE=ASYNC&SUBSET_DATA_LAYERS=/GLAH05/Data_1HZ/Packet_data/,/GLAH06/Data_40HZ/Reflectivity,/GLAH07/Data_5HZ/Transmit_Energy	EGI returns the order Id / request Id	Typically EGI will process this request in the background, though an "immediate" reply on order complete is possible.
3	Submit the following URLs, substituting the order id received from step 2, for order information and status, repeating as necessary until all jobs (3) are complete. http://f5eil01v:22500/egi_TS2/request/793 http://f5eil01v:22500/esir_TS2/793/ http://f5eil01v:22500/esir_TS2/793/requestSummary.txt	EGI & ESIR status pages	
4	Using the links provided in the esir status page (...esir_TS2/793), download the three output files to a convenient working directory.	Three files downloaded, with the names: subsetted_GLAH05_633_2107_002_0113_1_01_0001.H5 subsetted_GLAH06_633_2107_002_0113_1_01_0001.H5 subsetted_GLAH07_633_2107_002_0111_0_01_0001.H5	The names should include subsetted_GLAH##_ but otherwise may vary slightly depending upon which granules were selected for processing in step 1
5	Enter the following URL to bring up the Data Access GUI web page (substitute host, port and mode as necessary) http://f5dpl01v:22500/DataAccessGui_TS2/ . Enter the username and password for the Data Access GUI login. 	The Data Access GUI web page is displayed	
6	Click on the Monitor tab and click on the Show-All-Agents button (or click on the GLAS agent listed in the table above). Click on the Request Id column header twice to bring the recent requests to the top of the list. Click on the Request Id provided in Step 2.	The Jobs table to the right will show the three jobs associated with this request (one per granule)	
7	Click on each job listed and note for each the job information window that pops-up. Note in particular the URL that is displayed. Verify the subset_data_layer is different in each job and provides the correct subset_data_layer for each granule, as follows: GLAH05: Data_1HZ/Packet_data GLAH06: Data_40HZ/Reflectivity GLAH07: Data_5HZ/Transmit_Energy	The job URLs will show the Subset_Data_Layers have been distributed from the request to each job, based upon the collection of the granules, and with the shortname prefix removed from the Subset_Data_Layer	
8	Run the following command for each of the three processed files downloaded: 	This command dumps the datasets contained in the processed file. The	

#	Action	Expected Result	Notes
	 f5dpl01v{dauty}44: /tools/hdf5/bin/h5dump -n ./subsetted_GLAH06_633_2107_002_0113_1_01_0001.H5 | grep 'dataset' | awk '{print $2}'	datasets contained should exactly match the subset_data_layer constraint applied for that collection. E.g.: /Data_40HZ/Reflectivity/d_RecNrgAll /Data_40HZ/Reflectivity/d_reflctUC /Data_40HZ/Reflectivity/d_sDevNsOb1 /Data_40HZ/Reflectivity/d_satNrgCorr	

TEST DATA:

EXPECTED RESULTS:

6.18 End to End GLAS HDF5 parameter subsetting (ECS-ECSTC-639)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test GLAS collections are installed in the mode and exported to ECHO. e.g. GLAH05.033 snd GLAH07.033		
2	Ensure that the GLAS subsetter service is installed in the mode enabled for the test collections		
3	Ensure that the spatial and temporal subsetting and band subsetting is enabled for the collections for the GLAS subsetting service		
4	Navigate to the Reverb testbed GUI. http://testbed.echo.nasa.gov/reverb		
5	Search for granules for the test GLAS collections.		
6	Select at least a couple of granules from each test collection and View cart.		
7	Click on the Granule Detail icon (i) and download the OnlineAccess URL for		

#	Action	Expected Result	Notes
	the granules.		
8	Run /tools/common/test/DataAccess/glasdump.rb test_granule.H5 to create dumps for the latitude, longitude and utc times or manually run the h5dump as follows... To get the list of latitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lat' -o lat1hz.dat GLAH_INPUT_GRANULE.H5 To get the list of longtitudes h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/Geolocation/d_lon' -o lon1hz.dat GLAH_INPUT_GRANULE.H5 To get the utc times h5dump -w 20 -y -m '%.15f' -d 'Data_1HZ/DS_UTCTime_1' -o raw_time1hz.dat GLAH_INPUT_GRANULE.H5		
9	Record the spatial extent for the test granules from lat1hz.dat and lon1hz.dat from the glasdump tool or h5dump or from the info and map view of the granules in reverb gui		
10	Record the temporal range for the test granules from the raw_time1hz from the glasdump tool or from the info and temporal extent displayed in reverb		
11	Click on Perform Service and Verify that the form options are displayed correctly.		
12	Check the box for 'Use these values for all applicable items'		
13	Enter a valid email address		
14	Select GLAS processing tool		
15	Ensure that the spatial constraint spans full earth		
16	Ensure that the temporal constraint spans the temporal extents for all the granules in the cart.		
17	Select appropriate parameter subset options for granules in each dataset		
18	Submit Service Request and verify that the requests succeeded.		
19	Verify in the EGI/ESIR logs that stitching has been called for each of the collections with multiple granules.		
20	Verify that all the stitch links are listed under "Download the files stitched together by Product:"		
21	Verify that there is a url for enabling stitching displayed and there is one link for each collection that has multiple granules.		
22	Click on stitched output link for each test collection to download the stitched ouput.		
23	Verify that the request summary link displays the request summary with all		

#	Action	Expected Result	Notes
	the subsetting parameter information, the input and the output granule list		
24	Verify that an email is sent to the requested email address that the service request is complete with links to download the subsetted files		
25	Download the files from the output links and verify that the subsetting matches the parameter subset options selected in the request.You can verify it by using the hdfview tool		

TEST DATA:

EXPECTED RESULTS:

7 SDPS-477 RESTRICT GRANULE UTILITY

7.1 Add a Restriction Flag (ECS-ECSTC-3875)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use the Restrict Granule Utility to add a Restriction Flag in the ECS Database		EcDsAmRestrictGranule.py --mode DEV02 --add-flag 50 --flag-description this is flag 50
2	Verify the add-flag action is logged in the Restrict Granule log file	"Inserted New RestrictionFlag" in the log file	
3	Verify the flag was inserted into the AIM database		select * from dsmdrestrictionflag

TEST DATA:

EXPECTED RESULTS:

7.2 Use the Restrict Granule Utility to add restrictions with Hide = Y (ECS-ECSTC-3876)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use the Restrict Granule Utility to add a Restriction to Granules with Hide = Y		-Note the state of the isOrderOnly for the granule before restricting the granules -Use the RestrictionFlag created in the earlier test case
2	Verify that granules that are restricted logged in the log file		
3	Verify that the restricted granules were sent to the Unpublish Utility		
4	Verify that the new restricted granules, restriction flag and hide/unpublishflag are noted in the DsMdGranuleRestriction table		select * from dsmdgranulerestriction
5	Verify these granule were unpublished by the Unpublish Utility.	isOrderOnly = 'H' or 'Y'	select isOrderOnly from amcollection where granuleid in (1234, 2345)
6	Verify that the BMGT exports and event for the restricting the granule. If the granule was public before and now is hidden, verify there is a second export event for that granule		

TEST DATA:

EXPECTED RESULTS:

7.3 Use the Restrict Granule Utility to add restrictions with Hide = N (ECS-ECSTC-3878)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use the Restrict Granule Utility to add a Restriction to Granules with Hide = N		-Note the state of the isOrderOnly for the granule before restricting the granules -Use the RestrictionFlag created in the earlier test case
2	Verify that granules that are restricted logged in the log file		
3	Verify that the restricted granules were sent to the Publish Utility		
4	Verify that the new restricted granules, restriction flag and hide/unpublishflag (N) are noted in the DsMdGranuleRestriction table		select * from dsmdgranulerestriction
5	Verify these granule were published by the Publish Utility.	isOrderOnly = '' or 'B'	select isOrderOnly from amcollection where granuleid in (1234, 2345)
6	Verify that the BMGT exports and event for the restricting the granule. If the granule was hidden before and now is published, verify there is a second export event for that granule		

TEST DATA:

EXPECTED RESULTS:

7.4 Unrestrict granules using the Restrict Granule Utility (ECS-ECSTC-3879)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use the Restrict Granule Utility to unrestrict granules from the previous test cases		
2	Verify that granules that are being unrestricted are logged in the log file		
3	Verify the granules are sent to the Publish/Unpublish utility based on their	If the collection's	select publishbydefaultflag

#	Action	Expected Result	Notes
	collection's PublishByDefaultFlag	PublishByDefaultFlag is 'Y' the granule will be sent to the Publish Utility . If it is 'N' the granule will be sent to the Unpublish Utility.	from amcollection where collectionid in (select collectionid from amgranule where granuleid = 123)
4	Verify that the BMGT exports and event for unrestricting the granule. If the granule was public before and now is hidden or vise-versa, verify there is a second export event for that granule		

TEST DATA:

EXPECTED RESULTS:

7.5 Restrict Granules: End to End Test (ECS-ECSTC-3880)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login into EarthData Login as with no special privileges (create a new user if needed) and use EDSC to find a granule to restrict.		
2	Use the Restrict Granule Utility to add a Restriction to the previously identified granule		Create a new RestrictionFlag or reuse one to add restrictions to granules
3	Verify BMGT exported the restriction to CMR		
4	As an admin use the MMT to create an Access Control List (also called a “Permission”) that allows non-privileged registered users to see granules that have no restrictions.		
5	Verify the Non-privileged user created in step one cannot see the granule in EDSC because it is restricted		
6	Verify the Privileged user can see the granule in EDSC because user is authorized.		
7	Use the Restrict Granule Utility to unrestrict the granule in the previous step.		

#	Action	Expected Result	Notes
8	Verify BMGT exported the unrestriction to CMR		
9	Verify that both Privileged and Non-privileged user can see the unrestricted granule		

TEST DATA:

EXPECTED RESULTS: