ECS Regression Test Cases

Table of Contents

1 AIM 1
1.1 Test Case 1 - Install ESDTs with a variety of spatial search types (ECS-ECSTC-163) 1
1.2 Test Case 2 - Update multiple ESDTs, success case - [Needs to be rewritten - ODL vs XML] (ECS-ECSTC-164) 22
1.3 Test Case 4 - Search Granules for Deletion (ECS-ECSTC-166) 27
1.4 Test Case 6 -Granule Logical Deletion -- DFA (ECS-ECSTC-168) 28
1.5 Test Case 7 - Undelete -physical (ECS-ECSTC-169) 30
1.6 Test Case 8 - Undelete -- DFA (ECS-ECSTC-170) 31
1.7 [PVC] - Test Case 9 - Granule Deletion Phase II - DeletionCleanup - (Needs re-write) (ECS-ECSTC-171) 33
1.8 Test Case 12 - Multiple Update request files processed in time tag order, Success Case (ECS-ECSTC-173) 34
1.9 [PVC ONLY] - Test Case 13 - QA Update Performance run : DS_7E_01, Criterion 3200 (ECS-ECSTC-174) 37
1.10 Test Case 14 - XML Replacement Utility:Successfully replace a granule metadata file (ECS-ECSTC-175) 38
1.11 [PVC] - Test Case 16 - DPL â€“ AIM Inventory Validation - (Needs re-write) (ECS-ECSTC-177) 39
1.12 Test Case 17 - Tape Archive Repair: DP_7F_01 Criterion 800 (ECS-ECSTC-178) 47
1.13 Test Case 18 - On-line Archive Repair from non-resident tape: DP_7F_01 Criterion 320 (ECS-ECSTC-179) 48
1.14 [PVC] - Test Case 19 - Failure Recovery Performance, Failure in AIM Inventory DB : DS_7E_01_QAUU, Criterion 3210 (ECS-ECSTC-180) 50
1.15 Test Case 20 - Running QA Update utility while Data Pool Move Collection utility is running in the same mode; Error Case : DS_7E_01_QAUU, Criterion 3170 (ECS-ECSTC-181) 51
1.16 Test Case 21 - Fault recovery â€“ physical delete; continue with previous run : DS_7E_01_GD, Criterion 1280 (ECS-ECSTC-182) 52
1.17 Test Case 22 - Failure Recovery during Update, database access failure : DS_7E_01_ESDTGUI, Criterion 2330 - (Needs re-write) (ECS-ECSTC-183) 53
1.18 Test Case 23 - Failure Recovery during Update, ESDT Maintenance GUI failure : DS_7E_01_ESDTGUI, Criterion 2337 (ECS-ECSTC-184) 55
1.19 Test Case 24 - Error Handling Options: recoverOnly and recoverInvestigated: DS_7E_01_QAUU, Criterion 3230] (ECS-ECSTC-185) 56
1.20 Test Case 25 - Fault recovery - delete or DFA : DS_7E_01_GD, Criterion 1260 (ECS-ECSTC-186) 58
1.21 Test Case 26 - RunArchive Checksum Validation Utility (ACVU) providing a file with a list of dbIds without the -calculate option : CK_7F_01, Criterion 670 (ECS-ECSTC-187) 58
1.22 [PVC] - Test Case 27 - AIM XML Check Utility (ECS-ECSTC-188) 59
1.23 Test Case 28 - EcDsCheckXMLArchive utility (ECS-ECSTC-189) 60

1.24 Test Case 29 - On-Line Archive Repair including QA and PH links: DP_7G_01 Criterion 300] (ECS-ECSTC-190) 61
1.25 AIM Modified 62
1.25.1 Test Case 3 - Delete multiple ESDTs, success case (ECS-ECSTC-165) 62
1.25.2 [PVC] - Test Case 5 - Granule Logical Deletion -- physical (ECS-ECSTC-167) 65
1.25.3 [PVC] - Test Case 11 - QAUU: Update by ESDT and temporal range for specific measured parameters (ECS-ECSTC-172) 66
1.25.4 [PVC] - Test Case 15 - Archive Check Utility (ECS-ECSTC-176) 72

2 BMGT 73
2.1 BMGT Automated 73
2.1.1 Nominal Collection Export[S-1]: Manual Export (ECS-ECSTC-397) 73
2.1.2 ECS Collection Additional Metadata[S-01]: ECS CoordinateSystem (ECS-ECSTC-398) 77
2.1.3 Nominal Granule Export[S-2i]: Automatic Export: Publish Granule (ECS-ECSTC-400) 78
2.1.4 Nominal Granule Export[S-2j]: Automatic Export: Unpublish Granule (ECS-ECSTC-401) 83
2.1.5 Nominal Granule Export[S-2k]: Automatic Export: Link to Browse (ECS-ECSTC-402) 88
2.1.6 Nominal Granule Export[S-2l]: Automatic Export: Unlink from Browse (ECS-ECSTC-403) 90
2.1.7 Nominal Collection Export[S-2d]: Automatic Export: Enable Collection Export (ECS-ECSTC-413) 93
2.1.8 Nominal Granule Export[S-2a]: Automatic Export: Ingest Granule (ECS-ECSTC-414) 94
2.1.9 Nominal Granule Export[S-2b]: Automatic Export: Logically Delete Granule (ECS-ECSTC-415) 99
2.1.10 Nominal Granule Export[S-2c]: Automatic Export: Physically Delete Granule (ECS-ECSTC-416) 103
2.1.11 Nominal Granule Export[S-2d]: Automatic Export: DFA Granule (ECS-ECSTC-417) 107
2.1.12 Nominal Granule Export[S-2e]: Automatic Export: Hide a Granule (ECS-ECSTC-419) 109
2.1.13 Nominal Granule Export[S-2f]: Automatic Export: Restrict Granule (ECS-ECSTC-420) 112
2.1.14 Nominal Granule Export[S-2g]: Automatic Export: Unrestrict Granule (ECS-ECSTC-421) 114
2.1.15 Nominal Granule Export[S-2h]: Automatic Export: QAUpdate granule. (ECS-ECSTC-422) 117
2.1.16 Nominal Granule Export[S-2n]: Automatic Export: Move Collection (ECS-ECSTC-423) 119
2.1.17 Nominal Granule Export[S-2o]: Automatic Export: XML Replacement (ECS-ECSTC-424) 122
2.1.18 ECS Collection Additional Metadata[S-02]: ECS GranuleSpatialRepresentation (ECS-ECSTC-425) 124

2.1.19 ECS Collection Additional Metadata[S-03]: ISO CoordinateSystem (ECS-ECSTC-426) ... 127
2.1.20 ECS Collection Additional Metadata[S-04]: ISO GranuleSpatialRepresentation (ECS-ECSTC-427) ... 130
2.1.21 ECS Collection Additional Metadata[S-05]: backtrack (ECS-ECSTC-428) ... 133
2.1.22 ECS Collection Additional Metadata[S-06]: no backtrack (ECS-ECSTC-429) ... 136
2.1.23 ECS Collection Additional Metadata[S-07]: TwoDCoordinateSystem (ECS-ECSTC-430) ... 139
2.1.24 ECS Collection Additional Metadata[S-08]: no TwoDCoordinateSystem (ECS-ECSTC-431) ... 142
2.1.25 ECS Collection Additional Metadata[S-09]: Product Specific Attributes (PSAs) (ECS-ECSTC-432) ... 145
2.1.26 ECS Collection Additional Metadata[S-10]: DIF ID (ECS-ECSTC-433) ... 150
2.1.27 ECS Collection Additional Metadata[S-12]: null ECS CoordinateSystem (ECS-ECSTC-434) ... 154
2.1.28 ECS Collection Additional Metadata[S-13]: null ECS GranuleSpatialRepresentation (ECS-ECSTC-435) ... 157
2.1.29 Manual Export - Collection Deletes (ECS-ECSTC-438) ... 160
2.1.30 Manual Export - Collection Full Update (ECS-ECSTC-439) ... 161
2.1.31 Manual Export - Granule Insert (ECS-ECSTC-440) ... 163
2.1.32 Manual Export - Last Update Time (ECS-ECSTC-441) ... 165
2.1.33 Nominal Collection Export[S-2a]: Automatic Export: Insert Collection (ECS-ECSTC-443) ... 168
2.1.34 Nominal Granule Export[S-1]: Manual Export (ECS-ECSTC-445) ... 170
2.1.35 Manual Export - Insert Time (ECS-ECSTC-640) ... 175
2.1.36 Manual Export - "Fake" Collection Delete (ECS-ECSTC-641) ... 176
2.1.37 Nominal Collection Export[S-2c]: Automatic Export: Update Collection (ECS-ECSTC-642) ... 177
2.1.38 Manual Export - Collection Full Update (ECS-ECSTC-644) ... 179
2.1.39 Manual Export - Granule Insert (ECS-ECSTC-645) ... 180
2.1.40 Manual Export - Collection Inserts (ECS-ECSTC-647) ... 185
2.1.41 Manual Export - Collection Deletes (ECS-ECSTC-648) ... 186
2.1.42 Manual Export - Granule Delete (ECS-ECSTC-649) ... 191
2.1.43 Long Form Verification - Collection (ECS-ECSTC-650) ... 193
2.1.44 Long Form Verification - Granule (ECS-ECSTC-651) ... 194
2.1.45 Long Form Verification - Insert Time (ECS-ECSTC-652) ... 195
2.1.46 Long Form Verification - Last Update Time (ECS-ECSTC-653) ... 197
2.1.47 Long Form Verification - Granule Updates (ECS-ECSTC-654) ... 198
2.1.48 Nominal Collection Export[S-2c]: Automatic Export: Update Collection (ECS-ECSTC-412) ... 199
2.2 BMGT Modified ... 203
2.2.1 Long Form Verification - Invocation via Command Line - b) Granules in Granule File (ECS-ECSTC-399) ... 203
2.2.2 Incremental Verification - Nominal (ECS-ECSTC-404) ... 206
2.2.3 MISR Browse Links - Science Before Browse (ECS-ECSTC-406) ... 218
2.2.4 MISR Browse Links - Browse Before Science (ECS-ECSTC-407) ... 225

2.2.5 Contact ECHO Errors (ECS-ECSTC-409) ... 231
2.2.6 Manual Export - Science granule with not yet published Browse (ECS-ECSTC-473) 233
2.2.7 Manual Export - Collection Deletes (ECS-ECSTC-665) ... 237
2.2.8 Short Form Verification - Exclude Pending Exports (ECS-ECSTC-673) ... 243
2.2.9 ISO Nominal Granule Export: Granule Ingest (ECS-ECSTC-674) ... 246
2.2.10 Alert After Numerous Retries (ECS-ECSTC-675) ... 249

3 CLS ... 253
3.1 OBSOLETE - Order Status GUI: Display Order Status, OD_S6_05. Criterion 130 (ECS-ECSTC-191) ... 253
3.2 OBSOLETE - Order Status GUI: Display Order Status based on History Range, 0D_S6_05, Criterion 180 (ECS-ECSTC-192) ... 254
3.3 Validating HDF data files (ECS-ECSTC-193) ... 256

4 CSS ... 256
4.1 CSS Registry (ECS-ECSTC-194) ... 256

5 DMS ... 257
5.1 OBSOLETE- Test Case 3 - Single Granule External Processing Order (ECS-ECSTC-195) ... 257
5.2 Test Case 1 - Single Granule Order (ECS-ECSTC-196) ... 258
5.3 Test Case 2 - Cancelled and Failed Granules (ECS-ECSTC-197) ... 258
5.4 Test Case 4 - Single Granule Order with invalid ESDT or Granule ID (ECS-ECSTC-198) 259
5.5 Test Case 5 - Double granule order containing simple items with different media requirements (ECS-ECSTC-199) ... 259
5.6 Test Case 6 - Multiple Granules Order (ECS-ECSTC-200) ... 260
5.7 Test Case 7 - Duplicate Requests (ECS-ECSTC-201) ... 261
5.8 [OBSOLETE] Test Case 8 - Single Granule Order using the ASTER GDEM Form - (Needs to be rewritten) (ECS-ECSTC-202) ... 262
5.9 [OBSOLETE] Test Case 9 - Single Granule Order with associated PH, QA, Browse Granules - (Needs to be rewritten) (ECS-ECSTC-203) ... 263

6 DPL INGEST ... 264
6.1 Test Case 1 - Use the DPL Ingest GUI to configure Data Providers (ECS-ECSTC-244) 264
6.2 Test Case 2 - Use the DPL Ingest GUI to configure Polling Locations (ECS-ECSTC-245) 266
6.3 Test Case 5 - View and modify the DPL Ingest GUI configuration (ECS-ECSTC-248)267
6.4 [PVC] - Test Case 6 - Ingest SIPS granules without publishing to the Data Pool (ECS-ECSTC-249) ... 268
6.5 [PVC] - Test Case 7 - SIPS Ingest and publish to public Data Pool, Band Extraction (ECS-ECSTC-250) ... 271
6.6 Test Case 8 - Cross-DAAC Ingest (ECS-ECSTC-251) ... 273
6.7 [PVC] - Test Case 9 - Ingest EDOS Expedited EDRs (ECS-ECSTC-252) ... 274
6.8 [PVC] - Test Case 10 - Verifies EMOS Ingest (Polling without delivery record functionality) (ECS-ECSTC-253) ... 276

6.9 [PVC] - Test Case 12 - Verifies ASTER Ingest (ECS-ECSTC-255) 277
6.10 Test Case 13 - Verifies MISBR Ingest and linkages are handled correctly (ECS-ECSTC-256) 279
6.11 [PVC] - Test Case 14 - FTP Polling, Transfers, and Notification (ECS-ECSTC-257) .. 280
6.12 [PVC] - Test Case 15 - Local Polling, Transfers, and Email Notification (ECS-ECSTC-258) 281
6.13 [PVC] - Test Case 16 - SCP Polling, Transfers, and Notification (ECS-ECSTC-259) .. 281
6.14 Test Case 16b - HTTP Polling, Transfers, and Notification (ECS-ECSTC-498) 283
6.15 [PVC] - Test Case 17 - Browse granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-260) 283
6.16 Test Case 18 - Browse granule(s) in separate PDR from science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-261) 285
6.17 [PVC] - Test Case 19 - QA granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-262) 286
6.18 [PVC] - Test Case 20 - PH granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-263) 287
6.19 Test Case 21 - PH granule(s) in different PDR from science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-264) 288
6.20 Test Case 22 - Data Pool Registration Error â€“ Retried (ECS-ECSTC-265) 289
6.21 Test Case 23 - Data Pool Registration Error â€“ Failed. (ECS-ECSTC-266) 291
6.22 Test Case 24 - Add & Modifying Volume Groups (ECS-ECSTC-267) 293
6.23 Test Case 25 - Verifies correct linking of AST_L1A, AST_L1B, and Browse (ECS-ECSTC-268) 293
6.24 Test Case 26 - Checksum during DPL Ingest of Browse, QA, PH, DAP (ECS-ECSTC-269) 296
6.25 Test Case 27 - Fault Recovery (ECS-ECSTC-270) 297
6.26 DPL Ingest Modified 298
6.26.1 Test Case 3 - Use the DPL Ingest GUI to configure Remote Transfer Hosts (ECS-ECSTC-246) 298
6.26.2 Test Case 4 - Use the DPL Ingest GUI to configure SCP Hosts (ECS-ECSTC-247) 299
6.26.3 [PVC] - Test Case 11 - Verifies EDOS Ingest (ECS-ECSTC-254) 300

7 DATA ACCESS 301
7.1 DP_81_02_TP011 Configure and Retrieve SDPS Processing Options (ECS-ECSTC-376) 301
7.2 DP_81_02_TP013 SDPS Web API Request Error Handling (ECS-ECSTC-378) 304
7.3 DP_81_02_TP030 SDPS Web API request cleanup (ECS-ECSTC-383) 307
7.4 DP_81_02_TP031 SDPS Web API asynchronous requests (ECS-ECSTC-384) 308
7.5 DP_81_02_TP033 SDPS Web API suspend and resume request acceptance (ECS-ECSTC-385) 310
7.6 DP_81_02_TP035 SDPS Web API monitoring, management, and reporting (ECS-ECSTC-386) 312
7.7 DP_81_02_TP039 SDPS Web API capacity management and tuning (ECS-ECSTC-388) 314

7.8 DP_81_02_TP038 SDPS Web API enable and disable processing for collections (ECS-ECSTC-389) 316
7.9 DP_81_02_TP018 Ordering with Processing Instructions (ECS-ECSTC-379) 317
7.10 GDAL 321
7.10.1 DP_81_02_TP050 GDAL processing requests and errors (ECS-ECSTC-483) 321
7.10.2 Configure GDAL collection formats - validate formats and collections (ECS-ECSTC-507) 322
7.10.3 Configure GDAL collection formats - configure single collection and single format (ECS-ECSTC-509) 323
7.10.4 Configure GDAL collection formats - configure all collections for supported gdal formats (ECS-ECSTC-519) 325
7.10.5 Configure GDAL collection formats - options (ECS-ECSTC-525) 327
7.10.6 Configure GDAL collection formats - configure single collection for all supported gdal formats (ECS-ECSTC-526) 330
7.10.7 Configure GDAL collection formats - configure limited collections - with granule state changes (ECS-ECSTC-527) 331
7.10.8 Configure GDAL collections DataObjects All_Collections (ECS-ECSTC-503) ... 333
7.10.9 Configure DataObjects - options (ECS-ECSTC-528) 335
7.10.10 DataAccess GDAL End To End Testing (ECS-ECSTC-522) 341
7.10.11 Configure Collection for GDAL band subsetting (ECS-ECSTC-504) 342
7.11 GLAS 343
7.11.1 Processing GLAS through OMS (ECS-ECSTC-563) 343
7.12 EMS 344
7.12.1 Data Access EMS testing with duplicate filenames and special characters (ECS-ECSTC-655) 344
7.12.2 EMS reporting for GLAH requests (ECS-ECSTC-656) 345
7.12.3 DP_81_02_TP036a EMS SDPS Web API metrics (ECS-ECSTC-529) 347
7.12.4 DP_81_02_TP036 Export SDPS Web API metrics to EMS (ECS-ECSTC-387) ... 350
7.13 HEG 353
7.13.1 DP_81_02_TP012 OMS Invocation of HEG Tool Adapter (ECS-ECSTC-377) ... 353
7.13.2 DP_81_02_TP044 SDPS Web API HEG Regression - AMSR (ECS-ECSTC-394) 355
7.13.3 DP_81_02_TP043 SDPS Web API HEG Regression - MISR (ECS-ECSTC-392) 355
7.13.4 DP_81_02_TP042 SDPS Web API HEG Regression - ASTER (ECS-ECSTC-391) 365
7.13.5 DP_81_02_TP041 SDPS Web API HEG Regression - MODIS (ECS-ECSTC-390) 367
7.13.6 DP_81_02_TP003 HEG Error Handling (ECS-ECSTC-393) 375
7.13.7 DP_81_02_TP004 ESI HEG processing errors (ECS-ECSTC-381) 379
7.13.8 HEGModified 380
7.13.8.1 DP_81_02_TP001 Submit HEG processing requests (ECS-ECSTC-380) 380
7.13.8.2 HEG Service request (ECS-ECSTC-658) 383
7.13.8.3 TP001_10 (Revised, Automated) (ECS-ECSTC-660) 383
7.14 ESI 384
7.15 EGI 384

7.15.1 DP_81_02_TP065 EGI filesize, mimetype, and filetype check (ECS-ECSTC-605) 384
7.15.2 EGI Notification Level Tests (ECS-ECSTC-676) 385
7.16 ESIR 387
7.16.1 DP_81_02_TP055 - Disable Zip Generation (ECS-ECSTC-501) 387
7.16.2 DP_81_02_TP060 ESIR Zip File Generation (ECS-ECSTC-601) 389
7.17 SII 390
7.17.1 DP_81_02_TP009 Simple Inventory Interface (ECS-ECSTC-382) 390
7.18 DA GUI Configuration 393
7.18.1 Configure LDOPE Service (ECS-ECSTC-521) 393
7.18.2 Configure Convert Enabled Flag (ECS-ECSTC-523) 399

8 EMS 401
8.1 Test Case 1e - g - EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to specify an output directory for the flat files. (ECS-ECSTC-271) 401
8.2 Test Case 2 - EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to override the default 24-hour period execution with an alternative starting and ending date. (ECS-ECSTC-272) 402
8.3 Test Case 3 - EMS Extract Utility: The EMS Dataset Extraction Utility shall rename a flat file if it has been resent to the EMS. (ECS-ECSTC-273) 403
8.4 Test Case 4 - EMS Extract Utility: (see Note) Verify that the EMS extract utility can be run as a cron (ECS-ECSTC-274) 404
8.5 RECOVERY (see Note) EMS Extract Utility: Verify for the default configuration run of the EMS extract utility, thatif the script does not run for a few days, the script can determine when it last ran successfully by retrieving the last successful run infor (ECS-ECSTC-275) 405
8.6 Test Case 6 - RECOVERY (see Note) EMS Extract Utility: Verify that if a dataset file has been extracted to the extract directory, but not sftpâ€™d, a subsequent run of the utility will sftp this file and mark the file as sftpâ€™d (ECS-ECSTC-276) 406
8.7 RECOVERY (see Note) EMS Extract Utility: Verify that if a dataset file has been removed from the extract directory, but not sftpâ€™d, a subsequent run of the utility will not crash when an attempt to sftp this file is made. Instead, the file will be marke (ECS-ECSTC-277) 407
8.8 (see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to override the default 24-hour period execution with an alternative starting and ending date. The data will be output to extract files in 24-hour (ECS-ECSTC-278) 408
8.9 Test Case 9 - EMS Extract Utility: The EMS Dataset Extraction Utility will not output data in less than a 24-hour period (ECS-ECSTC-279) 409
8.10 [EDF OPS / PVC TS2]Test Case 11 - EMS Extract Utility:Any EMD code changes will need to be Verified against the EMS ICD and established Processes (ECS-ECSTC-281) 410
8.11 (see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to extract QA | PH data for DistFTP | DistHTTP (ECS-ECSTC-282) 410

8.12 EMS Extract Utility: The EMS Dataset Extraction Utilityshall allow operator to set a default date for subsequent runs of the utility either for all data sets or for one dataset (ECS-ECSTC-283) 411
8.13 (see Note) EMS Extract Utility: The EMS Dataset Extraction Utilityshall allow operator to set the LAG time for running of default dataset(s) (ECS-ECSTC-284) 412
8.14 (see Note) EMS Extract Utility: shall allow operator to set the DESTINATIONDIR for data that is sent to EMS (ECS-ECSTC-285) 413
8.15 (see Note) EMS Extract Utility: shall allow intendedUsage data to be sent to EMS for MSS/OMS data (ECS-ECSTC-286) 414
8.16 EMS Modified 415
8.16.1 EMS Extract Utility: If only the <begin date> is specified for the EMS Dataset Extraction Utility the script will output dataset files from that specific day forward to the current date minus 2 days (ECS-ECSTC-280) 415
8.17 Test Case 1 - EMS Extract Utility: Verify that flat files are created and exported for all datasets when Extract utility is run using default configuration (ECS-ECSTC-889) 416
8.18 Test Case 1a - EMS Extract Utility: The EMS Dataset Extraction Utility shall maintain a record of the status and starting/ending reporting period of each execution. (ECS-ECSTC-890) 417
8.19 Test Case 1b - EMS Extract Utility: The EMS Dataset Extraction Utility shall by default generate and export EMS flat file information for a 24-hour period (ECS-ECSTC-891) 418
8.20 Test Case 1c - EMS Extract Utility: The EMS Dataset Extraction Utility shall by default use the ending date of the previous reporting period to determine the starting and ending date for the current execution. (ECS-ECSTC-892) 418
8.21 Test Case 1d - EMS Extract Utility: The EMS Dataset Extraction Utility shall by default generate and export each of the following types of flat files per execution (ECS-ECSTC-893) 419
8.22 Test Case 1h - EMS Extract Utility: Verify that in extracted dataset files, there is one record per line and each record is divided into fields using '|&|' as a delimiter. (ECS-ECSTC-900) 420

9 HEG 421
9.1 Resample, swtif and gdtif conversion (ECS-ECSTC-287) 421

10 MSS 422
10.1 System Components Availability (ECS-ECSTC-288) 422
10.2 Alerting Capability (ECS-ECSTC-289) 422
10.3 Resource usage monitoring (ECS-ECSTC-290) 423
10.4 Disk Space Usage (ECS-ECSTC-291) 424
10.5 Custom Code Availability (ECS-ECSTC-292) 425
10.6 GUI â€“ Configure Operator and Roles (ECS-ECSTC-293) 426
10.7 DPL Ingest Business Process (ECS-ECSTC-294) 427
10.8 Resource Availability and Usage Reporting (ECS-ECSTC-295) 429
10.9 Ingest Queue Size Monitoring (ECS-ECSTC-296) 430
10.10 Ingest Alert and Intervention Monitoring (ECS-ECSTC-297) 432
10.11 Database Monitoring (ECS-ECSTC-298) 433
10.12 File Transfer Monitoring (ECS-ECSTC-299) 433

10.13 External Service Monitoring (ECS-ECSTC-300) 434
10.14 OMS Alert and Intervention Monitoring (ECS-ECSTC-301)..... 435
10.15 OMS Queue Monitoring (ECS-ECSTC-302)..... 436
10.16 Custom Code Metrics (ECS-ECSTC-303)..... 436
10.17 Custom Utility Monitoring (ECS-ECSTC-304)..... 437
10.18 Data Access Business Process (ECS-ECSTC-305)..... 438
10.19 Order Management Business Process (ECS-ECSTC-306) 440

11 OMS 442
11.1 Test Case 5 - DPL file system unavailable (ECS-ECSTC-311)..... 442
11.2 [OBSOLETE] Test Case 7 - Ftp Pull HEG Order (ECS-ECSTC-313)..... 443
11.3 [OBSOLETE] Test Case 8 - Ftp Push HEG Order (ECS-ECSTC-314)..... 444
11.4 Test Case 10 - Ftp Push/SCP Destination Automatic Retry (ECS-ECSTC-316)..... 445
11.5 Test Case 11 - Ftp Push/SCP Manual Destination Retry (ECS-ECSTC-317)..... 446
11.6 Test Case 12 - FTP Push/SCP: Granule Error Operator Intervention (ECS-ECSTC-318) 446
11.7 OBSOLETE - Test Case 13 - ODL to XML (ECS-ECSTC-319) 448
11.8 [PVC ONLY] Test Case 14 - External Subsetting Order (ECS-ECSTC-320)..... 448
11.9 [PVC ONLY] Test Case 16 - Fault Recovery â€“ Bouncing the Order Manager Server (ECS-ECSTC-322) 449
11.10 Test Case 17 - OMS GUI: Non Privileged User Test (ECS-ECSTC-323) 450
11.11 Test Case 20 - Global Non-Configured FtpPush Configuration (ECS-ECSTC-324) .. 450
11.12 Test Case 21 - Valgrind Memory Test (Optional) (ECS-ECSTC-325) 451
11.13 Tet Case 22 - DeleteFromArchive = H Granules (ECS-ECSTC-326)..... 452
11.14 EcOmCleanupOldData.pl (ECS-ECSTC-395)..... 452
11.15 EcOmDbCleanArchive.pl (ECS-ECSTC-396)..... 455
11.16 OMS Modified..... 457
11.16.1 [PVC] Test Case 4 - SCP Request for a Configured Destination (ECS-ECSTC-310) 457
11.16.2 Test Case 1 - HTTPS Request (ECS-ECSTC-307) 458
11.16.3 Test Case 2 - FtpPush Request for a Configured Destination (ECS-ECSTC-308) 459
11.16.4 [OBSOLETE] Test Case 3 - FtpPush Request for an Un-Configured Destination (ECS-ECSTC-309)..... 459
11.16.5 Test Case 6 - Suspend, Resume / Cancel, Retry (ECS-ECSTC-312)..... 460
11.16.6 Test Case 9 - Ftp Push/SCP Destination Manual Retry Suspension (ECS-ECSTC-315) 461
11.16.7 [PVC] Test Case 15 - Checksum during Data Distribution : CK_7F_01, Criterion 30 (ECS-ECSTC-321)..... 462
11.16.8 OMS GUI Config Parameters (ECS-ECSTC-444)..... 463
11.16.9 Ordering newly published replacement granules (ECS-ECSTC-496) 464
11.16.10 Resubmit request with failed granule(s) (ECS-ECSTC-562) 464
11.16.11 Partitioning (ECS-ECSTC-636)..... 465
11.16.12 Request Priority (ECS-ECSTC-637) 466
11.16.13 OMS GUI: Request Management (ECS-ECSTC-638)..... 466
11.16.14 OMS Startup Performance (ECS-ECSTC-657)..... 468
11.17 Test Case 23 - OMS GUI Distribution Request Filter (ECS-ECSTC-3884) 471

12 SMAP 471
12.1 SMAP Modified 471
12.1.1 Ingest XPath Validation (ECS-ECSTC-364) 471
12.1.2 Install ISO ESDT (ECS-ECSTC-363) 474
12.1.3 ISO Ingest Attribute Type Conversion (ECS-ECSTC-365) 478
12.1.4 ODL metadata files for ECS Distribution (ECS-ECSTC-366) 482
12.1.5 XML metadata files for ISO Distribution (ECS-ECSTC-367) 485
12.1.6 QA Update Failure Verification (ECS-ECSTC-368) 487

13 SSS 491
13.1 Test Case 1 - Enter Subscriptions via SSS GUI (ECS-ECSTC-327) 491
13.2 Test Case 3 - Unqualified Subscription with Email/FtpPush (ECS-ECSTC-329) 491
13.3 Test Case 5 - Qualified Subscription for Data Pool Insert (ECS-ECSTC-331) 492
13.4 Test Case 6 - Qualifying Metadata in Email Notification (ECS-ECSTC-332) 493
13.5 Test Case 7 - All Metadata in Email Notification (ECS-ECSTC-333) 493
13.6 Test Case 8 - Spatially Qualified Subscriptions with Spatial Search Type = GPolygon : NS_SY_01, Criterion 20 (ECS-ECSTC-334) 494
13.7 Test Case 9 - Spatially Qualified Subscriptions with Spatial Search Type = Bounding Rectangle : NS_SY_01 criterion 30 (ECS-ECSTC-335) 495
13.8 Test Case 10 - Spatially Qualified Subscriptions with Spatial Search Type = Orbit : NS_SY_01, criterion 40 (ECS-ECSTC-336) 496
13.9 Test Case 11 - Combinations of Qualifiers : NS_SY_01, criterion 90 (ECS-ECSTC-337) 498
13.10 Test Case 12 - Placing Bundling Order Subscription : OD_S3_03, criterion 10 (ECS-ECSTC-338) 499
13.11 Test Case 13 - Executing Bundling Order Subscription : OD_S3_03, criterion 110, steps a, b, c, i, and m (ECS-ECSTC-339) 499
13.12 Test Case 14 - Subscription Expiration : NS_SY_01 criterion 210 (ECS-ECSTC-340) 501
13.13 Test Case 15 - XDAAC (ECS-ECSTC-341) 501
13.14 SSS Modified 503
13.14.1 Test Case 2 - Unqualified Subscription with Email (ECS-ECSTC-328) 503
13.14.2 Test Case 4 - Unqualified Subscription with Email/FtpPull (ECS-ECSTC-330) . 503
13.14.3 Validate SSS GUI (ECS-ECSTC-446) 504
13.14.4 Deleting a failed subscription in SSS GUI (ECS-ECSTC-447) 505

14 DUPLICATE GRANULES 506
14.1 Duplicate Granule Detection and Replacement - Identical LocalGranuleIDs (ECS-ECSTC-474) 506

15 DBDM 510
15.1 pgcopy (ECS-ECSTC-500) 510

16 DPL 511
16.1 Test Case 3 - Publish with Theme (ECS-ECSTC-204) 511
16.2 Test Case 4 - [Data Pool Maintenance GUI – Check Batch Insert (ECS-ECSTC-3883)512
16.3 [OBSOLETE] - Test Case 8 - Update Granule Utility OBSOLETE ("4.8.5 Update Granule Deleted. Not applicable for Release 7.23.") (ECS-ECSTC-206) 512

16.4 [OBSOLETE - PVC] - Test Case 11 - Firewall FTP Access Statistics Utility (ECS-ECSTC-207) 513
16.5 [PARTIAL OBSOLETE] -Test Case 13 - Inventory Validation (Orphan, Phantom and Link Checking) - [PVC] (ECS-ECSTC-208) 514
16.6 Test Case 14 - Repair â€“ DP_7F_01, Criterion 300 partially (ECS-ECSTC-209) 520
16.7 [OBSOLETE] - Test Case 15 - Most Recent Inserts Utility (ECS-ECSTC-210)..... 521
16.8 [OBSOLETE] - Test Case 16 - Remap Collection to New Group, DP_7F_01, Criterion 2020 (ECS-ECSTC-211) 522
16.9 Test Case 17 - GUI Security (ECS-ECSTC-212) 524
16.10 [PVC] - Test Case 18 - S4 order (ECS-ECSTC-213)..... 524
16.11 Test Case 20 - Hidden Scrambler Utility (ECS-ECSTC-214) 525
16.12 Test Case 21 - Create Collections on DPL Maintenance GUI (ECS-ECSTC-215) 525
16.13 [OBSOLETE] - Test Case 22 - Detection of Illegal Access to Hidden Directories: OD_S5_06, Criterion 90 (ECS-ECSTC-216) 526
16.14 Test Case 23 - DPL Maintenance GUI (ECS-ECSTC-217)..... 527
16.15 Test Case 24 - Move Collection Utility (ECS-ECSTC-218)..... 528
16.16 Test Case 26 - Publishing Hidden Science Granule with Browse (ECS-ECSTC-219) 529
16.17 Test Case 27 - Backfill : DP_72_01, Criterion 170 (ECS-ECSTC-220) 529
16.18 Test Case 28 - Granule Replacement ON, Collection Configured For Public Data Pool Insert : DP_72_01, Criterion 180 (ECS-ECSTC-221)..... 530
16.19 Test Case 29 - Granule Replacement ON, Older Granule Does not Replace Newer Granule (ECS-ECSTC-222)..... 531
16.20 Test Case 30 - Granule Replacement OFF, Collection Configured For Public Data Pool Insert : DP_72_01, Criterion 185 with modification (ECS-ECSTC-223) 532
16.21 Test Case 32 - Publish After Validation, DP_7F_01, Criterion 130 (ECS-ECSTC-225) 533
16.22 [OBSOLETE] - Test Case 33 - On-Line Archive Bulk Repair, DP_7F_01, Criterion 330 (ECS-ECSTC-226) 534
16.23 Test Case 34 - Un-publish Granules that were Reported by Data Pool Inventory Validation, DP_7F_01, Criterion 90 (ECS-ECSTC-227)..... 535
16.24 [OBSOLETE] - Test Case 35 - Non-ECS insert: DP_S3_02, Criterion 110 (ECS-ECSTC-228) 536
16.25 [PVC] - Test Case 36 - Run DPCV against DPL, Provide a List of Granule IDs : CK_7F_01, Criterion 540 (ECS-ECSTC-229) 537
16.26 Test Case 37 -Run DPCV for a Single ESDT, with Granule Insert Date Range with Checksum Failures: CK_7F_01, Criterion 510 (ECS-ECSTC-230) 539
16.27 [PARTIAL OBSOLETE] - Test Case 39 -QA/PH DPL Ingest, Insert:DP_7G_01, Criterion 100 - [PVC - Web Access portion is Obsolete] (ECS-ECSTC-231)..... 540
16.28 Test Case 40 - QA/PH/Browse DPL Replacement (ECS-ECSTC-232) 541
16.29 Test Case 41 - QA/PH publishing and unpublishing (ECS-ECSTC-233) 547
16.30 [OBSOLETE] - Test Case 42 - On-Line Archive Repair including QA and PH links (ECS-ECSTC-234) 548
16.31 [PVC] - Test Case 43 - Move AMSR Collection (ECS-ECSTC-235)..... 550
16.32 [PVC] - Test Case 46 - Link Checking Restart After Fault, DP_7F_01, Criterion 1022 (ECS-ECSTC-236) 551

16.33 Test Case 47 - Fault Recovery â€“ Collection to Group Remapping utility (ECS-ECSTC-237) 551
16.34 [OBSOLETE] - Test Case 48 - Fault Recovery â€“ Most Recent Inserts Utility (ECS-ECSTC-238) 552
16.35 Test Case 49 - Fault Recovery â€“ Hidden Scrambler Utility (ECS-ECSTC-239) 553
16.36 [PVC] - Test Case 50 - Fault Recovery - Data Pool Un-publishing â€“ Restart after Fault, DP_7F_01, Criterion 1032 (ECS-ECSTC-240) 554
16.37 [PVC] - Test Case 51 - Fault Recovery - Bulk On-Line Archive Repair â€“ Restart after Fault, DP_7F_01 Criterion 1052 (ECS-ECSTC-241) 555
16.38 Test Case 52 - Fault Recovery - Data Pool Move Collection Utility when the Move of the Collection is Interrupted and then Restarted (ECS-ECSTC-242) 556
16.39 Test Case 53 - DPL XML Check Utility (ECS-ECSTC-243) 557
16.40 DP_81_01_TP055 DPL Cleanup with input file (ECS-ECSTC-342) 558
16.41 DP_81_01_TP056 DPL Cleanup on non-ECS granules (ECS-ECSTC-343) 560
16.42 DPL Modified 563
16.42.1 Data Pool Maintenance GUI - Check Batch Insert Status (ECS-ECSTC-205) 563
16.42.2 Publish Utility, DP_7F_01, Criterion 500 (ECS-ECSTC-224) 563
16.42.3 DataPool Maintenance GUI - QualitySummary URL should allow secure URL (ECS-ECSTC-495) 565
16.42.4 Inventory Validation(Orphan, Phantom and Link Checking) (CleanupFilesOnDisk(2)) (NCR 8051022, 8052031, 8052056) (ECS-ECSTC-663) 565
16.43 [OBSOLETE] - Test Case 1 - Qualified Subscription Insert (ECS-ECSTC-905) 567
16.44 [OBSOLETE] - Test Case 6 - Download Granules via Web (ECS-ECSTC-906) 568
16.45 [OBSOLETE] - Test Case 19 - HEG Order from Web Access (ECS-ECSTC-915) ... 569

17 LONG SHORTNAME AND POSTGRES 9.3 (ALL UTILITIES) 569
17.1 ESDT Installation (ECS-ECSTC-541) 569
17.2 Ingest a Granule (ECS-ECSTC-530) 570
17.3 BMGT Nominal Exports (ECS-ECSTC-545) 571
17.4 Order a granule (ECS-ECSTC-531) 572
17.5 Unpublish a granule (ECS-ECSTC-533) 572
17.6 Publish a granule (ECS-ECSTC-534) 573
17.7 Duplicate Granule (ECS-ECSTC-542) 574
17.8 RestoreOLAFromTape (ECS-ECSTC-535) 575
17.9 RestoreTapeFromOLA (ECS-ECSTC-536) 575
17.10 DPCV (ECS-ECSTC-537) 576
17.11 ACVU (ECS-ECSTC-539) 576
17.12 Data Access Nominal Test (ECS-ECSTC-540) 577
17.13 Move Collection (ECS-ECSTC-546) 578
17.14 Remap Collection Group (ECS-ECSTC-547) 579
17.15 QAUU (ECS-ECSTC-548) 579
17.16 XCU (ECS-ECSTC-549) 580
17.17 XRU (ECS-ECSTC-550) 580
17.18 EMS long shortname End to End test (ECS-ECSTC-551) 581
17.19 Map Granule Generation (ECS-ECSTC-552) 582

17.20 Cleanup Files on Disk (ECS-ECSTC-553) 583
17.21 Cleanup Granules (ECS-ECSTC-554) 583
17.22 Inventory Validation Tool (ECS-ECSTC-557) 584
17.23 Datapool Hidden Scrambler Utility (ECS-ECSTC-559) 584
17.24 Search and Delete Utilities (ECS-ECSTC-560) 585
17.25 Delete Collection (ECS-ECSTC-561) 586

18 BIG GRANULES 587
18.1 Ingest a Large Granule (ECS-ECSTC-347) 587
18.2 Order a large granule (ECS-ECSTC-349) 587
18.3 RestoreOLAFromTape (ECS-ECSTC-350) 589
18.4 RestoreTapeFromOLA (ECS-ECSTC-351) 590
18.5 Unpublish a large granule (ECS-ECSTC-352) 590
18.6 Publish a large granule (ECS-ECSTC-353) 591
18.7 Use DPCV to checksum a large granule (ECS-ECSTC-354) 592
18.8 Use CVS to checksum a large granule (ECS-ECSTC-355) 593
18.9 Use ACVU to checksum a large granule on tape (ECS-ECSTC-356) 593
18.10 Download a large granule using wu-ftp (with alternative compression methods) (ECS-ECSTC-357) 594
18.11 Download a large granule using apache (with alternative compression methods) (ECS-ECSTC-358) 595
18.12 Subset a large granule using ESI (ECS-ECSTC-359) 596

19 PUBLISH ALL BROWSE 597
19.1 DPL Cleanup 597
19.1.1 Public Browse and MODIS unpublish (ECS-ECSTC-369) 597
19.1.2 Public Browse and Unpublish of deleted MODIS Browse (ECS-ECSTC-370) 598
19.1.3 Public Browse and publishing of undeleted MODIS Science and Browse (ECS-ECSTC-371) 598
19.1.4 Public Browse and Browse only publishing of undeleted MODIS Science and Browse (ECS-ECSTC-372) 599
19.1.5 Public Browse and DataPool Cleanup orphan and phantom checking (ECS-ECSTC-373) 600
19.1.6 Public Browse and Data Pool recovery from publication errors (ECS-ECSTC-374) 601
19.2 IVT 602
19.2.1 Public Browse and hidden Science with public Browse (ECS-ECSTC-375) 602
19.3 Ingest 603
19.3.1 Public Browse and ASTER ingest (ECS-ECSTC-344) 603
19.3.2 Public Browse and MISRBR ingest (ECS-ECSTC-345) 604
19.3.3 Public Browse and MODIS ingest (ECS-ECSTC-346) 605
19.3.4 Public Browse and AMSR ingest (ECS-ECSTC-360) 605
19.3.5 Public Browse and AMSR Browse replacement (ECS-ECSTC-361) 607
19.3.6 Public Browse and MODIS browse replacement (ECS-ECSTC-362) 607

20 HTTPS TESTING 608
20.1 Submit egi/esi requests for ICESat-2 data (ECS-ECSTC-3858) 608
20.2 Submit egi/esi requests for FRI Service (ECS-ECSTC-3859) 609

20.3 Submit egi/esi requests for MOFIS Service (ECS-ECSTC-3860) 610
20.4 Submit egi/esi requests for HEG Service (ECS-ECSTC-3861) 610
20.5 Submit egi/esi requests for GDAL Service (ECS-ECSTC-3862)..... 611
20.6 Submit order request with no processing (ECS-ECSTC-3863)..... 612
20.7 Perform a Direct Download via the Earthdata Search Client (ECS-ECSTC-3864) 613
20.8 Cancelling requests in processing in Data Access via the GUI (ECS-ECSTC-3865).... 613
20.9 Uploading Service and Order Forms (ECS-ECSTC-3870) 614
20.10 Configure Dataset-Service Data Objects for a Collection (ECS-ECSTC-3871)..... 614
20.11 Submit egi/esi requests for SMAPL1L2 Service (ECS-ECSTC-3872) 615
20.12 Submit egi/esi requests for GLAS Service (ECS-ECSTC-3874)..... 616
20.13 Submit egi/esi requests for GEDI data (ECS-ECSTC-3885)..... 617

List of Figures

No table of figures entries found.

List of Tables

No table of figures entries found.

1 AIM

1.1 Test Case 1 - Install ESDTs with a variety of spatial search types (ECS-ECSTC-163)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Install ESDTs with a variety of spatial search types]		#comment
2	Verify that the descriptor ODL files for the ESDT being installed are moved from the installation source directory into the configured descriptor directory.		
3	Prepare descriptor files for several (at least 5) ESDTs covering a variety of ESDT types described below, such that the descriptors conform to the ECS data model and the XML schema validation rules, guaranteeing successful ESDT installation. Examples of such can be found in /sotestdata/DROP_721/DS_7E_01/Criteria/2030, and are listed as follows: 1. ESDT descriptor with a spatial search attribute of Rectangle MOD09CMG.005 MYD09CMG.005 2. ESDT descriptor with a spatial search attribute of GPolygon MOD14.005 MYD14.005 3. ESDT descriptor with a spatial search attribute of NotSupported AEPOE7W.001 PM1GBAD1.001 4. ESDT descriptor with a spatial search attribute of Orbit AE_Land.002 AE_Rain.002 5. ESDT descriptor with a spatial search attribute of Point g3bssp.007 g3bt.007 		
4	Ensure collections are not already installed.		
5	Ensure EcDsAmESDTMaint.properties has log.debug.level=XVERBOSE (for verification).		
6	Copy the descriptor files to the source directory (specified in the GUI).		
7	Verify that the descriptors are listed in the GUI.		
8	Get the current time to help search the logs.		
9	From the ESDT Maintenance GUI, select the ESDTs in the source directory and initiate installation.		
10	Upon completion of the ESDT installation, verify that the GUI displays a message indicating number of ESDTs successfully installed.		
11	Verify that the metadata elements provided in the descriptor files were		

#	Action	Expected Result	Notes
	validated by the XML Services library against the ESDR common schema, which conforms to the ECS data model. If you have the ESDT Maintenance GUI debug level set to XVERBOSE, you should get a message something like the following in the ESDT Maintenance GUI log: EcsFileProcessor.validateXmlFile DsESDTMoMOD09A1.005.xml Other than that, it might require looking at the actual code.		
12	Verify by inspection that the element types and lengths in the descriptors match those stated in the ESDT common schema which in turn matches the ECS data model. The ESDT common schema is under: /usr/ecs/MODE/CUSTOM/WWW/DSS/ESDTMaint/WEB-INF/lib/EcDsAmEsdtCommonSchemas.jar. 		Are the descriptors supposed to be compared to XML schema? Should verify that the descriptors in the configured descriptors directory diff with the originals. Jon Pals: " For Step 5, Yes, it is asking that you compare what is in the ODL ESDT descriptor file against the XML schema files. For this, I suggest doing some random sampling. Just take a few attributes from the ESDT descriptor file and find how those attributes are defined in the XML schema files." Need an automated way to compare ODL descriptor files to XML schemas.
13	Verify that an MCF file is generated for each ESDT being installed, and stored into the configured MCF directory location (locations pulled from /custom/ecs/TS3/CUSTOM/cfg/EcDsAmESDTMaint.properties) descriptor.target.dir = /stornext/smallfiles/TS3/descriptor descriptor.source.dir = /usr/ecs/TS3/CUSTOM/data/ESS mcf.target.dir = /stornext/smallfiles/TS3/mcf archive.metadata.dir = /stornext/smallfiles/TS3/metadata 		20130731t124819::f5dpl01v::cmshared::/stornext/smallfiles/TS3/mcf $ find . -type f -mtime -1 | sed 's/..\([^#]*\)#\([^.]*\).*/\1.\2/' MOD09CMG.005 MOD14.005 MYD14.005 MYD09CMG.005 AE_Land.002<br

#	Action	Expected Result	Notes
			/>AE_Rain.002 g3bssp.007 PM1GBAD1.001 g3bt.007 AEPOE7W.001
14	Verify that the generated MCF files are correct. 		(Define "correct": The MCFs should be a pretty close match with the INVENTORYMETADATA section and the ARCHIVEDMETADATA section in the ESDT descriptor file. The place where there should be a difference in in the AdditionalAttributes group. The ProductSpecificMetadata group in the ESDT descriptor file is replaced with a generic AdditionalAttributes group in the MCF.) Within the INVENTORYMETADATA group, the ProductSpecificMetadata group from the descriptor file should be replaced with this AdditionalAttributes block in the MCF file: GROUP = AdditionalAttributes OBJECT = AdditionalAttributesContainer Data_Location = "NONE" Mandatory =

#	Action	Expected Result	Notes
			"FALSE" CLASS = "M" OBJECT = AdditionalAttributeName Mandatory = "FALSE" CLASS = "M" Data_Location = "PGE" NUM_VAL = 1 TYPE = "STRING" END_OBJECT = AdditionalAttributeName GROUP = InformationContent CLASS = "M" OBJECT = ParameterValue Mandatory = "FALSE" Data_Location = "PGE" NUM_VAL = 1 TYPE = "STRING" END_OBJECT = ParameterValue END_GROUP = InformationContent END_OBJECT = AdditionalAttributesContainer END_GROUP = AdditionalAttributes MODIS MCF files should include the descriptor file's

#	Action	Expected Result	Notes
			ARCHIVEDMETADATA group. AE_Land#002.MCF, AE_Rain#002.MCF, and PM1GBAD1#001.MCF do not have this group
15	Verify that an XML schema file (*.xsd) is generated for each ESDT being installed, and stored into the descriptors directory.		
16	Verify that the generated XML schema files contain validation rules based upon the contents of the Inventory section of the Descriptor file and the default element rules. 		The "Inventory section" probably starts here: GROUP = INVENTORYMETADATA Jon Pals: For Step 9, take a look at the 'Mandatory =' lines in the INVENTORYMETADATA section of the ESDT descriptor file and compare that with the ESDT's .xsd file in the /stornext/smallfiles/<MODE>/descriptor directory. The INVENTORYMETADATA attributes with 'Mandatory = "FALSE"' lines should be listed in the .xsd file as having 'minOccurs="0"' . The INVENTORYMETADATA attributes with 'Mandatory = "TRUE"' lines should be listed in the .xsd file as having no 'minOccurs' or having a 'minOccurs="1"' .

#	Action	Expected Result	Notes
17	Verify that basic collection identification information and proper spatial search type have been added into the AIM Inventory database. 		
18	Verify that the PSA associations, if any, have been added into the AIM Inventory database. 		select aa.additionalattributename from amcollection c join dsmdcollectionaddnlattribsxref ax on c.collectionid = ax.collectionid join dsmdadditionalattributes aa on ax.attributeid = aa.attributeid where c.shortname = '${SHORTNAME}' and c.versionid = ${VERSIONID};
19	Verify that an insert event, and event qualifiers are inserted into the Spatial Subscription Server database for each ESDT being installed. 		Event qualifiers appear restricted to those defined in ecnbeventmetadataattrdef.attributename. EVENTPARMS: Extracted from descriptor file Online EVENTPARMS known by SSS will be added. -- What qualifiers does SSS know about? select * from ecnbeventmetadataattrdef where attributename in (${EVENTPARMS}); -- What qualifiers were added? select attributename from ecnbeventattrxref where esdt_id = '${SHORTNAME}' and versionid = $VERSIONID;<br

#	Action	Expected Result	Notes
			/>'RangeBeginningTime' and 'RangeEndingTime' appear to get combined into 'GranuleTimeRange'. 'RangeBeginningDate' and 'RangeEndingDate' appear to get combined into 'GranuleDateRange'. g3bssp.007 and g3bt.007 both lack date and time ranges, but both have GranuleTimeRange and GranuleDateRange in ecnbeventattrxref. example: select * from ecnbeventdefinition where (esdt_id = 'MOD14' and versionid = 5) or (esdt_id = 'MYD14' and versionid = 5) or (esdt_id = 'AEPOE7W' and versionid = 1) or (esdt_id = 'PM1GBAD1' and versionid = 1) or (esdt_id = 'AE_Land' and versionid = 2) or (esdt_id = 'AE_Rain' and versionid = 2) or (esdt_id = 'MOD09CMG' and versionid = 5) or (esdt_id = 'MYD09CMG' and versionid = 5) or (esdt_id = 'g3bssp' and versionid = 7) or (esdt_id = 'g3bt' and versionid = 7); eventtype | esdt_id |

#	Action	Expected Result	Notes
			versionid | eventid ----------+----------+----------+--------- INSERT | AE_Land | 2 | INSERT | AE_Rain | 2 | INSERT | AEPOE7W | 1 | INSERT | MOD09CMG| 5 | INSERT | MOD14 | 5 | INSERT | MYD09CMG | 5 | INSERT | MYD14 | 5 | INSERT | PM1GBAD1 | 1 | INSERT | g3bssp | 7 | INSERT | g3bt | 7 | (10 rows)
20	Verify that the ESDT Maintenance GUI logs the processing activities including time of request, action requested, ESDT ShortName, Version ID, descriptor file name, and the result of the operation, in a configured log file.		The ESDTMaintenanceGUI.* logs include a timestamp in each log entry, but it's not clear when a request actually began. These lines from the debug log may be the beginning of a request: 07.31.2013 12:02:09.563 : Thread ID [41] : VERBOSE : /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTAmAE_Land.002.desc dataModelType is: ECS 07.31.2013 12:02:09.563 : Thread ID [41] : VERBOSE : installing

#	Action	Expected Result	Notes
			ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTAmAE_Land.002.desc The action is not stated explicitly. In the debug log, one entry includes the word "installing": 07.31.2013 12:02:11.821 : Thread ID [41] : VERBOSE : installing ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc The ShortName and Version ID are part of the descriptor filename. They appear in several other log entries, but it's not clear why: 07.31.2013 12:02:09.651 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:10 String Value:AE_Land.002 Type:12 07.31.2013 12:02:09.702 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:1 String Value:AE_Land Type:1 07.31.2013 12:02:09.702 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:2 String Value:2

#	Action	Expected Result	Notes
			Type:4 The debug log reports the successful installation like this: 07.31.2013 12:02:10.358 : Thread ID [41] : INFORMATION : InstallESDTPage:Successfully installed ESDT DsESDTAmAE_Land.002.desc The ESDTMaintenanceGUI.ops0.log contains very little information. Example: From /usr/ecs/TS3/CUSTOM/logs/ESDTMaintenanceGUI.debug0.log: time of the installation request: 07.31.2013 12:02:11.821 : Thread ID [41] : VERBOSE : installing ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc action requested: (this shows the action taken) 07.31.2013 12:02:11.821 : Thread ID [41] : VERBOSE : installing ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc ShortName: 07.31.2013 12:02:11.862 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:1 String

#	Action	Expected Result	Notes
			Value:MOD09CMG Type:1 Version ID: 07.31.2013 12:02:11.862 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:4 String Value:5 Type:4 descriptor file name: 07.31.2013 12:02:11.821 : Thread ID [41] : VERBOSE : installing ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc result of the operation (successful): 07.31.2013 12:02:12.215 : Thread ID [41] : INFORMATION : InstallESDTPage:Successfully installed ESDT DsESDTMoMOD09CMG.005.desc
21	Verification Instructions		#comment
22	Verify that there are at least 5 descriptors that match the setup requirements in directory /sotestdata/DROP_721/DS_7E_01/Criteria/2030/. Verify that the value of the OBJECT = SpatialSearchType group is “rectangle”, “GPolygon”, “not supported”, “point”, or “orbit” accordingly: | ESDT | SpatialSearchType | Archiving DAAC | | ACR3L2SC.001 | NotSupported | ASDC | | AEPOE7W.001 | NotSupported | NSIDC | | AE_Land.002 | Orbit | NSIDC | | AE_Rain.002 | Orbit | NSIDC | | MB2LME.198 | Orbit | ASDC | | MIANRCCH.198 | NotSupported | ASDC | | MIL2ASOS.198 | Orbit | ASDC | | MIL3DAE.198 | Rectangle | ASDC | | MOD09CMG.005 | Rectangle | LP DAAC | |		

#	Action	Expected Result	Notes
	MOD14.005 | GPolygon | LP DAAC | | MOP02.003 | Rectangle | ASDC | | MYD09CMG.005 | Rectangle | LP_DAAC | | MYD14.005 | GPolygon | LP DAAC | | PM1GBAD1.001 | NotSupported | NSIDC | | TL3ATD.002 | Rectangle | ASDC | | g3assp.004 | Point | ASDC | | g3atb.004 | Point | ASDC |		
23	Return to the ESDT List Page. Select the check box next to the descriptor names set up in setup step 1. Select the “Install new ESDTs/Update existing ESDTs” button. Note the wall clock time of the start of the installation for later verification in the logs. Note the number of descriptors selected.		Login to the ESDT Maintenance GUI. Ensure no test ESDT is currently installed. If it is, delete its granules, clean up orphans, remove from the data pool (DPL GUI), and delete it (ESDT GUI). Click the "Install new ESDTs/Update existing ESDTs" button. Note the ESDT source directory. Copy the test ESDTs to the ESDT source directory. Login to the ESDT Maintenance GUI. Click the "Install new ESDTs/Update existing ESDTs" button. Select the test ESDTs. Click the "Proceed with installation/update" button. Wait for the screen to refresh.
24	Verify that the GUI displays the number of descriptors installed is equal to the number of descriptors selected.		
25	Verify that the metadata elements provided in the descriptor files were validated by the XML validation utility by viewing its application log in the mode. Verify for each descriptor that no failures occurred. 		

#	Action	Expected Result	Notes
26	Select one descriptor from each part of setup step 1. 		
27	Go to the directory where the generic descriptor schema is installed and view it.		
28	Verify by inspection that rules for elements in the generic schema are executed correctly for corresponding elements in the descriptor. A similar comparison should be performed between the elements of the generic descriptor schema and the ECS data model. 		What is the XML validation utility used, and where is its log? Where is "the directory where the generic descriptor schema is installed"?
29	Verify by inspection that rules for elements in the generic schema are executed correctly for corresponding elements in the descriptor. 		
30	Verify each element in the descriptor file that has a matching element in the generic schema gets added to the descriptor's .xsd file. 		"A similar comparison should be performed between the elements of the generic descriptor schema and the ECS data model." What does this mean? /usr/ecs/TS3/CUSTOM/logs/ESDTMaintenanceGUI.debug0.log shows the descriptor .xml and .xsd files are generated but not whether they are validated: 07.31.2013 12:02:12.021 : Thread ID [41] : xmlsvcs.schemagen.EcAmDescSchemaGenImpl : VERBOSE : START generateDescriptorSchemaFromOdlDescriptor odlDescriptor: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc outputSchema:

#	Action	Expected Result	Notes
			/stornext/smallfiles/TS3/descriptor/DsESDTMoMOD09CMG.005.xsd 07.31.2013 12:02:12.021 : Thread ID [41] : xmlsvcs.schemagen.EcAmDescSchemaGenImpl : VERBOSE : START generateDescriptorXmlFile inputODLFile: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc outputXMLFile: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc.xml 07.31.2013 12:02:12.050 : Thread ID [41] : xmlsvcs.schemagen.EcAmDescSchemaGenImpl : VERBOSE : END generateDescriptorXmlFile inputODLFile: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc outputXMLFile: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc.xml 07.31.2013 12:02:12.051 : Thread ID [41] : xmlsvcs.schemagen.EcAmDescSchemaGenImpl : VERBOSE : START generateDescriptorSchemaFromXmlDescriptor xmlDescriptor:

#	Action	Expected Result	Notes
			/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc.xml outputSchema: /stornext/smallfiles/TS3/descriptor/DsESDTMoMOD09CMG.005.xsd 07.31.2013 12:02:12.051 : Thread ID [41] : xmlsvcs.schemagen.EcAmSchemaGenerator : INFORMATION : createESDTSchema START: XML descriptor - /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc.xml, Schema - /stornext/smallfiles/TS3/descriptor/DsESDTMoMOD09CMG.005.xsd 07.31.2013 12:02:12.154 : Thread ID [41] : xmlsvcs.schemagen.EcAmSchemaGenerator : INFORMATION : createESDTSchema DONE: XML descriptor - /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc.xml, Schema - /stornext/smallfiles/TS3/descriptor/DsESDTMoMOD09CMG.005.xsd 07.31.2013 12:02:12.155 : Thread ID [41] : xmlsvcs.schemagen.EcAmDescSchemaGenImpl : VERBOSE : END generateDescriptorSchemaF

#	Action	Expected Result	Notes
			romXmlDescriptor xmlDescriptor: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc.xml outputSchema: /stornext/smallfiles/TS3/descriptor/DsESDTMoMOD09CMG.005.xsd 07.31.2013 12:02:12.155 : Thread ID [41] : xmlsvcs.schemagen.EcAmDescSchemaGenImpl : VERBOSE : END generateDescriptorSchemaFromOdlDescriptor odlDescriptor: /usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc outputSchema: /stornext/smallfiles/TS3/descriptor/DsESDTMoMOD09CMG.005.xsd
31	Find the mcf target directory (parameter mcf.target.dir) in the EcAmMaintenanceGui.properties file. Change the directory to the ESDT specific subdirectory (probably /stornext/smallfiles/<mode>/mcf/). Perform an ls to verify that the mcf file is found. Repeat for each MCF file generated from the ESDT installation.		/custom/ecs/${MODE}/CUSTOM/cfg/EcDsAmESDTMaint.properties mcf.target.dir=/stornext/smallfiles/${MODE}/mcf
32	Compare the descriptor file to the mcf file to verify that the MCF is correct. 		
33	Change the directory to this directory (probably /stornext/smallfiles/<mode>/descriptor/).		
34	Perform an ls to verify that the schema file is found. 		/custom/ecs/${MODE}/CUSTOM/cfg/EcDsAmESDTMaint.properties descriptor.target.dir=/storn

#	Action	Expected Result	Notes
			ext/smallfiles/${MODE}/descriptor
35	Make a one to one comparison of the groups in the Inventory Section of the Descriptor file with the corresponding groups in the schema.		
36	Verify that the values of the descriptor are valid according to the rules in the schema.		
37	Find the descriptor source and target directory (parameters descriptor.source.dir and descriptor.target.dir) in the EcAmMaintenanceGui.properties file. 		
38	Change to target directory (probably /stornext/smallfiles/<mode>/descriptor/).		
39	Perform an ls to verify that the descriptor file DsESDT<ShortName>.<VersionId>.desc is found. Repeat for each part descriptor installed in the setup step 1.		
40	Verify that the descriptor file was deleted in the source directory in the mode.		
41	Change to directory descriptor.source.dir.		
42	Verify that the descriptor was deleted by using the ls command and finding no files matching the descriptor's name. Also, repeat for each descriptor installed.		/custom/ecs/${MODE}/CUSTOM/cfg/EcDsAmESDTMaint.properties descriptor.source.dir=/usr/ecs/${MODE}/CUSTOM/data/ESS descriptor.target.dir=/stornext/smallfiles/${MODE}/descriptor This should be at least 2 steps.
43	In the AIM Inventory database, verify that the ShortName, VersionID, and insertTime are populated in the DsMdCollections table for each ESDT installed. select ShortName, VersionID, insertTime from DsMdCollections where ShortName = "<ShortName>" and VersionID= "<VersionID>" 		
44	Verify that the spatial searchSearchType matches the one specified in the descriptor. select configuredName, VersionID, spatialSearchType from DsMdESDTConfiguredType where configuredName = "<shortName>" and VersionID = "VersionID" 		
45	Repeat for each installed ESDT.		select c.inserttime,

#	Action	Expected Result	Notes
			esdt(c.shortname,c.versionid), c.collectionid, c.spatialsearchtype, e.spatialsearchtype from amcollection c join dsgeesdtconfiguredtype e on c.shortname = e.configuredname and c.versionid = e.versionid where (c.shortname = '${SHORTNAME}' and c.versionid = ${VERSIONID}) or (c.shortname = '${SHORTNAME}' and c.versionid = ${VERSIONID}) ... order by c.spatialsearchtype, c.shortname, c.versionid; dsgeesdtconfiguredtype replaces DsMdESDTConfiguredType. select c.inserttime, esdt(c.shortname,c.versionid), c.collectionid, c.spatialsearchtype, e.spatialsearchtype from amcollection c join dsgeesdtconfiguredtype e on c.shortname = e.configuredname and c.versionid = e.versionid where (c.shortname = 'MOD14' and c.versionid = 5) or (c.shortname = 'MYD14' and c.versionid =

#	Action	Expected Result	Notes
			5) or (c.shortname = 'AEPOE7W' and c.versionid = 1) or (c.shortname = 'PM1GBAD1' and c.versionid = 1) or (c.shortname = 'AE_Land' and c.versionid = 2) or (c.shortname = 'AE_Rain' and c.versionid = 2) or (c.shortname = 'MOD09CMG' and c.versionid = 5) or (c.shortname = 'MYD09CMG' and c.versionid = 5) or (c.shortname = 'g3bssp' and c.versionid = 7) or (c.shortname = 'g3bt' and c.versionid = 7) order by c.spatialsearchtype, c.shortname, c.versionid; inserttime	esdt	collectionid	spatialsearchtype	spatialsearchtype ----------+----------+----------+----------+----------+---------- 2013-07-31 12:02:12.560421	MOD14.005	198330	GPolygon	GPolygon 2013-07-31 12:02:13.716981	MYD14.005	198332	GPolygon	GPolygon 2013-07-31 12:02:11.403874	

#	Action	Expected Result	Notes
			AEPOE7W.001 | 198328 | NotSupported | NotSupported 2013-07-31 12:02:14.194661 | PM1GBAD1.001 | 198333 | NotSupported | NotSupported 2013-07-31 12:02:09.656776 | AE_Land.002 | 198326 | Orbit | Orbit 2013-07-31 12:02:10.765544 | AE_Rain.002 | 198327 | Orbit | Orbit 2013-07-31 12:02:14.865666 | g3bssp.007 | 198334 | Point | Point 2013-07-31 12:02:15.367958 | g3bt.007 | 198335 | Point | Point 2013-07-31 12:02:11.866531 | MOD09CMG.005 | 198329 | Rectangle | Rectangle 2013-07-31 12:02:13.12891 | MYD09CMG.005 | 198331 | Rectangle | Rectangle (10 rows)
46	In another terminal log into the Spatial Subscription Server (SSS) database. View the <ShortName>.<VersionId> descriptor in the mode and find the INSERT object under the EVENT group in the descriptor. Under the EVENTPARMS object are a list of parameters which should be populated in the SSS database.		
47	Verify that the event was inserted into the EcNbEventDefinition table. select * from EcNbEventDefinition where ESDT_Id in (<a quoted list of ShortNames of installed ESDTs>) 		

#	Action	Expected Result	Notes
48	Verify that each parameter EVENTPARM parameter is populated in the SSS database . select * from EcNbEventAttrXref where ESDT_Id = “ShortName” 		
49	Repeat this step for each descriptor installed		
50	Change directory to the log.dir directory of the mode.		
51	View the ESDT_Maint.log and search for the first ESDT installed.		
52	Verify the time of the installation request, the ShortName, Version ID, descriptor file name and the result of the operation (successful) is listed in the log file.		
53	Repeat for each ESDT installed.		time of the installation request: 07.31.2013 12:02:11.821 : Thread ID [41] : VERBOSE : installing ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoMOD09CMG.005.desc ShortName: 07.31.2013 12:02:11.862 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:1 String Value:MOD09CMG Type:1 Version ID: 07.31.2013 12:02:11.862 : Thread ID [41] : VERBOSE : AbstractServiceImpl.setObject4CallableStmt() Param Index:4 String Value:5 Type:4 descriptor file name: 07.31.2013 12:02:11.821 : Thread ID [41] : VERBOSE : installing ESDT:/usr/ecs/TS3/CUSTOM/data/ESS/DsESDTMoM

#	Action	Expected Result	Notes
			OD09CMG.005.desc result of the operation (successful): 07.31.2013 12:02:12.215 : Thread ID [41] : INFORMATION : InstallESDTPage:Successfully installed ESDT DsESDTMoMOD09CMG.005.desc

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.2 Test Case 2 - Update multiple ESDTs, success case - [Needs to be rewritten - ODL vs XML] (ECS-ECSTC-164)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Preconditions		#comment
2	Modify the ESDT descriptor ODL files for several (at least 6) previously installed ESDTs, such that the modifications include the following variety of ESDT updates: 1. removing one or more existing collection level attributes that are not restricted 2. removing DLLName parameter from the Collection metadata group 3. changing the ShortName in the CollectionAssociation group 4. changing a mandatory inventory attribute to optional 5 adding one or more optional inventory metadata attributes 6. adding one or more new qualifiers on existing events		#comment
3	Use descriptors in /sotestdata/DROP_721/DS_7E_01/Criteria/2060/Replacement as the initial		#comment

#	Action	Expected Result	Notes
	descriptors. Use only GLA01.013 ... GLA06.013; do not use GLA07.013.		
4	Use descriptors in /sotestdata/DROP_721/DS_7E_01/Criteria/2120 as the replacement descriptors.		#comment
5	Compare each replacement descriptor with its initial counterpart, and note the differences.		
6	Ensure none of the test collections are installed, according to the ESDT Maintenance GUI.		
7	Note the values associated with the following names in EcDsAmESDTMaint.properties: descriptor.target.dir descriptor.source.dir mcf.target.dir E.g., descriptor.target.dir=/stornext/smallfiles/OPS/descriptor descriptor.source.dir=/usr/ecs/OPS/CUSTOM/data/ESS mcf.target.dir=/stornext/smallfiles/OPS/mcf		
8	Install the initial test collections: Copy the descriptor files to the the ESDT Maintenance GUI host in ${descriptor.source.dir}. Log in to the ESDT Maintenance GUI. Click the "Install New ESDTs/Update Existing ESDTs" button. Select all the test descriptors. Click the "Proceed with installation/update" button. Ensure that all descriptors install successfully. Ensure the descriptor files are removed from the source directory.		
9	Verify an MCF file is created in ${mcf.target.dir} for each descriptor. 		
10	Copy the MCF files to a separate directory for later verification. 		
11	Verify each descriptor was moved to ${descriptor.target.dir}. 		
12	Copy the descriptor files to a separate directory for later verification. 		
13	Verify an XML schema file (*.xsd) was created for each descriptor in ${descriptor.target.dir}. 		
14	Copy the schema files to a separate directory for later verification. 		
15	Extract /usr/ecs/TS3/CUSTOM/lib/DSS/EcDsAmEsdtCommonSchemas.jar for later verification.		
16	Setup		#comment
17	Copy the descriptor files to the the ESDT Maintenance GUI host in ${descriptor.source.dir}.		
18	Log in to the ESDT Maintenance GUI.		
19	Click the "Install New ESDTs/Update Existing ESDTs" button.		
20	Select all the test descriptors.		

#	Action	Expected Result	Notes
21	Note the current time as t0.		
22	Click the "Proceed with installation/update" button.		
23	Upon completion of the ESDT update, verify that the GUI displays a message indicating the number of ESDTs successfully updated.		
24	Verify that an MCF file is created for each ESDT being updated, and stored into the configured MCF directory location replacing the previous version.		#comment
25	Verify for each updated descriptor an MCF file exists in ${mcf.target.dir} with a timestamp more recent than time t0.		
26	Verify that the MCF files are correct.		#comment
27	For each descriptor file, note the contents of the INVENTORYMETADATA section and the ARCHIVEDMETADATA section.		
28	Verify the MCF file has the same INVENTORYMETADATA group as the descriptor, with the exception that ProductSpecificMetadata group from the descriptor file is replaced with this static AdditionalAttributes block in the MCF file: GROUP = AdditionalAttributes OBJECT = AdditionalAttributesContainer Data_Location = "NONE" Mandatory = "FALSE" CLASS = "M" OBJECT = AdditionalAttributeName Mandatory = "FALSE" CLASS = "M" Data_Location = "PGE" NUM_VAL = 1 TYPE = "STRING" END_OBJECT = AdditionalAttributeName GROUP = InformationContent CLASS = "M" OBJECT = ParameterValue Mandatory = "FALSE" Data_Location = "PGE" NUM_VAL = 1 TYPE = "STRING" END_OBJECT = ParameterValue END_GROUP = InformationContent END_OBJECT = AdditionalAttributesContainer END_GROUP = AdditionalAttributes		
29	Verify in the MCF file INVENTORYMETADATA section is followed by the descriptor's ARCHIVEDMETADATA section.		
30	Verify the descriptor file for each ESDT is replaced by the updated descriptor file, and that the updated descriptor files are consistent with the ECS data model.		#comment
31	Verify each ESDT's updated descriptor file replaced the initial descriptor file in the ${descriptor.source.dir} directory, using diff.		

#	Action	Expected Result	Notes
32	Verify each ESDT validates against the extracted schema. For each ESDT, Log in to the ESDT Maintenance GUI. Click on the descriptor file in the list. Copy the XML version of the descriptor. Save the XML to a file. Validate the file against the extracted schema: xmllint --noout --schema /path/to/schema/Descriptor.xsd ../descriptor_file.xml		
33	Verify that an XML schema file is generated for each ESDT and stored into the configured location replacing the previous version		#comment
34	Verify that each ESDT has an XML schema file (*.xsd) in ${descriptor.source.dir} with a timestamp on or after time t0.		
35	Verify that the generated XML schema files contain validation rules based upon the contents of the Inventory section of the Descriptor file and the default element rules.		#comment
36	Diff the updated XML schema files against the initial XML schema files saved earlier.		
37	Verify the differences between updated and initial XML schema files correspond to the differences between updated ana initial descriptor files.		
38	Verify that collection-based tables in the AIM Inventory database as identified in the Operations Concept are populated with correct information.		#comment
39	For each updated ESDT, in the ecs database, in the amcollection table, verify lastupdate is after time t0 and each of LongName, CollectionDescription, SpatialSearchType, and RevisionDate match those in the descriptor file: select lastupdate, esdt(shortname,versionid), LongName, CollectionDescription, SpatialSearchType, RevisionDate from amcollection where shortname = '${SHORTNAME}' and versionid = ${VERSIONID};		
40	For each updated ESDT, in the ecs database, in the DsGeESDTConfiguredType table, verify descriptorFileNameBase matches the descriptor file name and spatialSearchType matches that in the descriptor file: select descriptorFileNameBase, spatialSearchType from DsGeESDTConfiguredType where configuredName = '${SHORTNAME}' and versionid = ${VERSIONID};		
41	For each updated ESDT, in the ecs database, verify the AdditionalAttributes are correctly cross-referenced and that each value matches that in the descriptor file in the appropriate AdditionalAttributesContainer section: select aa.attributeid, aa.additionalattributename, aa.additionalattributedescription, aa.additionalattributedatatype, 		

#	Action	Expected Result	Notes
	aa.parameterunitsofmeasure, aa.parameterrangebegin, aa.parameterrangeend, aa.parametervalueaccuracy, aa.valueaccuracyexplanation, aa.measurementresolution from amcollection c join dsmdcollectionaddnlattribsxref ax on c.collectionid = ax.collectionid join dsmdadditionalattributes aa on ax.attributeid = aa.attributeid where c.shortname = '${SHORTNAME}' and c.versionid = ${VERSIONID}; Note that if an additional attribute existed in the database prior to installing an ESDT, it will not be updated to reflect the new definition.		
42	Verify that the insert event and event qualifiers are updated correctly in the Spatial Subscription Server database for each ESDT being updated.		#comment
43	Verify each ESDT has an INSERT event definition: select * from EcNbEventDefinition where ESDT_Id = '${SHORTNAME}' and VersionID = ${VERSIONID}		

44	From each descriptor file, get a list of EVENTPARM names:
	
	sed -n '/^\s*GROUP\s*=\s*EVENT\s*$/,/^END_GROUP\s*=\s*EVENT\s*$/ {
	 /^\s*OBJECT\s*=\s*INSERT\s*$/,/^\s*END_OBJECT\s*=\s*INSERT\s*$/ {
	 /^\s*OBJECT\s*=\s*EVENTPARMS\s*$/,/^\s*END_OBJECT\s*=\s*EVENTPARMS\s*$/ {
	  /^\s*OBJECT\s*=\s*EVENTPARMS\s*$/ d
	 s/^\s*OBJECT\s*=\s*//p
	  }
	  }
	}' ${DESCRIPTOR_FILE}		

45	For each ESDT, find the EVENTPARMS that are eligible to be added to the Spatial Subscription Server database: select attributename from ecnbeventmetadataattrdef where attributename in (${EVENTPARMS}) order by attributename;		
46	For each ESDT, find the EVENTPARMS that were associated with that ESDT as event qualifiers: select attributename from ecnbeventattrxref where esdt_id = '${SHORTNAME}' and versionid = $VERSIONID order by attributename;		
47	Verify the associated event qualifiers match the eligible list with the exception that 'RangeBeginningTime' and 'RangeEndingTime' from the descriptor get combined into 'GranuleTimeRange' in the database; 'RangeBeginningDate' and 'RangeEndingDate' from the descriptor get combined into 'GranuleDateRange' in the database.		
48	Verify that, upon completion of updates, the ESDT Maintenance GUI displays a message saying that the ESDT changes will take effect only after the Ingest service is re-started.		

TEST DATA:
Any current data types.
Initial ESDTS: /sotestdata/DROP_721/DS_7E_01/Criteria/2060/Replacement
Updated ESDTS: /sotestdata/DROP_721/DS_7E_01/Criteria/2120
The 'update' ESDT descriptor files have been modified as follows:
1. removing one or more existing collection level attributes that are not restricted
DsESDTGlGLA01.013.desc
2. removing DLLName parameter from the Collection metadata group
DsESDTGlGLA02.013.desc
3. changing the ShortName in the CollectionAssociation group
DsESDTGlGLA03.013.desc
4. changing a mandatory inventory attribute to optional
DsESDTGlGLA04.013.desc
5 adding one or more optional inventory metadata attributes
DsESDTGlGLA05.013.desc
6. adding one or more new qualifiers on existing events
DsESDTGlGLA06.013.desc
QAPercentInterpolatedData

EXPECTED RESULTS:

1.3 Test Case 4 - Search Granules for Deletion (ECS-ECSTC-166)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Select a test collection.		
2	Find a datetime T_BEGIN that's later than the earliest beginningdatetime of granules in the test collection.		
3	Find a datetime T_END that's earlier than the latest beginningdatetime of granules in the test collection.		
4	Find the granules with beginningdatetime between T_BEGIN and T_END: select granuleid from amgranule where		

#	Action	Expected Result	Notes
	shortname = '${SHORTNAME}' and versionid = ${VERSIONID} and beginningdatetime >= '${T_BEGIN}' and beginningdatetime <= '${T_END}';		
5	To simplify the command line, set and export the following environment variables before running the search utility: MODE <MODE>		
6	Execute the EcDsBulkSearch utility with the following command line parameters: EcDsBulkSearch.pl -mode <MODE> -geoidfile <path/geoidfile_name> -name <ShortName> -version <VersionId> -acquirebegin <mm/dd/yyyy> [<hh:mm:ss>] -acquireend <mm/dd/yyyy> [<hh:mm:ss>]		
7	See AIM Granule Deletion 609 document for details.		#comment
8	Verify the utility runs without errors.		
9	Verify the geoid file file is created in the specified directory.		
10	Verify the geoid file contains the expected granule GEOIds (i.e., all granules for the specified Shortname.versionId which have a BeginningDateTime in the specified range).		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.4 Test Case 6 -Granule Logical Deletion -- DFA (ECS-ECSTC-168)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Logical DFA is combined with Unpublish and BMGT for an end to end test.		#comment
2	Preconditions		#comment

#	Action	Expected Result	Notes
3	Identify several granules in a single collection to delete. Ensure some granules have associated browse.		
4	To simplify the command line, set and export the following environment variables before running the delete utility: #csh setenv MODE <MODE> #bash export MODE=<MODE> See AIM Granule Deletion 609 document for details.		
5	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
6	Ensure BMGT is running: ./EcBmBMGTAppStart ${MODE}		
7	Ensure the granules' collection is enabled for BMGT collection and granule export. update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = '${SHORTNAME}' and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for all of its granules to finish exporting.		
8	Setup and Verification		#comment
9	Note the current time as t0.		
10	Run the EcDsBulkDelete utility with the –dfa option as follows: EcDsBulkDelete.pl -dfa -geoidfile <path/geoidfile_name> -log <log_file_name.log>		
11	Verify the utility runs successfully and the output is correct:		#comment
12	Verify all granule(s) in the path/geoidfile_name have been marked as DFA, i.e., each has DeleteFromArchive = 'Y' in the amgranule table.		
13	Verify the DFAed granule(s) have been logged in the specified log file.		
14	Verify the TCP proxy log shows that after time t0 BMGT exported a DELETE request for each granule.		
15	Note the current time as t1.		
16	Run the Unpublish utility with –aim option in the mode: ./EcDlUnpublishStart.pl -mode ${MODE} -aim -offset 1		
17	Verify the DFAed granules are moved to the hidden Data Pool.		
18	Verify the TCP proxy shows no requests for the unpublished granules after time t1.		

TEST DATA:

Any current data types

EXPECTED RESULTS:

1.5 Test Case 7 - Undelete -physical (ECS-ECSTC-169)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Logical undelete is combined with Publish and BMGT for an end to end test. Note: Granules cannot be undeleted after physical deletion.		#comment
2	Preconditions		#comment
3	Identify several granules in a single collection to delete. Ensure some granules have associated browse. Save the granules' geoids to a file.		
4	To simplify the command line, set and export the following environment variables before running the delete utility: #csh setenv MODE <MODE> #bash export MODE=<MODE> See AIM Granule Deletion 609 document for details.		
5	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
6	Ensure BMGT is running: ./EcBmBMGTAppStart ${MODE}		
7	Ensure the granules' collection is enabled for BMGT collection and granule export. update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = '${SHORTNAME}' and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for all of its granules to finish exporting.		
8	Ensure the granules are logically deleted, e.g., by running bulk delete with the -physical option: EcDsBulkDelete.pl -physical -geoidfile <path/geoidfile_name>		
9	Setup and Verification		#comment
10	Note the current time as t0.		
11	Run the EcDsBulkUndelete utility as follows: EcDsBulkUndelete.pl –physical -geoidfile <path/geoidfile_name> -log <log_file_name.log> 		
12	Verify the utility runs successfully and the output is correct:		#comment
13	Verify all granule(s) in the path/geoidfile_name have been undeleted, i.e.,		

#	Action	Expected Result	Notes
	each has a null DeleteEffectiveDate in the amgranule table.		
14	Verify the undeleted granules have been logged in the specified log file.		
15	Verify the TCP proxy log shows that after time t0 BMGT exported a PUT request for each granule.		
16	Note the current time as t1.		
17	Run the Publish utility with –aim option in the mode: ./EcDlPublishStart.pl -mode ${MODE} -aim -offset 1		
18	Verify the undeleted granules are moved to the public Data Pool.		
19	Verify the TCP proxy log shows that after time t1 BMGT exported a PUT request for each granule.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.6 Test Case 8 - Undelete -- DFA (ECS-ECSTC-170)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Logical unDFA is combined with Publish and BMGT for an end to end test. Note: Granules cannot be undeleted after physical deletion.		#comment
2	Preconditions		#comment
3	Identify several granules in a single collection to delete.		
4	Ensure some granules have associated browse.		
5	Save the granules' geoids to a file.		
6	To simplify the command line, set and export the following environment variables before running the delete utility: #csh setenv MODE <MODE> #bash export MODE=<MODE> See AIM Granule Deletion 609 document for details.		

#	Action	Expected Result	Notes
7	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
8	Ensure BMGT is running: ./EcBmBMGTAppStart ${MODE}		
9	Ensure the granules' collection is enabled for BMGT collection and granule export. update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = '${SHORTNAME}' and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for all of its granules to finish exporting.		
10	Ensure the granules are logically DFAed, e.g., by running bulk delete with the -dfa option: EcDsBulkDelete.pl -dfa -geoidfile <path/geoidfile_name>		
11	Setup and Verification		#comment
12	Note the current time as t0.		
13	Run the EcDsBulkUndelete utility as follows: EcDsBulkUndelete.pl –dfa -geoidfile <path/geoidfile_name> -log <log_file_name.log>		
14	Verify the utility runs successfully and the output is correct:		#comment
15	Verify all granule(s) in the path/geoidfile_name have been undeleted, i.e., each has a null DeleteFromArchive in the amgranule table.		
16	Verify the undeleted granules have been logged in the specified log file.		
17	Verify the TCP proxy log shows that after time t0 BMGT exported a PUT request for each granule.		
18	Note the current time as t1.		
19	Run the Publish utility with –aim option in the mode: ./EcDlPublishUtilityStart <MODE> -ecs -g <ecsId1>		
20	Verify the undeleted granules are moved to the public Data Pool.		
21	Verify the TCP proxy log shows that after time t1 BMGT exported a PUT request for each granule.		

TEST DATA:

Any current data types

EXPECTED RESULTS:

1.7 [PVC] - Test Case 9 - Granule Deletion Phase II - DeletionCleanup - (Needs re-write) (ECS-ECSTC-171)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Granule Deletion Phase II - DeletionCleanup] (including browse)		#comment
2	Preconditions		#comment
3	Select 2 ESDTs with granules to be deleted. Save the ESDTs to a file, one per line.		
4	Ensure granules for at least 2 ESDTs have been marked for deletion, using the BulkDelete utility. At least one of these ESDTs should be a MISR ESDT in the DsMdMisrProcessingCriteria table. Some granules should be marked for deletion using the -physical option, and some should be marked using the -dfa option.		
5	Ensure either that no other granules in the mode are marked for deletion or that it is OK with the lab lead and everyone else in the mode to physically delete them.		
6	Ensure the granules to be deleted are unpublished, e.g., by running the Unpublish utility with the -aim option. Note that Granule Deletion Phase II will skip granules that are in public Data Pool or on order.		
7	To simplify the command line, set and export the following environment variables before running the search utility: #csh setenv MODE <MODE> #bash export MODE=<MODE>		
8	Set up and verification		#comment
9	See AIM Granule Deletion 609 document for details. 1. All command line parameters are optional, if not supplied, required parameters will be prompted in runtime; 2. The batch size command line parameters can take default values if they are not specified by the operator; 3. If there are any granule(s) that have not been cleaned up in a previous run of the DeletionCleanup utility, the operator will be prompted to 1) complete the unfinished run only, or 2) complete the previous unfinished run and start a new run; 4. On the lag time menu, choose either option 1 or option 2. 5. On the Data Type menu, choose option 1, and use an input file of the ESDTs for which you have marked granules for		#comment

#	Action	Expected Result	Notes
	deletion.		
10	Run the EcDsDeletionCleanup utility with the following options: EcDsDeletionCleanup.pl -mode <MODE>		
11	On the lag time menu, choose either option 1 or option 2.		
12	On the Data Type menu, choose option 1, and use the input file of the ESDTs prepared earlier.		
13	Verify that the utility runs successfully and that the output is correct:		#comment
14	Verify all database entries have been cleaned up for granules marked for deletion with the –physical option. For the MISR granules this includes the DsMdMisrAttributes, DsMdMisrCamera and DsMdMisrBrowseGranuleXref tables.		
15	Verify that the database entries have NOT been cleaned up for granules marked for deletion with the –dfa option.		
16	Verify XML metadata files have been physically removed from the XML archive for granules marked for deletion with the –physical option.		
17	Verify XML metadata files have NOT been physical removed from the XML archive for granules marked for delete with the –dfa option.		
18	Verify data files have been physically removed from the archive for all granules marked for deletion (with either the –physical or –dfa options).		
19	Verify that the deleted granule(s) have been logged in the specified log file.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.8 Test Case 12 - Multiple Update request files processed in time tag order, Success Case (ECS-ECSTC-173)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Multiple Update request files processed in time tag order, Success Case] (See DS_7E_01, criterion 3100)		#comment
2	Prepare QA Update request files for 2 originating sites, with at least 3 request files per site.		
3	Each request file is prepared such that it would update QA flags and explanation for at least 10 distinct granules, which do not overlap with granules specified in other request files. (We can choose a different ESDT in each request file to achieve this).		
4	Make sure that the granules provided in each request exist in both the AIM inventory as well as in public Data Pool.		
5	The request files are named such that each file has a different time tag as part of its name.		
6	Place the request files in the configured request directory visible to the QA Update utility		
7	Run the QA Update utility to process all files in the request directory.		
8	Verify from the logs and the time stamps of the updated XML metadata files that the QA Update utility processed the request files for each site, in order of the time tag included as part of the request file name.		
9	Verify that the XML metadata files are updated correctly in the small file archive system. Specifically, verify that the QA flags and the corresponding explanation fields are updated correctly for the affected measured parameters.		
10	Verify that the metadata files in the public Data Pool file systems are replaced by the updated XML metadata files from the small file archive system.		
11	Verify that the updated QA attributes for the affected granules are correctly updated in the Data Pool database.		
12	Verify that the QA Update Utility appended the date and time of update accurate to the minute, to the explanation fields, and stored them in the corresponding metadata files in the small file archive system, as well as in the Data Pool database.		
13	Verify that lastUpdate time for each of the granules affected is updated correctly in the AIM Inventory database as well as in the Data Pool database.		
14	Verification Instructions		#comment
15	Use DPL Ingest GUI to control whether the granules for an ESDT go to the data pool and whether they go to the public or hidden Data Pool area. From DPL Ingest GUI, Configuration, Data Types, configure this ESDT to be inserted into Data Pool and whether it’s public or not.		

#	Action	Expected Result	Notes
16	Configure the ESDT as specified in the Test Data Requirement to be inserted into Data Pool, and ingest 60 granules, and 10 granules per ESDT		
17	Prepare 3 QA update request files per site for 2 originating sites. The update request updates the operational quality flags and explanations of ALL measured parameters of the ESDT by Granule UR. Each request file updates 10 distinct granules from the same ESDT.		
18	Name the QA Update Request file name with the following format: <MODE>_<SITE>_QAUPDATE.<timestamp> (example OPS_LDOPE_QAUPDATE.20010228122030) Note: Use different time tag for each request.		
19	Place the request files in the QA_REQUEST_DIR directory.		
20	Run query SearchWithDbID.ksh for each entry in the request file. Save the query results.		
21	Locate the granule xml files for the affected granules in the small file archive under /stornext/smallfiles/<mode>/metadata/<ShortName.VersionID>, and copy the xml files to a temporary location for later comparison.		
22	Run EcAmQAUUStart <MODE>		
23	In the QAUU log, verify QAUU processed the request files for each site based on the time tag included as part of the request file name.		
24	Locate the XML files for the successfully updated granules in /stornext/smallfiles/<mode>/metadata/<ShortName.VersionID>, check the time stamps of the updated xml metadata file, verify they are updated in the order of time tag included as part of the request file name.		
25	Compare the updated XML metadata file with the original XML files, and verify that the QA flags and explanations are updated correctly for the affected measured parameters.		
26	Locate the XML files for the successful updated granules in /datapool/<MODE>/user/<FILE_SYSTEM>/, and verify they are the same as the ones in the small file archive using diff.		
27	Query the Data Pool database GetDplMeasuredParameter.ksh <dbID> to verify the QA flags are correctly updated. The OperationalQualityFlagExplanation field has time tag appended for the updated QAFlags.		
28	View the updated XML files (both in the small archive and the DataPool), and verify the OperationalQualityFlagExplanation field has time tag appended for the updated QAFlags		

#	Action	Expected Result	Notes
29	Query the Inventory Database GetInDbLastUpdate.ksh <dbID> and the Data Pool database GetDplLastUpdate.ksh <dbID>, and verify that the lastUpdate is updated using the system.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.9 [PVC ONLY] - Test Case 13 - QA Update Performance run : DS_7E_01, Criterion 3200 (ECS-ECSTC-174)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[QA Update Performance run : DS_7E_01, Criterion 3200]		#comment
2	Prepare a QA update request file based on ESDT ShortName, Version ID, and temporal range, for at least two ESDTs that have granules in AIM Inventory database as well as in Data Pool.		
3	The request is designed such that a total of at least 20,000 granules, each with an average of 2 measured parameters, will qualify for update. This was criteria 3200 in the QAUU IPT.		
4	Place the request file in the configured request directory visible to the QA Update Utility.		
5	Run QA Update Utility to process the request file in the directory.		
6	Verify that the QA Update Utility is capable of updating metadata files in the XML archive at a rate of no less than 15,000 granules per hour.		
7	Verify that the QA Update Utility is capable of updating affected Data Pool XML metadata files at a rate of no less than 18,000 files per hour.		
8	Identify 20 000 granules in the PVC for two ESDTs as described in the Test Data Requirements, and make sure the granules are in the public Data Pool.		
9	Prepare a QA update request which updates the science quality flags of ALL measured parameters of the ESDT for a temporal range.		

#	Action	Expected Result	Notes
10	Name the QA Update Request file name with the following format: <MODE>_<SITE>_QAUPDATE.<timestamp> (example OPS_LDOPE_QAUPDATE.20010228122030)		
11	Copy the request file to QA_REQUEST_DIR		
12	Run EcAmQAUUStart <MODE> -file <QAUpdate Request File>		
13	Count the number of updated metadata files in the XML archive whose time stamps fall within an hour window, and verify the rate of updating metadata files is greater than 15 000 granules/hour.		
14	Count the number of updated metadata files in the public data pool area that have the time stamps within an hour window, and verify the rate of updating metadata files is greater than 18 000 granules/hour.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.10 Test Case 14 - XML Replacement Utility:Successfully replace a granule metadata file (ECS-ECSTC-175)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[XML Replacement Utility:Successfully replace a granule metadata file]		#comment
2	Identify a granule metadata file to replace. Query DsMdXMLFile and DsMdXMLPath in the inventory database for a metadata file. This should ensure us that the ESDT is installed and the xml schema is in the correct location.		
3	Go to the destination location and verify that the queried file exists and the permission allows cmshared to read and write. Also, open up the file to ensure the file is good.		
4	Copy the metadata file out and put it in a temporary place. <br		

#	Action	Expected Result	Notes
	/>IMPORTANT: Do not rename the copied out file, it must have the same name as the one in the destination directory		
5	Modify the copied out metadata file. Use the xml schema as a guide to help you select the attribute to modify.		
6	Go to the mode's utilities directory and execute EcDsAmXruStart -xmlfile <absolute path of the modified metadata file>		
7	Verification:		#comment
8	Verify that the modified file is removed from the source location		
9	Verify that the destination location contains the modified metadata file.		
10	Verify that the lastUpdate attribute is updated to a time close to when the replacement was executed.		
11	Verify that the granule's lastUpdate time in DsMdGranules is updated.		
12	Verify that a GRUPDATE event is inserted for the granule into the DsMdGrEventHistory table.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.11 [PVC] - Test Case 16 - DPL â€“ AIM Inventory Validation - (Needs re-write) (ECS-ECSTC-177)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	This criterion was adapted from criterion 200 in ticket DP_7F_01.		#comment
2	S-1 Execute a Data Pool inventory validation Ensure that the AIM Inventory Catalog includes at least two of each of the following cases: a. Collections defined in the DPL inventory and configured to be published during ingest as well as collections not so configured that contain ECS granules in the AIM inventory, including granules that are: logically deleted (non-NULL deleteEffectiveDate), flagged as deleted from archive		#comment

#	Action	Expected Result	Notes
	(DeleteFromArchive set to 'Y') and flagged as hidden (DeleteFromArchive set to 'H'), as well as granules that do not meet any of these conditions. In addition to granules from science collections, include at least one granule each from the DAP, QA, and PH collections that is logically deleted and one that is not. These granules are not meant to represent data integrity issues and are only used to supply conditions that should be ignored by the validation. b. Non-ECS granules in the DPL inventory (also not a data integrity issue). c. Collections defined in the DPL inventory and configured to be published during ingest that contain ECS granules in the AIM inventory whose Inventory entries indicate they are members of the Hidden Data Pool (this simulates a Publication failure or possibly the removal of a granule restriction or deletion). In addition to granules from science collections, include at least one granule each from the QA and PH collections. d. Collections with ECS granules that are public in the Data Pool and are currently flagged as logically deleted (i.e., have a non-NULL deleteEffectiveDate), deleted from archive (i.e., DeleteFromArchive set to 'Y'), or hidden from normal users (i.e., DeleteFromArchive set to 'H') in the AIM Inventory Catalog, as well as granules that do not meet any of these conditions (this simulates the result of running the AIM Granule Deletion utilities without running the DPL Un-Publish utility). e. Granules in the Public Data Pool that contain Inventory Catalog Restriction entries that indicate the Granules should not be Public. f. Science Granules in the Public Data Pool that should have been replaced by a replacement granule in the Hidden Data Pool that should be public (the database may be manually updated to create this situation). This simulates a publication failure on a replacement granule. g. Granules that are registered in the AIM Inventory Catalog but are not listed as Archived (the Inventory Catalog ArchiveTime is NULL). This could be the result of an Ingest failure prior to archiving. The state of the request/granule in Ingest (if it exists) should indicate that Ingest is not currently processing it (it should be in a terminal state). In addition to granules from science collections, include at least one granule each from the DAP, QA, and PH collections. h. Public and Hidden Science Granules that were manually deleted from the Data Pool using the EcDlCleanupGranules utility (both should be represented as hidden after the utility completes). i. Public Science Granules with associated Browse granules where the Browse Inventory Catalog entries indicate they are not in the Public Data Pool. j. Public Browse granules that are associated only with hidden Science Granules. k. Science Granules in the Public Data Pool that are replacements for granules		

#	Action	Expected Result	Notes
	in the Hidden Data Pool. This represents a normal replacement that should not be reported. l. Hidden granules with "not null" version numbers that were replaced but their file names are not consistent with their version numbers recorded in the Inventory Catalog. Specify the output location for the validation report. Perform the test while there are concurrent Ingest and Distribution activities, with at least twenty granules ingested and twenty granules being newly ordered and distributed during the test.</i>		
3	a) Ensure the AllowPublishFlag = 'Y' for datatype MOD29P1D.005 and in AllowPublishFlag = 'N' for datatype MOD29P1N.005 in the AmCollection table.		
4	Ingest the PDRs in the test requirement 100_A/MD29P1D.005 above.		
5	a.1) Use the query below to find geoids: select 'SC:' || esdt(shortname, versionid) || ':' || granuleid from AmGranule where ShortName = 'MOD29P1D' and VersionId = 5 and RegistrationTime > now()::date;		
6	a.2) Pick up the first granule to save in /usr/ecs/<MODE>/CUSTOM/utilities/Delete_Physical.txt Pick up the second granule to save in /usr/ecs/<MODE>/CUSTOM/utilities/Delete_DFA.txt Pick up the third granule to save in /usr/ecs/<MODE>/CUSTOM/utilities/Hidden.txt Pick up the third granule to save in /usr/ecs/<MODE>/CUSTOM/utilities/Hidden.txt		
7	a.3) Ingest the the PDRs in the test requirement 100_A/MYD29P1D.005 above then use the query in step a.1 for MYD29P1D.005 data type to obtain granules. Edit the Delete_Physical.txt in step a.2 to add the first granule above and save the file. Edit the Delete_DFA.txt in step a.2 above to add the second granule above and save the file. Edit the Hidden.txt in step a.2 above to add to add the third granule above and save the file.		
8	a.4) Ingest the the PDRs in the test requirement 100_A/MOD29P1N.005 above then use the query in step a.1 for MOD29P1N.005 data type to obtain granules. Edit the Delete_Physical.txt in step a.2 to add the first granule above and save the file. Edit the Delete_DFA.txt in step a.2 above to add the second granule above and save the file. Edit the Hidden.txt in step a.2 above to add to add the third granule above and save the file		

#	Action	Expected Result	Notes
9	a.5) Ingest the the PDRs in the test requirement 100_A/MYD29P1N.005 above then use the query in step a.1 for MYD29P1N.005 data type to obtain granules. Edit the Delete_Physical.txt in step a.2 to add the first granule above and save the file. Edit the Delete_DFA.txt in step a.2 above to add to add the second granule above and save the file. Edit the Hidden.txt in step a.2 above to add to add the third granule above and save the file		
10	Run EcDsBulkDelete.pl <MODE> –physical –geoidfile delete_physical.txt –server <server> -database <database name> –U <username> -P <user password>		
11	Run EcDsBulkDelete.pl <MODE> –dfa –geoidfile delete_dfa.txt –server <server> -database <database name> –U <username> -P <user password> Use the GranuleId in the Hidden.txt file to set DFA = 'H'.		
12	b) EcDlPublishUtilityStart <MODE> -nonecs –file <filename>		
13	c) Ingest the the PDRs in the test requirement 100_C above create one inputfile which contains granuleId in S-1c.		
14	Run unpublish script below. EcDlUnpublishStart.pl -mode <MODE> -f inputfile Inputfile contains QA and PH granuleIDs.		
15	d) Ingest the PDRs in the test requirement 100_D above. To logically delete 2 granules, run BulkDelete –physical. To DFA 2 granules, run BulkDelete –dfa. To hide 2 granules, use SQL to set DeleteFromArchive = 'H'. Leave 2 granules as is.		
16	e) Ingest the PDRs in the test requirement 100_e.		
17	Login to the ecs database. Exec InsertGranuleRestriction granuleId, RestrictionFlag, UnpublishFlag For example, Exec InsertGranuleRestriction 12345, 255, 'Y'		
18	Run EcDlUnpublishStart.pl script EcDlUnpublishStart.pl -mode <MODE> -g granuleId <step S-1e>		
19	f) Turn replacement on or off for an ESDT using the DPL maintenance GUI. Set ReplacementOn = 'N' for ESDT MOD29P1D.005 Ingest granule f1 and allow it to be published.		
20	Re-ingest d2. It will should publication and therefore remain in the hidden datapool.		
21	Set ReplacementOn = 'Y' for ESDT MOD29P1D.005 when the operator runs IVT it should identify the granule the operator ingested as a missing		

#	Action	Expected Result	Notes
	replacement.		
22	g) Set non-exist archive directory for data type in 100_G above then ingest the PDRs in the test requirement 100_G.		
23	h) Ingest granules.		
24	Get granuleId then run the script below. EcDlCleanupGranules.pl <MODE> -ecs -grans <granuleId> The ecsId_file contains ecsId.		
25	select ShortName, VersionId, granuleId, isOrderOnly, PublishTime, from AmGranule where GranuleId in (granuleId)		
26	j) Ingest PDRs in the test requirement 100_J for sc granules with browse. Manually update the IsOrderOnly to 'H' for the browse granule.		
27	k) Ingest sc granules with browse.		
28	Run the publish the browse using granuleIds above. The filename contains ecs GranuleId EcDlPublishUtilityStart MODE -ecs -file filename		
29	a. Ingest granules		
30	Using the SSS GUI, create a subscription order for MOD29P1D.005. Ingest all PDRs in directory the below. /sotestdata/DROP_801/DP_81_01/Criteria/100/100_Concurrent		
31	While the 40 granules are being ingested and 40 granules are being distributed, execute the validation utility below. EcDlInventoryValidationTool.pl <MODE> and save off the log and output files for verification later.		
32	V-1 Verify that non-ECS granules (S-1b) are not reported as missing from the Archive.		#comment
33	Open the EcDlInventoryValidationTool.log to verify that non-ECS granules are not reported as missing from Archive.		
34	V-2 Verify ECS granules that are eligible to be public but that are members of the Hidden Data Pool (S-1c) are logged by the utility and that the log includes their status in the AIM inventory and the time they were inserted into ECS.		#comment
35	Open the EcDlInventoryValidationTool.log to verify public ECS granules that are logged hidden (S-1c) are logged includes their status and ecsInsertTime for QA and PH granules. 2011/06/02 07:15:40.760 [GranuleId, DeleteEffectiveDate, DeleteFromArchive, ArchiveTime, ReplacementOnFlag]: 3002000469, , N, Jun 1 2011 3:20PM, Y		
36	V-3 Verify Public ECS granules that are logically deleted, DFA’ed, or		#comment

#	Action	Expected Result	Notes
	hidden (S-1d) are logged by the utility and that the log includes their status in the AIM inventory and the time they were inserted into ECS.		
37	Open the EcDlInventoryValidationTool.log to verify public ECS granules that are logged logically deleted, DFA'ed or hidden (S-1d) are logged includes their status and ecsInsertTime. Logically deleted: 2011/06/02 08:05:22.879 (GranuleId, DeleteEffectiveDate, DeleteFromArchive, ArchiveTime): 3002000471, Jun 2 2011 8:00AM, N, Jun 2 2011 7:49AM. DFA'ed: 2011/06/02 08:05:22.879 (GranuleId, DeleteEffectiveDate, DeleteFromArchive, ArchiveTime): 3002000470, , Y, Jun 2 2011 7:49AM. Hidden: 2011/06/02 08:05:22.879 (GranuleId, DeleteEffectiveDate, DeleteFromArchive, ArchiveTime): 3002000472, , H, Jun 2 2011 7:57AM. 		
38	V-4 Verify Public ECS granules that are Restricted with an option to un-publish (S-1e) are logged by the utility.		#comment
39	Open the EcDlInventoryValidationTool.log to verify public ECS granules that are Restricted with an option to un-publish (S-1e) are logged. 2011/06/02 07:51:43.122 (GranuleId, DeleteEffectiveDate, DeleteFromArchive, ArchiveTime): 3002000462, , N, May 31 2011 10:18AM.		
40	V-5 Verify Public ECS granules that have eligible replacement granules in the Hidden Data Pool (S-1f) are logged by the utility and that granules were replaced (S-1K) are not logged.		#comment
41	Open the EcDlInventoryValidationTool.log to verify granules that have eligible replacement granules in the Hidden Data Pool (S-1f) are logged and that granules were replaced (S-1K) are not logged.		
42	V-6 Verify that granules with NULL ArchiveTime (S-1g) are logged by the utility. Notes, granules that are currently being processed by Ingest (not in a "terminal" state) should not be reported.		#comment
43	Open the EcDlInventoryValidationTool.log to verify that granules with NULL ArchiveTime (S-1g) are logged.		
44	V-7 Verify that granules that were manually deleted from the Data Pool (S-1h) are not logged. They should still be represented as Hidden granules in the Inventory Catalog.		#comment
45	Open the EcDlInventoryValidationTool.log to verify that granules that were manually deleted from the Data Pool (S-1h) are not logged.		
46	V-8 Verify that browse granules that are missing from the Public Data Pool (S-1i) are logged by the utility.		#comment

#	Action	Expected Result	Notes
47	Open the EcDlInventoryValidationTool.log to verify that browse granules that are missing from the Public Data Pool (S-1i) are logged. For example, Start CheckInventoryI: Browse Granules that should be in public Datapool. 2011/05/31 16:03:28.409 (BrowseId, DeleteEffectiveDate, ArchiveTime): 3002000158, , Apr 21 2011 2:12PM.		
48	V-9 Verify that Public browse granules that are associated with Hidden Science granules only (S-1j) are logged by the utility.		#comment
49	EcDlPublishUtilityStart OPS -ecs -g browseId (298972) Check the EcDlPublishUtility.log to make sure there is browse id which was published from the command line.		
50	V-10 Verify that no DPL-AIM inventory discrepancies are logged other than those mentioned in V-2 through V-9.		#comment
51	Verify that no other discrepancy (from this data) is logged in this run.		
52	V-11 Verify that the inventory validation creates one or several output files reporting the discrepancies in the specified location.		#comment
53	The VALIDATION_OUTPUT_DIR defined in the EcDlInventoryValidationTool.CFG file. Go to the VALIDATION_OUTPUT_DIR to verify that Inventory Validation creates one or more output file suitable as input to the various repair utilities.		
54	V-12 Verify that the output report(s) include all the discrepancies that were logged and no others.		#comment
55	Verify that an output report is prepared that includes the above errors and no other errors.		
56	V-13 Verify that the validation exits with an exit code indicating that errors occurred.		#comment
57	Open the EcDlInventoryValidationTool.log to verify that the Inventory Validation Tool exits with an exit code that indicates some errors occurred.		
58	V-14 Verify that granules with version numbers that don't match their file names (S-1l) are logged		#comment
59	Verify that granules with version numbers that don't match their file names (S-1l) are logged.		

TEST DATA:

Crit id	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location
100_A/AE_Land 100_A/DAP 100_A/MOD29P1D 100_A/MYD29P1D 100_A/MOD29P1N 100_A/MYD29P1N		AE_Land.002 DAP.001 MOD29P1D.005 MYD29P1D.005 MOD29P1N.005 MYD29P1N.005 PH.001 QA.001		2 granules 2 daps 4 granules 4 granules 4 non- public granules 4 non-public granules 2 ph 2 qa		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_A/AE_Land.002 /sotestdata/DROP_801/DP_81_01/Criteria/100/100_A/DAP.001 /sotestdata/DROP_801/DP_81_01/Criteria/100/100_A/MOD29P1D.005 /sotestdata/DROP_801/DP_81_01/Criteria/100/100_A/MYD29P1D.005 /sotestdata/DROP_801/DP_81_01/Criteria/100/100_A/MOD29P1N.005 /sotestdata/DROP_801/DP_81_01/Criteria/100/100_A/MYD29P1N.005
100_B		ATSM2LSF.001		2 granules		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_B
100_C		AE_Land.002 MOD29P1D.005 MYD29P1D.005 PH.001 QA.001		2 granules 2 granules 2 granules 2 ph 2 qa		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_C
100_D		MOD29P1D.005 MYD29P1D.005		4 granules 4 granules		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_D
100_E		MOD29P1D.005		2 granules		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_E
100_F		MOD29P1D.005		2 granules + 2 replacement		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_F
100_G		AE_Land.002 DAP MOD29P1D.005 MYD29P1D.00		1 granules 1 granules 2 granules 2 non –public granules		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_G

Crit id	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location
		5 PH.001 QA.001		2 ph 2 qa		
100_H		MOD29P1D.005 MYD29P1D.005		2 granules 2 non –public granules		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_H
100_I		MOD29P1D.005 Browse.001		2 granules 2 browse		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_I
100_J		MOD29P1D.005 Browse.001		2 granules 2 browse		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_J
100_K		MOD29P1D.005		2 granules + 2 replacement		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_K
100_Concurrent		MOD29P1D.005 Browse.001		40 granules 40 browse		/sotestdata/DROP_801/DP_81_01/Criteria/100/100_Concurrent

EXPECTED RESULTS:

1.12 Test Case 17 - Tape Archive Repair: DP_7F_01 Criterion 800 (ECS-ECSTC-178)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Tape Archive Repair: DP_7F_01 Criterion 800] Perform a tape archive repair using the RestoreTapeFromOla utility for a list of at least ten (10) granules some of which reside in the public Data Pool and some in the hidden Data Pool.		#comment
2	At least one of the granules shall be a non-science granule, such as PH or		

#	Action	Expected Result	Notes
	DAP. The granules shall have been reported by the tape archive checksum verification utility (ACVU) and the output from that utility shall be used as input into the repair function.		
3	All of the granules shall have checksums in the Data Pool and AIM, but for one of the granules the checksums shall be invalid.		
4	The granules shall reside in at least two different archive volume groups.		
5	Verify that the repair of the granule that has an incorrect checksum fails and that the failure is logged.		
6	Verify that the repair of the granules with correct checksum succeeds and the granules are written to the correct volume groups.		
7	Retrieve the repaired granules via a command line copy operation to verify that they have been written to tape correctly.		
8	Verify that the repair exits with an exit code indicating that an error occurred.		

TEST DATA:

EXPECTED RESULTS:

1.13 Test Case 18 - On-line Archive Repair from non-resident tape: DP_7F_01 Criterion 320 (ECS-ECSTC-179)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	<i>[On-line Archive Repair from non-resident tape: DP_7F_01 Criterion 320] Create phantoms in the public and hidden Data Pool that satisfy the following conditions: a. Turn at least ten (10) public science granules into phantoms including MISR, MODIS and ASTER granules by removing their science and XML files. At least one of these granules shall be in use for order purposes; remove its hidden links, as well. b. At least one but not all of the science granules shall have a browse that is present in the public Data Pool. c. At least two of the phantom science granules with browse shall also be missing their browse links from the Data Pool		#comment

#	Action	Expected Result	Notes
	directories, including a MISR and a non-MISR granule. d. One of the science granules shall have experienced a file name collision in the hidden Data Pool, that is, its file names shall have been suffixed during Data Pool insert.</i>		
2	Also turn at least five (5) public Browse granules, including MISR, MODIS and ASTER browse into phantoms by removing their browse files from the public Data Pool.		
3	Save the removed XML and browse files for comparison during the test.		
4	Submit an on-line archive repair for these granules via the command line using the RestoreOlaFromTape utility, specifying the granules to be repaired in an input file; and request restoration of browse links that may be missing for granules whose files were restored (using the –restorelinks command line parameter).		
5	Ensure that the test requires access to granules on a tape that is resident in the silo as well as a tape that is not resident in the silo.		
6	Verify that all science granules that resided on tapes resident in the silo were repaired, i.e., their files are now in the correct public Data Pool locations and have the original file names		
7	Verify that the science granules that resided on tapes that were not resident in the silo were not repaired.		
8	Verify that the utility logs each of the granules that were not repaired, the reason for not repairing it, and the tape label.		
9	Verify that the utility exits with an exit code indicating the occurrence of errors.		
10	Verify that the utility sends a notice to the specified e-mail address indicating that a repair failed due to error, the nature of the repair, and the name and location of the log file.		
11	Make the missing tape resident in the silo and re-run the repair for the skipped granules. Verify that the remaining granules are now repaired		
12	Verify that the contents of all the restored XML files (including those referenced in V-6) match the contents of the original XML files.		
13	Verify that all browse granules were repaired and are present as jpeg files in their original disk location and with their original file names.		
14	Verify that the repaired jpeg files are identical to the original jpeg files.		
15	Verify that the missing browse links are restored.		
16	Verify that the hidden links required by the public granules in condition (a) that are also referenced by orders were restored.		

#	Action	Expected Result	Notes
17	Verify that any files that were copied from the ECS tape archive into the Data Pool during the repair were checksummed and that the checksum is recorded in the Data Pool inventory.		
18	Verify that any files that were copied from the ECS browse archive into the Data Pool during the repair had their checksum verified.		

TEST DATA:

EXPECTED RESULTS:

1.14 [PVC] - Test Case 19 - Failure Recovery Performance, Failure in AIM Inventory DB : DS_7E_01_QAUU, Criterion 3210 (ECS-ECSTC-180)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Failure Recovery Performance, Failure in AIM Inventory DB : DS_7E_01_QAUU, Criterion 3210]		#comment
2	Repeat the test used in Criterion 3200, but using a different time tag as part of the file name to distinguish the current updates from the updates made in Criterion 3200.		
3	Lock the Sybase Account ID for the QA Update service, and kill the Sybase process ID (spid) for the QA Update process to cause a Database failure.		
4	Verify that the QA Update run terminated with an incomplete status(not all granules were updated), after displaying and logging an appropriate error message.		
5	Unlock the Sybase Account ID for the QA Update service.		
6	Re-start the QA update run.		
7	Verify that the QA update utility displays a message about an incomplete run from a previous session, and prompts the operator to complete the run.		
8	Verify that after operator confirmation, the QA update utility resumes processing the rest of the granules in the request.		

#	Action	Expected Result	Notes
9	Verify that the QA update utility completed updates for all granules correctly.		
10	Verify that the run time difference between this test and the test conducted in Criterion 3200 is less than 10 minutes. (The total run time for the current test is computed as the sum of run times for the initial run up to failure, and the failure recovery run, not counting the pause between the two runs.)		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.15 Test Case 20 - Running QA Update utility while Data Pool Move Collection utility is running in the same mode; Error Case : DS_7E_01_QAUU, Criterion 3170 (ECS-ECSTC-181)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Running QA Update utility while Data Pool Move Collection utility is running in the same mode; Error Case : DS_7E_01_QAUU, Criterion 3170]		#comment
2	Start a Data Pool Move Collection utility run.		
3	While the Data Pool Move Collection run is in progress, attempt to start up a QA Update run in the same ECS mode.		
4	Verify that the QA Update Utility aborted with an error message indicating that the QA update is not allowed while Data Pool Move collection run is in progress.		
5	Verify that Data Pool Move Collection utility successfully runs to completion.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.16 Test Case 21 - Fault recovery â€“ physical delete; continue with previous run : DS_7E_01_GD, Criterion 1280 (ECS-ECSTC-182)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault recovery – physical delete; continue with previous run : DS_7E_01_GD, Criterion 1280]		#comment
2	Run the EcDsDeletionCleanup utility with a lag time of 0, and introduce a system fault during the run. The number of granules to be physically deleted during the run should be large enough that some but not all of the granules to be deleted are processed before the fault occurs.		
3	After the fault occurs, use EcDsBulkDelete utility with the physical option to mark additional files for deletion.		
4	Remove the fault condition, and restart the EcDsDeletionCleanup utility with a lag time of 0. When prompted whether to continue the previous run or start a new run, choose to continue the previous run.		
5	Verify that the utility can be successfully restarted after the fault (with the physical delete option).		
6	Verify that, on restart, the utility displays the number of granules that were not processed in the previous run, and allows the operator to continue with the previous run.		
7	Verify that all granules that were marked for deletion when the utility was initially run are deleted successfully from the inventory database, the xml archive and the science file archive.		
8	Verify that granules which were marked for deletion after the fault occurred are also deleted successfully from the inventory database, the xml archive and the science file archive.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.17 Test Case 22 - Failure Recovery during Update, database access failure : DS_7E_01_ESDTGUI, Criterion 2330 - (Needs re-write) (ECS-ECSTC-183)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Failure Recovery during Update, database access failure : DS_7E_01_ESDTGUI, Criterion 2330]		#comment
2	Modify several (at least 4) ESDTs making sure that they can be successfully updated.		
3	Place the descriptor files in a configured source directory.		
4	Lock a table in the Subscription Server database (e.g. EcNbEventDefinition table) to cause update to stall during database access for event definition updates.		
5	From the ESDT Maintenance GUI, select and update the ESDTs.		
6	Lock the Sybase account for the ESDT Maintenance GUI and kill the Sybase process ID (spid) to cause a database error.		
7	Verify that the ESDT Maintenance GUI detects the database errors and displays an appropriate error message.		
8	Verify that the ESDT that was affected by the database table locking did not complete update. Verify the following:		#comment
9	Verify that the descriptor file is still in the configured source directory.		
10	Verify that the “lastUpdate” attribute in the DsGeESDTConfiguredType table does not indicate a new update time for the ESDT.		
11	Verify that thedescriptor, MCF, and the ESDT specific schema in the respective configured directories are restored to their original versions prior to update.		

#	Action	Expected Result	Notes
12	Unlock the Sybase account ID for the ESDT Maintenance GUI that was locked in Setup step 5 to resume database processing		
13	Unlock the database table that was locked in Setup step 3.		
14	Verify that the ESDT Maintenance GUI displays a message about incomplete update request in a previous session, and prompts the operator to clean up the incomplete request.		
15	Operator cleans up the incomplete update request and resumes processing.		
16	Upon completion of the ESDT update, verify that the GUI displays a message indicating successful update.		
17	Verify that all ESDTs including the one that failed in the previous session are now updated. Verify the following:		#comment
18	Verify that the MCF files for the updated ESDTs are stored in the configured MCF directory location replacing the previous versions.		
19	Verify that the MCF files are correct		
20	Verify that the XML schema files are generated for the updated ESDTs and stored into the configured location replacing the previous versions.*Verify that the generated XML schema files contain validation rules based upon the contents of the Inventory section of the Descriptor file		
21	And verify the default element rules.		#comment
22	Verify that the descriptor ODL files for the ESDT being updated are moved from the source source directory into the configured descriptor directory replacing the previous versions.		
23	Verify that collection based tables in the AIM Inventory database as identified in the Operations Concept are populated with correct information.		
24	Verify that the CSDTs associated with the ESDTs updated are registered into the AIM Inventory database.		
25	Verify that an insert event, and event qualifiers are inserted into the Spatial Subscription Server database for each ESDT updated.		
26	Verify that, upon completion of update, the ESDT Maintenance GUI displays a message saying that the ESDT changes will take effect only after the Ingest service is re-started.		

TEST DATA:

Any current data types

EXPECTED RESULTS:

1.18 Test Case 23 - Failure Recovery during Update, ESDT Maintenance GUI failure : DS_7E_01_ESDTGUI, Criterion 2337 (ECS-ECSTC-184)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Failure Recovery during Update, ESDT Maintenance GUI failure : DS_7E_01_ESDTGUI, Criterion 2337]		#comment
2	Modify several (at least 4) ESDTs making sure that they can be successfully updated.		
3	Place the descriptor files in a configured source directory.		
4	Lock a table in the Subscription Server database (e.g. EcNbEventDefinition table) to cause update to stall during database access for event definition updates.		
5	From the ESDT Maintenance GUI, select and update the ESDTs.		
6	Shut down the ESDT Maintenance GUI by sending a kill signal to the ESDT Maintenance GUI process.		
7	Verify that the ESDT being updated while the database table was locked did not complete update. Verify the following:		#comment
8	Verify that the descriptor file is still in the configured source directory.		
9	Verify that the “esdtstate” attribute in the DsGeESDTConfiguredType table for the ESDT does not indicate installation completion status.		
10	Verify that there was no partial update of ESDT.		
11			#comment
12	Unlock the database table that was locked in Setup step 3 to allow the update process to continue.		
13	Restart the ESDT Maintenance GUI.		
14	Verify that the ESDT Maintenance GUI displays a message about incomplete update request in a previous session, and prompts the operator to clean up the incomplete request.		
15	Operator cleans up the incomplete ESDT update, and resumes ESDT update.		
16	Verify that the Verification steps 7 through 12 described in Criterion 2330 are		

#	Action	Expected Result	Notes
	satisfied.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.19 Test Case 24 - Error Handling Options: recoverOnly and recoverInvestigated: DS_7E_01_QAUU, Criterion 3230] (ECS-ECSTC-185)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Error Handling Options: recoverOnly and recoverInvestigated: DS_7E_01_QAUU, Criterion 3230]		#comment
2	Prepare a QA update request file to update at least 20 granules that are in the AIM Inventory and in the public Data Pool		
3	Temporarily make some (at least 10) of the affected XML metadata files in the public Data Pool file system read-only, to cause the XML metadata file updates to fail.		
4	Run QA Update Utility using ‘noexitonerr’ option, providing the input request file on command line.		
5	Verify that the QA updates are successfully completed in the AIM inventory database as well as in the XML archive file system.		
6	Verify that the QA Update Utility failed to update those granules in the Data Pool database, and the corresponding XML metadata files that have been made read-only.		
7	Verify that the QA Update Utility successfully completed updating the Data Pool database and the corresponding XML files in the public Data Pool, for other granules whose metadata file access was not altered.		
8	Verify that the QA update utility logs the time, DPL granule ID, and DPL file name for each granule that it failed to update in the Data Pool.		

#	Action	Expected Result	Notes
9	Verify that the QA update utility displays a message indicating failure in updating those read-only XML files in Data Pool		
10	Verify that the input request file is moved into the failed request directory.		
11	Verify that an email notification is sent to the configured e-mail address for the site as well as to the configured internal DAAC email address, indicating error in updating Data Pool.		
12	Verify in the AIM inventory database that the granules that failed the QA updates were flagged with “failed” status in the QA Update working table		
13	Correct access permission of the Data Pool XML metadata files for 5 of the granules that failed previously so they can be successfully updated in subsequent failure recovery run.		
14	Add another QA Update request file in the input request directory to perform QA update on a different set of granules. Prepare the request file such that it will succeed in performing the QA update		
15	Submit a QA update failure recovery run using a command-line option “recoverOnly” to attempt to recover from the previously failed QA Update run.		
16	Verify that the QA update run only attempts to recover the previously failed granules, and no new request get processed.		
17	Verify this time that the QA Update run successfully updated the Data Pool XML metadata files for the 5 granules whose errors have been corrected in V-3230-9		
18	Verify that the QA Update run again failed those granules whose errors were not corrected in V-3230-9, and that they are now flagged to indicate that they need investigation		
19	Correct the Data Pool file access permission for the remaining granules that have been flagged as needing investigation, so that they can be successfully updated during a subsequent recovery run.		
20	Re-submit a QA Update recovery run using a command line option “recoverInvestigated”.		
21	Verify that the QA Update run successfully updated the XML metadata files for those granules that have been corrected in V-3230-15.		
22	Verify that QA Update run successfully processed the new QA Update request file placed in the input request directory in V-3230-10.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.20 Test Case 25 - Fault recovery - delete or DFA : DS_7E_01_GD, Criterion 1260 (ECS-ECSTC-186)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault recovery - delete or DFA : DS_7E_01_GD, Criterion 1260]		#comment
2	Run the EcDsBulkDelete utility with the logical delete or DFA option, and introduce a system fault during the run. The utility's input file should be large enough that some but not all of the granules in the input file are processed before the fault occurs.		
3	Remove the fault condition, and restart the utility.		
4	Verify that when the fault is introduced, the utility logs an error and terminates gracefully		
5	Verify that the utility can be successfully restarted after the fault.		
6	Verify that all granules in the input file are processed successfully.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.21 Test Case 26 - RunArchive Checksum Validation Utility (ACVU) providing a file with a list of dbIds without the -calculate option : CK_7F_01, Criterion 670 (ECS-ECSTC-187)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[RunArchive Checksum Validation Utility (ACVU) providing a file with a list of dbIds without the -calculate option : CK_7F_01, Criterion 670]		#comment
2	Choose a list of dbIds that span all the possible checksum status and checksum verification time combinations. Run ACVU specifying a file of dbIds and days since last checksum without the calculate option.		
3	Verify the last checksum times of the appropriate granules were updated.		
4	Verify that none of the dbIds with null last checksum times were updated.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.22 [PVC] - Test Case 27 - AIM XML Check Utility (ECS-ECSTC-188)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[AIM XML Check Utility]		#comment
2	Note: Refer to the AIM XML Check Utility 609 document for detailed instructions.		#comment
3	Usage of EcDsAmXcu.pl: EcDsAmXcu.pl <MODE> [-days <NUMBER OF DAYS>] [-percent <PERCENT 1-100>] (-ESDT <ESDTS ex. MOD29.005> [-startdate <STARTDATE ex: Jan 1 2008> [-enddate <ENDDATE ex: Dec 25 2008>]] | -granuleid <GRANULEIDS> | -file <FILENAME>) [-outputDir <DIRECTORY>]		#comment
4	Run XCU to check the well-formedness of all the xml files or only the specified xml files (by using the optional parameters) in the small file archive		

#	Action	Expected Result	Notes
	in the mode.		
5	Verify that the utility reports the validation results and list the xml files which are not well-formed.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.23 Test Case 28 - EcDsCheckXMLArchive utility (ECS-ECSTC-189)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EcDsCheckXMLArchive utility] Note: Refer to the Check XML Archive 609 document for detailed instructions.		#comment
2	Ensure that there are at least 5 files in the AmMetadataFile table which are not in the XML archive, and at least 5 files in the XML archive for the mode which are not in the AmMetadataFile table in the AIM database,		
3	Run the EcDsCheckXMLArchive.pl utility in the mode.		
4	Verify that the utility produces two output files: MissingXMLArchive.report.YYYY.MM.DD and MissingXMLEntryInDB.report.YYYY.MM.DD.		
5	Verify that the MissingXMLArchive.report.YYYY.MM.DD file contains the entries in the DsMdXMLFile table that are missing files in the XML archive.		
6	Verify that the MissingXMLEntryInDB.report.YYYY.MM.DD file contains the files in the XML archive that are not in the DsMdXMLFile table.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.24 Test Case 29 - On-Line Archive Repair including QA and PH links: DP_7G_01 Criterion 300] (ECS-ECSTC-190)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	On-Line Archive Repair including QA and PH links: DP_7G_01 Criterion 300]		#comment
2	S-1: Turn at least ten public AMSR granules that have associated QA, PH, and Browse granules into phantoms, i.e., their Data Pool (and AIM) inventory entries are intact, but their science and XML files and their symbolic links are missing from the On-Line Archive. At least one of the AMSR granules shall be referenced by an order. Remove its hidden links.		
3	S-2: Also turn at least two public QA and two PH granules into phantoms, i.e., their Data Pool (and AIM) inventory entries are intact, but neither their files nor the symbolic links that point to them are still in the On-Line Archive, but their AMSR granules are present and public, and are not among the phantoms listed in S-1. At least one of the QA and one of the PH granules shall be referenced by an order. Remove their hidden links, as well.		
4	Submit an on-line archive repair for the AMSR granules and the QA/PH granules via the command line, specifying the granules to be repaired in an input file; and request restoration of links that may be missing for granules whose files are being restored.		
5	Verify that all AMSR science granules were repaired, i.e., their files and symbolic links are now in the correct public Data Pool locations, have the original file respectively, link names, and that the links point to the same QA, PH, and Browse files as before.		
6	Verify that all QA and PH granules were repaired, i.e., their files are now in the correct public Data Pool locations and the symbolic links that point to the files have been restored in the correct locations, as well.		
7	Verify that the hidden links referenced in S-1 and S-2 were restored.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.25 AIM Modified

1.25.1 Test Case 3 - Delete multiple ESDTs, success case (ECS-ECSTC-165)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	DS_7E_01 ESDT Maintenance GUI ITP: http://dmserver.gsfc.nasa.gov/release721/ESDT_Maint_GUI/ITP_DS_7E_01_ESDTGUI.doc¶		#comment
2	Preconditions		#comment
3	Ensure a database client is connected to the ecs database.		
4	Identify 3 ESDTs for deletion (C1, C2, C3).		
5	Save the collection IDs of C1, C2, C3. For each, select collectionid from amcollection where shortname = '${SHORTNAME}' and versionid = ${VERSIONID}		
6	Save the attribute IDs associated with each of C1, C2, C3. For each, select attributeid from amcollection c join dsmdcollectionaddnlattribsxref x on c.collectionid = x.collectionid where shortname = '${SHORTNAME}' and versionid = ${VERSIONID}		
7	Ensure C1, C2, C3 have no granules. For each, select esdt(shortname, versionid), granuleid from amgranule where shortname = '${SHORTNAME}' and versionid = ${VERSIONID}		
8	Ensure C1, C2, C3 are absent from the data pool. For each, select * from amcollection where groupid is null and shortname = '${SHORTNAME}' and versionid = ${VERSIONID}		
9	Ensure C1, C2, C3 are absent from the Spatial Subscription Server database. For each, select * from EcNbSubscription where esdt_id = '${C1_SHORTNAME}' and versionid = ${C1_VERSIONID}		

#	Action	Expected Result	Notes
10	S-1 From the ESDT Maintenance GUI select a group of ESDTs for deletion. Each of the ESDTs selected must satisfy the following conditions to for deletion to be possible: 1. the ESDT contain no granules in the AIM inventory database 2. the ESDT is not defined within the Data Pool 3. the Spatial Subscription service contains no active or inactive subscriptions referencing the ESDT		#comment
11	Note the current time as t0.		
12	On the ESDT Maintenance GUI "ESDT List" page, select C1, C2, and C3.		
13	Click the “Delete Selected ESDTs” button at the bottom of the page to begin the ESDT deletion.		
14	Select OK at the confirmation prompt.		
15	V-1 Verify that each ESDT was deleted successfully. 1. The MCF file was deleted 2. The ESDT XML schema file was deleted 3. The ESDT descriptor file was deleted. 4. The ESDT basic collections and PSA associations were deleted from the Inventory database 5. The insert events and event qualifiers for the ESDT were deleted from the Subscription Server database 6. The XML metadata file directories associated with the ESDT were removed from the XML archive 7. PSA definitions are removed if there are no other ESDTs associated with them		#comment
16	1. Verify no MCF files exist for C1, C2, or C3. For each ESDT, ls -l /stornext/smallfiles/${MODE}/mcf/${SHORTNAME}#${VERSIONID}.MCF For example, ls -l /stornext/smallfiles/OPS/mcf/MOD29P1D#005.MCF		
17	2. Verify no ESDT XML schema files exist for C1, C2, or C3. For each ESDT, /stornext/smallfiles/${MODE}/descriptor/*${SHORTNAME}.${VERSIONID}.xsd For example, ls -l /stornext/smallfiles/OPS/descriptor/*MOD29.003.xsd		
18	3. Verify no ESDT descriptor files exist for C1, C2, or C3. For each ESDT, /stornext/smallfiles/${MODE}/descriptor/*${SHORTNAME}.${VERSIONID}.desc For example, ls -l /stornext/smallfiles/OPS/descriptor/*MOD29.003.desc		
19	4. Verify the collections table has no rows for C1, C3, or C3. For each ESDT, select * from amcollection where shortname = '${C1_SHORTNAME}' and versionid = ${VERSIONID}		

#	Action	Expected Result	Notes
20	4. Verify additional attribute associations no longer exist for C1, C2, C3. select * from dsmdcollectionaddnlattribsxref where collectionid in (${COLLECTIONIDS})		
21	5. Verify Subscription Server database has no event definitions for C1, C2, or C3. For each, select * from ecnbeventdefinition where esdt_id = '${SHORTNAME}' and versionid = ${VERSIONID}		
22	5. Verify Subscription Server database has no event qualifiers for C1, C2, or C3. For each, select * from ecnbeventattrxref where esdt_id = '${SHORTNAME}' and versionid = ${VERSIONID}		
23	6. Verify no XML metadata directory exists in the small file archive for C1, C2, or C3. For each, ls -ld /stornext/smallfiles/${MODE}/metadata/${SHORTNAME}.${VERSIONID} For example, ls -ld /stornext/smallfiles/OPS/metadata/MOD29.003		
24	7. Verify the additional attribute definitions for C1, C2, C3 no longer exist: select attributeid from dsmdadditionalattributes where attributeid in (${ATTRIBUTEIDS}) If any previously associated attributes remain, verify they are associated with existing collections: select distinct(attributeid) from dsmdcollectionaddnlattribsxref where attributeid in (${REMAINING_ATTRIBUTEIDS}) should return the same list of ${REMAINING_ATTRIBUTEIDS}.		
25	V-2 Verify that the ESDT Maintenance GUI displays a message indicating number of ESDTs deleted.		#comment
26	Verify the ESDT Maintenance GUI displays a message indicating 3 ESDTs were deleted.		
27	V-3 Verify that the ESDT Maintenance GUI logs the processing activities, including time of request, action requested, ESDT ShortName, Version ID, descriptor file name, and the result of the operation, in a configured log file.		#comment
28	ssh to the ESDT Maintenance GUI host.		
29	Search the ESDT GUI log for the first mentions of C1, C2, and C3 on or after the time t0. /usr/ecs/${MODE}/CUSTOM/logs/ESDTMaintenanceGUI.*.log*		
30	Verify each of the following is logged for each of C1, C2, C3: request time action requested (delete) ShortName Version ID descriptor file name operation result		The descriptor file name is not logged in the ESDTMaintenanceGUI.debug0.log file. There is a

#	Action	Expected Result	Notes
			existence ncr8051131.Test Run - 1705:Delete multiple ESDTs, success case - Multiple Issues Found

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.25.2 [PVC] - Test Case 5 - Granule Logical Deletion -- physical (ECS-ECSTC-167)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Logical delete is combined with Unpublish and BMGT for an end to end test.		#comment
2	Preconditions		#comment
3	Identify several granules in a single collection to delete. Ensure some granules have associated browse.		
4	To simplify the command line, set and export the following environment variables before running the delete utility: #csh setenv MODE <MODE> #bash export MODE=<MODE> See AIM Granule Deletion 609 document for details.		
5	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
6	Ensure BMGT is running: ./EcBmBMGTAppStart ${MODE}		
7	Ensure the granules' collection is enabled for BMGT collection and granule export. update bg_collection_configuration set collectionexportflag = 'Y', granuleexportflag = 'Y' where shortname = '${SHORTNAME}' and versionid = ${VERSIONID} If the collection was newly enabled for export in this step, wait for all of its granules to finish exporting.		

#	Action	Expected Result	Notes
8	Setup and Verification		#comment
9	Note the current time as t0.		
10	Run the EcDsBulkDelete utility with the –physical option as follows: EcDsBulkDelete.pl -physical -geoidfile <path/geoidfile_name> -log <log_file_name.log>		
11	Verify the utility runs successfully and the output is correct:		#comment
12	Verify all granule(s) in the path/geoidfile_name have been marked for deletion, i.e., each has a non-null deleteEffectiveDate in the amgranule table.		
13	Verify the deleted granule(s) have been logged in the specified log file.		
14	Verify the TCP proxy log shows that after time t0 BMGT exported a DELETE request for each granule.		
15	Note the current time as t1.		
16	Run the Unpublish utility with –aim option in the mode: ./EcDlUnpublishStart.pl -mode ${MODE} -aim -offset 1		
17	Verify the granules marked for deletion are moved to the hidden Data Pool.		
18	Verify the TCP proxy shows no requests for the unpublished granules after time t1.		ncr8051949 Unpublished logical deleted a science granule associated with a browse causes BMGT exports the logical deleted a science granule to ECHO twice.

TEST DATA:
Any current data types

EXPECTED RESULTS:

1.25.3 [PVC] - Test Case 11 - QAUU: Update by ESDT and temporal range for specific measured parameters (ECS-ECSTC-172)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[QAUU: Update by ESDT and temporal range for specific measured parameters] (See Criterion 3000 in DS_7E_01)		#comment
2	Select an ESDT that has at least two measured parameters per granule. The ESDT has granules in AIM inventory, as well as in Data Pool, some of which are located in public Data Pool area, and others are in hidden Data Pool area. ESDT MOD10A1 is a good choice. select granuleid, beginningdatetime, isorderonly, deletefromarchive from amgranule where collectionid = order by beginningdatetime; If data is not found, Use DPL Ingest GUI to control whether the granules for an ESDT go to the data pool and whether they go to the public or hidden Data Pool area. a) From DPL Ingest GUI, Configuration, Data Types, configure this ESDT to be inserted into Data Pool and whether it's public or not. b) Configure the ESDT as specified in the Test Data Requirement to be inserted into Data Pool public area, and ingest 5 granules. c) Configure the ESDT to be inserted into Data Pool but not public area, and ingest another 5 granules.		
3	Set up an input file for the QA Update Utility to update the science quality flags of the ESDT for a temporal range, such that there are at least 10 granules within the time range. Set the temporal range so that it matches the boundary of at least one of the granules acquisition times. Specify a valid email address for the originating SCF site as "Requester ID" in the input request file. Specify science QA flag values and explanation for at least 2 measurement parameters. a) Prepare a QA update request which updates the science quality flags of the specific measured parameters of the ESDT for a temporal range. See the QAUU 609 section "4.8.9.3 Input File Format" for the format of Request with Temporal Range: http://edhs1.gsfc.nasa.gov/waisdata/eed/pdf/609eed001_File_6.pdf The measured parameters should be specified in the request file for this test. b) Specify some granules to have acquisition times at the exact boundary points of the temporal range. c) Specify the email address in the input request file.		

#	Action	Expected Result	Notes
	Put 'From ' as the first line of the request file for the request from GSFC. d) Specify science QA flag and explanations for at least 2 measured parameters. e) Name the QA Update Request file name with the following format: __QAUPDATE. (example OPS_GSFC_QAUPDATE.20010228122030) Sample contents of the request file ********** From GSFC begin QAMetadataUpdate Science ESDT MOD10A1 5 Nov 05 2007 00:00:00.000 Nov 05 2007 01:15:00.000 Snow_Albedo_Daily_Tile Being Investigated Verify ESDT Temp Option MOD10A1 5 Nov 05 2007 00:00:00.000 Nov 05 2007 01:15:00.000 Snow_Cover_Daily_Tile Being Investigated Verify ESDT Temp Option end QAMetadataUpdate **********		
4	Run query. select granuleid, beginningdatetime, isorderonly, from amgranule where shortname = 'MOD10A1' and versionid = 5 where beginningdatetime >= and beginningdatetime < Save the query results		
5	Locate the granule XML files for the affected granules in the small file archive under /stornext/smallfiles//metadata/, and copy the XML files to a temporary location for later comparison.		
6	Ensure that email addresses are in the config file Update the following parameters in the config file or add the following 3 parameters with appropriate values. _EMAIL_FROM_ADDRESSES=xyz@nasa.gov _EMAIL_REPLY_ADDRESS=xyz@nasa.gov _NOTIFICATION_ON_SUCCESS= Y		
7	Copy the request file to QA_REQUEST_DIR directory.		
8	Run EcAmQAUUStart -file -skipRecovery		
9	VERIFICATION		
10	Verify that the QA Update Utility displays the total number of granules that		

#	Action	Expected Result	Notes
	will be affected by this update and prompts for operator confirmation before performing the update.		
11	Verify that upon completion of the QA update run, the operator can view the following information: a. A list of affected granules identified by ESDT Short Name, Version ID, LGID, dbID, along with the measured parameters and the updated QA flag values b. Total number of granules updated		
12	Verify that a log file is created with a standard log file name in a configured log directory.		
13	Verify the QA Update Utility log file is created under /usr/ecs//CUSTOM/logs.		
14	Verify the total number of affected granules is displayed and correct by comparing with the number from the query in step 4, and confirm the updating after the prompt. Verify QAUU Log(EcDsAmQauu.ops0.log) for the following statements: Total number of granules updated: Total number of granules specified in the request: Total number of granules eligible to be updated after validation:		
15	Verify the QA update utility log recorded the processing activities.		
16	Verify that the granules listed are in the correct acquisition time range (inclusive of the boundary points) and the granule count is correct.		
17	Verify that the XML metadata files for each of the granules are updated correctly in the small file archive system.		
18	Specifically, verify that the QA flags and the corresponding explanation fields are updated correctly for the affected measured parameters. Locate the XML files for the affected granules in /stornext/smallfiles//metadata/, compare with the original XML files using diff, and verify that the QA flags and explanations are updated correctly for the affected measured parameters.		
19	View the updated XML files, and verify the ScienceFlagExplanation field has time tag accurate to the minute appended for the updated QAFlags.		
20	Verify the XML metadata files in the public and hidden Data Pool file systems are replaced by the corresponding updated XML metadata files in the small file archive system.		
21	Locate the XML files for the affected granules in public data pool		

#	Action	Expected Result	Notes
	/datapool//user//, and verify they are the same as the ones in the small file archive using diff.		
22	Verify that the QA Update Utility appended the date and time of update accurate to the minute, to the explanation fields, and stored them in the corresponding metadata files in the small file archive system, as well as in the Data Pool database.		
23	Verify that lastUpdatetime for each of the granules affected is updated correctly in the AIM Inventory database as well as in the Data Pool database using the system time. Run Query select granuleid, beginningdatetime, isorderonly, lastupdate from amgranule where shortname = 'MOD10A1' and versionid = 5 order by lastupdate desc; and verify that all the granules lastupdate is changed to the time when the utility was run.		
24	Verify that the QA update utility recorded the list of granules updated along with the updated QA values for the affected measured parameters in the AIM Inventory database (for use by BMGT) Run query select * from DsMdGrEventHistory order by eventtime desc, and verify all the updated granules have events recorded.		
25	Verify that the QA update request file is moved into the configured completed request directory location. Verify the QA Update Request file is moved from directory QA_REQUEST_DIR to QA_COMPLETED_REQUEST_DIR.		
26	Verify that a history file is generated and stored in the QA_HISTORY_DIR directory, which contains the granule IDs (db IDs) along with the measured parameters, the QA values, and the corresponding explanation fields before and after the update for each granule updated. Verify the history file is created under QA_HISTORY_DIR, and it contains the dbID, ParameterName, FlagName, NewQualityFlag, OldQualityFlag, NewQualityFlagExplan, and OldQualityFlagExplan		
27	Verify that the QA update utility logged the processing activities including the following: a. time stamp is included in each log entry b. the name of the request file is logged at the start of processing the update request c. Total number of granules requested.		

#	Action	Expected Result	Notes
	d. Total number of granules updated. e. Run completion status Verify in the log for the following statement : Processing request file Total number of granules updated: Total number of granules specified in the request: Total number of granules eligible to be updated after validation: RUN COMPLETED SUCCESSFULLY.		
28	Verify that the total number of granules requested is the same as the total number of granules updated. Verify in the QAUU log for both the numbers to match Total number of granules updated: Total number of granules specified in the request:		
29	Verify that the QAUU history file under /usr/ecs//CUSTOM/data/DSS/AIM/QAUUHistory contains the information about the affected granules and the log file contains total number of granules updated.		
30	Verify that an email notification is sent to the configured “Reply-to” e-mail address for the originating site, as well as to the configured internal DAAC email address, and that it contains the following information: a. The name of the request file processed. b. Total number of granules specified in the request c. Total number of granules updated. Verify the email notification is sent to email addresses specified in DAAC_EMAIL_ ADDRESSES and SITE_ReplyAddress. Verify the email contains the correct information and matches with information in the log file.		
31	Create an ESDT request with an invalid timestamp like 15:00:00:000pm Verify that the 'SOME ERRORS OCCURRED' is captured in the log		
32	Correct the timestamp error by using 15:00:00.000. Use the time range such that there are no granules associated with the request. Verify that "No non-smap granules associated with the following request data" is captured in the log.		

TEST DATA:

EXPECTED RESULTS:

1.25.4 [PVC] - Test Case 15 - Archive Check Utility (ECS-ECSTC-176)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Archive Check Utility]		#comment
2	Note: Refer to the EcDsAmArchiveCheckUtility.609 document for details.		#comment
3	Run EcDsAmArchiveCheckUtilityStart to get the usage of this utility. Usage: EcDsAmArchiveCheckUtilityStart [MODE] [-ar ArchiveRootPath] [-lo (Output Directory)] [-s | -a | -e [ESDT_LIST] | -d ["Mon DD, YYYY HH:MM [AM | PM] - Mon DD, YYYY HH:MM [AM | PM]"] | -v [VolumeGroupPath] | -o | -h | -vs [VolumeGroupPath] [-nx]		
4	Run EcDsAmArchiveCheckUtilityStart <MODE> to check consistency of the ECS data file archive and xml file archive for granules in all the volume groups with the specified ArchiveRoot (such as /stornext/snfs1/) in this mode.		
5	Verify that the utility reports the discrepancies between the archive and the database for all the volume groups with the specified ArchiveRoot. It also reports the phantom and orphan xml files for these granules.		
6	Verify that the report files are generated with detailed information about the discrepancies.		
7	Run EcDsAmArchiveCheckUtilityStart <MODE> with –e and –d options: For example: EcDsAmArchiveCheckUtilityStart DEV04 -e "MOD10C2*" -v /stornext/snfs1/DEV04/MODIS -d "Aug 27 2007 - May 05 2009".		The AmArchiveCheckUtility can can't report the missing Archive for AE_Land.002 and MOD29P1D.005 datatypes in the OPS.2014.09.24_15:22:04_f5dpl01v.notInArch

#	Action	Expected Result	Notes
			Files with missing archive records in VolumeGroup "/stornext/snfs1/OPS/AMSR It works for AE_5DSno.002 and AST_L1T.003 in EDF-OPS mode.
8	Verify that the utility reports the discrepancies between the archive and database for the specified ESDTs with the specified volume group path and date range. . It also reports the phantom and orphan xml files for these granules.		
9	Verify that the report files are generated with the detailed information about the discrepancies.		A new ncr8052052 EcDsAmArchiveCheckUtilityStart does not reported new ingested granules after removing archive files.

TEST DATA:

EXPECTED RESULTS:

2 BMGT

2.1 BMGT Automated

2.1.1 Nominal Collection Export[S-1]: Manual Export (ECS-ECSTC-397)

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.		
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>		

#	Action	Expected Result	Notes
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 --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 =	

#	Action	Expected Result	Notes
		<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> 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.		

2.1.2 ECS Collection Additional Metadata[S-01]: ECS CoordinateSystem (ECS-ECSTC-398)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S1.feature		

TEST DATA:

EXPECTED RESULTS:

2.1.3 Nominal Granule Export[S-2i]: Automatic Export: Publish Granule (ECS-ECSTC-400)

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> 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.		

2.1.4 Nominal Granule Export[S-2j]: Automatic Export: Unpublish Granule (ECS-ECSTC-401)

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.		

2.1.5 Nominal Granule Export[S-2k]: Automatic Export: Link to Browse (ECS-ECSTC-402)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_link_to_browse_S2k.feature		
5	Verify that the scenario is passed		

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.		

2.1.6 Nominal Granule Export[S-2l]: Automatic Export: Unlink from Browse (ECS-ECSTC-403)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_unlink_from_browse_S2l.feature		
5	Verify that the scenario is passed		

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.		

2.1.7 Nominal Collection Export[S-2d]: Automatic Export: Enable Collection Export (ECS-ECSTC-413)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/AutoExport_Collection_Enable.feature		
5	Verify that the scenario is passed		

TEST DATA:

Crit id	Crit ccr no	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
S-2d		1 collection to test enable collection export after collection	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
		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.		

2.1.8 Nominal Granule Export[S-2a]: Automatic Export: Ingest Granule (ECS-ECSTC-414)

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 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		

#	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/.		
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 multiple, redundant exports could result.		#comment
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 		#comment

#	Action	Expected Result	Notes
	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		
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 = <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.		

2.1.9 Nominal Granule Export[S-2b]: Automatic Export: Logically Delete Granule (ECS-ECSTC-415)

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		

#	Action	Expected Result	Notes
	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, 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

#	Action	Expected Result	Notes
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.		

2.1.10 Nominal Granule Export[S-2c]: Automatic Export: Physically Delete Granule (ECS-ECSTC-416)

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.		

#	Action	Expected Result	Notes
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 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}		

#	Action	Expected Result	Notes
	-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.		

2.1.11 Nominal Granule Export[S-2d]: Automatic Export: DFA Granule (ECS-ECSTC-417)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: features/bmgt/nominal_autoexport_dfa_granule_40_S2d.feature		
5	Verify that the scenario is passed		

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.		

2.1.12 Nominal Granule Export[S-2e]: Automatic Export: Hide a Granule (ECS-ECSTC-419)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_hide_granule_40_S2e.feature		
5	Verify that the scenario is passed		

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.		

2.1.13 Nominal Granule Export[S-2f]: Automatic Export: Restrict Granule (ECS-ECSTC-420)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: features/bmgt/nominal_autoexport_restrict_granule_40_S2f.feature		
5	Verify that the scenario is passed		

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.		

2.1.14 Nominal Granule Export[S-2g]: Automatic Export: Unrestrict Granule (ECS-ECSTC-421)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_unrestrict_granule_40_S2g.feature		
5	Verify that the scenario is passed		

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.		

2.1.15 Nominal Granule Export[S-2h]: Automatic Export: QAUpdate granule. (ECS-ECSTC-422)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_qaupd_granule_40_S2h.feature		
5	Verify that the scenario is passed		

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.		

2.1.16 Nominal Granule Export[S-2n]: Automatic Export: Move Collection (ECS-ECSTC-423)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_move_coll_40_S2n.feature		
5	Verify that the scenario is passed		

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.		

2.1.17 Nominal Granule Export[S-2o]: Automatic Export: XML Replacement (ECS-ECSTC-424)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_autoexport_xmlreplacement_40_S2o.feature		
5	Verify that the scenario is passed		
6			

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.		

2.1.18 ECS Collection Additional Metadata[S-02]: ECS GranuleSpatialRepresentation (ECS-ECSTC-425)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S2.feature		
5	Verify that the scenario is passed.		

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.		

2.1.19 ECS Collection Additional Metadata[S-03]: ISO CoordinateSystem (ECS-ECSTC-426)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S3.feature		
5	Verify that the scenario is passed		

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.		

2.1.20 ECS Collection Additional Metadata[S-04]: ISO GranuleSpatialRepresentation (ECS-ECSTC-427)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S4.feature		
5	Verify that the scenario is passed		

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.		

2.1.21 ECS Collection Additional Metadata[S-05]: backtrack (ECS-ECSTC-428)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S5.feature		
5	Verify that the scenario is passed		

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.		

2.1.22 ECS Collection Additional Metadata[S-06]: no backtrack (ECS-ECSTC-429)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S6.feature		
5	Verify that the scenario is passed		

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.		

2.1.23 ECS Collection Additional Metadata[S-07]: TwoDCoordinateSystem (ECS-ECSTC-430)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S7.feature		
5	Verify that the scenario is passed		

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.		

2.1.24 ECS Collection Additional Metadata[S-08]: no TwoDCoordinateSystem (ECS-ECSTC-431)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S8.feature		
5	Verify that the scenario is passed		

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.		

2.1.25 ECS Collection Additional Metadata[S-09]: Product Specific Attributes (PSAs) (ECS-ECSTC-432)

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-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 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',		

#	Action	Expected Result	Notes
	'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.		

2.1.26 ECS Collection Additional Metadata[S-10]: DIF ID (ECS-ECSTC-433)

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.		
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		#comment

#	Action	Expected Result	Notes
	ID for the collection for which one was provided, and contains no DIF ID for the other collection.		
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.		

2.1.27 ECS Collection Additional Metadata[S-12]: null ECS CoordinateSystem (ECS-ECSTC-434)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S12.feature		
5	Verify that the scenario is passed		

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.		

2.1.28 ECS Collection Additional Metadata[S-13]: null ECS GranuleSpatialRepresentation (ECS-ECSTC-435)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Ecs_Collection_Additional_Metadata_S13.feature		
5	Verify that the scenario is passed		

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.		

2.1.29 Manual Export - Collection Deletes (ECS-ECSTC-438)

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	login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		

#	Action	Expected Result	Notes
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Manual_Collection_Deletes.feature		
5	Verify that the scenario is passed		

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.		

2.1.30 Manual Export - Collection Full Update (ECS-ECSTC-439)

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
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		#comment

#	Action	Expected Result	Notes
	the full granule metadata for that granule.</i>		
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.		

2.1.31 Manual Export - Granule Insert (ECS-ECSTC-440)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/ManualExport_Granule_Insert.feature		
5	Verify that the scenario is passed		

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 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		

2.1.32 Manual Export - Last Update Time (ECS-ECSTC-441)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/ManualExport_LastUpdate.feature		
5	Verify that the scenario is passed		

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.		

2.1.33 Nominal Collection Export[S-2a]: Automatic Export: Insert Collection (ECS-ECSTC-443)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/AutoExport_Collection_Insert.feature		
5	Verify that the scenario is passed		

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 automatic export			1 Collection to be installed (C2)		/sotestdata/DROP_802/BE_82_01/Criteria/010/010_2_A	

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.		

2.1.34 Nominal Granule Export[S-1]: Manual Export (ECS-ECSTC-445)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/nominal_granule_manual_export_40_S1.feature		
5	Scenario should pass.		

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	
40S-2f			1 granule to restrict (g6)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_F	
40S-2g			1 granule to unrestrict (g7)	MYD14.005			/sotestdata/DROP_802/BE_82_01/Criteria/040/040_2_G	

Crit id	Crit ccrno	Test Data Description	Data Type Requirements	Metadata Requirements	Volume Requirements	Size Requirements	Data Location	Readiness Status
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 browse (b2) to be unlinked	1 browse granule linked to a science granule 1 browse granule linked to a different science			/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
				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.		

2.1.35 Manual Export - Insert Time (ECS-ECSTC-640)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/ManualExport_Insert_Time.feature		
5	Verify that the scenario is passed		

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.		

2.1.36 Manual Export - "Fake" Collection Delete (ECS-ECSTC-641)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Manual_Collection_Force_Delete.feature		
5	Verify that the scenario is passed		

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.		

2.1.37 Nominal Collection Export[S-2c]: Automatic Export: Update Collection (ECS-ECSTC-642)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/AutoExport_Collection_Update.feature		
5	Verify that the scenario is passed		

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.		

2.1.38 Manual Export - Collection Full Update (ECS-ECSTC-644)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Manual_Collection_Full_Update.feature		
5	Verify that the scenario is passed		

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.		

2.1.39 Manual Export - Granule Insert (ECS-ECSTC-645)

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
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		

#	Action	Expected Result	Notes
	<dbuser> -password <passwd> -geoidfile /path/to/geoids_g2_g3.txt		
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

#	Action	Expected Result	Notes
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 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		

#	Action	Expected Result	Notes
	<g4_granuleid>,<g5_granuleid>,<g6_granuleid>		
45	V-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		#comment
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		

2.1.40 Manual Export - Collection Inserts (ECS-ECSTC-647)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Manual_Collection_Inserts.feature		
5	Verify that the scenario is passed		

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	3 collections					

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

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).		

2.1.41 Manual Export - Collection Deletes (ECS-ECSTC-648)

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 = <C2_VERSIONID>	esdtstate != 'installed'	

#	Action	Expected Result	Notes
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>		
26	Ensure collection C5 is "installed": select esdtstate<br	esdtstate == 'installed'	

#	Action	Expected Result	Notes
	/>from DsGeESDTConfiguredType where configuredname = '<C5_SHORTNAME>' and versionid = <C5_VERSIONID>		
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.		
41	S-2 Repeat S-1, but specify that the operation should be an insert only		#comment

#	Action	Expected Result	Notes
	export.		
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.		
60	Verify the TCP proxy (or mock ECHO) log shows no export of collection C5.		

#	Action	Expected Result	Notes
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.		
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.		

2.1.42 Manual Export - Granule Delete (ECS-ECSTC-649)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/ManualExport_Granule_Delete.feature		
5	Verify that the scenario is passed		

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		

2.1.43 Long Form Verification - Collection (ECS-ECSTC-650)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Longform_Collection_Export.feature		
5	Verify that the scenario is passed		

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 .		

2.1.44 Long Form Verification - Granule (ECS-ECSTC-651)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Longform_Granule_Export.feature		
5	Verify that the scenario is passed		

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] .		

2.1.45 Long Form Verification - Insert Time (ECS-ECSTC-652)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Longform_InsertTime_Export.feature		
5	Verify that the scenario is passed		
6			

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.		

2.1.46 Long Form Verification - Last Update Time (ECS-ECSTC-653)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Longform_LastUpdate_Export.feature		
5	Verify that the scenario is passed		

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.		

2.1.47 Long Form Verification - Granule Updates (ECS-ECSTC-654)

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	Login as cmshared on f5oml01v and set view to AUTOTEST		
2	setenv MODE OPS		
3	cd /ecs/formal/COMMON/automation/autotest directory		
4	Run the following cucumber feature file: cucumber features/bmgt/Longform_GranuleUpdates_Export.feature		
5	Verify that the scenario is passed		
6			

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).		

2.1.48 Nominal Collection Export[S-2c]: Automatic Export: Update Collection (ECS-ECSTC-412)

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 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		

#	Action	Expected Result	Notes
	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.		
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<br	

#	Action	Expected Result	Notes
		/>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> 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 numbers with no leading zeroes.		#comment
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.		

2.2 BMGT Modified

2.2.1 Long Form Verification - Invocation via Command Line - b) Granules in Granule File (ECS-ECSTC-399)

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 = ${VERSIONID} If the collections were newly enabled for export in this step, wait for them and their granules to be exported.		
7	Ensure ECHO has the test collections' metadata. For each of C1, C2, C3, curl -k -H Echo-Token:${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

#	Action	Expected Result	Notes
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 '?semantic_diff=true'. For example, PUT /catalog-rest/providers/DEV08/granules/SC%3AD5TES1.001%3A87900?semantic_diff=true HTTP/1.1		
18	Create another file with invalid granuleids		
19	EcBmBMGTManualStart --long --metg -gf /path/to/invalid_granules.txt		
20	Verify that the TCP proxy log does not have any PUTs		
21	Verify that the manual driver logs do not have any exceptions		

TEST DATA:

refer to test case 669

EXPECTED RESULTS:

	V	492	1	Verify that the operation in S-1 results in the following exports: a. Single HTTP PUT. b. Single HTTP PUT. c. HTTP PUT for each granule in the collection. d. HTTP PUT for each granule in the collection. e. HTTP PUT for each specified collection. f. HTTP PUT for each specified collection. g. HTTP PUT for each granule and collection in		

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

2.2.2 Incremental Verification - Nominal (ECS-ECSTC-404)

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 = '<C10_SHORTNAME>' and versionid = <C10_VERSIONID>)		
8	Ensure collections C1..C1 exist in ECHO.		
9	Ensure all test granules exist in ECHO.		
10	Update bg_collection_status set lastupdateverified = now() where collectionid in ( )		
11	Ensure a TCP proxy or mock ECHO is capturing BMGT traffic.		
12			
13	Setup		#comment
14	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
15	Ensure that the collections C1..C10 have a large number of granules in AIM (1000 in the EDF; about 100 000 in the PVC)		

#	Action	Expected Result	Notes
16	Configure BMGT.Incremental.Duration = 10 (the interval is in days).		
17	Configure BMGT.Verification.MaxGranules to 1000 in the EDF or 10 000 in the PVC. Also set maxgranulestoverify = 50 per collection in bg_collection_configuration table		
18	In the BMGT GUI, reset verification status for all collections to be unverified or 0%.		
19	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
20	In the BMGT GUI System Status tab, verify that the overall verification percentage is displayed as 0%		
21	In the BMGT GUI System Status tab, verify that verification percentage is displayed as 0% for each collection and group in the mode		
22	S-2 Initiate incremental verification repeatedly until there are no more granules to verify.		#comment
23	Ensure that no other operations are going on in the system which would cause the update of any granules.		
24	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.		
25	Ensure that there are no ingest, delete, or update operations ocurring in the mode		
26	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.		
27	./EcBmBMGTManualStart <MODE> --incremental		
28	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
29	Verify that for the first two incremantal iterations, any granule that falls between min(lastupdateverified) and min(lastupdateverified) in bg_collection_status table + 10 * (BMGT.Incremental.Duration) in bg_configuration_property		
30	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.		
31	V-3 Inspect the log file to verify that for each Incremental verification		#comment

#	Action	Expected Result	Notes
	initiation, no more than the configured number of granules is enqueued.		
32	Verify that for each incremental iteration, the total number of granules is less than or equal to BMGT.Verification.MaxGranules		
33	Verify that for each incremental iteration, the total number of granules for each collection is less than or equal to 10 * bg_collection_configuration.maxgranulestoverify for the given collection.		
34	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
35	Verify that the bmgt log is written under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log		
36	Verify that the bmgt log indicates the time when the incremental verification was started.		
37	Verify that the bmgt log indicates the time when all request have been added to the queue		
38	Verify that the bmgt log indicated the time when verification export was started.		
39	Verify that the bmgt log indicated the time when verification export was completed.		
40	Verify that the bmgt log indicates the time span of the verified granules in the iteration.		
41	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
42	Verify the overall verification percentage increases after the first two iterations		
43	Verify the verification percentage for collections C1 - C10 increases after the first two iterations		
44	Verify the overall verification percentage = 100% after running multiple incremental iterations		
45	Verify that the verification percentage for Collections C1..C10 = 100% after		

#	Action	Expected Result	Notes
	running multiple incremental iterations.		
46	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
47	Save the TCP proxy log, and start a new one: Stop the TCP proxy. Move the log to a new name. Start the proxy.		
48	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		
49	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		
50	Verify the number of granules eligible for export per collection matches the number found in the TCP proxy log: diff -w eligible_counts.txt exported_counts.txt		
51	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
52	Verify that the TCP proxy shows that each verification export is a single HTTP PUT request.		
53	Verify that the TCP proxy shows that the request has the HTTP query parameter "semantic_diff=true" indicating it is for verification purposes.		
54	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

#	Action	Expected Result	Notes
55	Verify that the TCP proxy shows a HTTP PUT request with Collection metadata for Collections C1..C10.		
56	Verify that the TCP proxy shows that the request is for verification purposes.		
57	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
58	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.		
59	a) In the public datapool		#comment
60	Ingest Science granules g1..g10 into Collection C1 with default publishing on.		
61	b) Have browse links		#comment
62	Ingest Science granules g11..g20 into Collection C2 with default publishing on.		
63	c) Are restricted.		#comment
64	Choose or create a restriction flag: select * from dsmdrestrictionflag -OR- insert into dsmdrestrictionflag values(128, 'BE_82_01 Crit 520')		
65	Add the restriction flag to granules g21, g22: insert into dsmdgranulerestriction values(<GRANULE_ID>, 128)		
66	S-4 Initiate another incremental verification export.		#comment
67	./EcBmBMGTManualStart -mode <MODE> --incremental		
68	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
69	Verify that there are no granules queued for verification export		
70	S-5 Resume Automatic export and allow it to pick up and export the events in S-3.		#comment
71	Resume the Event queue in the BMGT GUI.		
72	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		
73	Verify that the bmgt logs also include the export of each of the granules in		

#	Action	Expected Result	Notes
	C1..C10 in Collection C1.		
74	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		
75	Verify that the bmgt logs also include the export of each of the granules in g11..g20 in Collection C2.		
76	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		
77	Verify that the bmgt logs also include the export of each of the granules in g21..g22 in Collection C3.		
78	S-6 Initiate another incremental verification export.		#comment
79	./EcBmBMGTManualStart <MODE> --incremental		
80	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
81	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		
82	Verify that the bmgt logs also include the export of each of the granules in C1..C10 in Collection C1.		
83	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		
84	Verify that the bmgt logs also include the export of each of the granules in C11..C20 in Collection C2.		
85	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		
86	Verify that the bmgt logs also include the export of each of the granules in g21,.g22 in Collection C3.		
87	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
88	Verify that for each incremental iteration, the total number of granules is less than or equal to BMGT.Verification.MaxGranules		
89	Verify that for each incremental iteration, the total number of granules for each collection is less than or equal to		

#	Action	Expected Result	Notes
	bg_collection_configuration.maxgranulestoverify for the given collection.		
90	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
91	Verify that the bmgt log is written under /usr/ecs/<MODE>/CUSTOM/logs/EcBmBMGTManualDriver.log		
92	Verify that the bmgt log indicates the time when the incremental verification was started.		
93	Verify that the bmgt log indicates the time when all request have been added to the queue		
94	Verify that the bmgt log indicated the time when verification export was started.		
95	Verify that the bmgt log indicated the time when verification export was completed.		
96	Verify that the bmgt log indicates the time span of the verified granules in the cycle.		
97	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
98	Verify the overall verification percentage = 100%		
99	Verify that the verification percentage for Collections C1..C10 = 100%		
100	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
101	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
102	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		

#	Action	Expected Result	Notes
	granule exports in the collection.		
103	If there are extra collection exports in the verification, these can be ignored.		
104	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
105	Verify that the TCP proxy shows that each verification export is a single HTTP PUT request.		
106	Verify that the TCP proxy shows that the request has the HTTP query parameter "semantic_diff=true" indicating it is for verification purposes.		
107	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
108	Verify that the TCP proxy shows a HTTP PUT request with Collection metadata for Collections C1...C10.		
109	Verify that the TCP proxy shows that the request is for verification purposeses (the url will contain a query parameter "xmldiff=true").		
110	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
111	Verify that the bmgt logs and database show when the request was generated for each verification request.		
112	Verify that the bmgt logs and database show when the metadata was generated for each verification request.		
113	Verify that the bmgt logs and database show when the export was sent to ECHO.		
114	Verify that the bmgt logs and database show when the response was received for each verification request.		
115	V-14 Verify that the BMGT GUI displays the completed export events, indicating that they were the result of an incremental verification export.		#comment

#	Action	Expected Result	Notes
	Verify that it indicates that the requests were successfully exported and indicates the time of export as well as granule or ID.		
116	In the BMGT GUI export request tab, filter the request to veiw only those items on the INCR queue		
117	Verify that each request is listed in the SUCCESS state.		
118	Verify that each request has the associated granule or collection id listed.		
119	Verify that each request has its completion time listed.		
120	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
121	In the BMGT GUI Export Request tab, filter to view only requests on the INCR queue.		
122	select the "Batch Job Summary" sub tab		
123	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items exported for verification.		
124	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items successfully verified (suiccess column).		
125	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).		
126	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.		
127	Verify that the "Batch Job Summary" sub tab lists for each incremental iteration, or batch, the number of items which were skipped.		
128	Ingest granules for any test collection		
129	Run incremental verification with the -metg option . ./EcBmBMGTManualStart -mode --incremental		
130	Verify that only granules metadata is exported.		
131	Verify that no collection metadata is exported		

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.		

2.2.3 MISR Browse Links - Science Before Browse (ECS-ECSTC-406)

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. 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	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:		#comment

#	Action	Expected Result	Notes
	/tools/common/test/BE_82_01/bin/xpath</i>		
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 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		

#	Action	Expected Result	Notes
	"//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 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).		#comment

#	Action	Expected Result	Notes
	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.		
13	S-2 Ingest the MISR Level 1, Level 2, and MIB2GEOP granules referenced in step S-1.		#comment
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.		

#	Action	Expected Result	Notes
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.		
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		

#	Action	Expected Result	Notes
	"//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>		
42	Re-ingest MISR L1 and L2 science granules that was used in step 15		
43	Repeat the verification steps. Ensure that the new MISR exported includes the MISR Browse URLs		
44	Re-ingest the MISR Browse granules that was ingested in step 26		
45	Verify that BMGT exports the new MISBR urls for the new MISR science granules		
46	Verify that the old MISR granules do not export MISRBR URLs		

TEST DATA:

Crit id	Crit ccrno	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			/sotestdata/DROP_802/BE_82_01/Criteria/110	

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

2.2.4 MISR Browse Links - Browse Before Science (ECS-ECSTC-407)

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 = 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/Or		

#	Action	Expected Result	Notes
	bitNumber/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 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		#comment

#	Action	Expected Result	Notes
	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 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		#comment

#	Action	Expected Result	Notes
	by the automatic polling process.		
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 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				

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

2.2.5 Contact ECHO Errors (ECS-ECSTC-409)

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	Ensure a mock ECHO or TCP proxy is capturing BMGT traffic . login as cmshared on f5oml01v. cd to /tools/common/test/BE_82_01/proxy/. run proxy_start		
6	Setup		#comment
7	Ensure a test Collection C1 has been installed in the mode. (ESDT verification script)		
8	Verify Collection C1 is enabled for Collection and Granule Export.		
9	With the dispatcher running, run proxy_stop, then proxy_start 500 to start a proxy that returns HTTP 500 errors.		
10	Make sure that the configuration for log4j.properties file under /usr/ecs//CUSTOM/data/BMGT/config/dispatcher is as follows: log4j.logger.emailLogger=info,notifier,notifierLog		
11	./EcBmBMGTManualStart <MODE> --metc --metg -c <C1_shortname>.<C1_versionid>		
12	Verify the BMGT GUI > Export Requests tab shows the request from S-1		

#	Action	Expected Result	Notes
	as PENDING.		
13	Verify in the BMT GUI, when filtering the requests with CompletionTime isNull, displays all the pending requests		
14	Verify that the dispatcher shows it is paused in the GUI		
15	Verify that no email is sent to the configured email address		
16	Wait for few minutes. Verify that the GUI display the error message in red, and is visible across all tabs.		
17	Repeat steps 10 through 17 using proxy_start 503 instead of proxy_start 500		
18	run proxy_stop to stop the proxy and restart proxy with proxy_start		
19	Veify that the requests are automatically exported		
20	Verify that the requests are successful		
21	Modify the log4j.properties file again : log4j.logger.emailLogger=DEBUG,notifier,notifierLog		
22	do proxy_stop and proxystart 500 and bounce the dispatcher		
23	./EcBmBMGTManualStart <MODE> --metc --metg -c <C1_shortname>.<C1_versionid>		
24	Verify in the BMT GUI, the Activity associated with the request from the previous step is marked as PENDING.		
25	Verify the BMGT GUI > System Status tab shows the Dispatcher is paused.		
26	Verify that an email is sent to the configured email address		
27	Verify that the email indicates the ECHO operator needs to be contacted		#comment
28	Wait for few minutes. Verify that the GUI display the error message in red, and is visible across all tabs.		
29	Verify that the bmgt GUI lists the export requests for granule g1 and Collection C1 as pending		
30	Verify in the BMT GUI, when filtering the requests with CompletionTime isNull, displays all the pending requests		
31	Verify that the requests are automatically exported		
32			

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
			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.		

2.2.6 Manual Export - Science granule with not yet published Browse (ECS-ECSTC-473)

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 granuleexportflag = 'Y', collectionexportflag = 'Y' where shortname = <SHORTNAME> and versionid = <VERSIONID>;		
5	Ensure granule g11 is in AIM: select shortname, versionid, granuleid from amgranule where localgranuleid = ${LOCALGRANULEID}		
6	Ensure granule g11 is associated with a browse granule b1: select * from ambrowsegranulexref where granuleid = ${GRANULEID} 		
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 not running. On the BMGT host: ps auxww | grep $MODE | grep EcBmAuto If needed, stop the		

#	Action	Expected Result	Notes
	Auto driver: ./EcBmBMGTAutoStop $MODE		
10	Ensure that the browse granule is published. Check ambrowse table. isordeonly should be null or 'B'		
11	Do a Manual Export of the science granule		
12	Verify that the export is successful and there is a PUT in the tcp.log for the granule		
13	Verify that the full metadata for the science granule is exported		
14	Verify granule g11's exported metadata contains a browse linkage URL: xpath "//OnlineResource[Type='BROWSE']" g11.xml		
15	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}.		
16	Save the original state of ambrowse granule. Update Ambrowse table. set isorderonly = 'H'		
17	do a manual export of g11		
18	Ensure the export succeeds . Check the tcp log for a HTTP PUT for g11		
19	Check the metadata exported. Only the science granule url should be exported		
20	No browse url should be exported		
21	Reset isorderonly flag in ambrowse back to the original value		

TEST DATA:

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.		

2.2.7 Manual Export - Collection Deletes (ECS-ECSTC-665)

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		

#	Action	Expected Result	Notes
	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 = <C2_VERSIONID>	esdtstate != 'installed'	
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>		

#	Action	Expected Result	Notes
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>		
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><br		

#	Action	Expected Result	Notes
	/><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.		
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.		

#	Action	Expected Result	Notes
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.		
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 ----------+----------+----------+----------+----------+----------	

#	Action	Expected Result	Notes
		--- 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.		

2.2.8 Short Form Verification - Exclude Pending Exports (ECS-ECSTC-673)

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.		

2.2.9 ISO Nominal Granule Export: Granule Ingest (ECS-ECSTC-674)

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 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		#comment

#	Action	Expected Result	Notes
	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: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		
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.		

2.2.10 Alert After Numerous Retries (ECS-ECSTC-675)

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 (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		

#	Action	Expected Result	Notes
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 ECHO interrupted in the middle (e.g. by bringing down a proxy) 4. ECHO terminates a persistent connection from ECHO.		#comment
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

#	Action	Expected Result	Notes
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 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.		#comment
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		

#	Action	Expected Result	Notes
	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.		

3 CLS

3.1 OBSOLETE - Order Status GUI: Display Order Status, OD_S6_05. Criterion 130 (ECS-ECSTC-191)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Request the status of a single, completed order that was handled by the OMS and that contains at least two requests by specifying a valid Order Id and user contact email address. Each request in the order should contain at least 100 granules that have NOT been processed by HEG or the external subsetter.		
2	Verify the following:		#comment
3	a. Order Id is correctly displayed.		
4	b. Submission date/time is correctly displayed.		
5	c. Order state is correctly displayed.		
6	d. Order completion date/time is correctly displayed.		
7	e. Request information (Request Id, request state, media type, number of granules, request completion date/time) is correctly displayed		
8	f. Order state and request states are presented in terms that an end user can understand.		
9	g. Order and request information are properly segmented across html pages in accordance with the user-configurable number of items per page.		
10	h. There is an indication that additional request details are available.		

TEST DATA:

EXPECTED RESULTS:

3.2 OBSOLETE - Order Status GUI: Display Order Status based on History Range, 0D_S6_05, Criterion 180 (ECS-ECSTC-192)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Request order history by specifying a starting and ending date, user contact email address, and one valid Order Id for the user. Ensure that at least 100 orders are displayed that represent a mix of orders for which request details are available for some orders and not available for others. Ensure that the date range requires the Order Status Interface to retrieve information from both the Order Manager operational tables and archive tables. Ensure that at least one of the orders contains granules that were processed by the external subsetter and one of the orders contains granules that were processed by HEG.		
2	Verify that the correct orders are returned and sorted by submission date/time.		
3	For each order verify the following:		#comment
4	a. Order Id is correctly displayed.		
5	b. Submission date/time is correctly displayed.		
6	c. Order state is correctly displayed.		
7	d. Order completion date/time is correctly displayed.		
8	e. Request information (Request Id, request state, media type, number of granules, request completion date/time) is correctly displayed, including the processing description for the granule to be processed by the external subsetter, in a manner which should be generally understandable by the user.		
9	f. Order state and request states are presented in terms that an end user can understand.		
10	g. Order and request information are properly segmented across html pages in accordance with the user-configurable number of items per page.		
11	h. An indication is provided when request details are not available for an order.		

TEST DATA:

EXPECTED RESULTS:

3.3 Validating HDF data files (ECS-ECSTC-193)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Get two HDF files with known differences.		
2	Use the HDIFF tool to compare them.		
3	Verify that the differences are detected.		

TEST DATA:

EXPECTED RESULTS:

4 CSS

4.1 CSS Registry (ECS-ECSTC-194)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Back up the registry tree.		
2	Remove a component from the mode.		
3	Bring up the registry GUI and verify that the component vanishes from the registry GUI.		
4	Re-install and configure the component.		
5	Run the registry population utility to populate the values for that component.		
6	Bring up the registry GUI and display the configured parameters and values for the component.		
7	Return the backed up values to the registry		

TEST DATA:
Any current data type

EXPECTED RESULTS:

5 DMS

5.1 OBSOLETE- Test Case 3 - Single Granule External Processing Order (ECS-ECSTC-195)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit an order through EWOC with a granule that requires spatial subsetting.		#Used to be done through HSA. HSA is obsolete.
2	Verify the following occurred:		#comment
3	a) A Submit message was received by the spatial subsetting external processor.		
4	b) The Submit message contained the order ID for this order and the request ID belonging to subsetting request.		
5	c) The request state has been updated to ‘pending’		
6	d) A secondary Submit message was received by EWOC from an External Processor.		
7	e) An order and request were registered, corresponding to the order submitted.		
8	f) The order is not marked as an ECHO order.		
9	g) The request is not marked as an External Processing request.		
10	h) A Submit response is received at the client that contains the order ID recorded in the database for this request and a successful status.		
11	i) Close Provider Order was not invoked in ECHO for the secondary Submit sent by an External Processor.		
12	j) The subsetted granule is shipped to the customer via FTP PULL and that the granule is subsetted correctly.		

TEST DATA:

EXPECTED RESULTS:

5.2 Test Case 1 - Single Granule Order (ECS-ECSTC-196)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit a single granule order through EWOC with the following media types: HTTPS, scp.		
2	Verify that the order is successfully registered in the OMS.		

TEST DATA:

EXPECTED RESULTS:

5.3 Test Case 2 - Cancelled and Failed Granules (ECS-ECSTC-197)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit an order through EWOC and then cancel the order in the OMS GUI.		
2	After the order has been canceled, verify that the EWOC sends an order update status message to ECHO indicating that the order has been canceled within 5 minutes of order status update in ECS.		
3	Repeat above test but this time fail the order in the OMS GUI.		

TEST DATA:

EXPECTED RESULTS:

5.4 Test Case 4 - Single Granule Order with invalid ESDT or Granule ID (ECS-ECSTC-198)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit a single granule order through EWOC for a granule that is not available in ECS.		
2	Verify that EWOC rejects the submission of the order specifying which granules failed validation.		

TEST DATA:

EXPECTED RESULTS:

5.5 Test Case 5 - Double granule order containing simple items with different media requirements (ECS-ECSTC-199)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit an order through EWOC for two products as follows,		#comment
2	a) Product 1 should be a simple granule of media type HTTPS. This is referred to as granule 1		
3	b) Product 2 should be a simple granule of media type SCP. This is		

#	Action	Expected Result	Notes
	referred to as granule 2		
4	Verify that the following occurred,		#comment
5	a) An order was registered in OMS DB corresponding to the order submitted.		
6	b) Two requests were registered with OMS corresponding to the order submitted. One request for granule 1 and one request for 2		

TEST DATA:

EXPECTED RESULTS:

5.6 Test Case 6 - Multiple Granules Order (ECS-ECSTC-200)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit order through EWOC for more than one product as follows,		#comment
2	a) Product 1 should be a simple granule of media type FTP PUSH. This is referred to as granule 1		
3	b) Product 2 should be a simple granule of media type SCP with a different set of push parameters (excluding user string) to product 1. This is referred to as granule 2		
4	e) Product 3 should be a simple granule of media type HTTPS. This is referred to as granule 3		
5	Verify that the following occurred,		#comment
6	a) An order was registered with OMS corresponding to the order submitted.		
7	b) Three requests were registered with OMS corresponding to the order submitted.		
8	d) One request in OMS contains 1 granule with the correct processing options for granule 1.		
9	e) One request in OMS contains 1 granule with the correct processing options for granule 2.		

#	Action	Expected Result	Notes
10	h) One request in OMS contains 1 granule with the correct processing options for granule 3.		
11	Verify that the following occurred,		#comment
12	All granules are shipped to the customer via the correct distribution method and have been processed (if applicable) correctly.		

TEST DATA:

EXPECTED RESULTS:

5.7 Test Case 7 - Duplicate Requests (ECS-ECSTC-201)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure EWOC to check for duplicate requests by modifying the check.duplicate property to YES in the EcDmEwoc.properties.		Restart the EWOC instance via the tomcat manager for the change to take effect.
2	Submit an order through EWOC for a regular single granule with distribution media type of FtpPush.		
3	Submit the same order again after a few seconds.		
4	Verify that the OMS shipped only one of the orders.		
5	Repeat the test after changing check.duplicate property to NO.		Restart the EWOC instance via the tomcat manager for the change to take effect.
6	Verify that the OMS shipped both orders.		

TEST DATA:

EXPECTED RESULTS:

5.8 [OBSOLETE] Test Case 8 - Single Granule Order using the ASTER GDEM Form - (Needs to be rewritten) (ECS-ECSTC-202)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit a single granule order using the ASTER_GDEM form through Reverb.		Reverb was replaced by EDSC and PUMP was replaced by MMT
2	Login to PUMP where the provider is set up (e.g. https://api-test.echo.nasa.gov/pump).		
3	Go to Provider Context and choose appropriate provider context.		
4	Click on Data Management and Option Definition to add the new ASTER_GDEM form.		
5	Click on Option Assignments to select the collection and assign the ASTER_GDEM form to the collection.		
6	Go to Reverb and search for the collection that was assigned the new form above.		
7	Add granules to be ordered to the shopping cart.		
8	When choosing options for granules, select the ASTER_GDEM form and choose a value in the intended usage field.		
9	Submit the order.		
10	Verify that the intended usage field is correctly populated in OmRequestGranule table.		
11	Verify the EMS Dataset Extraction Utility provides output of IntendedUsage values in flat file that is to be sent on to EMS.		
12	Run the following EMS extract utility command with or without the -v flag (verbose) with dates entered greater than 24-hour period, ie, one week apart: The time period specified should include the orders that have IntendedUsage specified for GDEM scenario. EcDbEMSdataExtractor.pl -m <MODE> -s "<start date>" -e "<end date>" -v -x DistFTP -o		

#	Action	Expected Result	Notes
13	Verify that output dataset file is created for dataset and date range and that the file includes the IntendedUsage values for the orders.		

TEST DATA:
ASTER GDEM

EXPECTED RESULTS:

5.9 [OBSOLETE] Test Case 9 - Single Granule Order with associated PH, QA, Browse Granules - (Needs to be rewritten) (ECS-ECSTC-203)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit a single granule order, using the form that enables ordering associated granules through Reverb.		Reverb was replaced by EDSC and PUMP was replaced by MMT
2	Login to PUMP where the provider is set up (e.g. https://api-test.echo.nasa.gov/pump).		
3	Go to Provider Context and choose appropriate provider context.		
4	Click on Data Management and Option Definition to add and associate the PH/QA ordering form.		
5	Click on Option Assignments to select the collection and assign the form to the collection.		
6	Go to Reverb and search for the collection that was assigned the new form above.		
7	Add granules to be ordered to the shopping cart.		
8	When choosing options for granules, select the new form and select order PH, order QA and order Browse field.		
9	Submit the order.		
10	Verify that the associated granules are ordered and delivered in OMS.		

TEST DATA:

EXPECTED RESULTS:

6 DPL INGEST

6.1 Test Case 1 - Use the DPL Ingest GUI to configure Data Providers (ECS-ECSTC-244)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies Data Providers can be configured using the DPL Ingest GUI		#comment
2	[View Data Providers]		#comment
3	As the ‘ingest admin’ operator, navigate to the Data Provider page on the Data Pool Ingest GUI.		
4	Verify that all provider information in the INGST CI database appears correctly on the Data Provider page.		
5	[Configure Data Providers]		#comment
6	Edit existing provider information (if any) and define new providers to fulfill the following requirements: At least one data provider should have an FTP notification method, at least one data provider should have an scp notification method, at least one data provider should have an http notification method, At least one data provider should have an SFTP notification method, at least one data provider should have an email notification method, at least one data provider should have a combination FTP/email notification method, and at least one data provider should have a combination scp/email notification method. and at least one data provider should have a combination http/email notification method. Scp type/cipher combinations to include in the test are: F-secure/None; OpenSSH/aes128; OpenSSH/3des. At least one data provider should have a combination SFTP/email notification method, At least one provider must use active		

#	Action	Expected Result	Notes
	mode. At least one should use passive mode.		
7	Verify that the Data Pool Ingest GUI allows the ‘ingest admin’ operator to edit data provider information.		
8	Verify that the Data Pool Ingest GUI prompts the operator to confirm changes to existing data providers before saving this information.		
9	Verify that the ingest admin operator has the authorization to define new data providers (i.e., has access to the data provider pages on the Data Pool Ingest GUI, and that information entered by this operator is stored in the database.)		
10	Verify that the Data Pool Ingest GUI allows all information in S-DPL-16110 to be entered.		
11	Verify that all of the notification methods in S-DPL-16150 can be entered or selected on the Data Pool Ingest GUI, as appropriate for the selected transfer method as per S-DPL-16110.		
12	For one provider with an FTP Notification method, verify that the Data Pool Ingest GUI allows the related FTP information to be entered (FTP host, destination directory, login id, password)		
13	Verify that the FTP password entered is not shown or stored in the clear.		
14	Verify that the FTP password entered is not shown in the Data Pool Ingest GUI log.		
15	For one provider with an email notification method, verify that the Data Pool Ingest GUI allows the related email address to be entered.		
16	Verify that the Data Pool Ingest GUI prompts the operator to confirm definitions of new data providers before saving this information.		
17	For one provider with an scp notification method, verify that the Data Pool Ingest GUI allows the related scp information to be entered (scp host, destination directory, login id, password, scp type, and cipher).		
18	Verify that the scp password entered is not shown or stored in the clear.		
19	Verify that the scp password entered is not shown in the Data Pool Ingest GUI log.		

TEST DATA:

Any current data types

EXPECTED RESULTS:

6.2 Test Case 2 - Use the DPL Ingest GUI to configure Polling Locations (ECS-ECSTC-245)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies Polling Locations can be configured using the DPL Ingest GUI		#comment
2	[Configure Polling Locations]		#comment
3	As the Ingest Admin operator use the Data Pool Ingest GUI to define all polling locations that will be used for testing this ticket. (Table of data provider to polling location mappings, and related S-DPL-16230 information for each polling location TBS as part of ITP). At least one polling location should have an FTP polling method, at least one polling location should have a local polling method, and at least one polling location should have an scp polling method, at least one polling location should have an http polling method, and at least one polling location should have an SFTP polling method. At least one polling location using FTP shall be for a provider using local transfers.		
4	Verify that the ingest admin operator has the authorization to define the polling locations (i.e., has access to the polling location pages on the Data Pool Ingest GUI, and that information entered by this operator is stored in the database.)		
5	Verify that the Data Pool Ingest GUI allows all information in S-DPL-16230 to be entered.		
6	Verify that all of the polling methods in S-DPL-16250 can be entered or selected on the Data Pool Ingest GUI.		
7	Verify that the Data Pool Ingest GUI prompts the operator to confirm definitions of new polling locations before saving this information.		
8	[Edit Polling Locations]		#comment
9	For one polling location, as the ‘ingest admin’ operator, edit all of its existing configuration parameters. (NOTE: After this criterion is complete, values of these configuration parameters should be reset to appropriate values for processing all PDRs in criterion 300).		
10	Verify that the Data Pool Ingest GUI allows the ‘ingest admin’ operator to edit polling location information.		

#	Action	Expected Result	Notes
11	Verify that the Data Pool Ingest GUI prompts the operator to confirm changes to existing polling locations before saving this information.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.3 Test Case 5 - View and modify the DPL Ingest GUI configuration (ECS-ECSTC-248)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies DPL Ingest GUI configuration can be viewed and modified correctly.		#comment
2	[View collection configuration]		#comment
3	As an ‘ingest admin’ operator, use the Data Pool Ingest GUI to list all data types.		
4	Verify that ”Ignore Validation Warnings” and “Public in Data Pool” are displayed for each ESDT.		
5	Verify that the Data Pool Ingest GUI provides the operator a method to quickly select or scan for a subset of existing Data Pool collections without having to enter the full ESDT name and version.		Since release 8.2, newly installed data types will be automatically added to the DPL Ingest InDataType table and visible in 'Data Types with DPL Ingest Configurations' table.
6	[Edit collection configuration]		#comment
7	As an ‘ingest admin’ operator, use the Data Pool Ingest GUI to edit publication policy configuration parameters for a Data Pool collection.		
8	Verify that the edited configuration parameters are correctly updated in the database.		

#	Action	Expected Result	Notes
9	[Reconfigure data type parameters.]		#comment
10	Log into the DPL Ingest GUI as an ‘ingest admin’ operator, and configure a data type (ESDT and Version) to be published in the public Data Pool upon insert.		
11	Submit several PDRs for the data type configured above. There needs to be a sufficient number of PDRs and granules such that some granules will complete ingest before the configuration change made during the test (see next step), and some granules will not start ingest until after the configuration change has been applied.		
12	After the first few granules completed ingest, re-configure the data type via the DPL Ingest GUI such that the data type no longer will be published in the Data Pool. Also change the minimum retention period sufficiently long so that the ingested granules are not cleaned up immediately after archiving, to allow time for the verification steps below.		
13	Verify that the first few granules that were ingested before the re-configuration are queued with the Data Pool Insert Service for insertion into the public Data Pool area.		
14	Verify that all granules that started ingest one minute or more after the re-configuration are inserted into the non-public Data Pool area, and are NOT queued with the Data Pool Insert Service for insertion into the public Data Pool area. NOTE: if the Science granule has an associated Browse the Browse will be published even if the Science granule is "hidden."		
15	Verify that ALL browse granules ingested as part of the test are recorded as public in the AIM schema and that the required browse JPEG files are created in the correct location in the public data pool (as recorded in the aim schema).		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.4 [PVC] - Test Case 6 - Ingest SIPS granules without publishing to the Data Pool (ECS-ECSTC-249)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies SIPS ingest without publishing to the Data Pool. This is the main regression test of DPL Ingest where all logging and ECS Service Host functionality should be tested.		#comment
2	[SIPS Ingest, but not to public Data Pool]		#comment
3	Filter the list of insert requests by request status SUCCESSFUL.		
4	Select one successful request from the filtered list where the collection in the request is configured not to be inserted into the public Data Pool.		
5	Verify that the request details are displayed on the Data Pool Ingest GUI, and include all information in S-DPL-16670.		
6	Verify that the request details include a list of all granules associated with the request, in the default order described in S-DPL-16690, and including all granule information listed in S-DPL-16690.		
7	Verify that all granules in the request are in a SUCCESSFUL state.		
8	Verify that the start and completion of PDR Validation, and the PDR Validation information in S-DPL-18345, are included in the application log for this request.		
9	Verify that the files for each granule in the request are transferred to a temporary directory on the file system configured for the collection.		
10	Verify that the files for each granule in the request are transferred using the configured transfer method.		
11	Verify that file transfers for all granules in the request were performed on one of the ECS platforms configured for that purpose.		
12	Verify that the start and completion of all file transfers, and the file transfer information in S-DPL-18350, are included in the application log for this request.		
13	Verify that the start and completion of ingest operations for the request, and all request information in S-DPL-18385, are included in the application log for this request.		
14	Verify that the start and completion of ingest operations for each granule in the request, and all granule information in S-DPL-18380, are included in the application log for this request.		
15	Verify that the Science granules in the request were inserted into a non-public directory on the Data Pool, and that all Data Pool inventory database information for non-public granules has been populated. Also verify all		

#	Action	Expected Result	Notes
	Browse granules ingested in this test are PUBLIC (AmGranule(IsOrderOnly) is NULL and the AmBrowseOnlineFile contains at least one entry for each Browse. Also verify the Browse JPEG files are stored in the appropriate public DataPool directory (as recorded in AmBrowseOnlineFile).		
16	Verify that queuing the Data Pool insert and all information in S-DPL-18375 are included in the application log for this request.		
17	Verify that all granules in the request are copied to the ECS archive, to the archive locations and volume groups that are configured for the collection in the AIM database. This includes all primary, backup, forward processing and reprocessing volume groups.		
18	Verify that the start and completion of archiving operations for all granules in the request, and all granule archiving information in S-DPL-18366, appears in the Data Pool Ingest Service application log.		
19	Verify that the start and completion of archiving operations for all files in the request, and all file archiving information in S-DPL-18367, appears in the Data Pool Ingest Service application log.		
20	Verify that the ECS archive file names of all granules in the request are consistent with the internal file naming convention. (Verify that the file names contain the ECS granule id which is the same as the ECS granule id in the inventory database and identify the physical file format correctly.		
21	Verify that the archive write operations for the granule were executed on one of ECS service hosts configured for that purpose.		
22	Verify that the XML metadata file is copied into the appropriate directory in the small file archive.		
23	Verify that the metadata for the granule is correctly inserted into the AIM database, and that the metadata was inserted in the AIM database AFTER the granule(s) in the request were copied to the ECS archive.		Verify the AmGranule.ArchiveTime for this granule is after the timestamp of the granule data file in the stornext archive. We check the timestamp of the data file instead of xml metadata file here, because 1). the xml file could be updated after archive, e.g. archive time update, browse linkage update. 2). the xml file is copied to the archive from

#	Action	Expected Result	Notes
			data pool at the same time when the data files are copied to archive.
24	Verify that the internal file(s) for the granule(s) are in the AIM metadata for the granule(s), and that this information is correct.		
25	Verify that the start and completion of metadata insert, and all metadata insert information in S-DPL-18370, appears in the Data Pool Ingest Service application log.		
26	Verify that an xml metadata file is stored in the non-public directory on the Data Pool for each granule in the request.		
27	Verify that the xml metadata file contents are correct for at least one granule of each ESDT used by the test.		
28	Verify that the ECSid of the granules in the request are stored in the "aim" schema, and in the Data Pool and AIM XML files.		
29	Verify that ALL browse granules ingested as part of the test are recorded as public in the AIM schema and that the required browse JPEG files are created in the correct location in the public data pool (as recorded in the aim schema).		

TEST DATA:
Any SIPS data type

EXPECTED RESULTS:

6.5 [PVC] - Test Case 7 - SIPS Ingest and publish to public Data Pool, Band Extraction (ECS-ECSTC-250)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies SIPS Data Pool Ingest and publication in the Data Pool works for a HEG-able granule.		#comment
2	[SIPS Ingest and to public Data Pool, Band Extraction]		#comment
3	Select one successful request from the filtered list where the collection in the		

#	Action	Expected Result	Notes
	request is configured to be inserted into the public Data Pool. THE GRANULES IN THE REQUEST MUST BE REAL HDFEOS DATA, BELONGING TO A HEG-ABLE COLLECTION. One of the granules shall have invalid band data; all other granules shall have valid band data.		
4	Verify that all granules in the request have been successfully queued with the Data Pool Insert Service for insertion into the public Data Pool.		
5	Verify that the queuing of the Data Pool insert and all information in S-DPL-18375 are included in the application log for this request.		
6	Verify, using the debug log or other method supplied in ITP, that the state of the request and the state(s) of all granules in the request were updated appropriately during processing of the request, and that each non-error state in S-DPL-18200 (for the request) and S-DPL-18210 (for granules) was logged in the correct order.		
7	Verify that start and completion of band extraction was logged for all granules, showing successful extraction for the granules with valid band information and a band extraction failure for the granule with invalid band information.		
8	For the granules with valid band information, verify that the band information was made available for insertion into the Data Pool inventory and is correct. (This can be verified, for example, by checking that the correct information is present in the Data Pool inventory or by completing a DPL publishing step and verifying that the granules can be ordered for HEG processing from the DPL Web GUI and the correct band subsetting options are offered.)		DPL Web GUI is retired after AIM-DPL database consolidation. We can check entries in aim.dlgranulehdfobjectsxref table.
9	Verify that all granules in the request are copied to the ECS archive.		
10	Verify that throughput statistics for archiving operations are stored in the Data Pool Ingest database, per the time interval configured in criterion 150, and that all statistics required in S-DPL-18850 are recorded (number of granules by archive, data volume by archive, number of granules per Data Pool file system, data volume per Data Pool file system).		

TEST DATA:

Any SIPS data type that is HEG-able

EXPECTED RESULTS:

6.6 Test Case 8 - Cross-DAAC Ingest (ECS-ECSTC-251)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies Cross-DAAC Ingest.		#comment
2	[Perform Cross-DAAC Ingest.]		#comment
3	This criterion may be verified by a separate test procedure or as part of the test procedure for criteria 300 to 395.		
4	This test uses the new capability that allows OMS to distribute metadata in .met file format.		From SSS GUI set the Media Type: scp
5	Identify collections that are transferred from one DAAC to another via cross DAAC ingest. Include at least two collections that represent ancillary granules and two collections that do not.		
6	Distribute at least one granule for each collection via OMS to a directory that is polled for Data Pool Ingest. Ensure that the corresponding DN includes a checksum and that the distribution includes an ODL metadata file (configure OMS accordingly).		
7	Verify that the corresponding Distribution Notices include a checksum.		
8	Verify that the distribution includes an ODL metadata file (configure OMS accordingly)		
9	Verify that the granules are ingested successfully.		
10	Reconfigure OMS to distribute metadata files to the destination in XML file format.		
11	Distribute at least one granule for each collection via OMS to a directory that is polled for Data Pool Ingest. Ensure that the corresponding DN includes a checksum and that the distribution includes a XML metadata file (configure OMS accordingly).		
12	Verify that the granules are ingested successfully.		
13	Verify that the AIM inventory contains the correct metadata.		

TEST DATA:

Any DDIST data type

EXPECTED RESULTS:

6.7 [PVC] - Test Case 9 - Ingest EDOS Expedited EDRs (ECS-ECSTC-252)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies Expedited EDOS / ASTER Ingest.		#comment
2	[Ingest EDOS Expedited EDRs]		#comment
3	With the Data Pool Ingest Service down, place all EDOS Expedited EDRs and signal files that will be used for testing in their appropriate polling directories. Functional testing will require a typical volume of Expedited operational data ingested from EDOS. (It is up to the tester to determine whether to perform testing with one subset of the full set of EDRs at a time, but all EDRs must ultimately be tested and be successfully processed.)		
4	Ensure the EDOS Expedited data types are configured not to be published upon insert.		
5	Ensure that mandatory checksum presence and 100% checksum verification is configured for the EDOS Expedited provider.		
6	Use the original data type and file type templates and source MCF information as defined in the INGST CI.		
7	Configure the maximum number of concurrent metadata extractions low enough to guarantee that the metadata extractions will run up against the configured maximum at least once during testing.		
8	Include at least one non-EDOS ingest request in the set of ingest requests that are included in the test and ensure that all the information needed to perform this ingest operation is configured.		
9	Set the default priority for the associated provider to be lower than the highest priority.		
10	Include at least one EDOS non-expedited request in the set of ingest requests that are included in the test and ensure that all the information needed to perform this ingest operation is configured. The provider for the EDOS		

#	Action	Expected Result	Notes
	non-expedited request should be the same as the provider for the EDOS expedited request The polling location may be the same or different.		
11	Set the default priority for this provider to be lower than the highest priority.		
12	Start the Data Pool Ingest Service. Wait until all requests are in a stopped state (Suspended, Successful, Failed, Partial_Failure) on the Request Page of the Data Pool Ingest GUI.		
13	Verify that the application log includes the start and completion of the metadata extraction steps and that no more than the configured maximum number of metadata extractions occurred simultaneously.		
14	[Regression, see criterion 340 in DP_S6_01.]		#comment
15	Select at least one successful EDOS Expedited ingest request and perform for it the verification steps 1, 2, 3, 5, 6, 11, 13, 16, 18, 19, 21, and 22 in criterion 340 in DP_S6_01. NOTE: for step 11 Browse granules included in the Ingest request will be public, on the Science Granules will be in the Hidden Data Pool.		
16	Verify that the start and completion of EDR Validation, and the EDR validation information corresponding to the PDR validation information in S-DPL-18345, are included in the application log for this request.		
17	[Regression, see criterion 340 step V-21.1 in DP_S6_01.]		#comment
18	For at least one granule of each of the EDOS Expedited data types included in the test verify that the metadata stored in the Data Pool XML file are correct and match what should have been extracted from the input granule.		
19	For at least one granule of each of the EDOS Expedited data types included in the test verify that the metadata stored in the AIM inventory are correct and match what should have been extracted from the input granule.		
20	Verify that all EDRs and PDRs are eventually processed successfully and that none of the EDRs or PDRs was processed twice.		
21	Verify that the EDOS Expedited data requests were processed at the highest possible priority, and that the non-EDOS and EDOS non-expedited requests were processed at the default priority configured for their associated data providers.		
22	Verify that for at least one successful EDOS Expedited EDR that a Expedited Acceptance Notice (EAN) was inserted into the DAAC configured local directory and was not inserted into the directory configured for EDOS notification.		
23	Verify for at least one successful EDOS Expedited ingest request that the EAN was constructed in compliance with ESDIS document 423-ICD-		

#	Action	Expected Result	Notes
	EDOS/EGS and follows the naming convention defined in that document.		
24	Verify that all temporary information (files, etc.) associated with the EDOS Expedited ingest request are removed after the request is complete and the EAN was deposited.		

TEST DATA:
Any Expedited data types

EXPECTED RESULTS:

6.8 [PVC] - Test Case 10 - Verifies EMOS Ingest (Polling without delivery record functionality) (ECS-ECSTC-253)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies EMOS Ingest (Polling without delivery record functionality).		#comment
2	[Submit EMOS Data Files]		#comment
3	Configure the EMOS Data Provider.		
4	Place several EMOS Data Files into the EMOS polling location.		
5	Start the Data Pool Ingest Service.		
6	Wait until all requests are in a stopped state (Suspended, Successful, Failed, Partial_Failure) on the Request page of the Data Pool Ingest GUI.		
7	Verify that the application log includes the start and completion of the metadata extraction steps for each granule.		
8	Verify that no more than the configured maximum number of metadata extraction steps were performed concurrently.		
9	Verify that the shortname and version id corresponding to the volume group used for archiving the granules are the shortname and version id configured for the EMOS data provider.		
10	For at least one EMOS granule included in the test, verify that the metadata		

#	Action	Expected Result	Notes
	stored in the Data Pool XML file are correct and match what should have been extracted from the input granule, according to the source MCF configuration in the Ingest database.		
11	For at least one EMOS granule included in the test, verify that the metadata stored in the AIM inventory database are correct and match what should have been extracted from the input granule, according to the source MCF configuration in the Ingest database.		
12	Verify that the shortname and version id for the granule in the AIM database are the shortname and version id configured for the EMOS data provider.		
13	Verify that all EMOS Data Files are eventually processed successfully and that none of the Data Files is processed twice.		
14	For at least one successful EMOS ingest request, verify that all temporary information (files, etc.) associated with the ingest request are removed after the request is complete.		
15	Verify no PAN is generated and no notification is queued in Ingest database for Notification Service.		
16	Verify the file size in DPL database and Inventory database match the real file size of the granule.		
17	Verify the file size saved in Datapool xml file and small file archive xml file match the real file size of the granule.		

TEST DATA:
ActSched.001

EXPECTED RESULTS:

6.9 [PVC] - Test Case 12 - Verifies ASTER Ingest (ECS-ECSTC-255)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies ASTER Ingest.		#comment

#	Action	Expected Result	Notes
2	[Submit all ASTER PDRs]		#comment
3	Configure ASTER L1A not to be published in the Data Pool.		
4	Use the original data type and file type templates and source MCF information as defined in the INGST CI.		
5	Submit several ASTER PDRs to an ASTER polling location (Provider's preprocessing type is NONE).		
6	Start Data Pool Ingest and wait until all requests are in a stopped state (Suspended, Successful, Failed, Partial_Failure) on the Request Page of the Data Pool Ingest GUI.		
7	Verify that the Data Pool Ingest Service application log includes information about the start and completion of metadata extraction steps.		
8	Using the application or performance log, verify that the number of concurrent metadata extraction steps did not exceed the configured maximum.		
9	Verify that the DAR identifications are included correctly in the metadata.		
10	Verify that the Local Granule ID is constructed correctly, that is, from the ESDT short name and version, the calendar date and time of day of the observation, and the production date and time, according to the following example: 'AST_L1A#00301032001055217_01162001155025.hdf'.		
11	Verify that the value of the product specific attribute ASTERGRANULEID matches the id of the ASTER granule provided in the PDR.		
12	For the ASTER Browse granule(s), verify that the Browse linkages have been performed correctly in the AIM database.		
13	Verify that the Data metadata and linkage information in the Data Pool inventory database and in the XML files is correct.		
14	Verify that the PAN was correctly transmitted as per configured notification method.		
15	Verify that the PAN was constructed in compliance with the applicable ICD.		
16	Verify that ALL browse granules ingested as part of the test are recorded as public in the AIM schema and that the required browse JPEG files are created in the correct location in the public data pool (as recorded in the aim schema).		

TEST DATA:

Any ASTER data types

EXPECTED RESULTS:

6.10 Test Case 13 - Verifies MISBR Ingest and linkages are handled correctly (ECS-ECSTC-256)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies MISBR Ingest and linkages are handled correctly.		#comment
2	[Ingest MISBR, MISR L1, MISR L2]		#comment
3	Ingest one L1, one L2, and one MISBR which should link together.		
4	Verify that the MISR granules are successfully ingested into ECS. Specifically, verify for at least one MISR Level 1 granule, one MISR Level 2 granule, and one MISBR granule that:		#comment
5		1. the granule information is recorded correctly in the AIM inventory database	
6		2. the granule XML metadata files are stored in XML Archive and their contents are correct	
7		3. the science data files are stored into archive	
8		4. the links between the MISR science granules and the matching MISBR granules are correctly recorded in the AIM database.	
9	Verify that ALL browse granules that were created (because the MISBR was configured as public and the MISBR matched the criteria of a L1 or L2 granule and thus generated browse granules) are recorded as public in the AIM schema and that the required browse JPEG files are created in the correct location in the public data pool (as recorded in the aim schema).		

TEST DATA:
MISR Level 1, MISR Level 2, and MISBR.005

EXPECTED RESULTS:

6.11 [PVC] - Test Case 14 - FTP Polling, Transfers, and Notification (ECS-ECSTC-257)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies successful Ingest is possible using FTP as the transferring mechanism.		#comment
2	[FTP Polling, Transfers, and Notification]		#comment
3	Use a provider which has an FTP polling location, the provider transfer type configured to FTP, and the Notification method set to FTP. Also ensure that Active FTP is used. If no such provider exists, then create a new provider or modify an existing provider.		
4	Place a PDR in the polling location and verify that an ingest request is created for that PDR within the configured polling interval.		
5	Verify in the Polling log that the file was transferred via FTP.		
6	Verify that eventually the request completes successfully.		
7	Verify in the Processing log that each file for each granule in the request was transferred via FTP.		
8	Verify that a PAN file is present in the notification directory.		
9	Verify in the Notification log file that the PAN file was transferred via FTP.		
10	Reconfigure the provider to use Passive FTP.		
11	Submit another PDR and verify that a new request is created and completes successfully.		
12	Also verify that a PAN file is present in the notification directory.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.12 [PVC] - Test Case 15 - Local Polling, Transfers, and Email Notification (ECS-ECSTC-258)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies successful Ingest is possible using local copies as the transferring mechanism. Also verifies email notification.		#comment
2	[Local Polling, Transfers, and Email Notification]		#comment
3	Use a provider which has a local polling location, the provider transfer type configured to Local, and the Notification method set to Email. If no such provider exists, then create a new provider or modify an existing provider.		The Polling Location can be a symbolic link to another directory.
4	Place a PDR in the polling location and verify that an ingest request is created for that PDR within the configured polling interval.		
5	Verify in the Polling log that the file was transferred locally (via a copy).		
6	Verify that eventually the request completes successfully.		
7	Verify in the Processing log that each file for each granule in the request was transferred via a local copy command.		
8	Verify that an email is received which contains the PAN file information.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.13 [PVC] - Test Case 16 - SCP Polling, Transfers, and Notification (ECS-ECSTC-259)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies successful Ingest is possible using SCP as the transferring mechanism.		#comment
2	[SCP Polling, Transfers, and Notification]		#comment
3	Use a provider which has an SCP polling location, the provider transfer type configured to SCP, and the Notification method set to SCP using F-Secure and no cipher. If no such provider exists, then create a new provider or modify an existing provider. Note that once a provider is configured for other than SCP, it cannot be changed to SCP.		
4	Place a PDR in the polling location and verify that an ingest request is created for that PDR within the configured polling interval.		
5	Verify in the Polling log that the file was transferred via SCP.		
6	Verify that eventually the request completes successfully.		
7	Verify in the Processing log that each file for each granule in the request was transferred via SCP.		
8	Verify that a PAN file is present in the notification directory.		
9	Verify in the Notification log file that the PAN file was transferred via SCP.		
10	Reconfigure the provider to use scp using OpenSSH and cipher aes128.		
11	Submit another PDR and verify that a new request is created and completes successfully. Also verify that a PAN file is present in the notification directory.		
12	Reconfigure the provider to use scp using OpenSSH and cipher des.		
13	Submit another PDR and verify that a new request is created and completes successfully. Also verify that a PAN file is present in the notification directory.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.14 Test Case 16b - HTTP Polling, Transfers, and Notification (ECS-ECSTC-498)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies successful Ingest is possible using HTTP as the transferring mechanism.		
2	[HTTP Polling, Transfers, and Notification]		
3	Use a provider which has an HTTP polling location, the provider transfer type configured to HTTP, and the Notification method set to HTTP. If no such provider exists, then create a new provider or modify an existing provider. 		
4	Place a PDR in the polling location and verify that an ingest request is created for that PDR within the configured polling interval.		
5	Verify in the Polling log that the file was transferred via HTTP.		
6	Verify that eventually the request completes successfully.		
7	Verify in the Processing log that each file for each granule in the request was transferred via HTTP.		
8	Verify that a PAN file is present in the notification directory.		
9	Verify in the Notification log file that the PAN file was transferred via HTTP.		
10	Add a new provider which has transfer type, polling method and notification type all set to HTTP. The http server doesn't need user authentication, so the Read/Write Login is not configured for this provider. Repeat step 4 to 9.		

TEST DATA:

EXPECTED RESULTS:

6.15 [PVC] - Test Case 17 - Browse granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-260)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies ingest of Browse granule in the same PDR as the science granule is successful and that the linkage is recorded successfully.		#comment
2	[Browse granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool]		#comment
3	Select one successful request from the filtered list where one or more science granules in the request have associated Browse granule(s) in the same PDR (e.g., AMSR Level 3), and the science collection is configured to be inserted in the public Data Pool.		
4	Verify that the browse linkage information is included in the AIM metadata (database) for the science granule.		
5	Verify that the science granule(s) are queued with the Data Pool Insert Service for insertion into in the public Data Pool.		
6	Verify that the original (hdfeos) version(s) of the associated browse granule(s) are stored in the Data Pool in the hidden directory structure.		
7	Verify that the science granule(s) and the original version(s) of the associated browse granule(s) are copied to the archive.		
8	Verify that the browse file name in the archive includes the ECS id that was pre-assigned to the browse granule.		
9	Verify that the science granule(s) and the original version(s) of the associated browse granule(s) are removed from the Data Pool hidden directory structure after their ingest request completes, subject to the rules in S-DPL-18455 and S-DPL-18460.		
10	Verify that the start and completion of preprocessing operations, and all preprocessing information in S-DPL-18365, is included in the Data Pool Ingest Service application log for this request. Note: all Browse granules should be public at the completion of the test. This means they should have JPEG files in the public data pool.		

TEST DATA:
Any data type with Browse

EXPECTED RESULTS:

6.16 Test Case 18 - Browse granule(s) in separate PDR from science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-261)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies correct linking when ingesting a browse granule with a linkage file.		#comment
2	[Browse granule(s) in separate PDR from science granule, where science granule is to be inserted in public Data Pool]		#comment
3	Submit two PDRs, the first for ingest of a science granule, and the second for ingest of (an) associated Browse granule(s), where the science collection is configured for public Data Pool insert (e.g., AMSR Level 2). The PDRs should be submitted far enough apart in time such that the science granule is archived and queued for insertion into the public Data Pool before the browse PDR is submitted.		
4	Verify that the browse linkage information is included in the AIM metadata (database) for the science granule.		
5	Verify that the science granule(s) are queued with the Data Pool Insert Service for insertion into in the public Data Pool.		
6	Verify that the original (hdf) version(s) of the associated browse granule(s) are stored in the Data Pool in the hidden directory structure.		
7	Verify that the science granule(s) and the original version(s) of the associated browse granule(s) are copied to the archive.		
8	Verify that the browse file name in the archive includes the ECS id that was pre-assigned to the browse granule.		
9	Verify that the science granule(s) and the original version(s) of the associated browse granule(s) are removed from the Data Pool hidden directory structure after their ingest request completes, subject to the rules in S-DPL-18455 and S-DPL-18460.		
10	Verify that the start and completion of preprocessing operations, and all preprocessing information in S-DPL-18365, is included in the Data Pool		

#	Action	Expected Result	Notes
	Ingest Service application log for this request.		
11	Verify that ALL browse granules ingested as part of the test are recorded as public in the AIM schema and that the required browse JPEG files are created in the correct location in the public data pool (as recorded in the aim schema).		

TEST DATA:
Any current data types with Browse

EXPECTED RESULTS:

6.17 [PVC] - Test Case 19 - QA granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-262)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies correct linking when ingesting a QA granule and science granule in the same PDR.		#comment
2	[QA granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool]		#comment
3	Select one successful request from the filtered list where one or more science granules in the request have associated QA granule(s) in the same PDR, and the science collection is configured to be archived and to be inserted in the public Data Pool. QA collection is configured to be inserted into public Data Pool as well.		
4	Verify that the QA linkage information is included in the AIM metadata (database) for the science granule.		
5	Verify that the science granule(s) are queued with the Data Pool Insert Service for insertion into the public Data Pool.		
6	Verify that the science granule(s) and the associated QA granule(s) are copied to the archive.		
7	Verify that the QA file name in the archive includes the ECS id that was pre-		

#	Action	Expected Result	Notes
	assigned to the QA granule.		
8	Verify that the science granule(s) and the QA granule(s) are removed from the Data Pool hidden directory structure after their ingest request completes, subject to the rules in S-DPL-18455.		

TEST DATA:
Any current data types with associated QA

EXPECTED RESULTS:

6.18 [PVC] - Test Case 20 - PH granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-263)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies correct linking when ingesting a PH granule and a science granule in the same PDR.		#comment
2	[PH granule(s) in same PDR as science granule, where science granule is to be inserted in public Data Pool]		#comment
3	Select one successful request from the filtered list where one or more science granules in the request have associated PH granule(s) in the same PDR, and the science collection is configured to be inserted in the public Data Pool. Assume PH will not be enabled for public Data Pool insert.		
4	Verify that the PH linkage information is included in the AIM metadata (database) for the science granule.		
5	Verify that the science granule(s) and associated PH granule(s) are stored in the Data Pool hidden directory structure.		
6	Verify that the science granule(s) and associated PH granule(s) are copied to the archive.		
7	Verify that the PH file name in the archive includes the ECS id that was pre-assigned to the PH granule.		

#	Action	Expected Result	Notes
8	Verify that the science granule(s) and the PH granule(s) are removed from the Data Pool hidden directory structure after their ingest request completes, subject to the rules in S-DPL-18455.		

TEST DATA:
Any current data types with associated PH

EXPECTED RESULTS:

6.19 Test Case 21 - PH granule(s) in different PDR from science granule, where science granule is to be inserted in public Data Pool (ECS-ECSTC-264)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies correct linking when ingesting a PH granule with a linkage file.		#comment
2	[PH granule(s) in different PDR from science granule, where science granule is to be inserted in public Data Pool]		#comment
3	Submit two PDRs, the first for ingest of a science granule, and the second for ingest of (an) associated PH granule(s), where the science collection is configured for public Data Pool insert. The PDRs should be submitted far enough apart in time such that the science granule is archived and queued for insertion into the public Data Pool before the PH PDR is submitted.		
4	Verify that the PH linkage information is included in the AIM metadata (database) for the science granule.		
5	Verify that the science granule(s) and associated PH granule(s) are stored in the Data Pool hidden directory structure.		
6	Verify that the science granule(s) and associated PH granule(s) are copied to the archive.		
7	Verify that the PH file name in the archive includes the ECS id that was pre-assigned to the PH granule.		

#	Action	Expected Result	Notes
8	Verify that the science granule(s) and the PH granule(s) are removed from the Data Pool hidden directory structure after their ingest request completes, subject to the rules in S-DPL-18455.		

TEST DATA:
Any current data types with associated PH

EXPECTED RESULTS:

6.20 Test Case 22 - Data Pool Registration Error â€“ Retried (ECS-ECSTC-265)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: This test verifies that the Data Pool Ingest system properly handles, reports on, and recovers from Data Pool Registration Errors:		#comment
2	[Data Pool Registration Error – Retried.]		#comment
3	This criterion may be verified by a separate test procedure or as part of the test procedure for criteria 300 to 395.		#comment
4	Ensure that an e-mail address for operator notification regarding ingest interventions is configured.		
5	Submit a valid PDR for at least two granules neither of which will be published in the Data Pool as part of the insert.		
6	Cause one of the granules to return a retriable error during Data Pool registration i.e. while trying to register the granule and its file in the Data Pool inventory.		
7	Verify that the Data Pool registration is retried using the configured retry interval and retry limit but ensure that the error condition persists.		
8	Verify that each error is logged in the ingest application log in accordance with S-DPL-18320.		
9	Verify that the granule is suspended after the configured retry limit is		

#	Action	Expected Result	Notes
	exhausted.		
10	Verify that the request state eventually changes to ‘Suspended’ [Note: the request may temporarily be in the ‘Partially_Suspended’ state depending on how quickly the retries are exhausted as compared to the time needed to complete the remaining granules.]		
11	Verify that an operator intervention is queued for the request after all remaining granules complete ingest and that the intervention appears in the monitoring screen.		
12	Verify that an e-mail is sent to the configured e-mail address regarding the intervention that provides the information mandated in S-DPL-17255.		
13	Verify that the operator intervention request is logged in the Data Pool Ingest Service application log.		
14	Verify that the intervention monitoring screen shows the information mandated in S-DPL-17265.		
15	Verify that the error type displayed in the monitoring screen is appropriate.		
16	Verify that it is possible to select an intervention and access its details from the intervention monitoring screen.		
17	Verify that the intervention details include a granule list showing suspended granules first.		
18	Verify that the granule list contains the information mandated by S-DPL-17330 and indicates the correct error type.		
19	Verify that it is possible to display granule details for a suspended granule as mandated by S-DPL-17340 and that the information includes information pertinent to the specific Data Pool registration error.		
20	Correct the problem that caused the Data Pool registration error and verify that it is possible to resume the granule.		
21	Verify that the closure of the intervention is recorded correctly as per S-DPL-17370.		
22	Verify that the closure of the intervention is correctly recorded in the request annotations as per S-DPL-17216.		
23	Verify that the request completes and the request state changes to ‘Successful’.		
24	Verify that a PAN is transmitted using the notification method configured for the provider.		
25	Verify that the PAN is formatted in compliance with the SIPS ICD.		
26	Verify that the PAN correctly identifies the request as having been successful as per SIPS ICD.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.21 Test Case 23 - Data Pool Registration Error â€“ Failed. (ECS-ECSTC-266)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Data Pool Registration Error – Failed.]		#comment
2	This criterion may be verified by a separate test procedure or as part of the test procedure for criteria 300 to 395.		#comment
3	Ensure that an e-mail address for operator notification regarding ingest interventions is configured.		
4	Submit a valid PDR for at least two granules neither of which will be published in the Data Pool as part of the insert.		
5	Cause one of the granules to return a retriable error during Data Pool registration i.e. while trying to register the granule and its file in the Data Pool inventory.		
6	Verify that the Data Pool registration is retried using the configured retry interval and retry limit but ensure that the error condition persists.		
7	Verify that each error is logged in the ingest application log in accordance with S-DPL-18320.		
8	Verify that the granule is suspended after the configured retry limit is exhausted.		
9	Verify that the request state eventually changes to ‘Suspended’ [Note: the request may temporarily be in the ‘Partially_Suspended’ state depending on how quickly the retries are exhausted as compared to the time needed to complete the remaining granules.]		
10	Verify that an operator intervention is queued for the request after all remaining granules complete ingest and that the intervention appears in the		

#	Action	Expected Result	Notes
	monitoring screen.		
11	Verify that an e-mail is sent to the configured e-mail address regarding the intervention that provides the information mandated in S-DPL-17255.		
12	Verify that the operator intervention request is logged in the Data Pool Ingest Service application log.		
13	Verify that the intervention monitoring screen shows the information mandated in S-DPL-17265.		
14	Verify that the error type displayed in the monitoring screen is appropriate.		
15	Verify that it is possible to select an intervention and access its details from the intervention monitoring screen.		
16	Verify that the intervention details include a granule list showing suspended granules first.		
17	Verify that the granule list contains the information mandated by S-DPL-17330 and indicates the correct error type.		
18	Verify that it is possible to display granule details for a suspended granule as mandated by S-DPL-17340 and that the information includes information pertinent to the specific Data Pool registration error.		
19	Verify that it is possible to select the suspended granule and fail it.		
20	Verify that the closure of the intervention was recorded correctly as per S-DPL-17370.		
21	Verify that the closure of the intervention is correctly recorded in the request annotations as per S-DPL-17216.		
22	Verify that the request state becomes ‘Partial_Failure’.		
23	Verify that the granule state becomes ‘PreprocErr’.		
24	Verify that a PAN is transmitted using the notification method configured for the provider.		
25	Verify that the PAN is formatted in compliance with the SIPS ICD.		
26	Verify that the PAN correctly identifies the disposition of each granule as specified in the SIPS ICD.		

TEST DATA:

Any current data types

EXPECTED RESULTS:

6.22 Test Case 24 - Add & Modifying Volume Groups (ECS-ECSTC-267)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies it is possible to add and modify volume groups using the DPL Ingest GUI and that the server correctly uses the new volume groups.		#comment
2	[Add & Modifying Volume Groups]		#comment
3	Using the DPL Ingest GUI add a primary and backup Volume Group for a collection.		
4	Ingest a granule and verify that it goes to the primary and backup volume groups.		
5	Modify Volume group and give it a new path		ncr8051952 Failing to alternate Primary Alternative and Backup Alternative volume group in EDF-TS2 mode.
6	Ingest a granule and verify that it goes to the new path.		
7	Click on reports and verified that the addition and modification from above are displayed in the report.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.23 Test Case 25 - Verifies correct linking of AST_L1A, AST_L1B, and Browse (ECS-ECSTC-268)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies correct linking of AST_L1A, AST_L1B, and Browse. Covers several use cases at LP DAAC including replacement scenarios.		#comment
2	[Ingest of AST_L1A with Browse]		#comment
3	Ensure AST_L1A.003 is not be configured for public Data Pool insert.		
4	Ingest an AST_L1A.003 granule with an associated Browse granule.		
5	Verify that the ingest requests for the AST_L1A and Browse granules are in the SUCCESSFUL state.		
6	Verify that both the AST_L1A granule and the associated Browse granule are successfully stored in the correct archive file location(s), corresponding to the open volume groups for the AST_L1A and Browse ESDTs respectively.		
7	Verify that the NDPIU stores metadata for both the AST_L1A and the Browse granule in the Inventory database, and that this metadata is complete and correct.		
8	Verify that the NDPIU stores complete and correct linkage information in the Inventory database, linking the correct AST_L1A granule to the Browse granule.		
9	Verify that the DPL Ingest Service stores Browse linkage information in the AST_L1A XML metadata file, including the UR of the Browse granule.		
10	[AST_L1B linkage]		#comment
11	Configure the AST_L1B.003 ESDT for insert into the public Data Pool.		
12	Ingest an AST_L1B granule produced from the first AST_L1A granule ingested. The AST_L1B granule does not have to be produced as part of this test.		
13	Verify that the ingest request for AST_L1B reaches a SUCCESSFUL state.		
14	Verify that the AST_L1B granule is successfully stored in the correct archive file location(s), corresponding to the open volume groups for the AST_L1B ESDT.		
15	Verify that the NDPIU stores complete and correct linkage information in the Inventory database, linking the AST_L1B granule to the Browse granule associated with the most recently inserted AST_L1A which matches the temporal range of the AST_L1B.		
16	Verify that the DPL Ingest Service stores Browse linkage information in the AST_L1B XML metadata file for the correct Browse file, including the UR of the Browse granule.		
17	[Duplicate AST_L1A]		#comment

#	Action	Expected Result	Notes
18	Submit a second PDR for the same AST_L1A.003 granule that was ingested previously, with the same associated browse granule.		
19	Verify that the ingest requests for the AST_L1A and Browse granules are in the SUCCESSFUL state.		
20	Verify that the NDPIU stores complete and correct linkage information in the Inventory database, linking the second Browse granule to the second AST_L1A granule, and not to the original AST_L1A granule inserted.		
21	Verify that the XML file for the second AST_L1A granule contains linkage information for the second Browse granule, that the XML file for the first AST_L1A inserted has not been updated, and that the XML file for the AST_L1B has not been updated.		
22	[AST_L1B produced from duplicate AST_L1A]		#comment
23	Ingest an AST_L1B granule produced from the second AST_L1A granule ingested.		
24	Ensure that Data Pool granule replacement is turned on for the AST_L1B collection.		
25	Verify that the ingest request for AST_L1B reaches a SUCCESSFUL state.		
26	Verify that the NDPIU stores complete and correct linkage information in the Inventory database, linking the AST_L1B granule ingested in this criterion to the second Browse granule ingested.		
27	Verify that the XML file for the AST_L1B granule ingested in this criterion contains linkage information for the second Browse granule ingested, and that the XML file for the first AST_L1B inserted has not been updated.		
28	[Browse replacement]		#comment
29	Ingest a new Browse granule that references the local granule id shared by the two AST_L1A granules.		
30	(This test simulates the LPDAAC practice of replacing a bad browse granule).		#comment
31	Verify that the Browse granule reaches a SUCCESSFUL state in Data Pool ingest.		
32	Verify that the Browse granule is stored in the browse archive.		
33	Verify that the NDPIU successfully stores all required metadata for the Browse granule in the Inventory database.		
34	Verify that the NDPIU links the Browse to the second AST_L1A and second AST_L1B granules.		
35	Verify that the NDPIU removes the existing links between the second AST_L1A and second AST_L1B granules, and the Browse granule linked to		

#	Action	Expected Result	Notes
	them.		
36	Verify that the Data Pool Ingest Service replaces the browse linkage information in the XML metadata files for the second AST_L1A and second AST_L1B granules with the new browse granule id.		
37	Verify that the existing links between the first AST_L1A and first AST_L1B granules ingested and the first Browse granule ingested are not updated.		
38	Verify that the XML file for the first AST_L1A inserted has not been updated, and that the XML file for the first AST_L1B inserted has not been updated.		
39	[Special PSA handling for ASTER data]		#comment
40	For the AST_L1A and AST_L1B granules ingested, verify that the IIU extracted the Product Specific Attribute named RadiometricDBVersion and stored it in the Inventory database.		
41	For the AST_L1A and AST_L1B granules ingested, verify that the IIU extracted the Product Specific Attribute named GeometricDBVersion and stored it in the Inventory database.		
42	For the AST_L1A and AST_L1B granules ingested, verify that the IIU extracted the Product Specific Attribute named DAR_ID and stored it in the Inventory database.		

TEST DATA:
AST_L1A.003, AST_L1B.003, Browse

EXPECTED RESULTS:

6.24 Test Case 26 - Checksum during DPL Ingest of Browse, QA, PH, DAP (ECS-ECSTC-269)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies DPL Ingest correctly calculates and stores checksum information for ancillary types.		#comment

#	Action	Expected Result	Notes
2	[Checksum during DPL Ingest of Browse, QA, PH, DAP]		#comment
3	Configure a data provider whose granules are public in the Data Pool to checksum 100% of the files ingested. Ingest at least one Browse and one of each ancillary (QA, PH, DAP).		
4	Verify the Ingest GUI allows for configuration of a default checksum algorithm of cksum or md5sum.		
5	Verify the checksum value of each browse and ancillary file is stored in the Ingest database along with the checksum algorithm used and a checksum origin of “DPLIngst”.		
6	Verify that the default configured checksum algorithm was used to calculate the checksum.		
7	Verify that one of the configured checksum service hosts was used to perform the checksum verification and that this was not the same service host used for the file transfer.		
8	Verify that the database tables in the DataPool database are populated with the appropriate checksum information for every data file.		
9	Verify that the database tables in the AIM Inventory database are populated with the appropriate checksum information for every data file.		
10	Verify that the checksum information is included in the datapool xml file for every ancillary file.		
11	Verify that the checksum is stored in the Data Pool Database for every ancillary files.		

TEST DATA:
Browse, QA, PH, DAP

EXPECTED RESULTS:

6.25 Test Case 27 - Fault Recovery (ECS-ECSTC-270)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies DPL Ingest correctly recovers granules and requests during a fault recovery situation.		#comment
2	[Fault Recovery]		#comment
3	Kill the Data Pool Ingest processing service while granules are actively being ingested.		
4	Bring the Data Pool Ingest processing service back up.		
5	Verify that all granules active at the time of shutdown complete successfully.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.26 DPL Ingest Modified

6.26.1 Test Case 3 - Use the DPL Ingest GUI to configure Remote Transfer Hosts (ECS-ECSTC-246)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies Remote Transfer Hosts can be configured using the DPL Ingest GUI		#comment
2	[Configure FTP hosts]		#comment
3	As the ‘ingest admin’ operator, navigate to the FTP Host page on the Data Pool Ingest GUI.		
4	Edit existing ftp host information (if any, from the INGST CI), per S-DPL-16260, and define new ftp hosts such that all ftp hosts which will be used for testing this ticket are defined. (Table of ftp hosts and related S-DPL-16260 information TBS as part of ITP). At least two ftp hosts should be defined that are not configured for the INGST subsystem.		
5	Verify that the Data Pool Ingest GUI allows the ‘ingest admin’ operator to		

#	Action	Expected Result	Notes
	edit ftp host information.		
6	Verify that the Data Pool Ingest GUI prompts the operator to confirm changes to existing ftp hosts and definitions of new ftp hosts before saving this information.		
7	Verify that the ingest admin operator has the authorization to define the ftp hosts (i.e., has access to the ftp host pages on the Data Pool Ingest GUI, and that information entered by this operator is stored in the database.)		
8	Verify that the Data Pool Ingest GUI allows all information in S-DPL-16260 to be entered.		
9	[Configure HTTP hosts]		
10	Configure a HTTP host use the step similar to 'Configure FTP hosts'.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.26.2 Test Case 4 - Use the DPL Ingest GUI to configure SCP Hosts (ECS-ECSTC-247)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies SCP Hosts can be configured using the DPL Ingest GUI		#comment
2	[Configure hosts for scp access]		#comment
3	As the ‘ingest admin’ operator, navigate to the host page on the Data Pool Ingest GUI.		
4	Edit existing information for hosts accessed via scp (if any, from the INGST CI), per S-DPL-16290, and define new scp hosts such that all scp hosts which will be used for testing this ticket are defined. (Table of hosts accessed via scp, and related S-DPL-16290 information TBS as part of ITP.) Scp type/cipher combinations to include in the test are: F-		

#	Action	Expected Result	Notes
	secure/None; OpenSSH/aes128; OpenSSH/3des, i.e., the test needs to involve several different providers.		
5	Verify that the Data Pool Ingest GUI allows the 'ingest admin' operator to enter and edit the scp host information in S-DPL-16290.		
6	Verify that the Data Pool Ingest GUI prompts the operator to confirm changes to existing scp host parameters and definitions of new scp host parameters before saving this information.		
7	Verify that the ingest admin operator has the authorization to define the scp host parameters in S-DPL-16290 (i.e., has access to the host pages on the Data Pool Ingest GUI, and that information entered by this operator is stored in the database.)		

TEST DATA:
Any current data types

EXPECTED RESULTS:

6.26.3 [PVC] - Test Case 11 - Verifies EDOS Ingest (ECS-ECSTC-254)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Description and Objectives: Verifies EDOS Ingest.		#comment
2	[Submit EDOS PDRs]		#comment
3	With the Data Pool Ingest Service down, place all EDOS PDRs and signal files (XFR files) that will be used for testing in the configured EDOS polling directories.		
4	Ensure the EDOS data types are configured not to be published upon insert		
5	Use the original data type and file type templates and source MCF information as defined in the INGST CI.		
6	Start the Data Pool Ingest Service. Wait until all requests are in a stopped state (Suspended, Successful, Failed, Partial_Failure) on the Request Page of		

#	Action	Expected Result	Notes
	the Data Pool Ingest GUI.		
7	Verify that the application log includes the start and completion of the metadata extraction steps.		
8	Use the application log to verify that no more than the configured maximum number of metadata extraction steps were performed concurrently.		
9	Verify that the metadata stored in the Data Pool XML file are correct and match what should have been extracted from the input granule.		
10	Verify that the metadata stored in the AIM inventory are correct and match what should have been extracted from the input granule.		
11	Verify that all PDRs are eventually processed successfully and that none of the PDRs was processed twice.		
12	Verify that for at least one successful EDOS PDR that a Product Acceptance Notice was inserted into the DAAC configured local directory and was not inserted into the directory configured for EDOS notification.		
13	Verify for at least one successful EDOS ingest request that the PAN was constructed in compliance with ESDIS document 423-ICD-EDOS/EGS and follows the naming convention defined in that document.		
14	Verify that all temporary information (files, etc.) associated with the ingest request are removed after the request is complete and the PAN was deposited.		

TEST DATA:
Any EDOS data types

EXPECTED RESULTS:

7 DATA ACCESS

7.1 DP_81_02_TP011 Configure and Retrieve SDPS Processing Options (ECS-ECSTC-376)

DESCRIPTION:
Test Case ID - 80
Run the tool to pre-populate the available processing options for the collections identified in Test Case 10.
Configure at least one collection which has HDF-EOS objects or dimensions that cannot be processed by HEG to use a processing tool other HEG.
For each of the processing options listed below configure at least one collection, but not all collections to support that processing option:
a. spatial subsetting via bounding box
b. temporal subsetting via one or a series of time intervals

c. list of objects, fields, bands, etc. which can be extracted
d. projections in which the output can be made available and the projection parameters that can be provided by the user
e. formats in which the output can be delivered and any formatting parameters that can be provided by the user
f. re-sampling options and any related parameters offered for the granules in this collection
Create and submit URLs to retrieve 'SDPS Processing Options Documents' for each of the collections configured.
Create and submit URLs to retrieve 'SDPS Processing Options Documents' for a granule belonging to each of the collections that have been configured. In addition choose two granules to test part c.) such that one granule contains all of the bands that are available for the collection and one is missing at least one of the available bands.

PRECONDITIONS:

Collections specified in Test Case ID 10 are present in the mode and contain public granules. There must be at least one public granule which does not contain all of the available bands for a collection.

STEPS:

#	Action	Expected Result	Notes
1	This test case spuns off of test case 10. Go back to test case 10, to see which granules of which collections you have ingested. Once the Configuration GUI is loaded the available collections should appear there. Once you click on a collection, this collection can be added to a service, by right clicking on the collection and then selecting Add New Service. Alternatively this collection can be added to an existing service. Once added one can see the available HDF objects that are available for that particular collection.	Verify that the ingested granules in collections in Test Case 10 appear in the Configuration Utility. Verify that for each collection there is a list of all available HDF objects. These HDF objects should also be present when one invokes the ESI url of that said granule and collection. Verify that the HDF objects are the same in each instance.	
2	Configure at least one collection which has HDF-EOS objects or dimensions that cannot be processed by HEG to use a processing tool other than HEG. For example select a one dimention HDF band to a swath collection, or selecting Position_in_Orbit object for any AE_DySno granule would do it. Disable all of the other HDF bands and enable only the one dimensional band.	Verify that only the enabled one-dimensional band is visible when the ESI form is brought up by going to the ESI form of a ingested granule of the configured collection.	
3	a. spatial subsetting via bounding box For this test, go back to the Configuration GUI, select the service that contains a different collection, and while in the configuration window, select the enable spatial subsetting checkbox.	Verify that the spatial subsetting section is visible on the form when invoked thru the ESI website by going to a granule that is part of the collection configured as part of this step.	
4	b. temporal subsetting via one or a series of time intervals: No use cases exist for this functionality, however, the ESI API supports it. Therefore we	Select a granule that is part of each collection. view the capabilities XML	

#	Action	Expected Result	Notes
	will only test the ability to turn temporal subsetting on and off. -In the configuration GUI, select a collection and then a service associated with that collection. Check the Enable Temporal Subsetting checkbox. Click the Update Service To Collection Mapping Button. -In the configuration GUI, select another collection and then a service associated with that collection. unheck the Enable Temporal Subsettign checkbox. Click the Update Service To Collection Mapping Button. 	for each granule. verify that the section for the service which was configured in the setup step has temporalSubsetting= true or temporalSubsetting = false consistent with what was configured.	
5	c. list of objects, fields, bands, etc. which can be extracted This test is very similar to step 2, but in this case select a collection that you have not selected before, create a service for it, and then select only a handful of all of the presented objects, fields and bands. Test at least two granules of the said collection.	Once configured head over to the ESI website and open a granule that is part of that collection. Only the objects, fields and bands selected in the configuration website should be visible. Nothing else should be present there.	
6	Further for the part above after testing two granules go back to the configuration gui and de-select every single objects, fields and bands, so that no objects, fields and bands are present. Then go to the ESI and process a granule of that collection. Then go back and add every single objects, fields and bands. Go back and test with the ESI once again.	Verify that whatever objects, fields and bands is selected is displayed on the ESI form, even if there are none selected or all selected. What is configured in the Configuration GUI should be present on the form	
7	d. projections in which the output can be made available and the projection parameters that can be provided by the user This option asks to select a new collection, and then selecting a few of all of the available projections. Just as before select a new collection, then add to a service, and then while on the configuration window, click on the Enabled Projections dropdown. This option will provide two fields - Available projections and selected projections. Select a few of all of the available projections. Once done head over to the ESI Website and select a granule that is part of that projection.	Verify that the selected projections appear on the ESI website. Also verify that all projection parameters also appear on the webform.	
8	e. formats in which the output can be delivered and any formatting parameters that can be provided by the user For this option select a new collection, and add a new service for it. When configured and the configuration window is shown, click on the Enabled Formats dropdown, and select a few of the available attributes. Once selected they will show in the Selected section. Make sure that you have some that are not selected. Once done proceed towards the ESI website, and select an ingested granule that is part of that collection for processing.	The selected ingested granule should have only the formats that were selected in the Configuration GUI. Anything that was not selected should not appear there.	
9	f. re-sampling options and any related parameters offered for the granules in this collection For this step select a new collection to add to a service.	Verify with the ESI that once saved the selected attributes are shown on	

#	Action	Expected Result	Notes
	When in the configuration window, click on the Enabled Resampling Types. There should be available attributes section, as well as selected attributes section. What appears in the Selected Attributes section should appear on the ESI webform. On the chosen collection, select a few attributes.	the ESI webpage for a granule of that collection.	
10	For the collection selected above select a really low value of the Select Timeout Seconds. This can be achieved by changing the timeout file of that service(Please consult with the Data Access admin for help and location of the config file). Once done, select a granule of that collection in the ESI website for processing. The granule should fail if the timeout is sufficient enough.	Verify that if a very low timeout value has been selected the timeout is indeed working, i.e. it times out when processed.	
11	As a final step select any other options on the config page as part of the already confugired collections from previous steps. Any selected scenario should reflect on the ESI webform.	Verify that any changes made on the Configuration GUI are reflected on the webform	

TEST DATA:

EXPECTED RESULTS:
Verify that the full list of objects, fields, and bands are populated for the collection.
Verify that it is possible to configure different collections to use different tools for processing.
Verify that it is possible to configure collections to offer each of the processing options listed in a. through f.
Verify that it is possible to configure collections to not offer each of the processing options listed in a. through f.
Verify that it is not possible to configure HEG processing of HDF-EOS objects and dimensions that cannot be subsetted or processed by HEG. Verify that it is possible to make available processing of HDF-EOS objects and dimensions that cannot be subsetted or processed by HEG if the tool is anything other than HEG.
Verify that the documents returned match the corresponding schema and contain the complete and correct description of the processing options for these collections.
Verify that the documents returned for the collection level and analogous granule level requests are identical for each collection with the exception of the request for the granule processing options which did not contain all of the bands.
Verify that for the request made on the granule with the missing bands the document returned does not contain those bands, but does contain the remaining available bands.

7.2 DP_81_02_TP013 SDPS Web API Request Error Handling (ECS-ECSTC-378)

DESCRIPTION:
Test Case ID - 30
Attempt to cover the potential URL request errors exhaustively. Submit processing requests that have the following errors:
a) Syntax errors in any one of the URL components needed to formulate the requests in Test Case 10.

b) Missing mandatory parameters (attempt to cover all mandatory parameters)
c) Data value errors in the URLs where the syntax is otherwise correct. Examples: invalid bounding boxes; invalid projection parameter values; missing projection parameter values that are conditionally required.
d) Reference to object, field, and band names that do not exist for granules in that collection.
e) Reference to a non-existing granule.
f) Reference to a granule that is in the AIM/DPL inventory but whose files are not available on-line.
g) Reference to a collection not supported for HEG processing.
h) Reference to a capability that is not supported, e.g., a projection.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	This test will cover various error test case scenarious. Use the online ESI/EGI Webform to submit the requests. Unless otherwise specified usy synchronous mode to process granules, and the wget EGI posted on the final processing page when submitting a granule. Alternatively for ESI testing one can use the attached urls in the spreadsheet which test all of the scenarious below. a) Syntax errors in any one of the URL components needed to formulate the requests in Test Case 10. Just as it written select a parameter(i.e. FILE_IDS) and change the parameter's name. Do this for all the parameters that are part of the url and submit each individual request.	Verify that whatever name you altered, i.e. not recognized by the system the appropriate error has been displayed.	
2	b) Missing mandatory parameters Submit a request which does not include any of teh following fields: • FILE_IDS • FILE_URLS • DATASET_ID 	Verify that when submitted an error corresponding to what is missing is generated.	
3	c) Data value errors in the URLs where the syntax is otherwise correct. Examples: invalid bounding boxes; invalid projection parameter values; missing projection parameter values that are conditionally required. For this scenario one would have to create the conditions described on the above. An easy way to construct illegal values is to go outside of possible allowed values. Range of latitude is -90o to 90o, and longitude -1800 to -180o. Providing custom values for projections that have those parameters as inputs will create an error. As for the bounding box(subsetting) an illegal coordinate would be again to fall outside of the coordinates that define the granule, or simply go over the extreme limits as outlined above.	Verify that for the error that is expected is actually produced and displayed.	
4	d) Reference to object, field, and band names that do not exist for granules in that collection. Just as described in prior steps change the name of a field, band and and objects.	Verify that the appropriate error has been displayed	
5	e) Reference to a non-existing granule. This part of the test deals with	Just as before verify that the correct	

#	Action	Expected Result	Notes
	the ?FILE_IDS= section of the url. All one has to do is to edit this to a non-existing GranuleId.	error message has been displayed	
6	f) Reference to a granule that is in the AIM/DPL inventory but whose files are not available on-line. For this scenario, when a granule has been ingested in AIM, it really means to delete, or for test purposes rename the filename of the granule on the AIM machine. A possible route is to rename the granule filename, and then run the url, and once an error has been observed to then change the name of the granule file to the original name.	Just as scenarios before verify that you are seeing the error that you are supposded to see. It should describe what went wrong.	
7	g) Reference to a collection not supported for HEG processing. This section deals with a collection that is not supported by the HEG. The collections that are supported currently by the HEG tool are outlined on the following webpage: http://newsroom.gsfc.nasa.gov/sdptoolkit/HEG/HEGProductList.html Choosing a collection that is not part of that list would have to create an error. One would also have to explicitly create the url since the web form only contains HEGable collections. To create the url obtain the GranuleId from EcInDb..AmGranule. Select a granule that is part of a collection not listed on the page above. 	Once the urls have been generated and run, expect to see errors displaying what went wrong. THe errors should be fairly descriptive	
8	h) Reference to a capability that is not supported, e.g., a projection. As in case g), there are certain capabilities that are not supported for certain granules, so for those granules one has to select capabilities that are part of the url but unsupported by the HEG. Alternatively one could also chose a fake projection name. The Heg server should fail.	As before verify that you obtain a descriptive error of what happened.	
9	For al of the processed requests above do at least 5-7 requests by using the gdal procesing tool. This can be achieved by selecting the gdal from the processing tool dropdown on the main form of the granule. This will simulate alternate internal API requests provided for in S-DPL-x0375.	Verify that you observe the same errors that you have observed with the HEG processing tool. There should be no distinctive difference.	

TEST DATA:

EXPECTED RESULTS:

Verify that the correct error response is returned. Though the test may use mostly one of the internal API implementations specified in S-DPL-x0375, repeat several (at least two) of the test cases using an alternate internal API implementations provided for in S-DPL-x0375.

7.3 DP_81_02_TP030 SDPS Web API request cleanup (ECS-ECSTC-383)

DESCRIPTION:

Test Case ID - 340
Configure the interval for keeping SDPS Web API outputs to 1 day.
Submit an on-demand processing request.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Go to the HEG Data Access Configuration Interface in the mode that you are operating in and click on the Application Configuration tab. Configure the interval for keeping SDPS Web API outputs(REQUEST_CLEANUP_AGE) to 1 day and the cleanup interval (REQUEST_CLEANUP_INTERVAL) to 1 hour.	The interval SDPS Web API time limit has been set	
2	Next open the ESI website and select a valid HEGable granule to process. Process the granule as normal.	Verify that a message has been seen in the output the displays that the user has 1 day to retrieve the output file. Also there should not be any errors in the output. Further verify that the output file shown in the output is present in the file system, as well as present as a download link.(Do not close the page, will be needed in step 5)	
3	For this step do asynchronous request by using an EWOC request(for assistance look at DP_81_02_TP019 Processing Services Operator Interventions).	Once the EWOC request has been executed verify that the output file is present, an email has been received, the granule request has been observed in the OMS GUI.	
4	Wait 1 day (or find a request that was completed >= 1 day ago)	Verify that the cleanup age has passed	
5	Go back to the file system and try to find the output files generated from step 2 and 3	The files should not be found. They should be deleted. Verify that also by clicking on the download link from step 2 (can also be retrieved by going to <egi_URL>/request/<requestID>) the browser should produce a	

#	Action	Expected Result	Notes
		file not found error	

TEST DATA:

EXPECTED RESULTS:
Verify that information is presented to the user specifying that the outputs will be unavailable after 1 day.
Verify that prior to 3 hours the files are available and accessible via the download link.
Verify that the outputs are removed from the file system after 1 day has passed (it is acceptable for the removal from disk to take up to 10 minutes).
Verify that clicking on the download link after the files have been removed results in an error indicating the files are not present.

7.4 DP_81_02_TP031 SDPS Web API asynchronous requests (ECS-ECSTC-384)

DESCRIPTION:
Test Case ID - 350
Submit several asynchronous web requests. Ensure that the requests will take enough time to complete to allow for monitoring. Ensure that at least one but not all requests specify email notification.
Submit one asynchronous web request which exceeds the configured request size limitation.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	In the Data Access GUI, go to the Application Configuration tab. Expand the EGI config group. Set the "MAX_GRANS_SINGLE_ASYNC_REQUEST" parameter to a value low enough that you can submit a request with more granules than this limit. Note that MAX_GRANS_FOR_SYNC_REQUEST is already set to 1. This value should not be changed.	Verify that it is possible to configure an asyncronous request size limitation. And that the synchronous request size limitation is hard coded to 1.	
2	In order to submit the requests listed in the below steps, use the ESI interface to drill down, select granule(s), and build a request. After clicking submit on the form, you will be presented with a number of links. Click the EGI link for synchronous requests, or copy/paste the provided wget command line into a terminal for async requests.		#comment
3	Submit several asynchronous web requests with granule counts not exceeding the configured "MAX_GRANS_SINGLE_ASYNC_REQUEST" value. Ensure that the requests will take enough time to complete to allow for	Verify that providing an email address can be included in an asynchronous web request.	

#	Action	Expected Result	Notes
	monitoring. Ensure that at least one but not all requests specify email notification.		
4	View the immediate response to the request, either in the browser or on the command line, depending on how the request was submitted.	Verify that all of the requests receive request tracking information in the immediate request response, regardless of whether an email address is specified in the request.	
5	Check the inbox of the email address that was provided in the requests.	Verify that all of the requests specifying email notification receive a request acceptance email which includes request identification and information on the parameters used to process the request.	
6	Go to the Monitoring tab of the Data Access GUI	Verify that the Data Access GUI displays all of the requests currently in progress including which backend services are being used for the request, number of granules, and current state.	
7	Wait a few minutes and then once again check the inbox for the email address used in the requests	Verify that all of the requests specifying email notification receive an email on completion of the request which includes the request identification and download link(s).	
8	Check the Monitoring tab of the Data Access GUI	Verify that all requests below the request size limitation complete successfully.	
9	Submit one asynchronous web request which exceeds the configured request size limitation. Note that for this request you have to submit this Asynchronous request through the webpage by clicking on the "Submit POST to EGI" button.	Verify that the request exceeding the request size limitation is rejected with an appropriate error message and not queued for processing.	

TEST DATA:

EXPECTED RESULTS:

Verify that it is possible to configure a request size limitation.
Verify that providing an email address can be included in an asynchronous web request.

Verify that the Data Access GUI displays all of the requests currently in progress including which backend services are being used for the request, number of granules, and current state.
Verify that all requests below the request size limitation complete successfully.
Verify that the request exceeding the request size limitation is rejected with an appropriate error message and not queued for processing.
Verify that all of the requests specifying email notification receive a request acceptance email which includes a request identification.
Verify that all of the requests receive request tracking information in the immediate request response, regardless of whether an email address is specified in the request.
Verify that all of the requests specifying email notification receive an email on completion of the request which includes the request identification and download link(s).

7.5 DP_81_02_TP033 SDPS Web API suspend and resume request acceptance (ECS-ECSTC-385)

DESCRIPTION:

Test Case ID - 370
Suspend the acceptance of requests via the SDPS Web API for two collections.
Submit one request for each of the two suspended collections and submit at least one request for two collections that are not suspended.
Perform Verification Set 1.
Suspend the acceptance of all requests via the SDPS Web API.
Submit at least one request for each of the 4 collections used previously.
Perform Verification Set 2.
Resume the acceptance of requests via the SDPS Web API for the two collections.
Submit at least one request for each of the 4 collections used previously.
Perform Verification Set 3.
Resume the acceptance of all requests via the SDPS Web API.
Submit at least one request for each of the 4 collections used previously.
Perform Verification Set 4.

PRECONDITIONS:

Need granules for at least four collections that can all be processed.

STEPS:

#	Action	Expected Result	Notes
1	Suspend the acceptance of requests via the SDPS Web API for two collections. This part can be achieved by going to the Configuration GUI, and going to the Collection Configuration Tab. Select a collection that has ingested granules that appear on the ESI/EGI website. Right click on that collection, and select Disable all services. This will disable processing for the collection in question. 	Verify that after that has been done the services under that collection show as greyed out.	
2	Perform Verification Set 1. Suspend the acceptance of all requests via	Verify that the requests for the two	

#	Action	Expected Result	Notes
	the SDPS Web API(Configuration GUI). Submit at least one request for each of the 4 collections(2 disabled and 2 enabled) used previously.	suspended collections are rejected with an appropriate error message. Verify that the requests for the two active collections complete successfully. Verify that the Data Access GUI displays the correct status for each of the four collections (two suspended, two active).	
3	Perform Verification Set 2. Resume(Enable) the acceptance of requests via the SDPS Web API for the two collections(the ones that were disabled in Step 2). Submit at least one request for each of the 4 collections used previously.	Verify that all requests have passed with an appropriate error message. Verify that the Data Access GUI displays the acceptance of all processing requests as suspended.	
4	Perform Verification Set 3. Go back to the Config GUI and disable all four collections(see step 1). Resume the acceptance of all requests via the SDPS Web API/EGI. Submit at least one request for each of the 4 collections used previously.	Verify that all requests are rejected with an appropriate error message. Verify that the Data Access GUI displays the acceptance of all processing requests as suspended.	
5	Perform Verification Set 4. Go back to the Config GUI and enable the four disabled collections in Step 4. Run 4 ESI/EGI requests on granules that are part of that collection.	Verify that the requests for all four collections complete successfully. Verify that the Data Access GUI displays all collections as active and the acceptance of processing requests as active.	

TEST DATA:

EXPECTED RESULTS:

Verification Set 1:
Verify that the requests for the two suspended collections are rejected with an appropriate error message.
Verify that the requests for the two active collections complete successfully.
Verify that the Data Access GUI displays the correct status for each of the four collections (two suspended, two active).
Verification Set 2:
Verify that all requests are rejected with an appropriate error message.
Verify that the Data Access GUI displays the acceptance of all processing requests as suspended.
Verification Set 3:
Verify that all requests are rejected with an appropriate error message.

Verify that the Data Access GUI displays the acceptance of all processing requests as suspended.
Verification Set 4:
Verify that the requests for all four collections complete successfully.
Verify that the Data Access GUI displays all collections as active and the acceptance of processing requests as active.

7.6 DP_81_02_TP035 SDPS Web API monitoring, management, and reporting (ECS-ECSTC-386)

DESCRIPTION:

Test Case ID - 390
Submit requests to the SDPS Web API such that there are requests processing throughout the duration of the test.
Perform verification set 1.
Cause the HEG service to be unavailable.
Perform verification set 2.
Cause the HEG service to be available again.
Verification set 3:
Cause the HEG service to be available again.

PRECONDITIONS:

S-MSS-x0010 The SDPS System Monitoring, Event Detection, and Response Service shall allow DAAC staff to monitor the capacity demands for internal resources that are used to process web access requests submitted via the SDPS Web API [DESIRABLE: separately for accesses from ECHO/Reverb vs. others.].
S-MSS-x0020 The SDPS System Monitoring, Event Detection, and Response Service shall allow DAAC staff to monitor the status (e.g., availability) of internal resources that are required to process web access requests submitted via the SDPS Web API.
S-MSS-x0030 The SDPS System Monitoring, Event Detection, and Response Service shall be able to alert DAAC staff when internal resources that are required to process web access requests submitted via the SDPS Web API have become unavailable and clear such alerts when the availability of such resources has been restored.

STEPS:

#	Action	Expected Result	Notes
1	Submit requests to the SDPS Web API such that there are requests processing throughout the duration of the test. For instance, submit a bunch or requests beforehand to build a queue, or have a script running which trickles requests in while performing the test. The idea is to replicate a normal operational environment.	Verify that requests are queued and processing in teh HEG service. Verify that it is possible in hyperic to: -monitor the number of requests in progress for the HegService. -monitor the number of queued requests -monitor the average completion time for requests -monitor the cpu and memory usage of the hosts where HegService, esi, egi, and other data access services are	

#	Action	Expected Result	Notes
		running. 	
2	Cause the HEG service to be unavailable using either the tomcat manager or thru Hyperic.	Verify that a hyperic alert is generated within one minute of the HEG service becoming unavailable and that: -The HEG Service status is displayed as unavailable in the Hyperic GUI. -An email has been dispatched and delivered from Hyperic. - Shut down the hyperic Agent 	
3	Cause the HEG service to be available again using either the tomcat manager or thru Hyperic.	Verify that once started, the HEG service is shown as started in Hyperic with 100% availability. Verify that the alert is marked as fixed in the Hyperic GUI within one minute of the HEG service becoming available and that: -The HEG Service status is displayed as available in the Hyperic GUI. -An email has been dispatched and delivered from Hyperic indicating that the HEG Service is now available. -Bring the hyperic Agent up	

TEST DATA:

EXPECTED RESULTS:

Use Hyperic for the verification steps
Verification set 1:
Verify that it is possible to monitor the number of HEG requests in progress.
Verify that it is possible to monitor the memory usage and CPU usage of the host on which the HEG service, esi, egi, and other services are running.
Verification set 2:
Verify that an alert is generated within one minute of the HEG service becoming unavailable.
Verify that the HEG status is displayed as unavailable.
Verification set 3:
Verify that the alert is marked as fixed within one minute of the HEG service becoming available.
Verify that the HEG status is displayed as available.

7.7 DP_81_02_TP039 SDPS Web API capacity management and tuning (ECS-ECSTC-388)

DESCRIPTION:

Test Case ID - 375
In the Data Access GUI, go to the “Application Configuration” tab.
Find the parameter “MAX_GRANS_SINGLE_ASYNC_REQUEST”. Set this parameter to 5.
Find the parameter “MAX_GRANS_QUEUE_ASYNC_REQUESTS”. Set this parameter to 9.
Find the parameter “MAX_GRANS_FOR_CHEAP_REQUEST”. Set this parameter to 4.
Find the parameter “MAX_GRANS_FOR_SYNC_REQUEST”. Set this parameter to 1.
In the Data Access GUI, go to the “Service Configuration” tab.
Double click on the service being tested (e.g. HEG).
Set the “Max Synchronous Jobs” and “Max Asynchronous Jobs” both to 2.
A) Submit an asynchronous request for 6 granules.
B) Submit a synchronous request for 2 granules.
C) Submit an asynchronous request for 5 granules, followed immediately by another request for 5 granules and by another synchronous request for 1 granule. Ensure that the all requests are received before the first one completes processing. You may need to choose processing options which are known to take a considerable amount of time.
D) Submit an asynchronous request for 5 granules, followed immediately by another request for 4 granules.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	In the Data Access GUI, go to the “Application Configuration” tab. Click to expand the "EGI" configuration group. Find the parameter “MAX_GRANS_SINGLE_ASYNC_REQUEST”. Set this parameter to 5. Find the parameter “MAX_GRANS_QUEUE_ASYNC_REQUESTS”. Set this parameter to 6. Find the parameter “MAX_GRANS_FOR_CHEAP_REQUEST”. Set this parameter to 4. Find the parameter “MAX_GRANS_FOR_SYNC_REQUEST”. Set this parameter to 1.	The specified config parameters should now be set.	
2	In the Data Access GUI, go to the “Service Configuration” tab. Double click on the service being tested (e.g. HEG). Set the “Max Synchronous Jobs” and “Max Asynchronous Jobs” both to 2.	The specified config parameter should now be set.	
3	In order to submit the requests listed in the below steps, use the ESI interface to drill down, select granule(s), and build a request. After clicking submit on the form, you will be presented with a number of links. Click the EGI link for synchronous requests, or click the EGI POST button for async requests.		#comment

#	Action	Expected Result	Notes
4	Submit an asynchronous request for 6 granules.	Verify that the request receives an error which indicates that the request exceeds the granule limit for ESI requests.	
5	Submit a synchronous request for 2 granules.	Verify that the request receives an error which indicates that the request exceeds the granule limit for synchronous ESI requests and will need to be resubmitted as an asynchronous request.	
6	Submit three asynchronous request for 5 granules each(each request must be sibmitted 10-15 seconds later), followed by another synchronous request for 1 granule. Ensure that the all requests are received before the first one completes processing. You may need to choose processing options which are known to take a considerable amount of time.	Verify that the first request is successfully added to the processing queue (and eventually is processed and results are returned). Verify that the second and/or third request receives an error specifying that the maximum number of queued granules has been exceeded and the request should be retried later. Verify that the fourth (sync) request is successfully processed and results in the url(s) to download the resulting files being returned.	
7	Submit an asynchronous request for 5 granules, followed immediately by another request for 4 granules.	Verify that both requests are successful, and that the first request is put on the queue for larger/slower requests and the second request is put on the queue for faster requests(this can be done by sorting the StartDate column in the Data Access Config GUI). The second request should complete before the first, or if not, shall at least have some of its granules processed without having to wait for the first request to complete.	

TEST DATA:

EXPECTED RESULTS:
A) Verify that the request in step A) receives an error which indicates that the request exceeds the granule limit for ESI requests.
B) Verify that the request in step B) receives an error which indicates that the request exceeds the granule limit for synchronous ESI requests and will need to be resubmitted as an asynchronous request.
C) Verify that the first request in step C) is successfully added to the processing queue (and eventually is processed and results are returned). Verify that the second request in step C) receives an error specifying that the maximum number of queued granules has been exceeded and the request should be retried later. Verify that the third request in step C) is successfully processed and results in the url(s) to download the resulting files being returned.
D) Verify that both requests in step D) are successful, and that the first request is put on the queue for larger/slower requests and the second request is put on the queue for faster requests. The second request should complete before the first, or if not, shall at least have some of its granules processed without having to wait for the first request to complete.

7.8 DP_81_02_TP038 SDPS Web API enable and disable processing for collections (ECS-ECSTC-389)

DESCRIPTION:
Test Case ID - 87
A)
Disable all of the processing options for a collection.
Submit one request require processing for the collection that has been disabled and one request for a collection for which processing is enabled.
B)
Enable the processing options for the previously disabled collection.
Submit a request requiring processing for that collection.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Open the Data Access Configuration Interface GUI, and select a HEGable valid collection((while in the Collection Configuration tab)(that has ingested granules) in the mode that you are working in	The collection has been selected.	
2	Right click on the collection and select Disable All Mapped Services. This will disable that collection for processing.	The collection has been disabled. Verify that after right clicking again, the Disable All Mapped Services cannot be selected again.	
3	Go to the ESI/EGI website, and go to the collection, and select a single egi granule processing that is part of that collection. Execute the autogenerated wget statement in any box in the testing environment	Verify that since the collection is disabled an error has been observed specifying that the collection is	

#	Action	Expected Result	Notes
		disabled and no processing can be done on the granule.	
4	Perform verification set 2. Enable the processing options(see step 1) for the previously disabled collection. Submit another egi request for a granule part of the same collection requiring processing for that collection.	Verify that the request against the collection which was previously disabled in step 2 completes successfully.	

TEST DATA:

EXPECTED RESULTS:
A)
Verify that the request against the enabled collection in step A) completes successfully.
Verify that the request against the disabled collection in step A) is rejected with an appropriate error indication IAW the SDPS-ECHO Web Interface ICD.
B)
Verify that the request submitted in step B) against the collection which was previously disabled in step A) completes successfully.

7.9 DP_81_02_TP018 Ordering with Processing Instructions (ECS-ECSTC-379)

DESCRIPTION:
Test Case ID - 200
Submit one FTP Pull and one FTP push order using Reverb such that processing is requested for granules from two different collections. Ensure that a different backend service is to be invoked for the two different collections. Ensure one of the orders includes granules for processing and also granules for which no processing is to be performed.
Note: For Increment III it is sufficient to use a simulator to submit the orders rather than Reverb, but in Increment IV Reverb must be used.

PRECONDITIONS:
Collections must be configured such that the two collections use different backend services for processing.
Granules must be available for ordering for the two collections - if using Reverb the granules must be exported to ECHO.
Performance logging needs to be enabled and the operations log level must be set appropriately to verify logging requirements.

STEPS:

#	Action	Expected Result	Notes
1	To perform this test select two collections. One collection should be HEGABLE and the other collection should be processable by GDAL. Choose 2 granules from the HEGABLE Collection and 3 granules from the GDAL Collection	The Collections and granules should be selected	

#	Action	Expected Result	Notes
2	Create an FTP Push order for One granule from the HEGABLE Collection and one granule from the GDAL collection. Apply HEG processing to the first granule and GDAL processing(convert to JPEG) to the second granule. Create an FTP Pull order for the remaining 3 granules. Apply HEG processing to the remaining HEGABLE Granule. Apply GDAL processing(convert to JPEG) to one on the remaining granules and no processing on the last granule.	For Increment 3, We create 2 request properties files that have the appropriate information.	
3	Submit The FTP Push Request. For increment 3 the command would look like EcDmEwocTestClientStart DEV01 f4eil01 22500 FTPPushRequest.properties n Verify that the orders are submitted to EWOC and the processing instructions are received. 	Check that you get the an Order ID back and the following: SubmitAcknowledgement.SubmitAcceptance is present SubmitAcknowledgement.OrderRejection is not present SubmitAcknowledgement.ProviderTrackingId : <orderid eg.0600114795> SubmitAcknowledgement.StatusInformation : Order received	
4	Verify that EWOC submits both orders to OMS in the same fashion (ie they are both placed in the same queue, the orders are not treated differently by EWOC as a result of using two different backend services).	In the OMS GUI, verify that only one request is created and that the request details show both granules.	
5	Verify that OMS submits all processing requests via the SDPS-DAAC Internal On-Demand Processing API.	Check in the OMS logs that it calls the Internal processing requests for HEGService and GDAL	
6	Verify that all granules complete successfully and the orders complete successfully.	In the OMS GUI Check that the request goes to the Shipped state	
7	Verify that the outputs for the FTP push request are successfully delivered after all of the granules have been processed.	Check the FTP push destination and make sure all files are there.	
8	Verify that the distribution request size for the order accounts for the total size of the output data once the processing has completed (i.e. for processed granules ensure that the size of the outputs is used to calculate the order size and not the size of the inputs).	Add up the sizes of the files in the FTP push destination for the granules. Make sure the size is equal to the output size of the granule stated in the OMS GUI	
9	Verify that any temporary files OMS created in support of processing are cleaned up after the orders have been successfully processed. NOTE: that OMS may not generated temp files for processing requests, in which case this step can be ignored.	Find what temporary files OMS creates and make sure the are no longer there after the request is shipped.	
10	Verify the following operations are logged - dispatching of a processing	Check the OMS log file and make sure	

#	Action	Expected Result	Notes
	request, completion of a processing request, starting work on a distribution request involving processing, completing work on a distribution request involving processing, sending a DN for a distribution request involving processing	all steps mentioned are logged. (log file usually located at /usr/ecs/DEV01/CUSTOM/logs/EcOmOrderManager.ALOG) 	
11	Verify all items listed in S-OMS-x0370 are logged for each processing request.	Check the OMS log file and make sure all items are logged.	
12	Verify entries are present in the performance log which can be used to determine the execution time of individual processing requests.	Check the OMS performance log. Make sure you can tell how long it took to process the individual requests.	
13	Submit The FTP Pull Request. For increment 3 the command would look like EcDmEwocTestClientStart DEV01 f4eil01 22500 FTPPullRequest.properties n Verify that the orders are submitted to EWOC and the processing instructions are received.	Check that you get the an Order ID back and the following: SubmitAcknowledgement.SubmitAcceptance is present SubmitAcknowledgement.OrderRejection is not present SubmitAcknowledgement.ProviderTrackingId : <orderid eg.0600114795> SubmitAcknowledgement.StatusInformation : Order received	
14	Verify that EWOC submits both orders to OMS in the same fashion (ie they are both placed in the same queue, the orders are not treated differently by EWOC as a result of using two different backend services).	In the OMS GUI, verify that only two request s are created and that one request details show both granules that have processing applied and the other request has the granule without processing.	
15	Verify that OMS submits all processing requests via the SDPS-DAAC Internal On-Demand Processing API.	Check in the OMS logs that it calls the Internal processing requests for HEGService and GDAL	
16	Verify that all granules complete successfully and the orders complete successfully.	In the OMS GUI Check that both requests go to the Shipped state	
17	Verify that a DN is delivered once the processing of all granules has completed for the FTP pull order (and verify that the DN is not queued until all processing has completed).	Check that you receive an email for each of the 2 requests stating that your order is complete. The 2 requests should have the same orderid Verify that the DN email contains the input granule ID and URL where the user can obtain the status of the	

#	Action	Expected Result	Notes
		order.	
18	Verify that the distribution request size for the order accounts for the total size of the output data once the processing has completed (i.e. for processed granules ensure that the size of the outputs is used to calculate the order size and not the size of the inputs).	Add up the sizes of the files in the FTP pull location for the granules. Make sure the size is equal to the output size of the granule stated in the OMS GUI	
19	Verify that any temporary files OMS created in support of processing are cleaned up after the orders have been successfully processed.	Find what temporary files OMS creates and make sure the are no longer there after the request is shipped.	
20	Verify the following operations are logged - dispatching of a processing request, completion of a processing request, starting work on a distribution request involving processing, completing work on a distribution request involving processing, sending a DN for a distribution request involving processing	Check the OMS log file and make sure all steps mentioned are logged.	
21	Verify all items listed in S-OMS-x0370 are logged for each processing request. 	Check the OMS log file and make sure all items are logged.	
22	Verify entries are present in the performance log which can be used to determine the execution time of individual processing requests.	Check the OMS performance log. Make sure you can tell how long it took to process the individual requests.	

TEST DATA:

EXPECTED RESULTS:

Verify that the orders are submitted to EWOC and the processing instructions are received.
Verify that EWOC submits both orders to OMS in the same fashion (ie they are both placed in the same queue, the orders are not treated differently by EWOC as a result of using two different backend services).
Verify that OMS submits all processing requests via the SDPS-DAAC Internal On-Demand Processing API.
Verify that all granules complete successfully and the orders complete successfully.
Verify that a DN is delivered once the processing of all granules has completed for the FTP pull order (and verify that the DN is not queued until all processing has completed).
Verify that the DN contains the input granule ID and URL where the user can obtain the status of the order.
Verify that the outputs for the FTP push request are successfully delivered after all of the granules have been processed.
Verify that the distribution request size for both orders accounts for the total size of the output data once the processing has completed (i.e. for processed granules ensure that the size of the outputs is used to calculate the order size and not the size of the inputs).
Verify that any temporary files OMS created in support of processing are cleaned up after the orders have been successfully processed.

Verify the following operations are logged - dispatching of a processing request, completion of a processing request, starting work on a distribution request involving processing, completing work on a distribution request involving processing, sending a DN for a distribution request involving processing
Verify all items listed in S-OMS-x0370 are logged for each processing request.
Verify entries are present in the performance log which can be used to determine the execution time of individual processing requests.

7.10 GDAL

7.10.1 DP_81_02_TP050 GDAL processing requests and errors (ECS-ECSTC-483)

DESCRIPTION:
Submit requests that will result in a variety of GDALService statuses.

Examples:
a) Conversion of a granule to a supported format.
b) Conversion of a granule with band subsetting.
c) Conversion of a granule with spatial subsetting.
d) Conversion of a granule to an unsupported format.
e) Conversion takes too long. Time out case
f) Conversion granule/file does not exist.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Note, requests in this test can be submitted via EGI, ESI or directly to the GDALService. The easisest way to build requests is to use the ESI inventory drilldown and then use the links provided.		#comment
2	Submit a request for conversion of a granule to a supported format like Gtiff.	The request should succeed and the converted files should appear in the results. Verify the converted files have names that identify the data-object component processed (and not sequential number or other non-identifying naming convention).	
3	Submit a request for conversion of a granule to Gtiff format with band subsetting. Select one or multiple bands.	The request should succeed and the converted files should appear in the results. The number of files should	

#	Action	Expected Result	Notes
		correspond to the number of bands you selected. Verify the converted files have names that identify the data-object component processed (and not sequential number or other non-identifying naming convention).	
4	Submit a request for conversion of a granule to Gtiff format with spatial subsetting.	The request should succeed and the converted files should appear in the results.	
5	Submit a request for conversion of a granule to an unsupported format.	The request should fail/error out with InvalidParameterValue and a message stating that the format used was not valid.	
6	Submit a request for conversion for a granule whose files are missing from the datapool.	The request should fail/error out with InternalError and a message similar to InputHDFEOSFileNotExistErr	
7	Submit a request for conversion that will take too long to process so that a timeout occurs. You can modify the GdalService.application.timeout parameter in the GdalServiceConfig.properties file to 1 to make timeouts happen very fast. Make sure to bounce GdalService if you modify the properties file.	The request should fail/error out with InternalError and message similar to Cancelled (Timeout). Also there should not be a stray gdal_translate process for this request on the GdalService host.	

TEST DATA:

EXPECTED RESULTS:
Verify that the Internal API returns:
a) The list of converted files in the case of success or
b) A correct classification of the error and/or an error description

7.10.2 Configure GDAL collection formats - validate formats and collections (ECS-ECSTC-507)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login to f5dpl01v host as cmshared. cd /usr/ecs/<MODE>/CUSTOM/utilities		
2	./EcDlDaGdalColFmtCfgStart --host f4dbl03 --port 5431 --collectionLimit 0 --formatLimit 0		
3	cd /usr/ecs/<MODE>/data/DPL/logs. Check the formats.log.	Verify that there are no errors in the formats.log	
4	Check the number of formats logged in formats.out. Check /tools/gdal/bin/gdalinfo --formats to get the valid gdal formats.	Verify that the formats.out has formats that match the supported gdal formats	current version of gdal supports 111 formats
5	Check collections.log. QA,PH, DAP collections and Any input files with the following filetypes are skipped: .ph, .dap, .DAT, .bin, .xml, .tar	Verify that there are no errors in collections.log	
6	Check the number of collections that are logged in collections.out. select distinct esdt(shortname, versionid) from amgranule; gives number of unique collections with granules in the amgranule table. QA,PH, DAP collections are skipped. grep Invalid collections.out returns those 3 skipped ancillary collections. Any input files with the following filetypes are skipped: .ph, .dap, .DAT, .bin, .xml, .tar. grep # collections.out returns the skipped filetypes.	The number of entries in collections.out should match the collections in amgranule table. Verify that the collections.out has all the collections that have granules excluding the ancillary collections and the unsupported filetypes	
7	cd /usr/ecs/<MODE>/CUSTOM/logs Check matrix....log	Verify that the matrix...log does not have any errors	

TEST DATA:

EXPECTED RESULTS:

7.10.3 Configure GDAL collection formats - configure single collection and single format (ECS-ECSTC-509)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test collection is installed with granules that are not deleted.		

#	Action	Expected Result	Notes
	e.g. MIL3DAE.004 in mode OPS		
2	Login to f5dpl01v host as cmshared. cd /usr/ecs/OPS/CUSTOM/utilities. Note current time as t0.		
3	./EcDlDaGdalColFmtCfgStart --mode OPS --host f4dbl03 --port 5431 --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats --retainHistory		
4	cd /usr/ecs/OPS/CUSTOM/logs. Check the matrix...log	Verify that the matrix...log does not have errors. Verify that matrix...log completes with a report indicating completion of the collection format configuration run	
5	cd /usr/ecs/OPS/CUSTOM/data/DPL Check output logs generated.	Verify that no collections.out or formats.out are generated in the output directory	
6	Check the amdagranuleformatconfig has the results of gdal_translate. for only the one collection and format with a starttime is greater than t0. select * from amdagranuleformatconfig where granuleid in (select granuleid from amgranule where collectionid = (select collectionid from amcollection where shortname='AE_Ocean' and versionid=2)); 	Verify that the amgranuleformatconfig table has an entry for only the one collection and format with a starttime that is greater than t0	
7	Check amdaformatsrvxref for the collection and format. select * from amdaformatsrvxref where formatid = (select formatid from amdaformat where name = 'GTiff') and serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that amdaformatsrvxref has an entry for the format being configured for gdal service	
8	select * from amdacolformatsrvxref where collectionid in (select collectionid from amcollection where shortname = 'MIL3DAE' and versionid = 4) and serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that the amdacolformatsrvxref has an entry for the collection and format and enabledflag = 'Y'	
9	select * from amdadatasetconfig where collectionid in (select collectionid from amcollection where shortname = 'MIL3DAE' and versionid = 4) and serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that the amdadatasetconfig has an entry for the collectionid and gdal serviceid and enabledflag = 'N'. Verify that the enablespatialFlag, enablebandFlag and enableformatFlag = 'Y'. Verify that all the other flags are set to 'N'	
10	Click on the DataAccess GUI, Service Configuration tab, open GDAL folder.	Verify that the collection is displayed under the GDAL folder.	

#	Action	Expected Result	Notes
11	Click on the Service Configuration tab, when you click on the collection under GDAL=> Configure Service: GDAL for Collection. Select Enabled formats.	Verify that the successful gdal formats for the collection are displayed under Selected Formats	

TEST DATA:

EXPECTED RESULTS:

7.10.4 Configure GDAL collection formats - configure all collections for supported gdal formats (ECS-ECSTC-519)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login into Data Access GUI http://f5dpl01v:22500/DataAccessGui/		
2	Select Service configuration tab. Under Services - Collections, click on GDAL folder. On the Edit Service:GDAL, Ensure that the hostname is set to f5dpl01v and the service url is http://f5dpl01v:22500/GdalService		
3	On the Service configuration tab. Ensure that band subsetting is enabled, spatial subsetting is enabled and temporal subsetting is disabled.		
4	On the Service Configuration tab. Under Services - Collections, click and open GDAL folder. The collections mapped to GDAL will be displayed. Right click on GDAL icon and select Unmap all collections.		
5	Run ./EcDlDaGdalColFmtCfgStart --mode OPS --host f4dbl03 --port 5431		
6	cd usr/ecs/OPS/CUSTOM/data/DPL . Check formats.out for the valid gdal formats	Verify that the formats.out has all the supported gdal formats	
7	Check the granuleids for all the collections in the mode from collections.out	Verify that the collections.out has a granuleid for every collection with a valid granule file in the mode	
8	Tail the output log to check for errors and progress of the run. cd to the logs directory for the mode. tail -f matrix.....log	Verify that matrix...log completes with a report indicating completion of the collection format configuration run 	

#	Action	Expected Result	Notes
9	Check amdaformatsrvxref table. select formatid from amdaformatsrvxref where serviceid in (select serviceid from amdaservice where name = 'GDAL')	Verify that there is an entry in amdaformatsrvxref for every valid gdal format.	
10	Check amdagranuleformatconfig table. select granuleid, formatid from amdagranuleformatconfig	Verify that there is an entry for each valid esdt in the mode indicated in collections.out and each gdal supported format in the amdagranuleformatconfig table.	if the location of the datafilee changed during publish or unpublish or a deletion after the start of the run, those granules will be skipped.
11	If there is a master table from previous runs with collection, format matrix , compare the results of amdagranuleformatconfig with the master table	Any successful collection format combination should not have a fail in amdagranuleformatconfig.	
12	Check the datapool directory. select getdatafile(granuleid) for the granuleid used.	Verify that no aux.xml files are generated in the datapool directories.	
13	Check amdacolformatsrvxref. select granuleid, formatid from amdagranuleformatconfig where success = 't'); select collectionid, formatid, serviceid from amdacolformatsrvxref where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that there is an entry in amdacolfmtsrvxref for each collection and format that succeeded in amdagranuleformatconfig	
14	select enabledflag from amdacolformatsrvxref where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that all the gdal service format entries have enabledflag = 'Y' for every successful collection format in amdagranuleformatconfig	
15	Check amdadatasetconfig table. select collectionid from amgranule where granuleid in (select granuleid from aim.amdagranuleformatconfig where success = 't'); select collectionid from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that there is an entry in amdadatasetconfig for every successful granule format entry in amdagranuleformatconfig.	
16	select allowtemporalflag, allowspatialflag, allowbandflag from amdaservice where name = 'GDAL'; select allowtemporalflag, allowspatialflag, allowbandflag from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify the flags are inherited from the amdaservice table.	
17	select formatflag, enabledflag from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that the formatflag = 'Y' and enabledflag = 'N' for all entries in the table	
18	select reprojectflag, resamplingflag, interpolationflag from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that all these flags are set to 'N'	
19	Click on the DataAccess GUI, Service Configuration tab, open GDAL folder.	Verify that all collections that are under the GDAL folder are displayed disabled.	

#	Action	Expected Result	Notes
20	Click Service Configuration tab, Click GDAL => Edit Service. Click on the Enabled formats	Verify that all successful gdal formats for any collection are displayed in the selected formats panel.	
21	Click on the Service Configuration tab, when you click on a collection under GDAL=> Configure Service: GDAL for Collection. Select Enabled formats.	Verify that the successful gdal formats for the collection are displayed under Selected Formats and all unsuccessful gdal formats for the collection are displayed under the Available formats panel.	

TEST DATA:

EXPECTED RESULTS:

7.10.5 Configure GDAL collection formats - options (ECS-ECSTC-525)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that four test collections are installed with granules that are not deleted. e.g. MIL3DAE.004, MIL3MAE004 in mode OPS		
2	Login to f5dpl01v host as cmshared. cd /usr/ecs/OPS/CUSTOM/utilities. Note current time as t0.		
3	#--noCollections and --noFormats option		
4	./EcDlDaGdalColFmtCfgStart --mode OPS --host f4dbl03 --port 5431 --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats		
5	cd /usr/ecs/OPS/CUSTOM/logs. Check the matrix...log	Verify that the matrix...log does not have errors. Verify that matrix...log completes with a report indicating completion of the collection format configuration run	

#	Action	Expected Result	Notes
6	cd /usr/ecs/OPS/CUSTOM/data/DPL Check output logs generated.	Verify that no collections.out or formats.out are generated in the output directory	
7	Check the amdagranuleformatconfig has the results of gdal_translate for the test collection and format	Verify that the amgranuleformatconfig table has an entry for the test collection and format with a starttime that is greater than t0	
8	#--retainHistory option		
9	Record current time as t1		
10	./EcDlDaGdalColFmtCfgStart --mode OPS --port 5431 --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats --retainHistory		
11	Check amdagranuleformatconfig. select * from amdagranuleformatconfig where starttime > t0	Verify that there are 2 entries in the table for the test collection and format	
12	select * from amdagranuleformatconfig where starttime > t1	Verify that there is one entry in the table with a starttime greater than t1	
13	# --directory option		
14	Ensure the output directory exists..eg. /workingdata/OPS		
15	./EcDlDaGdalColFmtCfgStart --mode OPS --port 5431 --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats --directory /workingdata/OPS	Verify that the matrix*.out and the matrix*.log files , GDAL_TRANSLATE and GDAL_TRANSLATE_OUT directories are created under /workingdata/OPS	
16	# --timeout option		
17	Check the duration d (in millisecs) for the granule from amdagranuleformatconfig.		
18	Select a timeout t to be about half the duration d. The timeout is in secs. e.g. 3		
19	./EcDlDaGdalColFmtCfgStart --mode OPS --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats --timeout 30	Verify that amdagranuleformatconfig has an entry for the collection and format with success = 'f'	
20	# --preview		
21	Ensure that there is no entry for the collection and format in the amdacolformatsrvxref table		
22	./EcDlDaGdalColFmtCfgStart --mode OPS --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats --preview	Verify that the amdagranuleformatconfig table has a successful entry for the collection and	

#	Action	Expected Result	Notes
		format and there is no update to the amdacolformatsrvxref table	
23	#--retainConfig		
24	On the DataAccess GUI, enable test collection MIL3DAE for GDAL service	Verify that the amdadatasetconfig table has enabledflag = 'Y"	
25	./EcDlDaGdalColFmtCfgStart --mode OPS --port 5431 --singleCollection MIL3DAE.004 --singleFormat GTiff --noCollections --noFormats --retainConfig	Verify that the test collection is still enabled in the GUI and the amdadatasetconfig table has the enabledflag = 'Y'	
26	#--collectionLimit and --formatLimit		
27	./EcDlDaGdalColFmtCfgStart --mode OPS --port 5431 --collectionLimit 2 --formatLimit 2 --retainConfig	Verify that the formats.out and collections.out are created. Verify that the matrix*.log indicates 2 collections and 2 formats were processed. Verify that output files in the GDAL_TRANSLATE directory are deleted	use --retainConfig to not delete the configuration of all other collections
28	#--noDelete		
29	./EcDlDaGdalColFmtCfgStart --mode OPS --port 5431 --collectionLimit 2 --formatLimit 2 --noDelete --retainConfig	Verify that the formats.out and collections.out are created. Verify that the matrix*.log indicates 2 collections and 2 formats were processed. Verify that output files in the GDAL_TRANSLATE directory contents are NOT deleted	
30	#--database host and port		
31	Logon to the p5dpl01 as cmshared. Run the format script with database host and port options	Verify that the matrix*.log indicates that the format config script ran successfully and there are no errors	
32	./EcDlDaGdalColFmtCfgStart mode OPS --port 5431 --singleCollection MIL3DAE.004 --singleFormat GTiff ----host p4dbl03 --port 5431 --mode TS2		

TEST DATA:

EXPECTED RESULTS:

7.10.6 Configure GDAL collection formats - configure single collection for all supported gdal formats (ECS-ECSTC-526)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login into Data Access GUI http://f5dpl01v:22500/DataAccessGui/		
2	Select Service configuration tab. Under Services - Collections, click on GDAL folder. On the Edit Service:GDAL, Ensure that the hostname is set to f5dpl01v and the service url is http://f5dpl01v:22500/GdalService		
3	On the Service configuration tab. Ensure that band subsetting is enabled, spatial subsetting is enabled and temporal subsetting is disabled.		
4	On the Service Configuration tab. Under Services - Collections, click and open GDAL folder. The collections mapped to GDAL will be displayed. Right click on GDAL icon and click on Remove Service from collection	Verify that there are no entries for this collection in amdadatasetconfig and amdacolformatsrvxref	
5	Run ./EcDlDaGdalColFmtCfgStart --mode OPS --host f4dbl03 --port 5431 --singleCollection MIL3DAE.004 --noCollections --retainConfig		
6	cd /workingdata/<MODE>/data/DPL . Check formats.out for the valid gdal formats	Verify that the formats.out has all the supported gdal formats	
7	Check collections.out	Verify that the collections.out is not updated.	
8	Tail the output log to check for errors and progress of the run. cd to the logs directory for the mode. tail -f matrix.....log	Verify that matrix...log completes with a report indicating completion of the collection format configuration run 	
9	Check amdagranuleformatconfig table. select granuleid, formatid from amdagranuleformatconfig	Verify that there is an entry for the test granule in the collection for each each gdal supported format in the amdagranuleformatconfig table.	
10	Check amdacolformatsrvxref. select granuleid, formatid from amdagranuleformatconfig where success = 't'); select collectionid, formatid, serviceid from amdacolformatsrvxref where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify that there is an entry in amdacolfmtsrvxref for each collection and format that succeeded in amdagranuleformatconfig for this collection	

#	Action	Expected Result	Notes
11	select enabledflag from amdacolformatsrvxref where serviceid = (select serviceid from amdaservice where name = 'GDAL') and collectionid = (select collectionid from amgranule where shortname = 'MIL3DAE' and verisonid = 4);	Verify that all the gdal service format entries have enabledflag = 'Y' for every successful format for this collection	
12	Check amdadatasetconfig table. select collectionid from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL') and collectionid = (select collectionid from amgranule where shortname = 'MIL3DAE' and verisonid = 4);	Verify that there is an entry in amdadatasetconfig for the test collection	
13	select allowtemporalflag, allowspatialflag, allowbandflag from amdaservice where name = 'GDAL'; select allowtemporalflag, allowspatialflag, allowbandflag from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL');	Verify the flags are inherited from the amdaservice table.	
14	select formatflag, enabledflag from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL') and collectionid = (select collectionid from amgranule where shortname = 'MIL3DAE' and verisonid = 4);;	Verify that the formatflag = 'Y' and enabledflag = 'N' for the test collection	
15	select reprojectflag, resamplingflag, interpolationflag from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'GDAL') and collectionid = (select collectionid from amgranule where shortname = 'MIL3DAE' and verisonid = 4);;	Verify that all these flags are set to 'N'	
16	Click on the DataAccess GUI, Service Configuration tab, open GDAL folder.	Verify that the test collection under the GDAL folder is displayed disabled.	
17	Click on the Service Configuration tab, when you click on a collection under GDAL=> Configure Service: GDAL for Collection. Select Enabled formats.	Verify that the successful gdal formats for the test collection are displayed under Selected Formats	

TEST DATA:

EXPECTED RESULTS:

7.10.7 Configure GDAL collection formats - configure limited collections - with granule state changes (ECS-ECSTC-527)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login into Data Access GUI http://f5dpl01v:22500/DataAccessGui/		
2	Select Service configuration tab. Under Services - Collections, click on GDAL folder. On the Edit Service:GDAL, Ensure that the hostname is set to f5dpl01v and the service url is http://f5dpl01v:22500/GdalService		
3	On the Service configuration tab. Ensure that band subsetting is enabled, spatial subsetting is enabled and temporal subsetting is disabled.		
4	Run ./EcDlDaGdalColFmtCfgStart --mode OPS --host f4dbl03 --port 5431 --collectionLimit 0 --formatLimit 0 (assuming OPS mode)		
5	cd /workingdata/OPS/data/DPL .	Verify that the formats.out has all the supported gdal formats	
6	Check the granuleids for all the collections in the mode from collections.out	Verify that the collections.out has a granuleid for every collection with a valid granule file in the mode	
7	Tail the output log to check for errors and progress of the run. cd to the logs directory for the mode. tail -f matrix.....log	Verify that matrix...log completes with a report indicating completion of the collection format configuration run 	
8	Pick three granuleids from collections.out from the top 10 collections. Unpublish a granule that is public, Publish a hidden granule and logically delete a third granule		
9	Run ./EcDlDaGdalColFmtCfgStart --mode OPS --host f4dbl03 --port 5431 --collectionLimit 10 --noCollections --noFormats --retainConfig (assuming OPS mode)	Verify that the collections.out and formats.out are not regenerated.	
10	Tail the output log to check for errors and progress of the run. cd to the logs directory for the mode. tail -f matrix.....log		
11	Check amdagranuleformatconfig table for the test granules	Verify that the the three granules that were published, unpublished and deleted are not in the table	if the location of the datafilee changed during publish or unpublish or a deletion after the start of the run, those granules will be skipped.

TEST DATA:

EXPECTED RESULTS:

7.10.8 Configure GDAL collections DataObjects All_Collections (ECS-ECSTC-503)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Clear out the table AmDaGranuleObjSrvXref by running "delete from AmDaGranuleObjSrvXref"		If you want to preserve the data do so by select * into AmDaGranuleObjSrvXref_temp. After this test you can clear out the table and load the old data back in.
2	Clear out the table DlDaColHdfObjectSrvXref by running "delete from DlDaColHdfObjectSrvXref"		Again if you want to preserve the existing data copy it into a temp table to reload into this one after the test is finished.
3	Navigate a browser to : http://f5dpl01v:22500/DataAccessGui_DEV07/config/datasetService/dataObjects/GDAL/All_Collections	After a couple minutes the browser will display "success"	
4	Run the following SQL to check which to verify that granules have had their SUBDATASET_NAMES extracted :		
5	select s.name, esdt(ShortName, VersionId) as esdt, objectname from AmGranule g join AmDaGranuleObjSrvXref x on g.granuleId = x.granuleId join DlHdfObjects o on x.objectid = o.objectid join AmDaService s on s.serviceid = x.serviceid order by s.name, esdt	This may return 0 rows for this initial release	
6	Run the following SQL to verify that collections were configured for SUBSETDATALAYERS processing:		
7	select esdt(ShortName, VersionId), s.name, o.objectname, f.fieldname from DlDaColHdfObjectSrvXref x join AmDaService s on x.serviceId = s.serviceId join DlHdfObjects o on o.objectId = x.objectid join DlHdfFields f	There should be multiple entries in this table. Verify that the objectname field does not contain the full hdf pathnames.	

#	Action	Expected Result	Notes
	on f.fieldId = x.fieldId join AmCollection c on c.CollectionId = x.CollectionId;		
8	If this is the first time this test is run create a baseline table using :		
9	Check to see if the baseline table exists in this mode:		
10	select count(*) from DlDaColHdfObjectSrvXref_baseline	table should exist	If table doesn't exist you can create it using "select * into dldacolhdfobjectsrvxref_baseline from dldacolhdfobjectsrvxref. All you can do is spot check the results and mark this test as complete.
11	Verify that the count of elements in the baseline table matches the count in the newly populated table.		
12	Verify that the following query does not contain any null columns:		
13	select * from (select esdt(ShortName, VersionId) as esdt, s.name, o.objectname, f.fieldname from DlDaColHdfObjectSrvXref x join AmDaService s on x.serviceId = s.serviceId join DlHdfObjects o on o.objectId = x.objectid join DlHdfFields f on f.fieldId = x.fieldId join AmCollection c on c.CollectionId = x.CollectionId) as new full outer join (select esdt(ShortName, VersionId) as esdt, s.name, o.objectname, f.fieldname from DlDaColHdfObjectSrvXref_baseline x join AmDaService s on x.serviceId = s.serviceId join DlHdfObjects o on o.objectId = x.objectid join DlHdfFields f on f.fieldId = x.fieldId join AmCollection c on c.CollectionId = x.CollectionId) as baseline on new.esdt = baseline.esdt and new.name = baseline.name and new.objectname = baseline.objectname and new.fieldname = baseline.fieldname	This shouldn't return any rows.	
14	Choose one of the collections in DlDaColHdfObjectSrvXref (e.g. MIL2ASAE.002) and bring up the DA GUI to enable band subsetting.		You may have to associate the collection with the

#	Action	Expected Result	Notes
			GDAL service. If so, all the rows in DlDaColHdfObjectSrvXref may get cleared out (NCR?). Simply navigate to http://f5dpl01v:22500/DataAccessGui_DEV07/config/datasetService/dataObjects/GDAL/ to repopulate DlDaColHdfObjectSrvXref
15	Export the capabilities for this collection and verify you can perform band subsetting in the Reverb GUI.		

TEST DATA:

EXPECTED RESULTS:

7.10.9 Configure DataObjects - options (ECS-ECSTC-528)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure DataObjects for a collection with no granules. f5dpl01v:22500/DataAccessGui/config/datasetService/dataObjects/GDAL/AE_L2A.002	Verify that no exception is thrown in the log. Verify that the web page reports success and there are no hdf objects in the hdf objects table	
2	Configure DataObjects for a collection with all granules option. f5dpl01v:22500/DataAccessGui/config/dataObjects/MOD10CM.005?useAllCollectionGranules	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
3	Configure DataObjects for a collection with sample granules.. eg. sql: delete	Verify that the browser shows success	

#	Action	Expected Result	Notes
	from AmDaSampleGranule where collectionId = 3000118315; insert into AmDaSampleGranule (collectionId, granuleId) values (3000118315, 3000594046); delete from AmDaGranuleObjSrvXref f5dpl01v:22500/DataAccessGui/config/dataObjects/Mod10CM.005?useSampleGranules	and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
4	Configure DataObjects for a collection with a granuleid : f5dpl01v:22500/DataAccessGui/config/dataObjects/Mod10CM.005?granuleId=3000594046	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
5	Configure DataObjects for a collection with esdt: e.g. sql: delete from AmDaGranuleObjSrvXref f5dpl01v:22500/DataAccessGui/config/dataObjects/Mod10CM.005 # granule is max-granuleId in collection	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
6	Configure DataObjects for a collection without servicetable: e.g. sql: select gx.* from DlGranuleHdfObjectsXref gx join aim_dev02.AmGranule gr on gx.granuleId = gr.granuleId where gr.collectionId = 3000118315 f5dpl01v:22500/DataAccessGui/config/dataObjects/Mod10CM.005&useServiceTable=false # granule is max-granuleId in collection sql: select gx.* from aim_dev02.DlGranuleHdfObjectsXref gx join aim_dev02.AmGranule gr on gx.granuleId = gr.granuleId where gr.collectionId = 3000118315	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
7	Configure DataObjects for a collection with agent name: e.g. sql: delete from AmDaGranuleObjSrvXref f5dpl01v:22500/DataAccessGui/config/dataObjects/MIL3DAE.004?agent=gdalinfo # granule is max-granuleId in collection, 9 fields, 2,3,4,5 dim, HegTool retruns details, Gdal does not	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	

#	Action	Expected Result	Notes
	sql: select * from aim_dev02.AmDaGranuleObjSrvXref (1 granule, 9 Obj/Field, no bands) Row granuleid objectid bandid dimid fieldid serviceid timestamp 1 3000604430 177 (null) (null) 652 0 2/4/2014 10:57:32 AM 2 3000604430 177 (null) (null) 653 0 2/4/2014 10:57:32 AM 3 3000604430 177 (null) (null) 654 0 2/4/2014 10:57:32 AM 4 3000604430 177 (null) (null) 655 0 2/4/2014 10:57:32 AM 5 3000604430 177 (null) (null) 656 0 2/4/2014 10:57:32 AM 6 3000604430 177 (null) (null) 657 0 2/4/2014 10:57:32 AM 7 3000604430 177 (null) (null) 658 0 2/4/2014 10:57:32 AM 8 3000604430 177 (null) (null) 659 0 2/4/2014 10:57:32 AM 9 3000604430 177 (null) (null) 660 0 2/4/2014 10:57:32 AM sql: delete from aim_dev02.AmDaGranuleObjSrvXref f5dpl01v:22500/DataAccessGui/config/dataObjects/MIL3DAE.004?agent=hegtool # granule is max-granuleId in collection, 9 fields, 2,3,4,5 dim, HegTool retruns details, Gdal does not sql: select * from AmDaGranuleObjSrvXref (1 granule, 9 Obj/Field, 2,3,4,5 dim bands) Row granuleid objectid bandid dimid fieldid serviceid timestamp 1 3000604430 177 123 11 652 0 2/4/2014 10:57:32 AM 2 3000604430 177 124 12 653 0 2/4/2014 10:57:32 AM 3 3000604430 177 125 (null) 654 0 2/4/2014 10:57:32 AM 4 3000604430 177 (null) (null) 655 0 2/4/2014 10:57:32 AM 5 3000604430 177 (null) (null) 656 0 2/4/2014 10:57:32 AM 6 3000604430 177 (null) (null) 657 0 2/4/2014 10:57:32 AM 7 3000604430 177 (null) (null) 658 0 2/4/2014 10:57:32 AM 8 3000604430 177 126 11 659 0 2/4/2014 10:57:32 AM 9 3000604430 177 127 11 660 0 2/4/2014 10:57:32 AM sql: delete from AmDaGranuleObjSrvXref f5dpl01v:22500/DataAccessGui/config/dataObjects/MIL3DAE.004 # granule is max-granuleId in collection, 9 fields, 2,3,4,5 dim, HegTool returns details, Gdal does not		

#	Action	Expected Result	Notes
	sql: select * from aim_dev02.AmDaGranuleObjSrvXref (1 granule, 9 Obj/Field, 2,3,4,5 dim bands) Row granuleid objectid bandid dimid fieldid serviceid timestamp 1 3000604430 177 123 11 652 0 2/4/2014 10:57:32 AM 2 3000604430 177 124 12 653 0 2/4/2014 10:57:32 AM 3 3000604430 177 125 (null) 654 0 2/4/2014 10:57:32 AM 4 3000604430 177 (null) (null) 655 0 2/4/2014 10:57:32 AM 5 3000604430 177 (null) (null) 656 0 2/4/2014 10:57:32 AM 6 3000604430 177 (null) (null) 657 0 2/4/2014 10:57:32 AM 7 3000604430 177 (null) (null) 658 0 2/4/2014 10:57:32 AM 8 3000604430 177 126 11 659 0 2/4/2014 10:57:32 AM 9 3000604430 177 127 11 660 0 2/4/2014 10:57:32 AM		
8	Configure DataObjects for a collection with agent name using datasetService: e.g. sql: select * from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 7 and collectionid = 3000595510 delete from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 7 and collectionid = 3000595510 f5dpl01v:22500/DataAccessGui/config/datasetService/dataObjects/GDAL/MIL3DAE.004?agent=gdalinfo # granule is max-granuleId in collection, 9 fields, 2,3,4,5 dim, Gdal does not return band-info # Note gdalinfo is agent by default. sql: select * from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 7 and collectionid = 3000595510 (1 granules, 9 Obj/Field ea., no bands) Row collectionid serviceid objectid fieldid bandid dimid enabledflag bandenabledflag dimenabledflag 1 3000595510 7 105 652 (null) (null) Y N N 2 3000595510 7 105 653 (null) (null) Y N N 3 3000595510 7 105 654 (null) (null) Y N N 4 3000595510 7 105 655 (null) (null) Y N N 5 3000595510 7 105 656 (null) (null) Y N N 6 3000595510 7 105 657 (null) (null) Y N N 7 3000595510 7 105 658 (null) (null) Y N N 8 3000595510 7 105 659 (null) (null) Y N N 9 3000595510 7 105 660 (null) (null) Y N N	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
9	Configure DataObjects for a collection with an ESDT using datasetService	Verify that the browser shows success	

#	Action	Expected Result	Notes
	for GDAL: e.g. sql: select * from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 7 and collectionid = 3000595510 delete from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 7 and collectionid = 3000595510 f5dpl01v:22500/DataAccessGui/config/datasetService/dataObjects/GDAL/MIL3DAE.004 # granule is max-granuleId in collection, 9 fields, 2,3,4,5 dim, Gdal does not return band-info # Note gdalinfo is agent by default. sql: select * from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 7 and collectionid = 3000595510 (1 granules, 9 Obj/Field ea., no bands) Row collectionid serviceid objectid fieldid bandid dimid enabledflag bandenabledflag dimenabledflag 1 3000595510 7 105 652 (null) (null) Y N N 2 3000595510 7 105 653 (null) (null) Y N N 3 3000595510 7 105 654 (null) (null) Y N N 4 3000595510 7 105 655 (null) (null) Y N N 5 3000595510 7 105 656 (null) (null) Y N N 6 3000595510 7 105 657 (null) (null) Y N N 7 3000595510 7 105 658 (null) (null) Y N N 8 3000595510 7 105 659 (null) (null) Y N N 9 3000595510 7 105 660 (null) (null) Y N N	and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	
10	Configure DataObjects for a collection with ESDT using datasetService for HEG: e.g. sql: select * from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 6 and collectionid = 3000595510 (MIL3DAE.004) Row collectionid serviceid objectid fieldid bandid dimid enabledflag bandenabledflag dimenabledflag 1 3000595510 6 97 652 99 11 Y Y Y 2 3000595510 6 97 653 100 12 Y Y Y 3 3000595510 6 97 654 99 (null) Y Y N 4 3000595510 6 97 655 (null) (null) Y N N 5 3000595510 6 97 656 101 5 Y Y Y 6 3000595510 6 97 657 101 5 Y Y Y 7 3000595510 6 97 658 101 5 Y Y Y 8 3000595510 6 97 659 99 11 Y Y Y 9 3000595510 6 97 660 99 11 Y Y Y	Verify that the browser shows success and there are no errors in the DA_Gui.log . Verify that the AmDaGranuleObjSrvXref has the bands populated.	

#	Action	Expected Result	Notes
	delete from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 6 and collectionid = 3000595510 f5dpl01v:22500/DataAccessGui/config/datasetService/dataObjects/HEG/MIL3DAE.004 # granule is max-granuleId in collection, 9 fields, 2,3,4,5 dim, # Note gdalinfo is agent by default. sql: select * from aim_dev02.DlDaColHdfObjectSrvXref where ServiceId = 6 and collectionid = 3000595510 (1 granules, 9 Obj/Field ea., no bands) Row collectionid serviceid objectid fieldid bandid dimid enabledflag bandenabledflag dimenabledflag 1 3000595510 6 177 652 123 11 Y Y Y 2 3000595510 6 177 653 124 12 Y Y Y 3 3000595510 6 177 654 125 (null) Y Y N 4 3000595510 6 177 655 (null) (null) Y N N 5 3000595510 6 177 656 (null) (null) Y N N (should be 99 11 Y Y Y) 6 3000595510 6 177 657 (null) (null) Y N N (should be 99 11 Y Y Y) 7 3000595510 6 177 658 (null) (null) Y N N (should be 99 11 Y Y Y) 8 3000595510 6 177 659 126 11 Y Y Y 9 3000595510 6 177 660 127 11 Y Y Y Note some Band & Dim4 info is not reported because Fields are 5-Dimension Hegtool does not report correctly		
11			
12	Configure DataObjects for a collection with ESDT usign datasetService for GDAL with useAllCollectionGranules and when some of the granules get deleted during the config run. Pick a collection with over 500 granules. Run the following: e.g.f5dpl01v:22500/DataAccessGui/config/dataObjects/MOD10CM.005?useAllCollectionGranules. Using BulkDelete, Unpublish and DeletionCleanup to delete the last few granules.	Verify that the DA_Gui.log reports that the granuleid was not found. But the config run is still successful/	
13	Configure DataObjects for a collection with a granuleid using datasetService for GDAL for a granule that is configured to be Golden	Verify that the config dataobjects still succeeds.	

TEST DATA:

EXPECTED RESULTS:

7.10.10 DataAccess GDAL End To End Testing (ECS-ECSTC-522)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Find a collection which is enabled for both HEG and GDAL processing in Data Access. Ensure that the service options forms have been uploaded to ECHO.		
2	Go to ECHO and add 3 granules from this collection to the cart.		
3	In the cart select "perform service".		NOTE: If it is grayed out, you will need to go back to the Collection Configuration tab of the DataAcces Configuration GUI and upload the echo form for the collection. Then, log into to the PUMP to update the virtual tags. This is done on the "Service Entries" page under "Service Management" in the "Provider Context" tab. To complete the process of updating the virtual tags, under the "Virtual" tab in the "Tags" section, select a DATASET (e.g. PVC_TS2). From the

#	Action	Expected Result	Notes
			list of datatypes, verify that the datatype used for the test is checked, then upload the tag.
4	Set different order options for each of the three granules:		
5	4a. select HEG processing with GeoTiff as the output format		
6	4b. select GDAL processing with NetCDF as the output format		
7	4c. select GDAL processing with png as the output format		
8	Submit the requests and verify that they succeed.		
9	Verify that the output files were processed correctly by opening and viewing them.		NOTE: To view the netCDF output, go to http://www.giss.nasa.gov/tools/panoply/download_gen.html to download and install the netCDF viewer called Panoply.

TEST DATA:

EXPECTED RESULTS:

7.10.11 Configure Collection for GDAL band subsetting (ECS-ECSTC-504)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Navigate to http://host/DataAccessGui_mode in a browser.		
2	Find the collection you want to configure by using the Filter Pattern		The collection should have no more than 3,000 granules in it to insure that gdal_info can run in less than 5 minutes. (10

#	Action	Expected Result	Notes
			granules/second)
3	Click on the collection. Hold mouse on top of "Configure Collection for Service".		chrome 33.0.1750.117 incompatibility.
4	Choose GDAL		
5	Service Configuration tab -> Services - Collections -> GDAL		
6	double click on GDAL		
7	click Enable Band Subsetting checkbox		
8	click Update Service Button		
9	expand GDAL tab		
10	double click collection		
11	choose Enable Band Subsetting checkbox		
12	click Update Service to Collection mapping		
13	navigate to http://host/DataAccessGui_mode/config/datasetObjects/GDAL/collection?useAllCollectionGranules	success is returned after in less than 5 minutes.	
14	Export capabilities to ECHO		
15	navigate to http://testbed.reverb.nasa.gov/reverb		
16	search for your collection		
17	Perform services on your collection	GDAL successfully processes your requests.	

TEST DATA:

EXPECTED RESULTS:

7.11 GLAS

7.11.1 Processing GLAS through OMS (ECS-ECSTC-563)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Order a GLAS using Reverb GUI and click the "set" button to specify spatial and temporal subsets.		

TEST DATA:

EXPECTED RESULTS:

7.12 EMS

7.12.1 Data Access EMS testing with duplicate filenames and special characters (ECS-ECSTC-655)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Create 2 EGI requests. Make one request with one granule. When it completes successfully modify the AmDaOutputFile.filepath for the request to have characters like (,),[,],{, and }. Make sure the actual file matches and has the same characters. Create another request with 2 granules. When it completes successfully, modify the AmDaOutputFile.filepath's of one of the granules to have the same filename as the other. Their directory paths can stay what they were or be modified if needed. Modify the actual filenames on disk so they match as well.	2 successful EGI requests	
2	Download the zip files of the 2 requests via ESIR		
3	ssh f5eil01v		
4	Run the Rollup of the Apache log data into the database eg. /usr/ecs/[MODE]/CUSTOM/utilities/EcDlDaRollupApacheLogs.ksh [MODE]		
5	Run select * from ecemsextracttypedaterange		
6	update the startdate and enddate so they fall in the range of the date when you downloaded the zip files		
7	Run "select * from EMSDistHTTPTransfer_view"	Make sure you can see rows in this	

#	Action	Expected Result	Notes
		table that show the output products in the 2 EGI requests you have made above.	

TEST DATA:

EXPECTED RESULTS:

7.12.2 EMS reporting for GLAH requests (ECS-ECSTC-656)

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

#	Action	Expected Result	Notes
			!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		
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	the end date has to be at

#	Action	Expected Result	Notes
		in ecemsextracttypedaterange	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		
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:

7.12.3 DP_81_02_TP036a EMS SDPS Web API metrics (ECS-ECSTC-529)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Verify that the mode is configured to log Apache events. Make an Data Access request for processing. Download the downloadURL in the response from ESIR.	On the x4eil01 machine run: grep CustomLog /usr/ecs/OPS/COTS/apache/conf/httpd.conf | grep Check to make sure the file: x5eil01:/usr/ecs//CUSTOM/data/DPL/EcDlDataAccessApache.log is updated with information pertaining to what you downloaded from ESIR.	
2	Run a series of Data Access requests through to capture the following scenarios from ESIR : 1) Download of individual output files (SCIENCE and METADATA (.xml)) by clicking on downloadURL in request response. 2) Download of gzip file containing all output files 3) Modify the download URL by changing the filename to something that does not actually exist for a given request 4) Specify an output file for a request that did not successfully process	Scenarios 1 and 2 should end up with EMS entries. Scenario 3 and 4 should not. Be sure to actually download the files, not just view the links. Also make sure that you save the download urls in a temporary location.	
3	Run the following script to read the entries in EcDlDataAccessApache.log into the aim_..AmDaTransfer database table. x5eil01:/usr/ecs//CUSTOM/utilitiesEcDlDaRollupApacheLogs.ksh	Verify that this script moved the data in : x5eil01:/usr/ecs//CUSTOM/data/DPL/EcDlDataAccessApache.log to a similar file with a timestamp appended. The above file should now be zero bytes. Also Verify the script populated AmDaTransfer by running the following sql in the aim_ database:	

#	Action	Expected Result	Notes
		select * from AmDaTransfer where TransferDateTime > TIMESTAMP	
4	Now run the EMS data extractor to create a flat file that can be exported to EMS: x5dpl01:/usr/ecs//CUSTOM/utilities/EcDbEMSdataExtractor.pl -mode -extracttype DistHTTP -startdate 'Mar 18 2014' -enddate 'Mar 19 2014' Where start date is yesterday's date and enddate is todays date.		
5	Now look at the flat file that was generated by the EMS script. The location of this file can be found by running the following command in the mode's logs directory: grep Transfer *EMS*	Verify that the flat file conforms to the specification located at: http://newsroom.gsfc.nasa.gov/esi/dev/schemas/EMSDistribution.xsd	
6	For this part while keeping some of the download urls delete the granule and then retry the download request.	Verify that the event is logged and that the granule has been deleted.	Go to the Gdal/Heg DefaultOutput directory to remove the files. Then go back to the DataAccess to Browser to click on file link such as AMSR_E_L3_DailySnow_V09_20020926_1.gif
7	Use these sql statements to validate the flat file created: select * from AmDaTransfer select * from AmDaRequest where RequestId = (in the flat file) select * from AmDaJob where RequestId = select * from AmDaJobDetail where JobId = select * from AmDaOutputFile select JobVolumeRatio from EMSDistHTTPTransfer_View where RequestId = "		

TEST DATA:

EXPECTED RESULTS:

7.12.4 DP_81_02_TP036 Export SDPS Web API metrics to EMS (ECS-ECSTC-387)

DESCRIPTION:

Test Case ID - 400
Submit ESI requests for collections specified in test case 10 such that there are a combination of both synchronous and asynchronous web requests which require both the HEG and GDAL services. In addition submit several OMS orders requesting processing using both the HEG and GDAL services.
Run the EMS extraction utility and send the resulting flat file to EMS for verification.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	The first set of test procedures can be done locally without a crontab setup		#comment
2	Verify that the mode is configured to log Apache events. Make an Data Access request for processing. Download the downloadURL in the response from ESIR. 	On the x5eil01 machine run: grep CustomLog /usr/ecs/OPS/COTS/apache/conf/httpd.conf | grep <MODE> Check to make sure the file: x4eil01:/usr/ecs/<MODE>/CUSTOM/data/DPL/EcDlDataAccessApache.log is updated with information pertainting to what you downloaded from ESIR.	
3	Run a series of Data Access requests through to capture the following scenarios from ESIR : 1) Download of individual output files (SCIENCE and METADATA (.xml)) by clicking on downloadURL in request response. 2) Download of gzip file containing all output files 3) Modify the download URL by changing the filename to something that does not actually exist for a given request 4) Specify an output file for a request that did not successfully process <br		

#	Action	Expected Result	Notes
	/>Scenarios 1 and 2 should end up with EMS entries. Scenario 3 and 4 should not. Be sure to actually download the files, not just view the links. Also make sure that you save the download urls in a temporary location.		
4	Run the following scrip to read the entries in EcDlDataAccessApache.log into the EcInDb_<MODE>..AmDaTransfer database table. x5eil01:/usr/ecs/<MODE>/CUSTOM/utilitiesEcDlDaRollupApacheLogs.ksh	Verify that this script moved the data in : x4eil01:/usr/ecs/<MODE>/CUSTOM/data/DPL/EcDlDataAccessApache.log to a similar file with a timestamp appended. The above file should now be zero bytes. Also Verify the script populated AmDaTransfer by running the following sql in the EcInDb_<MODE> database: select * from AmDaTransfer where TransferDateTime > dateadd(dd, -<days since last time Apache log was imported into database>, getdate())	
5	Now run the EMS data extractor to create a flat file that can be exported to EMS: x5dpl01:/usr/ecs/<MODE>/CUSTOM/utilities/EcDbEMSdataExtractor.pl -mode <MODE> -extracttype DistHTTP -startdate YYYYMMDD -enddate YYYYMMDD Where start date is yesterdays date and enddate is todays date.		
6	Now look at the flat file that was generated by the EMS script. The location of this file can be found by running the following command in the mode's logs directory: grep Transfer *EMS* 	Verify that the flat file conforms to the specification located at: http://newsroom.gsfc.nasa.gov/esi/dev/schemas/EMSDistribution.xsd 	
7	For this part while keeping some of the download urls delete the granule and then retry the download request.	Verify that the event is logged and that the granule has been deleted.	
8	The tester will need to obtain EMS credentials in order to view the EMS metrics after export.		#comment
9			
10	Use these sql statements to validate the flat file created: select * from AmDaTransfer select * from AmDaRequest where RequestId = (in the flat file) select * from AmDaJob where		

#	Action	Expected Result	Notes
	RequestId = select * from AmDaJobDetail where JobId = select * from AmDaOutputFile select JobVolumeRatio from EMSDistHTTPTransfer_View where RequestId = " 		
11	This test is for End to End testing with EMS		#comment
12	Determine the designated frequency of exports to EMS. Verify that there is a cron job or other mechanism set up to export metrics on that specified schedule.	EMS export has been properly configured	
13	Note the date and time of the requests specified below. Ensure that they all fall within the same EMS metrics period. This will make it easier to find the metrics for these requests in later steps.	additional info is obtained to help in later verification steps.	
14	Submit ESI requests for collections specified in test case 10 such that there are a combination of both synchronous and asynchronous web requests which require both the HEG and GDAL services. In addition submit several OMS orders requesting processing using both the HEG and GDAL services.	Verify that all submitted requests are successful.	
15	Submit a number of requests which result in error.	Verify that these requests fail.	
16	Submit a request for the output files of each of the successful processing requests. For some of these, request the output in a zip file.	Verify that the requested files are received (both single files and zip)	
17	Submit a few requests for output files of processing requests which will result in an error (e.g. a request for a valid processing request but specifying a non existance ile, oir a request for the output of a non existant request)	Verify that the requests fail with an error.	
18	Allow the EMS extraction script to run on its normal schedule.	Verify that the EMS script ran successfully.	
19	Log in to the EMS and view metircs for the provider representing the Mode where this test is being run.	Verify that metrics exist for the time frame in which the test requests in this test were performed.	
20	View the metrics for successful processing requests during the time frame in which the requests were performed.	Verify that a record exists for each successful processing request. Verify that the record indicates whether the request was synchronous or asynchronous, the number of granules included, the tool used (e.g. HEG or GDAL), the processing options, and the processing duration among other things.	
21	View the metrics for failed processing requests during the time frame in which the requests in this test were performed.	Verify that a record exists for each failed processing request. Verify that the record includes the same items	

#	Action	Expected Result	Notes
		listed for successful requests, but also an indication as to teh reason for the failure (e.g. an error code).	
22	View the metrics for successful data transfer requests during the time frame in which the requests in this test were performed.	Verify that a record exists for each successful file transfer request. Verify that the record indicates the type of request(single file or zip), the IP address of the requestor, and the transfer duration among other things.	
23	View the metrics for failed data transfer requests during the time frame in which the requests in this test were performed.	Verify that a record exists for each successful file transfer request. Verify that the record includes all of the fields listed for successful request, and in addition an indication of why the request failed (e.g. error code).	

TEST DATA:

EXPECTED RESULTS:
Verify that all metrics are successfully processed by EMS for synchronous web requests, asynchronous web requests, and orders for both the HEG and GDAL services.

7.13 HEG

7.13.1 DP_81_02_TP012 OMS Invocation of HEG Tool Adapter (ECS-ECSTC-377)

DESCRIPTION:
Test Case ID - 130
Submit the following order requests via the SDPS Web Service:
a.) Single granule specifying spatial subsetting, band subsetting, reformatting, reprojection, and resampling
b.) Order for 2 granules each for 3 collections - processing options for the 6 granules should match the following :

1. Spatial subsetting, band subsetting, reformatting, reprojection, and resampling
2. Spatial subsetting only
3. Band subsetting only
4. Reformatting only
5. Resampling only
6. Reprojection and resampling

PRECONDITIONS:
Collections must be configured and contain the number of required public granules needed for the test.

STEPS:

#	Action	Expected Result	Notes
1	To perform this test select three collections, one of which should be swath and at least one should be grid. Once you have selected your collections, select two HEG-able granules from each collection.	The collections and granules have been selected.	
2	To complete the first part a) of the test select one of the granules thru the online WEB gui and process it using the spatial subsetting, band subsetting, reformatting, reprojection, and resampling options.	Verify that the the order has been submitted, shown on the OMS Gui and processed thru the stages	
3	For part b, submit 2 granules for 3 collections, and then perform the actions outlined below. This totals equal to 30 orders b.) Order for 2 granules each for 3 collections - processing options for the 6 granules should match the following : 1. Spatial subsetting, band subsetting, reformatting, reprojection, and resampling 2. Spatial subsetting only 3. Band subsetting only 4. Reformatting only 5. Resampling only 6. Reprojection and resampling Note: Alternatively one can use the attached urls to generate the requests. There are 3 collections selected, with all urls generated. One could also run a wget script which can execute all the urls at once and process the granules	Once the orders have been submitted verify that no error is observed on the Web Gui. Next verify that the order have been seen on the OMS GUI under Request management -> Distribution Requests. Next verify that the orders have gone thru all the stages and verify that there are no orders gone to operator interventions. To verify that the granules have been processed correctly verify that with the standalone HEG tool, in either Linux or Windows envionments	
4			

TEST DATA:

EXPECTED RESULTS:
Verify that the SDPS Web Service submits the orders for OMS processing.
Verify that OMS dispatches the request to the HEG Tool adapter via the SDPS-DAAC internal on-demand processing API.
Verify that the HEG tool adapter executes the correct HEG executable and performs the correct processing based on the options provided by the request.
Verify that the outputs produced are correct.
Verify that the HEG tool adapter returns successful execution results in accordance with the SDPS-DAAC internal on-demand processing API.
Verify that all OMS orders are shipped successfully.

7.13.2 DP_81_02_TP044 SDPS Web API HEG Regression - AMSR (ECS-ECSTC-394)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Data can be found in /sotestdata/DROP_801/HEG/MODIS_Land It may need to be ingested prior to test. In order to find if the data in question is already ingested, and to find its granuleId use the following SQL: select GranuleId, LocalGranuleID from AmGranule where LocalGranuleID like '%<IDENTIFIER>%' and ShortName = '<SHORTNAME>' and VersionId = <VERSIONID> where <IDENTIFIER> is the ID used below to define the different granules and <SHORTNAME> and <VERSIONID> are the shortname and versionID of the collection in question.		
2	For instructions on how to construct and submit a request see Test case 150 (DP_81_02_TP000.1). It may be useful to save off the URLs or wget commands for each request for later reuse. The mechanism of submital (syncronous/asynchronous, EWOC/EGI) is not too important for this test, but ideally. you shoudl do some of each.		

TEST DATA:

EXPECTED RESULTS:

7.13.3 DP_81_02_TP043 SDPS Web API HEG Regression - MISR (ECS-ECSTC-392)

DESCRIPTION:

This test will exercise common MODIS HEG processing use cases.

<u>Background:</u>

Sources:

- MISR data product specifications Revision S from Sep 2010
- EMS data to derive granule counts and sizes

- Feedback from Abe Taaheri and Jon Pals
  Notes:
- 2011 Dist is the number of distributed granules Jan – Apr,2011.
- Size (MB) reflects the average size of a distributed granule. Note that for MISR, granule sizes can vary widely and large granules can be more than twice as large as the average.
- MIL2ASAE.2, MIL2ASLS.2 were chosen because they are popular; use the HDF-EOS Stacked-Block Grid posing HEG challenges, and users likely want re-projection, spatial and science parameter subsetting
- MI1B2E.3, MI1B2T.3 were included because they are popular; are examples of MISR data with 3-D fields that have XDim, YDim, and SOMBlockDim only; and are accessed by 2 HEG orders at ASDC; and are candidates for geographic subsetting, re-projection
- MIL3DAE.4 was included because it is among the most popular examples of Level 3 products using a geographic grid
- MIL3MRD.5 was selected as an example of a very large product using a geographic grid.

PRECONDITIONS:

Granules for test:

ESDT	Selected Granules	Granule Characteristics
MIL2ASAE.002	P177_O058716 P024_O058721 P072_O058724	Russia to South Africa Canada to Mexico Alaska to New Zealand
MIL2ASLS.002	P177_O058716 P024_O058721 P072_O058724	Russia to South Africa Canada to Mexico Alaska to New Zealand
MI1B2E.003	P177_O058716 (AN) P177_O058716 (DF) P024_O058721 (AN) P024_O058721 (DF) P072_O058724 (AN) P072_O058724 (DF)	Russia to South Africa (AN camera) Russia to South Africa (DF camera) Canada to Mexico (AN camera) Canada to Mexico (DF camera) Alaska to New Zealand (AN camera) Alaska to New Zealand (DF camera)
MI1B2T.003	P177_O058716 (AN) P177_O058716 (DF) P024_O058721 (AN) P024_O058721 (DF) P072_O058724 (AN) P072_O058724 (DF)	Russia to South Africa (AN camera) Russia to South Africa (DF camera) Canada to Mexico (AN camera) Canada to Mexico (DF camera) Alaska to New Zealand (AN camera) Alaska to New Zealand (DF camera)
MIL3DAE.004	JAN_01_2010 APR_01_2010 JUL_01_2010 OCT_01_2010	January April July October
MIL3MRD.005	JAN_2010	January

	APR_2010 JUL_2010 OCT_2010	April July October

STEPS:

#	Action	Expected Result	Notes
1	Data can be found in /sotestdata/DROP_801/HEG/MODIS_Land It may need to be ingested prior to test. In order to find if the data in question is already ingested, and to find its granuleId use the following SQL: select GranuleId, LocalGranuleID from AmGranule where LocalGranuleID like '%<IDENTIFIER>%' and ShortName = '<SHORTNAME>' and VersionId = <VERSIONID> where <IDENTIFIER> is the ID used below to define the different granules and <SHORTNAME> and <VERSIONID> are the shortname and versionID of the collection in question.		
2	For instructions on how to construct and submit a request see Test case 150 (DP_81_02_TP000.1). It may be useful to save off the URLs or wget commands for each request for later reuse. The mechanism of submital (syncronous/asynchronous, EWOC/EGI) is not too important for this test, but ideally. you shoudl do some of each.		
3	MIL2ASAE.002		#comment
4	P177_O058716	Band Subset: “WaterLeavEqReflExp” (1.1km), Reproject to Geographic Band Subset: “RegBestEstimateSpectralOptDepth” (17.6km), Reproject to Geographic Band Subset: “WaterLeavEqReflExp” (1.1km), “RegBestEstimateSpectralOptDepth” (17.6km), Reproject to Geographic Band Subset: “WaterLeavEqReflExp” (1.1km), Spatial Subset: West Bound 15, South Bound -30, East Bound 35, North Bound 30, Reproject to Lambert Equal Area Azimuthal Band Subset: “RegBestEstimateSpectralOptDepth” (17.6km), Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal	

#	Action	Expected Result	Notes
		Area Azimuthal Band Subset: “WaterLeavEqReflExp” (1.1km), Spatial Subset: 15E, 30S, 35E, 30N, Reproject to Polar Stereographic Band Subset: “RegBestEstimateSpectralOptDepth” (17.6km), Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic	
5	P024_O058721	Band Subset: “WaterLeavEqReflExp” (1.1km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “RegBestEstimateSpectralOptDepth” (17.6km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “WaterLeavEqReflExp” (1.1km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic Band Subset: “RegBestEstimateSpectralOptDepth” (17.6km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic	
6	P072_O058724	Band Subset: “WaterLeavEqReflExp” (1.1km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “RegBestEstimateSpectralOptDepth” (17.6km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “WaterLeavEqReflExp” (1.1km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic Band Subset:	

#	Action	Expected Result	Notes
		“RegBestEstimateSpectralOptDepth” (17.6km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic	
7	MIL2ASLS.002		#comment
8	P177_O058716	Band Subset: “NDVI” (1.1km), Reproject to Geographic Band Subset: “NormBlkSfcIrrad” (17.6km), Reproject to Geographic Band Subset: “NDVI” (1.1km), “NormBlkSfcIrrad” (17.6km), Reproject to Geographic Band Subset: “NDVI” (1.1km), Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal Area Azimuthal Band Subset: “NormBlkSfcIrrad” (17.6km), Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal Area Azimuthal Band Subset: “NDVI” (1.1km), Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic Band Subset: “NormBlkSfcIrrad” (17.6km), Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic	
9	P024_O058721	Band Subset: “NDVI” (1.1km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “NormBlkSfcIrrad” (17.6km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “NDVI” (1.1km), Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic Band Subset: “NormBlkSfcIrrad” (17.6km), Spatial Subset: 0N, 105W, 50N, 80W,	

#	Action	Expected Result	Notes
		Reproject to Polar Stereographic	
10	P072_O058724	Band Subset: “NDVI” (1.1km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “NormBlkSfcIrrad” (17.6km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “NDVI” (1.1km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic Band Subset: “NormBlkSfcIrrad” (17.6km), Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic	
11	MI1B2E.003		#comment
12	P177_O058716 (AN)	Band Subset: “Red Radiance/RDQI”, Reproject to Geographic Band Subset: “Red Radiance/RDQI”, “Blue Radiance/RDQI”, “Green Radiance/RDQI”, Reproject to Geographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic	
13	P177_O058716 (DF)	Band Subset: “Red Radiance/RDQI”, Reproject to Geographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic	
14	P024_O058721 (AN)	Band Subset: “Red Radiance/RDQI”,	

#	Action	Expected Result	Notes
		Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to State Plane 4802	
15	P024_O058721 (DF)	Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to State Plane 4802	
16	P072_O058724 (AN)	Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic	
17	P072_O058724 (DF)	Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic	
18	MI1B2T.003		#comment
19	P177_O058716 (AN)	Band Subset: “Red Radiance/RDQI”, Reproject to Geographic Band Subset: “Red Radiance/RDQI”, “Blue Radiance/RDQI”, “Green	

#	Action	Expected Result	Notes
		Radiance/RDQI”, Reproject to Geographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic	
20	P177_O058716 (DF)	Band Subset: “Red Radiance/RDQI”, Reproject to Geographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 30S, 15E, 30N, 35E, Reproject to Polar Stereographic	
21	P024_O058721 (AN)	Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to State Plane 4802	
22	P024_O058721 (DF)	Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to Polar Stereographic Band Subset: “Red Radiance/RDQI”, Spatial Subset: 0N, 105W, 50N, 80W, Reproject to State Plane 4802	
23	P072_O058724 (AN)	Band Subset: “Red Radiance/RDQI”,	

#	Action	Expected Result	Notes
		Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic	
24	P072_O058724 (DF)	Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Lambert Equal Area Azimuthal Band Subset: “Red Radiance/RDQI”, Spatial Subset: 50S, 170E, 35S, 180E, Reproject to Polar Stereographic	
25	MIL3DAE.004		#comment
26	JAN_01_2010	Band Subset: “Optical depth average” Band Subset: “Optical depth average”, “Single-scatter albedo” Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
27	APR_01_2010	Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
28	JUL_01_2010	Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W Band Subset: “Optical	

#	Action	Expected Result	Notes
		depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
29	OCT_01_2010	Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Optical depth average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
30	MIL3MRD.005		#comment
31	JAN_2010	Band Subset: “Average” Band Subset: “Average”, “Covariance” Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
32	APR_2010	Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
33	JUL_2010	Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W<br	

#	Action	Expected Result	Notes
		/>Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	
34	OCT_2010	Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Transverse Mercator Band Subset: “Average”, Spatial Subset: 20N, 180W, 70N, 65W, Reproject to Lambert Conformal Conic	

TEST DATA:

EXPECTED RESULTS:
The results of each test should be viewed in a GeoTiff capable image viewer such as ENVI or eog. The images should be inspected to ensure that they contain viewable data and that data seems to be in the expected projection and there are no obvious errors present. This may require some consultaion with the Science Office of HEG developers to gain an idea of what 'expected' means for each ESDT/projection/ band/ etc.

7.13.4 DP_81_02_TP042 SDPS Web API HEG Regression - ASTER (ECS-ECSTC-391)

DESCRIPTION:
This test will exercise common MODIS HEG processing use cases.
Background:
ASTER L1B is the only product proposed for inclusion in the benchmark suite.

PRECONDITIONS:
Granules for test:

ESDT	Selected Granules	Granule Characteristics
AST_L1B.003	05272010220817 02192011160332	Alaska Maryland, DC, Northern Virginia

	02012011211726 01082011050521	Pacific off Hawaii (no land pixels) Boulder (Night)

STEPS:

#	Action	Expected Result	Notes
1	Data can be found in /sotestdata/DROP_801/HEG/MODIS_Land It may need to be ingested prior to test. In order to find if the data in question is already ingested, and to find its granuleId use the following SQL: select GranuleId, LocalGranuleID from AmGranule where LocalGranuleID like '%<IDENTIFIER>%' and ShortName = '<SHORTNAME>' and VersionId = <VERSIONID> where <IDENTIFIER> is the ID used below to define the different granules and <SHORTNAME> and <VERSIONID> are the shortname and versionID of the collection in question.		
2	For instructions on how to construct and submit a request see Test case 150 (DP_81_02_TP000.1). It may be useful to save off the URLs or wget commands for each request for later reuse. The mechanism of submital (syncronous/asynchronous, EWOC/EGI) is not too important for this test, but ideally. you shoudl do some of each.		
3	AST_L1B.003		#comment
4	5272010220817	Band Subset: “ImageData1”, Pixel Size, Reproject to Geographic	
5	2192011160332	Band Subset: “ImageData1”, Reproject to Geographic Band Subset: “ImageData1”,”ImageData2”, Pixel Size, Reproject to State Plane 1900	
6	2012011211726	Band Subset: “ImageData1”, Pixel Size, Reproject to Geographic	
7	1082011050521	Band Subset: “ImageData10”, Pixel Size, Reproject to State Plane 0501 	

TEST DATA:

EXPECTED RESULTS:

The results of each test should be viewed in a GeoTiff capable image viewer such as ENVI or eog. The images should be inspected to ensure that they contain viewable data and that data seems to be in the expected projection and there are no obvious errors present. This may require some consultaion with the Science Office of HEG developers to gain an idea of what 'expected' means for each ESDT/projection/ band/ etc.

7.13.5 DP_81_02_TP041 SDPS Web API HEG Regression - MODIS (ECS-ECSTC-390)

DESCRIPTION:

This test will exercise common MODIS HEG processing use cases.

Background:

Sources:

- MODIS product descriptions and linked documents available at the LPDAAC web site,
- EMS data to derive granule counts and sizes,
- Feedback from Abe Taaheri and Jon Pals

Notes:

i. 2011 Dist is the number of distributed granules Jan – Mar, 2011.
j. Size (MB) reflects the average size of a distributed compressed granule.
k. Where there was a choice between MYD / MOD products, I picked the one with the higher distribution.
l. Based on feedback from Abe, MCD43A3.5 and MCD43C1.5 are not suitable substitutes for MOD43B3.4 and MOD43C1.4 and have been removed.
m. Based on feedback from Jon, MCD43A1.005 has been added because it includes a 3-D field.
n. MOD09GQ was chosen over MOD09A1 and MOD09GA, even they have higher distribution numbers) because of the higher resolution and number of pixels.
o. MOD11A1.5 and MOD15A2.5[JLP1] were chosen strictly because of their popularity.
p. MOD11_L2.5 was included because it is a swath product and the most popular fire product.

MOD09CMG.5 was included because it uses a geographic grid.

[JLP1]This isn’t present in the table. Was it intentionally deleted?

PRECONDITIONS:

Granules for test:

ESDT	Selected Granules	Granule Characteristics
MOD09GQ.005	A2010033.h09v02 A2010033.h10v02 A2010033.h12v05 A2010046.h18v00 A2010033.h25v04	Alaska + Russia (crosses 180° Longitude) Alaska, State Plane 5006 (Transverse Mercator) Maryland, State Plane 1900 (Lambert Conformal Conic) North Pole (very few land pixels) Mongolia (no open water pixels)
MOD11A1.005	A2010033.h09v02 A2010033.h10v02	Alaska + Russia (crosses 180° Longitude) Alaska, State Plane 5006 (Transverse Mercator)

	A2010033.h12v05 A2010046.h18v00 A2010033.h25v04	Maryland, State Plane 1900 (Lambert Conformal Conic) North Pole (very few land pixels) Mongolia (no open water pixels)
MCD43A1.005	A2010033.h09v02 A2010033.h10v02 A2010033.h12v05 A2010041.h18v00 A2010033.h25v04	Alaska + Russia (crosses 180° Longitude) Alaska, State Plane 5006 (Transverse Mercator) Maryland, State Plane 1900 (Lambert Conformal Conic) North Pole (very few land pixels) Mongolia (no open water pixels)
MOD11_L2.005	A2010033.0050 A2010033.0715 A2010033.0210 A2010033.0045 A2011033.0415	Alaska + Russia (crosses 180° Longitude) Alaska, State Plane 5006 (Transverse Mercator) Maryland, State Plane 1900 (Lambert Conformal Conic) North Pole (no land pixels) Mongolia (no open water pixels)
MOD09CMG.005	A2010001 A2010091 A2010182 A2010274	January April July October

NSIDC Granules

MOD10A1.005	A2005177.h12v12 A2007313.h16v16 A2007321.h15v01 A2008361.h24v05 A2009001.h07v03 A2009001.h18v04 A2009100.h18v04	
MOD10A2.005	A2005177.h12v12 A2007313.h16v16 A2007321.h15v01 A2008361.h24v05 A2009001.h07v03 A2009001.h18v04 A2009097.h18v04	
MOD10CM.005	A2006001	
MYD10A1.005	A2005177.h12v12 A2007313.h16v16 A2007321.h15v01	

	A2008361.h24v05 A2009001.h07v03 A2009001.h18v04 A2009100.h18v04	
MYD10CM.005	A2006001	

STEPS:

#	Action	Expected Result	Notes
1	Data can be found in /sotestdata/DROP_801/HEG/MODIS_Land It may need to be ingested prior to test. In order to find if the data in question is already ingested, and to find its granuleId use the following SQL: select GranuleId, LocalGranuleID from AmGranule where LocalGranuleID like '%<IDENTIFIER>%' and ShortName = '<SHORTNAME>' and VersionId = <VERSIONID> where <IDENTIFIER> is the ID used below to define the different granules and <SHORTNAME> and <VERSIONID> are the shortname and versionID of the collection in question.		
2	For instructions on how to construct and submit a request see Test case 150 (DP_81_02_TP000.1). It may be useful to save off the URLs or wget commands for each request for later reuse. The mechanism of submital (syncronous/asynchronous, EWOC/EGI) is not too important for this test, but ideally. you shoudl do some of each.		
3	MOD09GQ.005		#comment
4	A2010033.h09v02	Band Subset: “sur_refl_b01_1”, Spatial Subset: 60N, 170W, 63N, 160W, Reproject to Geographic Band Subset: “sur_refl_b01_1”, Spatial Subset: 60N, 170W, 63N, 160W, Reproject to UTM	
5	A2010033.h10v02	Band Subset: “sur_refl_b01_1”, Spatial Subset: 60N, 160W, 70N, 150W, Reproject to State Plane 5006 Band Subset: “sur_refl_b01_1”, Spatial Subset: 60N, 160W, 70N, 150W, Reproject to TM	
6	A2010033.h12v05	Band Subset: “sur_refl_b01_1”,	

#	Action	Expected Result	Notes
		Spatial Subset: 37N, 80W, 40N, 75W, Reproject to State Plane 1900 Band Subset: “sur_refl_b01_1”, Spatial Subset: 37N, 80W, 40N, 75W, Reproject to Lambert Conformal Conic	
7	A2010046.h18v00	Band Subset: “sur_refl_b01_1”, Spatial Subset: 75N, 0E, 90N, 70E, Reproject to Geographic Band Subset: “sur_refl_b01_1”, Spatial Subset: 75N, 0E, 90N, 70E, Reproject to UTM	
8	A2010033.h25v04	Band Subset: “sur_refl_b01_1”, Reproject to Geographic Band Subset: “sur_refl_b01_1”,”sur_refl_b02_1”, Reproject to Geographic Band Subset: “sur_refl_b01_1”, Spatial Subset: 40N, 100E, 50N, 110E, Reproject to Geographic Band Subset: “sur_refl_b01_1”, Spatial Subset: 40N, 100E, 50N, 110E, Reproject to UTM	
9	MOD11A1.005		#comment
10	A2010033.h09v02	Band Subset: “LST_Day_1km”, Spatial Subset: 60N, 170W, 63N, 160W, Reproject to Geographic Band Subset: “LST_Day_1km”, Spatial Subset: 60N, 170W, 63N, 160W, Reproject to UTM	
11	A2010033.h10v02	Band Subset: “LST_Day_1km”, Spatial Subset: 60N, 160W, 70N, 150W, Reproject to State Plane 5006 Band Subset: “LST_Day_1km”, Spatial Subset: 60N, 160W, 70N, 150W, Reproject to TM	
12	A2010033.h12v05	Band Subset: “LST_Day_1km”,	

#	Action	Expected Result	Notes
		Spatial Subset: 37N, 80W, 40N, 75W, Reproject to State Plane 1900 Band Subset: “LST_Day_1km”, Spatial Subset: 37N, 80W, 40N, 75W, Reproject to Lambert Conformal Conic	
13	A2010046.h18v00	Band Subset: “LST_Day_1km”, Spatial Subset: 75N, 0E, 90N, 70E, Reproject to Geographic Band Subset: “LST_Day_1km”, Spatial Subset: 75N, 0E, 90N, 70E, Reproject to UTM	
14	A2010033.h25v04	Band Subset: “LST_Day_1km”, Reproject to Geographic Band Subset: “LST_Day_1km”,”Emis_31”, Reproject to Geopgraphic Band Subset: “LST_Day_1km”, Spatial Subset: 40N, 100E, 50N, 110E, Reproject to Geographic Band Subset: “LST_Day_1km”, Spatial Subset: 40N, 100E, 50N, 110E, Reproject to UTM	
15	MCD43A1.005		#comment
16	A2010033.h09v02	Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 60N, 170W, 63N, 160W, Reproject to Geographic Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 60N, 170W, 63N, 160W, Reproject to UTM	
17	A2010033.h10v02	Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 60N, 160W, 70N, 150W, Reproject to State Plane 5006 Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 60N, 160W, 70N,	

#	Action	Expected Result	Notes
		150W, Reproject to TM	
18	A2010033.h12v05	Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 37N, 80W, 40N, 75W, Reproject to State Plane 1900 Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 37N, 80W, 40N, 75W, Reproject to Lambert Conformal Conic	
19	A2010041.h18v00	Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 75N, 0E, 90N, 70E, Reproject to Geographic Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 75N, 0E, 90N, 70E, Reproject to UTM	
20	A2010033.h25v04	Band Subset: “BRDF_Albedo_Parameters_vis”, Reproject to Geographic Band Subset: “BRDF_Albedo_Parameters_vis”, ”BRDF_Albedo_Parameters_nir”, Reproject to Geographic Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 40N, 100E, 50N, 110E, Reproject to Geographic Band Subset: “BRDF_Albedo_Parameters_vis”, Spatial Subset: 40N, 100E, 50N, 110E, Reproject to UTM	
21	MOD11_L2.005		#comment
22	A2010033.0050	Band Subset: “LST”, Spatial Subset: 65N, 170W, 75N, 150W, Reproject to Lambert Equal Area Azimuthal Band Subset: “LST”, Spatial Subset:	

#	Action	Expected Result	Notes
		65N, 170W, 75N, 150W, Reproject to Polar Stereographic	
23	A2010033.0715	Band Subset: “LST”, Spatial Subset: 55N, 160W, 70N, 130W, Reproject to Lambert Equal Area Azimuthal Band Subset: “LST”, Spatial Subset: 55N, 160W, 70N, 130W, Reproject to Polar Stereographic	
24	A2010033.0210	Band Subset: “LST”, Spatial Subset: 30N, 77W, 40N, 63W, Reproject to Lambert Equal Area Azimuthal Band Subset: “LST”, Spatial Subset: 30N, 77W, 40N, 63W, Reproject to Polar Stereographic	
25	A2010033.0045	Band Subset: “LST”, Spatial Subset: 70N, 130W, 80N, 70W, Reproject to Lambert Equal Area Azimuthal Band Subset: “LST”, Spatial Subset: 70N, 130W, 80N, 70W, Reproject to Polar Stereographic	
26	A2011033.0415	Band Subset: “LST”, Project to Geographic Band Subset: “LST”, “Emis_32”, Project to Geographic Band Subset: “LST”, Spatial Subset: 40N, 90E, 50N, 110E, Reproject to Lambert Equal Area Azimuthal Band Subset: “LST”, Spatial Subset: 40N, 90E, 50N, 110E, Reproject to Polar Stereographic	
27	MOD09CMG.005		#comment
28	A2010001	Band Subset: “Course Resolution Surface Reflectance Band 2” Band Subset: “Course Resolution Surface Reflectance Band 1”, “Course Resolution Surface Reflectance Band 2” Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N,	

#	Action	Expected Result	Notes
		65W, Reproject to Lambert Conformal Conic Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Transverse Mercator	
29	A2010091	Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Lambert Conformal Conic Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Transverse Mercator	
30	A2010182	Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Lambert Conformal Conic Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Transverse Mercator	
31	A2010274	Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Lambert Conformal Conic Band Subset: “Course Resolution Surface Reflectance Band 2”, Spatial Subset: 25N, 125W, 50N, 65W, Reproject to Transverse Mercator	

TEST DATA:

EXPECTED RESULTS:

The results of each test should be viewed in a GeoTiff capable image viewer such as ENVI or eog. The images should be inspected to ensure that they contain viewable data and that data seems to be in the expected projection and there are no obvious errors present. This may require some consultaion with the Science Office of HEG developers to gain an idea of what 'expected' means for each ESDT/projection/ band/ etc.

7.13.6 DP_81_02_TP003 HEG Error Handling (ECS-ECSTC-393)

DESCRIPTION:

Test Case Id 40
For at least a portion of this test ensure that the HEG tool runs on a platform different from the one used by the SDPS External Web Service API and simulate a situation where the tool platform is not attached to the StorNext file systems.
Submit requests that will result in HEG processing errors or no outputs. Use the list of NCRs and HEG improvements made to recognize such errors as a guide to complete the list below. Examples:
a) Geographic subsetting for an area not covered by the granule in question.
b) Re-projections with parameters that are invalid given the geographic coverage of the granule.
c) Subsetting for bands that cannot be subsetted.
d) The HEG tool adapter cannot be contacted.
e) The HEG tool command fails.
f) The HEG tool execution times out.

PRECONDITIONS:

Use test data identified in test cases 41,42,43,44

STEPS:

#	Action	Expected Result	Notes
1	See the linked test cases below to identify collections and granules to use for this test	Test data has been identified	
2	@151 [TC-151]		#comment
3	Test data can also be found at : @156 [TC-156] but this test case is not yet ready		#comment
4	@153 [TC-153]		#comment
5	@152 [TC-152]		#comment
6	Set up the Data Access environment to ensure that there are two versions of the HEG tool adapter running: one on a machine that is connected to the SAN/Datapool, and another which is not(this service will also need to be configured slightly differently). Verify that these services are configured to be available for the collections being used in this test.	The required processing services have been configured	
7	Follow the intsructions below to build and submit a number of processing		

#	Action	Expected Result	Notes
	requests. Make sure that some of them are synchronous and others asynchronous; that some are submitted to EGI via a POST, others a GET, and others are submitted via OMS; and that some are submitted to the SAN connected service and others to the unconnected service.		
8	You will use the ESI inventory drilldown service to build a request. If you already have EWOC driver properties files in place, you can skip this step.		
9	Go to http://<host>:<port>/esi_<MODE>/inventory	A list of collections should appear. This list contains all colelctions which have ConversionEnabledFlag set in the database and have at least 1 public granule. The granule count next to each collection comes from the datapool webaccess statistics tables which are updated periodically, so may not always be totally accurate.	
10	Click one of the collections	A list of granules will appear. Note that this is a subset of all granules in the collection as displaying thousands of granules here is not very useful.	
11	Click the "Order Form" button next to one of the granules.	A simple HTML form appears.	
12	Click the link for the service you wish to use	You will be taken to the section of the page for the tool you have selected.	
13	In the order form select your order options. All fields are optional and can be left blank. The available options will be based on the configuration in the Data Access GUI (http://<host>:<port>/DataAccessGui_<MODE>/).		
14	Request Type:	Select: -Synchronous if you are submitting a single granule and want it to be processed and returned directly to you, rather than getting back an order ID. A synchronous request will generally take longer to get a response, but there are less steps involved to get the data. -Asynchronous if you are submitting more than one granule and/or want a quicker response containing the order ID instead of waiting for the	

#	Action	Expected Result	Notes
		processing to complete first. Asynchronous is required if the request will be submitted to OMS.	
15	Format:	Select the desired output format. Note that currently, if the service being used is a GDAL service, Format is the only processing option you can select (cannot select projection, bands, resampling, or spatial subsetting).	
16	Projection/Projection Parameters:	Select the desired output projection and its parameters if reprojection is desired	
17	Bands:	Select the desired band to extract if band subsettign is desired	
18	Spatial Subset:	Specify a bounding box if spatial subsetting is desired	
19	Resampling:	Select a resample dimension and value if resampling is desired	
20	Email Address:	Enter your email address to receive status information about your request. This is required for asynchronous requests.	
21	Additional Granule IDs:	If the request type is Asynchronous and multiple granules from teh same collection are to be included in the same request (with all the same order options), Enter them here, separated by commas.	
22	Click the submit button that appears at the top and bottom of the section for each tool.	A page will appear with a number of submittal options.	
23	External EGI Request URL (POST): This is the simplest way to submit a request directly to EGI. This option works for both synchronous and asynchronous request types. Click the "Submit POST" button to submit the request Alternatively, you can copy the wget command line script listed below the submit button. This command should be run from the command line. This is useful as it can be saved in a file for later use or scripted easily.		
24	Response will be an XML file containing the order Id of the request along		

#	Action	Expected Result	Notes
	with a URL to get the status of the request (for asynchronous) or a list of download URLs (for synchronous).		
25	External EGI Request URL (GET) [synchronous only]: This only appears if Request Type is Synchronous, but it is useful in that it can be copied to a browser address bar and run that way. It is also easy to same in a text file or book mark for later use. Click the link to submit	Response will be an XML file containing download URLs for the results of the requested processing.	
26	EWOC parameter file optionselect line [asynchronous only]: This is the line you would add to an EWOC test driver properties file in order to perform the requested processing via OMS. The order will be submitted as an FTP Pull order. Once the provided line is in the properties file, there are some other things taht need to be added there. see an existing example. Once the file is complete, run: ./EcDmEwocTestClientStart DEV01 f4eil01 22500 <filename> n	Response to running EWOC test client is the created order id.	
27	ESI Request URL (GET): This should only be used if you are specifically submitting a request to the internal ESI service. This request will not be tracked in the database and avoids EGI and OMS. This is only for very specific testing cases.		
28	Submit requests that will result in HEG processing errors or no outputs. Use the list of NCRs and HEG improvements made to recognize such errors as a guide to complete the list below. Examples: a) Geographic subsetting for an area not covered by the granule in question. b) Re-projections with parameters that are invalid given the geographic coverage of the granule. c) Subsetting for bands that cannot be subsetted. d) The HEG tool adapter cannot be contacted. e) The HEG tool command fails. f) The HEG tool execution times out.	Verify that the Internal API returns, for each of the listed requests: a) A correct classification of the error b) An error description 	

TEST DATA:

EXPECTED RESULTS:

Verify that the Internal API returns, for each error case described above:
a) A correct classification of the error
b) An error description

7.13.7 DP_81_02_TP004 ESI HEG processing errors (ECS-ECSTC-381)

DESCRIPTION:

Test Case ID - 30

Submit requests that will result in HEG processing errors or no outputs. Use the list of NCRs and HEG improvements made to recognize such errors as a guide to complete the list below. Examples:

a) Geographic subsetting for an area not covered by the granule in question.

b) Re-projections with parameters that are invalid given the geographic coverage of the granule.

c) Subsetting for bands that cannot be subsetted.

d) The HEG tool adapter cannot be contacted.

e) The HEG tool command fails. The HEG tool execution times out.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Note, all requests in this test should be submitted to the EGI service as either synchronous or asycnchronous requests. The easiest way to build requests is to use the ESI inventory drilldown and then click the EGI link or use the provided wget command line.		#comment
2	Submit requests that will result in HEG processing errors or no outputs. a) Geographic subsetting for an area not covered by the granule in question. For this part first find teh granule to be used for this test. Use the Simple Inventory Interface to get some information about the granule: e.g. http://f4hel01:22500/SimpleInventoryInterface_DEV01/geofile/290275.xml for granule 290275 in the DEV01 mode. The elements in the XML named BBox will list the allowable bounding box dimensions for this granule. Any requests for subsetting outside of those coordinates would produce an error since the granule does not have that information.	The maximum allowed bounding box has been determined for the granule.	
3	Submit a request to EGI with coordinates that fall outside of the granule's range.	The request should fail and an error message should be produced.	
4	b) Re-projections with parameters that are invalid given the geographic coverage of the granule. This part deals with a reprojection to a different projection with invalid selected parameters. Such examples could be selecting negative numbers or out of bounds values for otherwise bounded parameters. For more help on selecting those values contact the HEG administrator. When ready head over to the ESI website and submit a valid granule processing with a re-projection to a different projection by selecting a projection and supplying invalid projection values.	The ESI website produces an error showing what went wrong.	

#	Action	Expected Result	Notes
5	c) Subsetting for bands that cannot be subsetted. For this part of the test select a band that is not supported for the granule. For more datails on selecting one contact the HEG support specialist. One example is the AE_Ocean collection, the Position_in_Orbit and Scan_quality_flag bands.	When processing an error is seen displaying what the problem is.	
6	d) The HEG tool adapter cannot be contacted. In order to do this stem one would have to stop the HegService tool adapter from the Tomcat Manager. Open the Tomcat Manager GUI for the mode that you are working for and find HegService_<MODE>. Click on the stop link to stop the tool adapter. Once the tool adapter has been stopped navigate back to the ESI website and try to process a granule.	Since the tool adapter has been stopped the granule should not be processed and a valid error message should be returned.	
7	e) The HEG tool command fails. The HEG tool execution times out. For this part select a granule that is part of collection that will take processing for a long enough time so that the HEG systems times out. If necessary set the timeout value sufficiently low in the HegService properties file such taht teh execution will not complete in the allowed time.	The system will produce the appropriate error - it will time out.	

TEST DATA:

EXPECTED RESULTS:

Verify that the Internal API returns:
a) A correct classification of the error
b) An error description

7.13.8 HEGModified

7.13.8.1 DP_81_02_TP001 Submit HEG processing requests (ECS-ECSTC-380)

DESCRIPTION:

Test Case ID - 10
Submit at least 50 and no more than 100 processing requests via the External Web API. Request the return of URLs pointing to the output files. Cover the processing options below (the list below can be expanded to list the precise set of requests to be submitted):
a) Spatial subsetting to an LLBOX with which the target granule overlaps without data subsetting.
b) Data subsetting at the object level, at the field level, and at the band level with and without spatial subsetting.
c) Re-projection to a projection that is valid for the target collection and granule, covering all types of projections as part of this test, with and without spatial and data subsetting
d) Resampling employing all resampling parameters that are offered by the API, covering granules that were spatially and data subsetted and re-projected, as well as some granules to which not all of these conditions apply.

e) Re-formatting to GeoTIFF, including multi-band GeoTIFF, as well as leaving the result in HDF-EOS format.
f) No change, i.e., obtaining the original granule without modification.

PRECONDITIONS:

For this test use at least two granules each from the following categories, including both public and hidden granules:
AMSR: AE_DySno, AE_L2A, AE_Ocean, AE_SI12
NSIDC MODIS: MOD10A1, MOD10A2, MOD10CM, MYD10_L2, MYD_29.5, MYD29P1N
Current EDC MODIS collections and at least one current MISR collection for each of the MISR processing levels L1B, 2, and 3 shall also be used in this test.

STEPS:

#	Action	Expected Result	Notes
1	Before starting this test go to the Data Access Configuration Interface website.	Verify that the collections that are to be processed are added as a part of a service. If they are not refer to the ITP that shows on how to add an existing collection to a service. Once verified that the collection is part of a service, it can be processed.	
2	Using the ESI/EGI website, submit at least 50 and no more than 100 processing requests via the External Web API(refer to the Pre-Conditions to see a list of Collections that need be tested). Request the return of URLs pointing to the output files. Over all of the requests divide them equaly amongst the next 6 processing steps. Using a working processing tool part of the ESI website, cover the processing option below: a) Spatial subsetting to an LLBOX with which the target granule overlaps without data subsetting. This can be achieved by putting values in the Spatial Subset section of the form.	Verify that the processing for a granule under the listed collections executes and processes with no errors. Also verify that the correctness of the result matches the one produced by the HEG standalone tool. To install the standalone tool and learn how to use it navigate to http://newsroom.gsfc.nasa.gov/sdptoolkit/HEG/HEGHome.html	
3	Using a working processing tool part of the ESI website, cover the processing option below: b) Data subsetting at the object level, at the field level, and at the band level with and without spatial subsetting. This option can be achieved by selecting any options from the Bands section tree of a granule processing. Try a number of options for each granule	Just as in the preceding step verify that there are no errors produced. Also verify that the corectness of the images produced(select GeoTiff in the Formats Option of the form) matches the picture produced from the standalone tool.	
4	Using a working processing tool part of the ESI website, cover the processing option below: c) Re-projection to a projection that is valid for the target collection and granule, covering all types of projections as part of this test, with and without spatial and data subsetting. This step can be done by	Verify that there are no errors produced. Also verify that the corectness of the images produced(select GeoTiff in the	

#	Action	Expected Result	Notes
	selecting any of the projections outlined on the ESI webform. If no parameters are put in the parameter boxes then the defaults will be chosen. Try to cover all scenarios, with empty boxes as well as boxes filled in with custom parameters.	Formats Option of the form) matches the picture produced from the standalone tool. The images produced for the same granule but with different projections should differ. Verify that this is the case.	
5	Using a working processing tool part of the ESI website, cover the processing option below: d) Resampling employing all resampling parameters that are offered by the API, covering granules that were spatially and data subsetted and re-projected, as well as some granules to which not all of these conditions apply. This option really combined all of the three steps above. Try and combine what was done before in one request.	Verify that there are no errors produced. Also verify that the corectness of the images produced(select GeoTiff in the Formats Option of the form) matches the picture produced from the standalone tool. The images produced for the same granule but with different resampling options should differ. Verify that this is the case.	
6	Using a working processing tool part of the ESI website, cover the processing option below: e) Re-formatting to GeoTIFF, including multi-band GeoTIFF, as well as leaving the result in HDF-EOS format. This option can be achieved by selecting GeoTiff from the Format dropdown. Try a few granules with both options only. Everything else should be left at its defult state.	Verify that there are no errors produced.	
7	Using a working processing tool part of the ESI website, cover the processing option below: f) No change, i.e., obtaining the original granule without modification. Process a granule with just the default options produced on the ESI form. Nothing should be selected	Verify that there are no errors produced.	

TEST DATA:

EXPECTED RESULTS:

For each of these cases, verify that the resulting HEG executions are correct, for example, by verifying that the HEG parameter file is correct and the correct executable will be invoked; or by verifying the resulting outputs against the result of the same HEG processing requests performed via the current software. Verify that the correct URLs are returned.

7.13.8.2 HEG Service request (ECS-ECSTC-658)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes

TEST DATA:

EXPECTED RESULTS:

7.13.8.3 TP001_10 (Revised, Automated) (ECS-ECSTC-660)

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 and HDF and GeoTiff 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/TP001_10/*.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/. If there is data already there, check if it needs to be moved to establish baseline-results for a previous release or test run.		

#	Action	Expected Result	Notes
3	Run TP001_10.rb (no parameters)	Console will show Curl command executions and will run for quite a while (~4 hrs.). 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:
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:
Generally - no HEG or HegService errors, correct and verified outputs.

7.14 ESI

7.15 EGI

7.15.1 DP_81_02_TP065 EGI filesize, mimetype, and filetype check (ECS-ECSTC-605)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Check HegService.properties and make sure HegService.application.LocalGetFileURL is not populated.		
2	Submit an EGI HEG request that will complete successfully	Status of request will be success	

#	Action	Expected Result	Notes
3	Check the AmDaOutputFile table for the files for the above request. Compare the file size in the table with the size on disk. Make sure they match. Also make sure filetype and mimetype in the table are not null.	file sizes match and filetype/mimetype are not null.	
4	Change HegService.properties and populate HegService.application.LocalGetFileURL with "HegService.application.LocalGetFileURL = http://f5dpl01v:22500/getfile_?FILE_URLS=". Change the host and mode parts accordingly.		
5	Repeat steps 2 and 3.		

TEST DATA:

EXPECTED RESULTS:

7.15.2 EGI Notification Level Tests (ECS-ECSTC-676)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Login to Reverb. If you do not have an account, you would need to create one.	Be logged in to Reverb.	
2	Go to your Account settings and change your Order Preferences, Receive order notification settings to "Always".		You may need to log out and log back in to have the settings take effect. bug# ECHO-182
3	Submit an EGI request through Reverb that will succeed.		
4	Make sure you get 2 emails. A pending/processing email and a completion email. (GLAS requests that require stitching may receive an additional "stitching complete" email.)		
5	Now change your Order Preferences, Receive order notification settings to Never.		You may need to log out and log back in to have the settings take effect. bug#

#	Action	Expected Result	Notes
			ECHO-182
6	Submit an EGI request through Reverb that will succeed.		
7	Make sure you do not get any emails for this request.		
8	Now change your Order Preferences, Receive order notification settings to "When orders reach a final state."		You may need to log out and log back in to have the settings take effect. bug# ECHO-182
9	Submit an EGI request through Reverb that will succeed.		
10	Make sure you get 1 email. You should only get the completion email. (GLAS requests that require stitching may receive an additional "stitching complete" email.)		
11	Now change your Order Preferences, Receive order notification settings to "When orders fail or are rejected"		You may need to log out and log back in to have the settings take effect. bug# ECHO-182
12	Submit an EGI request through Reverb that will succeed.		
13	Make sure you do not get any emails for this request.		
14	Submit an EGI request through Reverb that will fail.		
15	Make sure you only get the failure email.		
16	Now change your Order Preferences, Receive order notification settings to "When orders change to any state"		You may need to log out and log back in to have the settings take effect. bug# ECHO-182
17	Submit an EGI request through Reverb that will succeed.		
18	Make sure you get 2 emails. A pending/processing email and a completion email. (GLAS requests that require stitching may receive an additional "stitching complete" email.)		

TEST DATA:

EXPECTED RESULTS:

7.16 ESIR

7.16.1 DP_81_02_TP055 - Disable Zip Generation (ECS-ECSTC-501)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes

TEST DATA:

2014/02/20
==========

REGRESSION TEST - NCR_8050576 - DISABLE ZIP GENERATION

==========

Pre-Conditions
==========
DataAccess must be up-to-date in the mode used for testing.
The inventory drilldown (http://f5dpl01v:22500/<IF MODE=OPS:"esi" ELSE "esi_MODE">/) must be accessible.
Be able to edit EcDlDaEsir.properties and delete files from the zip staging area.
Access the Tomcat Application Manager.

Setup
==========
#S-1 [Create A Request]

From the inventory drilldown, perform a service on a granule:
> http://f5dpl01v:22500/esi_DEV02/inventory
> DEV02
> AE_Ocean.002
> Granules: [Order Form]
> HEG: [Submit Request to HEG]
> External EGI Request URL (POST): [Submit POST to EGI] OR External User Request (EGI): <URL>

> Note the request ID. It'll look like:
    http://f5eil01v:22500/esir_DEV02/<REQUEST_ID>.zip

Verify
==========
#V-1 [Download Service Products]

Retrieve the variable "ZIP.STAGING.AREA" from:
> x5eil01v:/usr/ecs/<MODE>/CUSTOM/cfg/EcDlDaEsir.properties

Download the files for your request.

Note the contents of this directory, particularly the zip file size:
> ls -ltr /usr/ecs/DEV02/CUSTOM/data/DPL/DaZipStaging | grep <REQUEST_ID>
    -rw-rw-r-- 1 cmshared cmshared   <ZIP_SIZE> Feb 21 07:00 <REQUEST_ID>.zip

Delete the zip file:
> rm /usr/ecs/DEV02/CUSTOM/data/DPL/DaZipStaging/<REQUEST_ID>.zip

---
#V-2 [Test Zip Generation Limit]

Edit EcDlDaEsir.properties:
> ZIP.GENERATION.LIMIT=1

Restart ESIR in Tomcat.

Attempt to download the zip files for the same request, using the URL from step [S-1].

#V-3 [Expected Output]

<REQUEST_ID> was obtained in step [S-1]. <ZIP_SIZE> was obtained in step [V-1].

No zip file generated for request: <REQUEST_ID>. The file exceeds the zip generation limit.
[INFO|ajp-bio-22067-exec-7|55|2014:02:21 16:45:57.190][esi.rest.result.resources.EsirResource|streamRequestFilesAsZip] Zip file is greater than configured limit for request: 15 where zip size of <ZIP_SIZE> is greater than max zip size: 1.0

EXPECTED RESULTS:

7.16.2 DP_81_02_TP060 ESIR Zip File Generation (ECS-ECSTC-601)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Create two succesful DataAccess EGI requests or use twi existing successful EGI requests. Make sure one request contains about 1GB of output files. The other request can have much less(around 200mb). Also make sure the zip files for these requests have not been generated or have been cleaned up.	Two completed egi requests with output files of about 1GB and 200MB.	You can manually delete the zip files if needed.
2	Using two different browsers(or two different browser windows) Request the zip generation of the request with 1GB of output files on both browsers. You can make the requests about 5-10 seconds apart.	The first browser that made the first request should finish first. The second browser with the second request should finish after the first request. The ESIR log should contain text that says that the ZIP file was already generated for the request.	
3	Manually clean up the zip files in step 2	Two completed egi requests with output files of about 1GB and 200MB.	
4	Using two different browsers(or two different browser windows) Request the zip generation of the request with 1GB of output files. Within 10 seconds, request the zip generation of the request with 200MB of output files on the second browser.	The 200MB request should complete before the 1GB request.	
5	After both requests complete in step 4. Make another request for the zip generation of the 200MB request.	The 200MB request should complete and the ESIR log should contain text that says that the ZIP file was already generated for the request.	

TEST DATA:

EXPECTED RESULTS:

7.17 SII

7.17.1 DP_81_02_TP009 Simple Inventory Interface (ECS-ECSTC-382)

DESCRIPTION:

Test Case ID - 100
Submit at least two requests for each type of SDPS inventory information supported by the simplified 'SDPS Inventory Web API', using that API:
a) Granule inventory information
b) Service information
c) Granule HDF information
Also submit requests that will result in these errors:
a) Requests for information about a granule that is not in the AIM/DPL inventory
b) Requests for a service that is not configured in ESI

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit at least two requests for granule inventory information by pointing your browser to: http://<host>:<port>/SimpleInventoryInterface_<MODE>/granule/<GranuleId>.xml	verify that you get a response containing XML conforming to the schema defined by http://newsroom.gsfc.nasa.gov/esi/8.1/schemas/SimpleInventoryInterface.xsd and http://newsroom.gsfc.nasa.gov/esi/8.1/schemas/doc/sii/index.html . Verify that the information in this XML validates against the schema and is correct.	
2	Submit at least two requests for granule inventory information by pointing your browser to: http://<host>:<port>/SimpleInventoryInterface_<MODE>/granules.html?lgid=/<LocalGranuleId>.xml	verify that you get a response containing an html table, containing the granule Ids for the specified local-granule-id	
3	Submit at least two requests for Service information by pointing your browser to http://<host>:<port>/SimpleInventoryInterface_<MODE>/service/<Service name>.xml	Verify that you get an XML response containing information about the specified ESI service. There is currently no XML schema defined for this response, but verify that it is well formed XML and looks like : <ServiceInfo><Service>&l	

#	Action	Expected Result	Notes
		t;ServiceId>1</ServiceId><Name>HEG</Name><Description>HDF-EOS to GeoTiff Converter(HEG)</Description><ServicePath>http://f4hel01:22500/HegService_DEV01</ServicePath><RequestType>sync</RequestType><MaxActiveAsyncJobs>4</MaxActiveAsyncJobs><MaxActiveSyncJobs>4</MaxActiveSyncJobs><Timeout>30</Timeout><AllowTemporalFlag>N</AllowTemporalFlag><AllowSpatialFlag>Y</AllowSpatialFlag><Host>i4oml01:45443</Host></Service></ServiceInfo>	
4	Submit at least two requests for Granule HDF information by pointing your browser to http://<host>:<port>/SimpleInventoryInterface_<MODE>/DataObjects/<granuleId>.xml and http://<host>:<port>/SimpleInventoryInterface_<MODE>/DataObjects/<granuleId>.xml?AGENT=gdalinfo	Verify that you get an XML response containing information information about the contents of the HDF file(s) associated with the specified granule. . There is currently no XML schema defined for this response, but verify that it is well formed XML and looks like : <DataObjects><File_URL>http://f4hel01:22500/getfile_DEV01?FILE_URLS=/datapool/DEV01/user/FS2/MOST/MOD10CM.005/2010.03.01/MOD10CM.A2010060.005.2010096215323.hdf</File_URL><Format>HDF4</Format><DataObject><DataObjectName>MOD_CMG_Snow_5km:Snow_Cover_Monthly_CMG</DataObjectName><Projection>GEOGRAPHIC</Projection><Projection_Parameters/><BBox>-90.0, -180.0, 90.0,	

#	Action	Expected Result	Notes
		180.0</BBox><DataOrg>EOS_GRID</DataOrg><hegable>true</hegable></DataObject><DataObject><DataObjectName>MOD_CMG_Snow_5km:Snow_Spatial_QA</DataObjectName><Projection>GEOGRAPHIC</Projection><Projection_Parameters/><BBox>-90.0, -180.0, 90.0, 180.0</BBox><DataOrg>EOS_GRID</DataOrg><hegable>true</hegable></DataObject></DataObjects>	
5	Submit a request for granule inventory information for a non existant granule by pointing your browser to : http://<host>:<port>/SimpleInventoryInterface_<MODE>/granule/<GranuleId>.xml where <GranuleId> refers to a granule that is not in the inventory.	Verify that an error is returned indicating that the specified granule was not found	
6	Submit a request for Service information for a non existant granule by pointing your browser to http://<host>:<port>/SimpleInventoryInterface_<MODE>/service/<Service name>.xml where Service name is not a configured service in ESI.	Verify that an error is returned indicating that the specified service was not found	
7	Submit a request for Granule HDF information for a non existent granule by pointing your browser to http://<host>:<port>/SimpleInventoryInterface_<MODE>/DataObjects/<granuleId>.xml where granuleId is not a valid granule in the inventory.	Verify that an error is returned indicating that the specified granule was not found	

TEST DATA:

EXPECTED RESULTS:
Verify the correctness of each result

7.18 DA GUI Configuration

7.18.1 Configure LDOPE Service (ECS-ECSTC-521)

DESCRIPTION:

PRECONDITIONS:

INTRODUCTION

There are several steps to configuring the LDOPE service for use with ECHO forms:
- Update the database with a service and a projection.
- Associate the service and the projection with various collections, using the Data Access GUI. (The LDOPE service will be used with the MOD09 QA Tool.)
- If not existing, create a tool adapter with which to process the granules. Otherwise, associate the tool adapter with the service in the GUI.
- (Generate a capabilities document/pull granules from above collections with ESI)
- View the results in ECHO or the Data Access Testbed.

Before completing this task, you will need to know:
- service name
- service host
- tool adapter URL/path
- supported request types
- supported collections

STEPS:

#	Action	Expected Result	Notes
1	LDOPE projections cannot be configured in the GUI until they have been added to the database. If desired, a service can be created in the GUI. If not, it will automatically be created during this step. Adding a projection to the database is best accomplished using the Korn shell script, EcDlDaAddLDOPEService, which to uses SQL statements to: - Create a service with which to associate the new projection (needed for ECHO forms) - Add rows to the tables: AmDaService, AmDaProjection, AmDaProjectionParam, AmDaProjectionParamValue, AmDaProjectionSrvXref The usage: EcDlDaAddLDOPEService.ksh $MODE $SERVICE_NAME [Note: If you opted to create a service in the GUI, $SERVICE_NAME should exactly match the name of the created service.]	The output: Mode: DEV02; Service: LDOPE /tools/postgres/9.2.4/bin/psql ecs -U EcDlDaDataAccess -h f4dbl03 -p 5432 set_search_path ---------- (1 row) DO	

#	Action	Expected Result	Notes
	An example: f5dpl01v{mbrown_8051146}~/Assignments/2013-08-30 > EcDlDaAddLDOPEService.ksh DEV02 LDOPE		
2	SELECT s.name AS service_name, p.name AS projection_name FROM AmDaService AS s JOIN AmDaProjectionSrvXref AS x ON s.serviceid=x.serviceid JOIN AmDaProjection AS p ON p.projectionid=x.projectionid WHERE s.name = 'LDOPE';	service_name | projection_name ----------+---------- LDOPE | LDOPE_Custom_Projection (1 row)	The number of rows may vary.
3	SELECT * FROM AmDaProjection WHERE projectionid = (SELECT projectionid FROM AmDaProjectionSrvXref WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE'));	projectionid | name | description ----------+----------+---------- 50 | LDOPE_Custom_Projection | LDOPE_Custom_Service (1 row)	The number of rows may vary.
4	SELECT * FROM AmDaProjectionParam WHERE projectionid = (SELECT projectionid FROM AmDaProjectionSrvXref WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE'));	paramid | projectionid | name | description | unit | projparamdatatype | minimum | maximum | defaultvalue | requiredflag | enumerationflag ---------+----------+------+----------+----------+----------+----------+----------+----------+----------+---------- 185 | 51 | Mask | Data mask type | Unitless | xs:string | | | Ideal | Y | Y (1 row)	The number of rows may vary.
5	SELECT * FROM AmDaProjectionParamValue WHERE paramid = (SELECT paramid FROM AmDaProjectionParam WHERE projectionid = (SELECT projectionid FROM AmDaProjectionSrvXref WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE')));	paramid | typevalue ---------+---------- 185 | Ideal 185 | Ideal/Good (2 rows)	The number of rows may vary.
6	Once LDOPE has been successfully added to the database as a service and a projection, then the GUI is needed. The following database tables will be affected:		

#	Action	Expected Result	Notes
	AmDaColProjectionSrvXref, AmDaDataSetConfig, AmDaServiceProvider, DlDaColHDFObjectSrvXref Using a browser, navigate to the Data Access GUI (l5dpl01:22500/DataAccessGui_). Once the GUI has loaded, open the tab "Service Configuration." > Click [Add New Service] > In the new service form, fill the text fields: > Request Type [Drop-down] > [Host & URL] OR [Service Path]* *Keep in mind that the path should include a Tomcat Server and a port, if the service depends upon it.		
7	Then, to map the services to the required collections, open the tab "Collection Configuration." > While holding the "Ctrl"/"Control" key, click on each of the collections that need to be mapped to the service.* > While selected, right-click, then choose: > [Configure New Service For Collection] > [LDOPE] > In the form, click [Update Service] *In this case, any collection beginning with "MOD09" will do for LDOPE.	Check the "Service Configuration" tab; the mapped collections/datasets should appear under the service(s) to which they were mapped. Double click any one of them to edit HDF Objects or any other variable.	
8	To verify your changes: SELECT * FROM AmDaServiceProvider WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE');	serviceproviderid | serviceid | servicepath | requesttype | timeout | host ----------+----------+----------+----------+----------+---------+------ 160 | 47 | | sync | 60 | (1 row)	The number of rows may vary.
9	SELECT * FROM AmDaColProjectionSrvXref WHERE projectionid = (SELECT projectionid FROM AmDaProjectionSrvXref WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE'));	collectionid | projectionid | serviceid | enabledflag ----------+----------+----------+---------- 21180 | 51 | 47 | Y 18424 | 51 | 47 | Y	The number of rows may vary.

#	Action	Expected Result	Notes
		3000003073	51	47	Y 3000003069	51	47	Y 3000003067	51	47	Y 3000118281	51	47	Y (6 rows)	
10	SELECT * FROM AmDaDataSetConfig WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE');	collectionid | serviceid | allowprojectionflag | allowformatflag | allowresamplingflag | allowinterpolationflag | allowbandflag | allowspatialflag | allowtemporalflag | enabledflag | asynconlyflag ----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+---------- 3000003073	47	Y	Y	Y	Y	Y	Y	Y	Y	N 3000003069	47	Y	Y	Y	Y	Y	Y	Y	Y	N 21180	47	Y	Y	Y	Y	Y	Y	Y	Y	N 3000003067	47	Y	Y	Y	Y	Y	Y	Y	Y	N 18424	47	Y	Y	Y	Y	Y	Y	Y	Y	N 3000118281	47	Y	Y	Y	Y	Y	Y	Y	Y |	The number of rows may vary.

#	Action	Expected Result	Notes
		Y | N (6 rows)	
11	 To link the adapter to the service, visit the "Service Configuration" tab of the GUI, and double-click on the "LDOPE" service. In the "Edit Service" dialog, edit the "Host" and "URL/Service Path" text fields: - If using a web-service adapter (not applicable to this delivery): Host: x5dpl01 URL/Service Path: http://x5dpl01:22500/HegService *If {MODE} is not "OPS", then the URL should be http://x5dpl01:22500/LdopeService_{MODE} - If using a command-line adapter: Host: x5dpl01:22500 URL/Service Path: /usr/ecs/{MODE}/CUSTOM/utilities/tempService.sh As before, click [Update Service] to save your changes. To verify the database edits, use the following query, substituting '%LDOPE%' for '%%' if needed:		
12	SELECT * FROM AmDaServiceProvider WHERE serviceid IN (SELECT serviceid FROM AmDaService WHERE name LIKE '%LDOPE%');	serviceproviderid | serviceid | servicepath | requesttype | timeout | host ----------+----------+----------+----------+----------+----------+----------+---------- 162 | 49 | l5dpl01:/usr/ecs/{MODE}/CUSTOM/utilities/tempService.sh | sync | 1 | f5dpl01v:22500 (1 row)	
13	 For the next part of the test, move to the "Collection Configuration" tab of the DA GUI. In it, associate "LDOPE" (not "LDOPE_CMD") with a collection by:	select * from amdadatasetconfig where serviceid = (select serviceid from amdaservice where name = 'LDOPE');	

#	Action	Expected Result	Notes
	     - Right-click on a collection - Click on "Configure New Service for Collection" - In the pop-up menu to the right, click on "LDOPE" To verify, use the following query: SELECT * FROM AmDaDatasetConfig WHERE serviceid = (SELECT serviceid FROM AmDaService WHERE name = 'LDOPE');	collectionid | serviceid | allowprojectionflag | allowformatflag | allowresamplingflag | allowinterpolationflag | allowbandflag | allowspatialflag | allowtemporalflag | enable dflag | asynconlyflag ----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+---------- 23986 | 5 | Y | Y | Y | Y | Y | Y | Y | Y | N (1 row)	
14	 In the DataAccess Testbed (x5dpl01:22500/esi)... ...select the appropriate mode beneath "Inventory Drilldown," which will provide a list of collections in that mode. The chosen collection should be one of those linked to the new service (LDOPE, in this case) through the DA GUI. Click the link. Two sections should appear: "Capabilities" and "Granules." Capabilities: The "Capabilities" section merely displays the XML documents that are processed by ECHO for display. Granules: The "Granules" section provides usable documents that will actually send and display data from various processing units. To verify that the command line adapter works, select "[Order Form]" for any granule. The next page should list "LDOPE" under "Available Processing Tools." Click the link to automatically scroll to the appropriate section.		

#	Action	Expected Result	Notes
	     Choose a "Synchronous" "Request Type", then [Submit request to LDOPE]. Open the link under "ESI Request URL (GET)" (the last option). The expected result is XML containing all of the parameters sent to tempService.sh as the order. Observe the stylesheet changes under the section "Format". The allowed parameters should include "Ideal" and "Ideal/Good". To simplify the test, a synchronous request is preferred. Choose it under "Request Type." Then click "[Submit Request to LDOPE]." Four options exist for submitting the granule order, three of which do not involve a command-line interface; the first three involve EGI, which will return a message indicating the success or failure of the request and provide download URLs. The fourth option is an ESI Request URL, which will submit the parameters in a HTTP GET request; in the dominant example discussed throughout this document (tempService.sh), this option will display the output of the tool adapter--which has been defined to simply return the parameters (given as arguments) in XML format. 		

TEST DATA:

EXPECTED RESULTS:

7.18.2 Configure Convert Enabled Flag (ECS-ECSTC-523)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	ESI Drill down:		It should list all the ECS collections that either have Convert-Enabled flag set to "Y" or have service associations. And the collections should include hidden granules.
2	Pick an ECS collection with Convert-Enabled flag set to 'Y'		
3	Map it to all services in DataAccess GUI	Verify that the collection is listed in the ESI drill down.	
4	Unmap the collection from all services in DataAccess GUI	Verify that the collection is also listed in the ESI drill down.	
5	Pick an ECS collection with Convert-Enabled flag set 'N'		
6	Map it to all services in DataAccess GUI	Verify the collection is listed in the ESI drill down.	
7	Pick an ECS collection with hidden granules as well as public granules		
8	Set the Convert-Enabled flag to 'Y'	Verify the hidden granules are listed, together with public granules	
9	Data Access GUI:		When associating a service with all the ECS collections or individual collections, Convert-Enabled value is ignored.
10	Identify at least one ECS collection with Convert-Enabled flag set to 'Y' and one ECS collection with Convert-Enabled flag set to 'N',		
11	Create a new service		
12	Associate it with all collection by clicking on "map-all-collections"	Verify that all the collections can be associated with the service, regardless of their Convert-Enabled value.	
13	Pick two ECS collections with no service associations		
14	Set the Convert-Enabled flag to 'Y' for one collection and 'N' for the other.	Verify that both collections can be associated with a service individually.	

TEST DATA:

EXPECTED RESULTS:

8 EMS

8.1 Test Case 1e - g - EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to specify an output directory for the flat files. (ECS-ECSTC-271)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to specify an output directory for the flat files.]		#comment
2	L4 Requirement 80		
3	Verify that the directory listed for the EMSEXTRACTDIR tag in the EcDbEMSdataExtractor.CFG file is the directory where the EMS extract utility has extracted the dataset files to. The EcDbEMSdataExtractor.CFG file is located in the /usr/ecs/<MODE>/CUSTOM/cfg directory.		
4	DataSets: All [EMS Extract Utility:The EMS Dataset Extraction Utility shall apply filenames that uniquely distinguish each flat file based on the following: a)Beginning date for the reporting period b)DAAC name c)EMS flat file type]		
5	L4 Requirement 90		
6	Verify that the Name given to a file is the same name listed in the EcEMSextractRecord table for that extract run and that the name contains the Beginning date for the reporting period, the DAAC name, and the EMS flat file type.		
7	DataSets: All [EMS Extract Utility: A record of the Extract utility run shall be kept in a log file and in the EcEMSextractRecord table]		
8	L4 Requirement 100		
9	Run the EMS extract utility using a command similar to the following:		
10	EcDbEMSdataExtractor.pl -m <MODE> -s "start date" -e “end date” -v -x DistFTP –o		

#	Action	Expected Result	Notes
11	Verify that a record of the run can be found in the log that is created and that a record for the run can be found in the EcEMSextractRecord table.		
12	DataSets: All [EMS Extract Utility: Verify that in extracted dataset files, there is one record per line and each record is divided into fields using '|&|' as a delimiter.]		
13	Visually inspect the files that have been created by the extract script.		
14	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.2 Test Case 2 - EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to override the default 24-hour period execution with an alternative starting and ending date. (ECS-ECSTC-272)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to override the default 24-hour period execution with an alternative starting and ending date.]		#comment
2	L4 Requirement 50		#comment
3	Run the following EMS extract utility command with or without the –v flag (verbose) with dates entered > than 24-hour period, i.e., one week apart: EcDbEMSdataExtractor.pl -m <MODE> -s "<start date>" -e "<end date>" -v -x <Dataset> -o		
4	This command can be run using any of the dataset ExtractType tags: Meta, searchExp, Ing, DistFTP, DistHTTP, or Arch.		
5	Verify that one (1) output dataset file is created for dataset and date range and that the file is scp'd to the configured destination.		
6	Verify that a record for this dataset run has been entered in the		

#	Action	Expected Result	Notes
	EcEMSextractRecord table and that the starting and ending reporting period matches the start date and end date entered in the command.		
7	Verify that the ExecutionMode for the record is recorded as override in the table.		
8	[(see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to specify a single flat file type for generation and export.]		#comment
9	L4 Requirement 70		#comment
10	Run the following EMS extract utility command with or without the –v flag (verbose) with dates entered for 24-hour period, or larger date range: Data for the specified dataset will be extracted in files for each 24-hour period. EcDbEMSdataExtractor.pl -m <MODE> -s "<start date>" -e "<end date>" -v -x <Dataset>		
11	This command can be run using any of the dataset ExtractType tags: Meta, searchExp, Ing, DistFTP, DistHTTP or Arch.		
12	Note: This capability allows the user to recover missing data for a specific dataset between a specified start date and end date. This test can be verified with test 2.		
13	Verify that the command run above has specified a single flat file type for generation and export for each 24-hour period between the start date and end date.		
14	Verify that a single record for this dataset run has been entered in the EcEMSextractRecord table.		

TEST DATA:

EXPECTED RESULTS:

8.3 Test Case 3 - EMS Extract Utility: The EMS Dataset Extraction Utility shall rename a flat file if it has been resent to the EMS. (ECS-ECSTC-273)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utility shall rename a flat file if it has been resent to the EMS.]		#comment
2	L4 Requirement 100		
3	Run the EMS extract utility using a command similar to the following:		
4	EcDbEMSdataExtractor.pl -m <MODE> -s "start date" -e “end date” -v -x DistFTP –o		
5	Run the command again and verify that the file name created is the same except for “.rev1” appended to the end of the name.		
6	Verify that for each subsequent run of the command, the rev number is incremented 1.		
7	Note: this capability ensures that if a flat file for a dataset has already been sent to EMS, that subsequent runs of the data extract script will not overwrite the previously sent file.		
8	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.4 Test Case 4 - EMS Extract Utility: (see Note) Verify that the EMS extract utility can be run as a cron (ECS-ECSTC-274)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: (see Note) Verify that the EMS extract utility can be run as a cron]		#comment
2	L4 Requirement 100		
3	Log into LINUX as the user that the cron will be run as. Set-up the cron by running the crontab –e command on the server that the cron will be run.		
4	The command will be something like: 51 16 * * 2 (export		

#	Action	Expected Result	Notes
	LD_LIBRARY_PATH=/tools/sybOCv12.5.1/lib:/home/cmops/lib;/bin/csh -c "cd /usr/ecs/TS1/CUSTOM/utilities; EcDbEMSdataExtractor.pl -m TS1")		
5	Note: command may be different based on the configuration for the server. This command should only be run after setting the defaults for execution of the script in default mode. If test 1 is set-up to run as a cron, this test is verified.		
6	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.5 RECOVERY (see Note) EMS Extract Utility: Verify for the default configuration run of the EMS extract utility, thatif the script does not run for a few days, the script can determine when it last ran successfully by retrieving the last successful run infor (ECS-ECSTC-275)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[RECOVERY (see Note) EMS Extract Utility: Verify for the default configuration run of the EMS extract utility, thatif the script does not run for a few days, the script can determine when it last ran successfully by retrieving the last successful run information from the EcEMSextractRecord table and automatically create files for missing days for each 24-hour period since the last successful run.]		#comment
2	This can be verified and simulated by setting the defaults for execution of the script in default mode about a week earlier than the current date. Then the Execution script can be run using the following parameters:		
3	EcDbEMSdataExtractor.pl -m TS1		
4	Verify that after execution files have been created for each dataset, upto the current date. Verify that the file is located in the extract directory and a record		

#	Action	Expected Result	Notes
	has been created in the EcEMSextractRecord table.		
5	Note: this capability of the EMS extract utility ensures that if the utility has not been run for a period of time, that the script can start running from the time it was previously run.		
6	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.6 Test Case 6 - RECOVERY (see Note) EMS Extract Utility: Verify that if a dataset file has been extracted to the extract directory, but not sftpâ€™d, a subsequent run of the utility will sftp this file and mark the file as sftpâ€™d (ECS-ECSTC-276)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[RECOVERY (see Note) EMS Extract Utility: Verify that if a dataset file has been extracted to the extract directory, but not sftp’d, a subsequent run of the utility will sftp this file and mark the file as sftp’d]		#comment
2	From the files that have been recently created during testing, find a record for the file in the EcEMSextractRecord table. In this table for that record set the FTPcompletionTime to NULL.		
3	Run the EMS Extract utility using the following command.		
4	EcDbEMSdataExtractor.pl -m TS1 -s "start date" -e "end date"		
5	Verify that the files that have had the FTPcompletionTime set to NULL have been scp’d to the sftp site and that the FTPcompletionTime has been updated.		
6	Note: This simulates files that have not previously been sftp’d, picked up automatically by the Extract script. This capability of the EMS extract script allows files to be sent that may not have been sent previously because of a		

#	Action	Expected Result	Notes
	down link between the DAAC and EMS.		
7	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.7 RECOVERY (see Note) EMS Extract Utility: Verify that if a dataset file has been removed from the extract directory, but not sftpâ€™d, a subsequent run of the utility will not crash when an attempt to sftp this file is made. Instead, the file will be marke (ECS-ECSTC-277)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[RECOVERY (see Note) EMS Extract Utility: Verify that if a dataset file has been removed from the extract directory, but not sftp’d, a subsequent run of the utility will not crash when an attempt to sftp this file is made. Instead, the file will be marked sftp’d and a note documenting this will be written to the log.]		#comment
2	From the files that have been recently created during testing, find a record for the file in the EcEMSextractRecord table. In this table for that record set the FTPcompletionTime to NULL.		
3	Locate this file in the bcp extract directory and remove it.		
4	Run the EMS Extract utility using the following command.		
5	EcDbEMSdataExtractor.pl -m TS1 -s "start date" -e "end date" -v		
6	Verify that the files that have had the FTPcompletionTime set to NULL and removed from the directory have had their FTPcompletionTime updated. Verify that a corresponding note for this file has been place in the log. Verify that the script finishes.		
7	Note: This simulates files that have not previously been sftp’d and have been		

#	Action	Expected Result	Notes
	removed or moved from the location that has been documented in the EcEMSextractRecord table. This capability ensures that the EMS extract script will continue to run, even if it is unable to locate the file that is to be sent to EMS.		
8	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.8 (see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to override the default 24-hour period execution with an alternative starting and ending date. The data will be output to extract files in 24-hour (ECS-ECSTC-278)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[(see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to override the default 24-hour period execution with an alternative starting and ending date. The data will be output to extract files in 24-hour periods between the date range specified]		#comment
2	Run the following EMS extract utility command with or without the –v flag (verbose) with dates entered > than 24-hour period, ie, one week apart:		
3	EcDbEMSdataExtractor.pl -m <MODE> -s "<start date>" -e "<end date>" -v		
4	To verify this, specify a date range of about one week.		
5	Verify that output dataset files are created for each dataset and the date range specified and that the file is scp’d to the configured destination. Verify that a dataset file is created for each 24-hour period. Verify that a record for this dataset run has been entered in the EcEMSextractRecord table and that the starting and ending reporting period matches the start date of each dataset file. Verify that the ExecutionMode for the record is recorded as override in		

#	Action	Expected Result	Notes
	the table.		
6	Note: This EMS extract script capability allows files for datasets to be generated for a user specified period of time.		
7	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.9 Test Case 9 - EMS Extract Utility: The EMS Dataset Extraction Utility will not output data in less than a 24-hour period (ECS-ECSTC-279)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utility will not output data in less than a 24-hour period]		#comment
2	Run the following EMS extract utility command with or without the –v flag (verbose) with dates entered less than than 24-hour period:		
3	EcDbEMSdataExtractor.pl -m <MODE> -s "<start date>" -e "< end date >" -v		
4	To verify this, specify a date range less than 24-hours. Verify that no output datasets are created and there is no entry in the EcEMSextractRecord table for this specified date range.		
5	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.10 [EDF OPS / PVC TS2]Test Case 11 - EMS Extract Utility:Any EMD code changes will need to be Verified against the EMS ICD and established Processes (ECS-ECSTC-281)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility:Any EMD code changes will need to be Verified against the EMS ICD and established Processes]		Need to be tested in EDF OPS or PVC TS2. PVC-TS2 used to be setup as a provider. EDF-OPS is set up to talk to EMS UAT.
2	Note: This testing will be conducted in PVC between EMD and EMS and at the DAACs between DAAC and EMS. Testing between PVC and EMS must be coordinated with EMS.		

TEST DATA:

EXPECTED RESULTS:

8.11 (see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to extract QA | PH data for DistFTP | DistHTTP (ECS-ECSTC-282)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[(see Note) EMS Extract Utility: The EMS Dataset Extraction Utility shall provide the operator the capability to extract QA | PH data for DistFTP | DistHTTP]		#comment
2	Run the following EMS extract utility command with or without the –v flag		

#	Action	Expected Result	Notes
	(verbose) with dates entered for 24-hour period, or larger date range: The range provided should include DataPool Web accesses to QA and PH data and FTP OMS orders of QA | PH granules. Data for the specified dataset will be extracted in files for each 24-hour period.		
3	EcDbEMSdataExtractor.pl -m <MODE> -s "<start date>" -e "<end date>" -v -x <Dataset>		
4	This command can be run using the following dataset ExtractType tags: DistFTP, DistHTTP.		
5	Note: The HTTP scripts may no longer be running at DAACs.		
6	Verify that the command run above has specified a single flat file type for generation and export for each 24-hour period between the start date and end date. Review the flat file created and ensure that the flat file contains QA | PH data.		
7	Compare with record from database and ensure that the data in flat file matches the record in database. Comparison should be made for OMS and DPL QA | PH.		
8	DataSets:		

TEST DATA:

EXPECTED RESULTS:

8.12 EMS Extract Utility: The EMS Dataset Extraction Utilityshall allow operator to set a default date for subsequent runs of the utility either for all data sets or for one dataset (ECS-ECSTC-283)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utilityshall allow operator to set a default date for subsequent runs of the utility either for all data sets or for one dataset]		#comment
2	The following datasets can be used when running this test: a) Product		

#	Action	Expected Result	Notes
	attribute metadata: Meta b) Product attribute search: searchExp c) Data Ingest: Ing d) Data Archive: Arch e) distribution orders DistHTTP (Datapool, Transfer) f) distribution orders DistFTP (DataPool, FtpPush, FtpPull, Scp)		
3			
4	This test sets the initial date that the default configuration of the utility should begin processing. See “Initializing the EMS Extract script prior to default execution” at end of document. A possible set of the initial default datasets could be one week earlier than the current date. This can be run for one dataset or for all datasets; “default” is written to the ExecutionMode field in the EcEMSextractRecord table and subsequent runs of the utility in default configuration (see test 1).		
5	Run one of the following EMS extract utility commands:		
6	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v –i		
7	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v –I –x <dataset>		
8	Verify that for command run above that “default” has been written to the EcEMSextractRecord table ExecutionMode column for the dataset.		
9	DataSets:		

TEST DATA:

EXPECTED RESULTS:

8.13 (see Note) EMS Extract Utility: The EMS Dataset Extraction Utilityshall allow operator to set the LAG time for running of default dataset(s) (ECS-ECSTC-284)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	<i>[(see Note) EMS Extract Utility: The EMS Dataset Extraction Utilityshall		#comment

#	Action	Expected Result	Notes
	allow operator to set the LAG time for running of default dataset(s)]</i>		
2	The following datasets can be used when running this test: a) Product attribute metadata: Meta b) Product attribute search: searchExp c) Data Ingest: Ing d) Data Archive: Arch distribution orders DistFTP (DataPool, FtpPush, FtpPull, Scp)		
3	DistHTTP (DataPool, Transfer) 		
4	This test sets the LAG time for subsequent runs of the EMS Extract script in default mode. This allows a variation in how current the data is that is sent to EMS. The LAG time is set by modifying the value for LAG in the EcDbEMSdataExtractor.CFG file.		
5	Note: The default LAG time set in configuration file is -1, meaning if the EMS extract script is run today for default configuration, data will be provided up to the day before yesterday. Setting the value to 0 will provide data up to yesterday. Setting the value to 1 will provide data up to current time		
6	Run one of the following EMS extract utility commands after setting the LAG time and setting the default for the dataset (see test 13) :		
7	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v –i		
8	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v –I –x <dataset>		
9	Verify that for command run, datasets have been run up to the time specified by the LAG configuration variable.		
10	DataSets:		

TEST DATA:

EXPECTED RESULTS:

8.14 (see Note) EMS Extract Utility: shall allow operator to set the DESTINATIONDIR for data that is sent to EMS (ECS-ECSTC-285)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[(see Note) EMS Extract Utility: shall allow operator to set the DESTINATIONDIR for data that is sent to EMS]		#comment
2	The following datasets can be used when running this test: a) Product attribute metadata: Meta b) Product attribute search: searchExp c) Data Ingest: Ing d) Data Archive: Arch e) distribution orders DistFTP (DataPool, FtpPush, FtpPull, Scp)		
3	f) DistHTTP (DataPool, Transfer)		
4	This test sets the DESTINATIONDIR for data that is sent to EMS.		
5	Note: The default DESTINATIONDIR for data sent to EMS is the home directory of the account. This allows for sending data to a subdirectory. The full path for the subdirectory should be specified.		
6	Run one of the following EMS extract utility commands after setting the DESTINATIONDIR in the configuration file.		
7	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v		
8	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v –x <dataset>		
9	Verify that for command run, the dataset has been sent to the specified directory.		
10	DataSets:		

TEST DATA:

EXPECTED RESULTS:

8.15 (see Note) EMS Extract Utility: shall allow intendedUsage data to be sent to EMS for MSS/OMS data (ECS-ECSTC-286)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[(see Note) EMS Extract Utility: shall allow intendedUsage data to be sent to EMS for MSS/OMS data]		#comment
2	The following datasets can be used when running this test: a) distribution orders DistFTP (DataPool, FtpPush, FtpPull, Scp)		
3	b) DistHTTP (DataPool, Transfer) 		
4	This test provides data that has intendedUsage set for MSS/OMS data.		
5	Note: The intendedUsage is set in the OmRequestGranule table.		
6	Run the following EMS extract utility commands for data range that contains data populating intendedUsage .		
7	EcDbEMSdataExtractor.pl -m TS1 -s "<start date>" -e "<end date>" -v –x <dataset>		
8	Verify that for command run, the dataset contains intendedUsage data.		
9	DataSets:		

TEST DATA:

EXPECTED RESULTS:

8.16 EMS Modified

8.16.1 EMS Extract Utility: If only the <begin date> is specified for the EMS Dataset Extraction Utility the script will output dataset files from that specific day forward to the current date minus 2 days (ECS-ECSTC-280)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: If only the <begin date> is specified for the EMS Dataset Extraction Utility the script will output dataset files from that specific day forward to the current date minus 2 days]		#comment
2	Verify that the lag time is properly set. Go to the EMS host and view the		

#	Action	Expected Result	Notes
	EcDbEMSdataExtractor.CFG configuration file. Verify that the LAG is set to -1		
3	Run the following EMS extract utility command with or without the –v flag (verbose) with dates entered as indicated:		
4	EcDbEMSdataExtractor.pl -m <MODE> -e "< end date >" -v (It's no longer valid to pass in end date alone, without begin date.)		
5	EcDbEMSdataExtractor.pl -m <MODE> -s "< begin date >" -v		
6	Verify that dataset files are created for that date forward and that there is a record of the run process in the EcEMSextractRecord table.		
7	DataSets: All		

TEST DATA:

EXPECTED RESULTS:

8.17 Test Case 1 - EMS Extract Utility: Verify that flat files are created and exported for all datasets when Extract utility is run using default configuration (ECS-ECSTC-889)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: Verify that flat files are created and exported for all datasets when Extract utility is run using default configuration]		#comment
2	L4 Requirement 10		#comment
3	The following datasets are extracted and exported using the extraction utility: a) Product attribute metadata: Meta b) Product attribute search: searchExp c) Data Ingest: Ing d) Data Archive: Arch		

#	Action	Expected Result	Notes
	e) Physical media distribution orders DistFTP (DataPool, FtpPush, FtpPull, Scp) DistHTTP (DataPool)		
4	Note: add a test for extract type ArchDel . EMS script –d option. Before running this test, set the initial date that the default configuration of the utility should begin processing. See Initialize default running of EMS extract utility at end of document. A possible set of the initial default datasets could be one week earlier than the current date.		
5	Run the following EMS extract utility command with or without the –v flag (verbose): EcDbEMSdataExtractor.pl -m -v		
6	Verify that output dataset files are created for datasets and that these files are scp'd to the configured destination. This test can be set up to run as a cron.		

TEST DATA:

EXPECTED RESULTS:

8.18 Test Case 1a - EMS Extract Utility: The EMS Dataset Extraction Utility shall maintain a record of the status and starting/ending reporting period of each execution. (ECS-ECSTC-890)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	EMS Extract Utility: The EMS Dataset Extraction Utility shall maintain a record of the status and starting/ending reporting period of each execution.		
2	L4 Requirement 20		

#	Action	Expected Result	Notes
3	For the default EMS script execution run above, verify that a record for these dataset runs has been entered in the EcEMSextractRecord table and that the starting and ending reporting period is entered for each execution record.		

TEST DATA:

EXPECTED RESULTS:

8.19 Test Case 1b - EMS Extract Utility: The EMS Dataset Extraction Utility shall by default generate and export EMS flat file information for a 24-hour period (ECS-ECSTC-891)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utility shall by default generate and export EMS flat file information for a 24-hour period.]		
2	L4 Requirement 30		
3	For the default EMS script execution run above, verify that the period of time used for each record is listed as 24 hours.		

TEST DATA:

EXPECTED RESULTS:

8.20 Test Case 1c - EMS Extract Utility: The EMS Dataset Extraction Utility shall by default use the ending date of the previous reporting period to determine the starting and ending date for the current execution. (ECS-ECSTC-892)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: The EMS Dataset Extraction Utility shall by default use the ending date of the previous reporting period to determine the starting and ending date for the current execution.]		
2	L4 Requirement 40		
3	For the default EMS script execution run above, verify that the period of time used for each record is consecutive periods of 24 hours.		

TEST DATA:

EXPECTED RESULTS:

8.21 Test Case 1d - EMS Extract Utility: The EMS Dataset Extraction Utility shall by default generate and export each of the following types of flat files per execution (ECS-ECSTC-893)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	EMS Extract Utility: The EMS Dataset Extraction Utility shall by default generate and export each of the following types of flat files per execution:		
2	a) Product attribute metadata b) Product attribute search c) Data Ingest d) Data Archive e) Physical media distribution orders		
3	L4 Requirement 60		
4	The following datasets are extracted and exported using the extraction utility: a) Product attribute metadata: Meta b) Product attribute search: searchExp		

#	Action	Expected Result	Notes
	c) Data Ingest: Ing d) Data Archive: Arch e) Physical media distribution orders DistFTP (DataPool, FtpPush, FtpPull, Scp) DistHTTP (DataPool)		
5	For the default EMS script execution run above, verify that output dataset files are created for all datasets listed above and that these files have been scp'd to the configured destination.		

TEST DATA:

EXPECTED RESULTS:

8.22 Test Case 1h - EMS Extract Utility: Verify that in extracted dataset files, there is one record per line and each record is divided into fields using '|&|' as a delimiter. (ECS-ECSTC-900)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[EMS Extract Utility: Verify that in extracted dataset files, there is one record per line and each record is divided into fields using '|&|' as a delimiter.]		
2	Visually inspect the files that have been created by the extract script.		

TEST DATA:

EXPECTED RESULTS:

9 HEG

9.1 Resample, swtif and gdtif conversion (ECS-ECSTC-287)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Resample, swtif and gdtif conversion]		#comment
2	Use the HEG Automated Testing Script (HATS) to perform these tests.		#comment
3	Refer to the HEG_Regression_Test_Spreadsheet for a list of tests.		#comment
4	Generate an XML file for each row in the spreadsheet.		
5	Once all XML files (each XML file represents a request) have been generated and a flat file with a list of these XML files has been created, invoke the EcDlHATSClientStart script by passing in the <MODE> and location of the flat file. The HATS GUI will load up and begin processing each request.		
6	Once all requests finish, perform verification.		
7	Examine each request as follows: (1) verify that the HATS GUI displays a blue status for the request, (2) verify that the number of met files is equal to the number of hdf or tif files, and (3) view each hdf and tif file using either HDFView (for hdf files) or xv (for tif files) to make sure the file displays an appropriate image.		

TEST DATA:

ASTER_DEM, AST_L1B, AE_Ocean, AE_DyOcn, AE_DySno, AE_SI12, MOD10C1, MOD11_L2, MOD09A1, MOD43B1, and MOD15A2, MI1B2E. (see spreadsheet)

EXPECTED RESULTS:

10 MSS

10.1 System Components Availability (ECS-ECSTC-288)

DESCRIPTION:
Verifies System Components can be successfully monitored.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Up or down status of system components – monitoring, event detection, notification		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the up or down status of the following system components: a. SDPS custom application host (x5oml01, x5dpl01, etc) b. inetd c. dns d. http e. ssh g. nfs f. sendmail g. Tomcat/Apache h. Oracle iPlanet Java System Web Server i. System mount point		
3	Verify that it is possible to configure the service to monitor all of the components for up or down status.		
4	Verify that the System Monitoring, Event Detection, and Response GUI displays the current status of all components which are being monitored, and that the status is correctly displayed as up.		
5	Bring down each of the monitored components.		
6	Verify that the GUI displays the current status of all of the components as down within one monitoring interval.		

TEST DATA:

EXPECTED RESULTS:

10.2 Alerting Capability (ECS-ECSTC-289)

DESCRIPTION:
Verifies alerting capabilities including email and text message.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure the System Monitoring, Event Detection and Response service to send an email two at least two recipients when a custom code server has a change in status from up to down.		
2	Configure the System Monitoring, Event Detection and Response service to escalate the alert if it is not fixed within a configurable period of time. Set it up in the following manner: a. email to at least two recipients different than the original two b. text message to at least two recipients		
3	Configure the primary alert notification to fire only once when the alert is detected.		
4	Verify that it is possible to configure the alerts and escalations.		
5	Verify that it is possible to configure the frequency at which the primary alert notification will fire.		
6	Verify that all email messages are received and that the text of all notification messages correctly identifies the alert.		
7	Allow the server to remain down for a period longer than the time period configured for the escalation.		
8	Verify that text messages are received and that the text of all notification messages correctly identifies the alert.		

TEST DATA:

EXPECTED RESULTS:

10.3 Resource usage monitoring (ECS-ECSTC-290)

DESCRIPTION:
Verifies capability to monitor resource usage metrics.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure the System Monitoring, Event Detection, and Response service to monitor the resource usage on one SDPS host.		
2	Configure the service to capture all of the metrics listed (from S-MSS-05120): 1) Memory usage 2) CPU usage 3) I/O characteristics (e.g., I/O rates, counts, queue size, service time, wait time, average I/O size) 4) Paging rate 5)Swap space 		
3	Verify that all of the metrics are captured and displayed on the GUI according to the collection interval defined for the metrics.		

TEST DATA:

EXPECTED RESULTS:

10.4 Disk Space Usage (ECS-ECSTC-291)

DESCRIPTION:
Verifies disk usage can be monitored correctly.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Space utilization monitoring and event detection		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor space utilization for each of the following ECS components: a. Data Pool file system b. Archive file system (e.g. snfs, amfs) c. Postgres database device d. Custom log directory		
3	Verify that it is possible to configure the service to monitor the space utilization of each of the components.		
4	Verify that the GUI displays the current space utilization of all of the components configured according to the monitoring interval.		

TEST DATA:

EXPECTED RESULTS:

10.5 Custom Code Availability (ECS-ECSTC-292)

DESCRIPTION:
Verifies ability to monitor the up and down status of custom code servers as well as restart of a custom code server.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Up or down status of Custom code and COTS services – monitoring, event detection, response		#comment
2	NOTE: This must be done in a controlled mode (e.g., OPS, TS3), because Hyperic does not understand ClearCase.		#comment
3	Configure the System Monitoring, Event Detection, and Response service to monitor the up or down status of each of the following services: a. Quick Servers b. Ingest servers c. OMS server d. EPD e. DPAD f. SSS g. AIM processes (IIU, XVU) h. BMGT servers i. EWOC j. DataAccess k. Order Status		
4	Configure the System Monitoring, Event Detection, and Response service to restart one of the custom code servers listed in S-40-1 (Step 1) when the server becomes unavailable.		
5	Ensure that all services being monitored are currently up.		
6	Verify that it is possible to configure the System Monitoring, Event Detection, and Response service to monitor services for up or down status.		
7	Verify that it is possible to configure the restart of the custom code server in S-40-2 (Step 2).		
8	Verify that the System Monitoring, Event Detection, and Response GUI displays the current status of all services which are being monitored, and that the status is correctly displayed as up.		
9	Bring down all of the services being monitored.		
10	Verify that the GUI displays the current status of all of the services as down within the configured monitoring interval.		

#	Action	Expected Result	Notes
11	Verify that the custom code server which was configured to be restarted in S-40-2 (Step 2) is restarted.		
12	Verify that at the next collection interval the status of the custom code server configured in S-40-2 (Step2) is displayed as up.		
13	Verify that the execution of the restart of the custom code server is logged and that the log entry includes the time stamp.		

TEST DATA:

EXPECTED RESULTS:

10.6 GUI â€“ Configure Operator and Roles (ECS-ECSTC-293)

DESCRIPTION:
Verify the ability to configure Hyperic operators and roles.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	GUI user access		#comment
2	Log in to the System Monitoring, Event Detection, and Response GUI with a user that has ‘security admin’ privileges.		
3	Create the following types of operator logins: a. View only b. Two mode monitoring administrators (two different modes) c. Security admin		
4	Verify that the ‘security admin’ operator is able to successfully create each of the operator logins.		
5	GUI user roles		#comment
6	Login to the System Monitoring, Event Detection, and Response GUI as a user with ‘security admin’ privileges.		
7	Use existing users or create new users such that there are at least four different operators.		
8	Using the System Monitoring, Event Detection, and Response GUI create the following user roles: a. System Administrator b. Database		

#	Action	Expected Result	Notes
	Management c. ECS operations		
9	Configure the users such that one user belongs to the System Administrator role, one user belongs to the Database Management role, one user belongs to the ECS operations role, and one user has all three roles.		
10	Verify that it is possible to create each of the user roles.		
11	Verify that it is possible to assign a single role to a single user.		

TEST DATA:

EXPECTED RESULTS:

10.7 DPL Ingest Business Process (ECS-ECSTC-294)

DESCRIPTION:
Verify the ability to configure and monitor the DPL Ingest business process.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Business Process Monitoring – DPL Ingest		#comment
2	[S-60-1] Ensure that the following resources are setup to be monitored by the System Monitoring, Event Detection, and Response service: a. DPL Ingest host b. ECS service hosts used for checksumming, archiving, and file transfers for the given mode c. Polling Service d. Processing Service e. Notification Service f. Quickserver on each ECS service host g. All Data Pool file systems h. All StorNext file systems used for Ingest i. Postgres host and database j. Apache / Tomcat web server hosting DPL Ingest GUI k. StorNext primary metadata server		
3	Using the System Monitoring, Event Detection, and Response GUI login as a ‘monitoring admin’ operator and create the DPL Ingest business process.		
4	Configure the DPL Ingest business process to include all of the resources listed in S-60-1 (Step 1).		
5	Configure the DPL Ingest business process to include the monitoring of an		

#	Action	Expected Result	Notes
	individual data provider queue size and throughput metrics.		
6	[S-60-4] Configure the business process rules to mark the DPL Ingest process as ‘Down’ if any of the following occur: a. DPL Ingest host is unavailable b. Postgres is unavailable c. Notification, Polling, or Processing Services are down d. All ECS service hosts used for checksumming are unavailable e. All ECS service hosts used for archiving are unavailable f. All ECS service hosts used for file transfers are unavailable 		
7	[S-60-5] Configure the business process rules to mark the DPL Ingest process as ‘Degraded’ if any of the following occur: a. At least one but not all Quickservers on ECS service hosts used for checksumming are unavailable b. At least one but not all Quickservers on ECS service hosts used for archiving are unavailable c. At least one but not all Quickservers on ECS service hosts used for file transfer are unavailable d. StorNext primary metadata server is down e. The Ingest queue is larger than N number of granules and the granules per minute throughput is less than M granules per minute. Note: For e. above the tester may configure values for N and M to be small in order to test this criterion in a non-performance mode.		
8	[S-60-6] Configure the business process rules to mark the DPL Ingest process as ‘Inactive’ if any of the following occur: a. Ingest throughput over the previous hour is 0.		
9	Ensure that at the start of the test all resources are available and that no Ingest has occurred during the previous hour.		
10	Verify that the current status of the DPL Ingest business process is displayed on the GUI as ‘Inactive’.		
11	Verify that it is possible to view the status of all of the DPL Ingest resources grouped together.		
12	Verify that it is possible to configure each of the business rules in S-60-4.		
13	Verify that it is possible to configure each of the business rules in S-60-5.		
14	Verify that it is possible to configure each of the business rules in S-60-6.		
15	Verify that the current status of one of the ingest resources is ‘Alert Pending’ and there is an explanation which identifies the Ingest throughput as being 0.		
16	Verify that all other resources are marked as available.		
17	Begin trickling in granules for Ingest at a rate greater than M (as configured in S-60-5 part e.) granules per minute. Once the granules begin to complete Ingest successfully:		

#	Action	Expected Result	Notes
18	Verify that the DPL Ingest business process status is changed from ‘Inactive’ to ‘Active’ within the configured monitoring interval.		
19	Verify that all of the DPL Ingest resources are displayed as available.		
20	Bring down one, but not all of the Quickservers used for file transfers. Verify that the DPL Ingest business process status is changed from ‘Active’ to ‘Degraded’ within the configured monitoring interval.		
21	Verify that there is a resource with an alert pending which identifies the Quickserver that is unavailable.		
22	Bring down the remaining Quickservers used for file transfers such that they are all unavailable.		
23	Verify that the DPL Ingest business process status is changed from ‘Degraded’ to ‘Down’ within the configured monitoring interval.		
24	Verify that on the business process page displaying the DPL Ingest resources, the Quickservers are displayed as down.		
25	Bring up all of the Quickservers used for file transfers.		
26	Verify that the DPL Ingest business process status is changed from ‘Down’ to ‘Active’ within the configured monitoring interval.		
27	Verify that all of the DPL Ingest resources are displayed as available.		

TEST DATA:

EXPECTED RESULTS:

10.8 Resource Availability and Usage Reporting (ECS-ECSTC-295)

DESCRIPTION:
Verifies resource availability and usage reporting.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Custom Code Server Availability Report		#comment
2	[S-70-1] Log in as the ‘security admin’ operator and configure the System		

#	Action	Expected Result	Notes
	Monitoring, Event Detection, and Response GUI to persistently store metrics information for a period of no less than 10 days.		
3	Generate a report detailing the availability of one of the custom code servers over a time period which is less than the value configured in S-70-1. The custom code server must have the following characteristics: a. Availability data must be present for the custom code server over the entire time period b. There must be both uptime and downtime for the server during that time range (availability must be greater than 0% and less than 100%).		
4	Verify that it is possible to configure the number of days to persistently store historic metric data.		
5	Verify that the GUI displays the total uptime and total downtime over the specified time period.		
6	Verify that the GUI displays a graph detailing the availability over the specified time period.		
7	Verify that is possible to save and view the report outside of the GUI.		
8	Resource Usage Report		#comment
9	Generate a report detailing the CPU usage of one of the SDPS code hosts over a time period less than the value configured in S-70-1. Availability data must be present for the host over the entire time period.		
10	Verify that the GUI displays the min, max, and average CPU usage over the specified time period.		
11	Verify that the GUI displays a graph detailing the CPU usage over the specified time period.		
12	Verify that is possible to save and view the report outside of the GUI.		

TEST DATA:

EXPECTED RESULTS:

10.9 Ingest Queue Size Monitoring (ECS-ECSTC-296)

DESCRIPTION:
Verifies capability to monitor the size of Ingest queues.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ingest Request Queue Size Monitoring		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the size of the Ingest request queue for two individual providers as well as the overall system.		
3	Ensure that there are requests queued for both of the providers for the duration of the test.		
4	Verify that the System Monitoring, Event Detection, and Response GUI displays the name of the Ingest request queue and the size of the queue by provider and overall.		
5	Ingest Granule Queue Size Monitoring		#comment
6	Configure the System Monitoring, Event Detection, and Response service to monitor the size of the Ingest granule queue for two individual providers as well as the overall system.		
7	Ensure that there are granules queued for both of the providers for the duration of the test.		
8	Verify that the System Monitoring, Event Detection, and Response GUI displays the name of the Ingest granule queue and the size of the queue by provider and overall.		
9	Ingest Notification Queue Size Monitoring		#comment
10	Configure the System Monitoring, Event Detection, and Response service to monitor the size of the Ingest notification queue for two providers as well as the overall system.		
11	Ingest data to ensure the size of the notification queue is greater than zero for each provider. Note: It is acceptable to bring the notification service down to prevent the actions from being worked off.		
12	Verify that the System Monitoring, Event Detection, and Response GUI displays the name of the Ingest notification queue and the size of the queue by provider and overall.		
13	Ingest Throughput Monitoring		#comment
14	[S-200-1] Configure the System Monitoring, Event Detection, and Response service to monitor the following Ingest throughput metrics for two individual providers as well as the overall system: a. Average number of science granules ingested per minute over the previous five minute sample. This will be referred to as metric A throughout the remainder of the test. b.		

#	Action	Expected Result	Notes
	Average MB per minute ingested over the previous five minute sample. c. Estimated time to complete ingest for all items currently in the queue based and current throughput as calculated in metric A.		
15	Configure the collection interval for the metrics in S-200-1 to be less than five minutes. Note: If the collection interval is greater than five minutes the metrics will not cover the entire time period.		
16	Verify for each of the metrics that the System Monitoring, Event Detection, and Response GUI displays the name of the metric and its current value by provider as well as the overall system.		
17	Verify that the current value of each metric is correct.		

TEST DATA:

EXPECTED RESULTS:

10.10 Ingest Alert and Intervention Monitoring (ECS-ECSTC-297)

DESCRIPTION:
Verifies Ingest alert and intervention monitoring.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ingest Alert and Intervention Counts with Warning Notifications		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the following for two individual providers as well as the overall system: a. Number of open Ingest alerts b. Number of open Ingest interventions		
3	Cause alerts and interventions such that the values are nonzero.		
4	Verify that it is possible to configure the service to monitor the number of open Ingest interventions and the number of open Ingest alerts by provider and for the overall system.		
5	Verify that the System Monitoring, Event Detection, and Response GUI displays the correct counts for the number of open Ingest interventions and		

#	Action	Expected Result	Notes
	the number of open Ingest alerts by provider and for the overall system.		

TEST DATA:

EXPECTED RESULTS:

10.11 Database Monitoring (ECS-ECSTC-298)

DESCRIPTION:
Verifies database monitoring capabilities.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure the System Monitoring, Event Detection, and Response service to monitor the main Postgres server		
2	Configure it such that the database connection to the Postgres server is monitored.		
3	Cause an issue such that you cannot connect to the database server.		
4	Verify that the database server connection is marked as unavailable.		
5	Bring down the database server.		
6	Verify that the database server process is marked as unavailable.		

TEST DATA:

EXPECTED RESULTS:

10.12 File Transfer Monitoring (ECS-ECSTC-299)

DESCRIPTION:
Verifies Wu-FTP monitoring.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	File transfer monitoring		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor transfers via Wu-FTP		
3	Transfer several files via Wu-FTP and verify that the Hyperic GUI displays the number of open sessions, number of files transferred, and the average throughput of files transferred.		
4	Configure the System Monitoring, Event Detection, and Response service to monitor transfers via HTTP downloads		
5	Transfer several files via HTTP and verify that the Hyperic GUI displays the number of open sessions, number of files transferred, and the average throughput of files transferred.		

TEST DATA:

EXPECTED RESULTS:

10.13 External Service Monitoring (ECS-ECSTC-300)

DESCRIPTION:
Verifies monitoring of external services.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	External service monitoring		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the up or down status of the following system components: a. Order Status GUI b. DPL DataAccess		
3	Configure the System Monitoring, Event Detection, and Response service to monitor the up or down status of the following external components: a. HSA b. ECHO Reverb c. ECHO API		

#	Action	Expected Result	Notes
4	Ensure that all components being monitored are currently up.		
5	Verify that it is possible to configure the service to monitor components for up or down status.		
6	Bring down each of the monitored components.		
7	Verify that the GUI displays the current status of all of the components as down.		

TEST DATA:

EXPECTED RESULTS:

10.14 OMS Alert and Intervention Monitoring (ECS-ECSTC-301)

DESCRIPTION:
Verifies capability to monitor OMS alerts and interventions.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	OMS alert and intervention monitoring		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the following: a. Number of OMS Alerts b. Number of OMS Interventions		
3	Cause OMS interventions and alerts such that the numbers are nonzero and the values are different from one another.		
4	Verify that it is possible to configure the service to monitor the number of open OMS interventions and the number of open OMS alerts.		
5	Verify that the System Monitoring, Event Detection, and Response GUI displays the correct counts for the number of open OMS interventions and the number of open OMS alerts.		

TEST DATA:

EXPECTED RESULTS:

10.15 OMS Queue Monitoring (ECS-ECSTC-302)

DESCRIPTION:
Verifies ability to monitor OMS queues.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	OMS queue size monitoring		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the OMS Granule Queue size and OMS Request Queue Size.		
3	Order granules such that the queue sizes are nonzero and different values from one another.		
4	Verify that the System Monitoring, Event Detection, and Response GUI displays the name of the OMS granule and request queues and the size of the queues.		

TEST DATA:

EXPECTED RESULTS:

10.16 Custom Code Metrics (ECS-ECSTC-303)

DESCRIPTION:
Verifies capability to monitor all custom code server metrics.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Custom Code Server Monitoring – Metrics		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the following custom code servers ensuring that all of the metrics defined in C-MSS-05265 are captured: a. Action Driver b. All Quick Server instances c. SSS Subscribed Event Driver d. SSS Delete Request Driver e. SSS Action Driver f. SSS Recovery Driver g. Order Manager server h. DPL Ingest Notification Service i. DPL Ingest Processing Service j. DPL Ingest Polling Service k. BMGT Dispatcher Service l. BMGT Auto Service m. BMGT Monitor Service 		
3	Verify that the metrics collected for each server can be viewed via the System Monitoring, Event Detection, and Response service.		

TEST DATA:

EXPECTED RESULTS:

10.17 Custom Utility Monitoring (ECS-ECSTC-304)

DESCRIPTION:
Verifies capability to monitor various custom code utilities.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Custom Code Utility Monitoring		#comment
2	Configure the System Monitoring, Event Detection, and Response service to monitor the following custom code utilities ensuring that CPU utilization Memory and utilization metrics (defined in C-MSS-05265) are captured for each of the following: a. Granule Deletion utilities b. QA Update utility c. Online Archive utilities d. Move Collection utility e. Checksum Verification utilities f. Data Pool cleanup utilities		

#	Action	Expected Result	Notes
3	Verify that the metrics collected for each utility can be viewed via the System Monitoring, Event Detection, and Response service.		

TEST DATA:

EXPECTED RESULTS:

10.18 Data Access Business Process (ECS-ECSTC-305)

DESCRIPTION:
Verifies ability to configure and monitor the Data Access business process.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Business Process Monitoring – Data Access		#comment
2	[S-600-1] Ensure that the following resources are setup to be monitored by the System Monitoring, Event Detection, and Response service: a. x5eil01 b. wu-ftp server c. Apache/Tomcat on x5eil01 d. DPL Web Access GUI e. Postgres Database server f. Web Order Status GUI g. DPL file systems h. ECHO/REVERB interface i. Local DAAC web page		
3	Using the System Monitoring, Event Detection, and Response GUI login as a ‘monitoring admin’ operator and create the Data Access business process.		
4	Configure the Data Access business process to include all of the resources listed in S-600-1 (Step 1).		
5	Configure the DPL Ingest business process to include the monitoring of the following metrics: a. Number of wu-ftp sessions b. Number of file transfers via ftp c. ftp transfer rates d. Number of file transfers via http e. http transfer rates f. Number of DPL data access sessions		
6	Configure the business process rules to mark the Data Access process as ‘Down’ if any of the following occur: a. The x5eil01 host is unavailable b. Postgre is unavailable c. The DPL file systems		

#	Action	Expected Result	Notes
	are unavailable		
7	Configure the business process rules to mark the Data Access process as ‘Degraded’ if any of the following occur: b. The wu-ftp server is unavailable c. Apache/Tomcat on x5eil01 is unavailable f. The Web Order Status GUI is unavailable g. The ECHO/REVERB interface is unavailable		
8	Configure the business process rules to mark the Data Access process as ‘Inactive’ if the following occur:		
9	Number of ftp transfers and number of http transfers over the previous monitoring interval is 0.		
10	Ensure that at the start of the test all resources are available and that no data accesses have occurred during the previous hour.		
11	Verify that the current status of the Data Access business process is displayed on the GUI as ‘Inactive’.		
12	Verify that it is possible to view the status of all of the Data Access resources grouped together.		
13	Verify that it is possible to configure each of the business rules in S-600-4 (Step 4).		
14	Verify that it is possible to configure each of the business rules in S-600-5 (Step 5).		
15	Verify that it is possible to configure each of the business rules in S-600-6 (Step 6).		
16	Verify that all Data Access resources are marked as available.		
17	Order data via the DPL Data Access.		
18	Verify that the Data Access business process is changed from ‘Inactive’ to ‘Active within the monitoring interval.		
19	Bring down the x5eil01 host.		
20	Verify that the Data Access business process is changed from ‘Active’ to Down’ within the configured monitoring interval.		
21	Verify on the Data Access business processing page that the x5eil01 host as well as all resources running on those hosts are displayed as down.		
22	Restart the x5eil01 host.		
23	Manually start the custom code servers on those hosts.		
24	Verify that the Data Access business process status is changed from ‘Down’ to ‘Active’ within the configured monitoring interval.		
25	Verify that all of the Data Access resources are displayed as available.		

TEST DATA:

EXPECTED RESULTS:

10.19 Order Management Business Process (ECS-ECSTC-306)

DESCRIPTION:
Verifies ability to configure and monitor the Order Management business process.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Business Process Monitoring – Order Management		#comment
2	Ensure that the following resources are setup to be monitored by the System Monitoring, Event Detection, and Response service: a. OMS host b. HEG processing host c. ftp push host (x5eil01 at most locations) d. Order Manager server e. EWOC f. EPD server g. HSA h. Postgres i. DPL file systems j. Copy Server (currently x5eil01, but it is configurable)		
3	Using the System Monitoring, Event Detection, and Response GUI login as a ‘monitoring admin’ operator and create the Order Management business process.		
4	Configure the Order Management business process to include all of the resources listed in S-610-1 (Step 1).		
5	Configure the Order Management business process to include the monitoring of the following metrics: a. OMS request queue size b. OMS granule queue size c. Order throughput average over the previous five minutes (by granule count and data volume per destination and ftp transfer type) d. Number of open OMS alerts e. Number of open OMS interventions f. Number of suspended destinations		
6	Configure the business process rules to mark the Order Management process as ‘Down’ if any of the following occur: a. OMS host is unavailable b. Postgres is unavailable c. The Order Manager server is down		

#	Action	Expected Result	Notes
7	Configure the business process rules to mark the Order Management process as ‘Degraded’ if any of the following occur: a. The HEG processing host is unavailable b. The EWOC is unavailable c. The EPD server is unavailable d. The HSA is unavailable e. The OMS granule queue is larger than N number of granules and the granules per minute order throughput is less than M granules per minute. f. There are open alerts g. There are open interventions h. There are suspended destinations Note: For e. above the tester may configure values for N and M to be small in order to test this criterion in a non-performance mode.		
8	Configure the business process rules to mark the Order Management process as ‘Inactive’ if all of the following occur: a. Order throughput average over the previous five minutes is 0 b. OMS request queue size is 0 c. OMS granule queue size is 0.		
9	Ensure that at the start of the test all resources are available, that no orders have been processed over the previous five minutes, and that the OMS queues are 0.		
10	Verify that the current status of the Order Management business process is displayed on the GUI as ‘Inactive’.		
11	Verify that it is possible to view the status of all of the Order Management resources grouped together.		
12	Verify that it is possible to configure each of the business rules in S-610-4 (Step 4).		
13	Verify that it is possible to configure each of the business rules in S-610-5 (Step 5).		
14	Verify that it is possible to configure each of the business rules in S-610-6 (Step 6).		
15	Submit ftp push orders to the system and verify that the status of the Order Management business process is changed from ‘Inactive’ to ‘Active’ within the monitoring interval.		
16	Verify that the order throughput average for the configured destination for orders placed in S-610-3 becomes greater than 0 within the monitoring interval. (Step 3)		
17	Suspend a configured destination and verify that the status of the Order Management business process is changed from ‘Active’ to ‘Degraded’ within the monitoring interval.		
18	Verify that there is a resource with an alert pending which identifies that the		

#	Action	Expected Result	Notes
	configured destination is suspended.		
19	Shut down the Order Manger server and verify that the status of the Order Management business process is changed from ‘Degraded’ to ‘Down’ within the monitoring interval.		
20	Verify on the Order Manager business processing page that the Order Manager server is displayed as down.		
21	Restart the Order Manager server and re-enable the suspended destination.		
22	Verify that the Order Management business process status is changed from ‘Down’ to ‘Active’ within the configured monitoring interval.		
23	Verify that all of the Order Management resources are displayed as available.		

TEST DATA:

EXPECTED RESULTS:

11 OMS

11.1 Test Case 5 - DPL file system unavailable (ECS-ECSTC-311)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	DPL file system unavailable		#comment
2	Submit multiple requests for granules with ESDTs on different file systems.		
3	Use the DPM GUI to make one of the Data Pool file systems unavailable.		
4	Verify the Operator Alerts page in the OMS GUI shows that the file system is suspended.		
5	Verify that an email is sent to the configured address for operator alerts containing details about the suspended file system.		
6	Verify that granules that reside on the suspend file system and their associated requests are not promoted to staging.		

#	Action	Expected Result	Notes
7	Resume the Data Pool file system using the DPM GUI.		
8	Verify that the alert is cleared from the OMS GUI.		
9	Verify the granules that reside on the previously suspended file system are promoted to staging and all requests get Shipped.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.2 [OBSOLETE] Test Case 7 - Ftp Pull HEG Order (ECS-ECSTC-313)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ftp Pull HEG Order		#comment
2	Select five HEGable granules that allow band subsetting and spatial subsetting. select 'SC:'||esdt(t1.shortname,t1.versionid)||':'||t1.granuleid as geoid from amgranule t1 join amdadatasetconfig t2 on t1.collectionid = t2.collectionid where t2.allowbandflag = 'Y' and t2.allowspatialflag = 'Y' and t1.publishtime is not null and t2.enabledflag = 'Y' order by t1.granuleid desc limit 5;		
3	Using the EWOC client, submit a Ftp Pull order to EGI for each granule using “GeoTiff” conversion and “Geographic” projection for some of the granules. For at least one granule select spatial subsetting coordinates that do not match the band subsetting choice.		
4	Verify that an Acceptance email is sent to user.		
5	Verify that the Acceptance email contains processing instructions.		
6	Use the OMS GUI distribution page and the OMS pending Requests page to track progress.		
7	Once the request state is shipped, use the OMS GUI distribution page to		

#	Action	Expected Result	Notes
	verify that a Distribution Notice Email is sent to the user.		
8	Verify that the links in the distribution notice page are functional.		
9	Verify that the links can be used to download the files.		
10	Verify that the granule with the spatial/band subsetting discrepancy was marked as failed in the OMS GUI and was not distributed to the user.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.3 [OBSOLETE] Test Case 8 - Ftp Push HEG Order (ECS-ECSTC-314)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ftp Push HEG Order		#comment
2	Select a HEGable granule. select 'SC:'||esdt(t1.shortname,t1.versionid)||':'||t1.granuleid as geoid from amgranule t1 join amdadatasetconfig t2 on t1.collectionid = t2.collectionid where t2.allowbandflag = 'Y' and t2.allowspatialflag = 'Y' and t1.publishtime is not null and t2.enabledflag = 'Y' order by t1.granuleid desc limit 1;		
3	Using the EWOC client, submit a Ftp Push order to EGI for the granule using “GeoTiff” conversion and “Geographic” projection.		
4	Verify that an Acceptance email was sent to the user.		
5	Verify that the Acceptance email contains processing instructions.		
6	Use the OMS GUI distribution page and the OMS pending Requests page to track progress.		
7	Once the request state is shipped, use the OMS GUI distribution page to verify that a Distribution Notice Email is sent to the user.		

#	Action	Expected Result	Notes
8	Verify that the destination directory contains files.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.4 Test Case 10 - Ftp Push/SCP Destination Automatic Retry (ECS-ECSTC-316)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ftp Push/SCP Destination Automatic Retry		#comment
2	Create 3 requests so that they contain an error that causes the FtpPush operations to fail.		
3	Submit the 3 FtpPush orders for granules to be sent to a single FtpPush destination configured for automatic retry.		
4	After the destination has been suspended and the alert has been generated, update the distribution requests such that they will now succeed, using the update feature specified in S-OMS-06270 from ticket OD_S4_01.		
5	Verify the automatic retry executes.		
6	Verify the alert is removed from the list of pending alerts after the next screen refresh.		
7	Verify the FtpPush operations complete successfully.		
8	Repeat this test for SCP media type.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.5 Test Case 11 - Ftp Push/SCP Manual Destination Retry (ECS-ECSTC-317)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ftp Push/SCP Manual Destination Retry		#comment
2	Create 3 requests so that they contain an error that causes the FtpPush operations to fail.		
3	Submit the 3 FTP Push orders for granules to be sent to a single FtpPush destination configured for automatic retry.		
4	After the destination has been suspended and the alert has been generated, use the OM GUI to switch the destination to manual suspension.		
5	Update the distribution requests such that they will now succeed, using the update feature specified in S-OMS-06270 from ticket OD_S4_01.		
6	Verify the automatic retry does not execute.		
7	Resume the destination manually.		
8	Verify the alert is removed from the list of pending alerts after the next screen refresh.		
9	Verify the FtpPush operations complete successfully.		
10	Repeat this test for SCP media type.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.6 Test Case 12 - FTP Push/SCP: Granule Error Operator Intervention (ECS-ECSTC-318)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	FTP Push/SCP: Granule Error Operator Intervention		#comment
2	Submit an FtpPush distribution request that has several granules.		
3	Cause one (and only one) of the granules to run into an FtpPush error that does not cause the destination to be suspended (e.g., delete or rename one of its files after it was staged but before it is pushed).		
4	Verify the destination does not suspend.		
5	Verify an intervention is queued.		
6	Verify the status of the destination is displayed correctly.		
7	Verify it is possible to fail the granule.		
8	Verify after failing the problem granule and closing the Operator intervention, any remaining granules are pushed, but not the failed granule. Note that failing the granule would not affect any concurrent FtpPush operation for other files belonging to that granule, i.e., there is no requirement that they be cancelled.		
9	Verify after all the remaining files have been pushed, a correct DN email is sent for FtpPush (see S-OMS-06673 from ticket OD_S4_01 for FtpPush) and the SCP notification is left on the destination for SCP distribution.		
10	Verify a DN email is sent, describing the failed granule.		
11	Verify the DN explains that this granule is the only one that has not yet completed pushing.		
12	Verify the DN identifies the failure.		
13	Repeat this test for SCP media type.		
14	Verify that the SCP directory contains a notification file, describing the failed granule that matches the email DN.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.7 OBSOLETE - Test Case 13 - ODL to XML (ECS-ECSTC-319)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	ODL to XML		Now done for S4PM by default.
2	Submit an FtpPull request via SSS or EWOC, using an email address configured for ODL to XML metadata conversion.		Http replaced Ftppull
3	Verify the requests Stage and Ship.		
4	Verify DN emails are sent.		
5	Verify the request directory is created in the DPL FtpPull area.		
6	Verify data file links are created in the DPL FtpPull area.		
7	Login to the ftp server (via anonymous login).		Obsolete
8	Verify the directory permissions of the created directory.		
9	Verify the symlinks are not visible.		
10	Verify that other request directories are not visible.		
11	Download the files from the ftp server (via anonymous login).		
12	Verify that all files listed are retrieved successfully.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.8 [PVC ONLY] Test Case 14 - External Subsetting Order (ECS-ECSTC-320)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	External Subsetting Order		#comment
2	As ECSGuest, submit an order via EWOC for External Processing (via s4pm or HSA) with FtpPush as the media type.		# HSA is obsolete. This test is done through S4PM on demand during a checkout or part of the performance run
3	Repeat the order as another registered user.		
4	Verify the requests Stage and Ship.		
5	Verify DN emails are sent.		
6	Verify data gets pushed to the ftp push destination.		
7	Verify the UserId for the configured user is shown on the OMS GUI Distribution Request List Page.		
8	Verify that all output products are cleaned up with request.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.9 [PVC ONLY] Test Case 16 - Fault Recovery â€“ Bouncing the Order Manager Server (ECS-ECSTC-322)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Fault Recovery – Bouncing the Order Manager Server		#comment
2	Stop and start the Order Manager server three times under load.		
3	Verify that the requests do not go into operator intervention.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.10 Test Case 17 - OMS GUI: Non Privileged User Test (ECS-ECSTC-323)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	OMS GUI: Non Privileged User Test		#comment
2	Verify that the OM GUI on startup asks for a login from the operator.		
3	Verify the OM GUI sets appropriate access privileges. I.e. It won't allow "full access" features to be used until a valid password is used.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.11 Test Case 20 - Global Non-Configured FtpPush Configuration (ECS-ECSTC-324)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Global Non-Configured FtpPush Configuration		#comment
2	Note: Make sure that the Min Throughput is set to a reasonable (or very high) value and will not affect this test. A value of 100+ should be good.		#comment

#	Action	Expected Result	Notes
3	Time Out:		#comment
4	On the host configured as the FtpServiceEndpoint in EcOmOrderManager.CFG Replace the EcDlFtpService with an executable that only sleeps for 20 seconds and returns 0.		
5	Set the Timeout to a number below 20 seconds (5-10 seconds is preferred).		
6	Submit an FtpPush request to a non-configured destination.		
7	Verify that the FtpPush actions resulted in a timeout via the Request Detail screen in OMGUI.		
8	Set the Timeout above 20 seconds (preferably 30 seconds).		
9	Resubmit the request.		
10	Verify that all the granules are marked as shipped.		

TEST DATA:

EXPECTED RESULTS:

11.12 Test Case 21 - Valgrind Memory Test (Optional) (ECS-ECSTC-325)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Valgrind Memory Test (Optional)		#comment
2	Repeat the set-up steps for test numbers 1 through 14 while running OMS through valgrind with --leak-check=full option and –log-file=<filename>. Verification steps for the repeated tests can be ignored.		
3	Verify there are no significant memory leaks: Check the valgrind logs for memory that has been definitely lost. If this number exceeds 25kBs, then it constitutes a significant memory leak.		

TEST DATA:

EXPECTED RESULTS:

11.13 Tet Case 22 - DeleteFromArchive = H Granules (ECS-ECSTC-326)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	DeleteFromArchive = H Granules		#comment
2	Mark a granule DeleteFromArchive = 'H' in AmGranule.		
3	Submit an FtpPull, FtpPush, or SCP request for this granule using the Spatial Subscription GUI, EWOC client, or the acquire utility.		
4	Verify that the request goes into Operator Intervention with appropriate message.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.14 EcOmCleanupOldData.pl (ECS-ECSTC-395)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure OMS configuration is current and correct (config files, properties files, database settings, etc.); Have cmshared access on x5oml01v.		

#	Action	Expected Result	Notes
3	Ensure the following file exists: /usr/ecs/<MODE>/CUSTOM/dbms/postgres/OMS/scripts/EcOmCleanupOldData.pl For reference, read: /usr/ecs/<MODE>/CUSTOM/dbms/postgres/OMS/OmCleanupOldData.sp		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/9.2/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('OMS', $MODE); show search_path;	Should list the schemas installed in the mode. Should include 'oms' for OPS or 'oms_<MODE>'.	"readonly" is an acceptable username.
5	Ensure that an OMS GUI is available for the mode that is being tested (file:///home/jsoohoo/home.html)		
6	Setup		#comment
7	S-1 [Verify Old Data] In PostgreSQL, ensure that test data exists in the table "OmRequest," under the schema "oms" or "oms_<MODE>." Configure the OMS GUI appropriately.		#comment Because test data is not always readily available, the following steps are forumulated to remove as little data from the pool as possible.
8	[Q-1] Query the database to verify that there exists more than 1 entry in the "OmRequest" table (query includes count of the records created on each day).	select count(requestid), to_char(date_trunc('day', finishdatetime), 'YYYY/MM/DD') from OmRequest group by to_char(date_trunc('day', finishdatetime), 'YYYY/MM/DD') order by to_char(date_trunc('day', finishdatetime), 'YYYY/MM/DD');	
9	[Q-2] Query the database to verify that the table "OmRequest" has entries with the range of "finishdatetime" field spanning more than 1 day.	select count(distinct to_char(date_trunc('day', finishdatetime), 'YYYY/MM/DD')) from OmRequest;	
10	[Q-3] Query the database to verify that you have a number that represents the interval that will remove the fewest records at a time from "OmRequest."	select now() - (select min(finishdatetime) from OmRequest) as interval;	The result of the Q-3 will be used in S-2 for configuration.
11	[Q-4] Query the database to verify that that there exists at least 1 entry in the "OmRequest" table that is not in a terminal state.	select count(requestStatus) from OmRequest where requestStatus not in ('Failed','Done','Cancelled','Expired','Shipped','Abort','Aborted','Canceled','Te	

#	Action	Expected Result	Notes
		rminated');	
12	[Q-5] Query the database to verify that the table "OmRequest" has entries with the range of "finishdatetime" field spanning more than 1 day, not including requests in a terminal state.	select count(requestStatus), to_char(date_trunc('day', finishdatetime), 'YYYY/MM/DD') from OmRequest where requestStatus not in ('Failed','Done','Cancelled','Expired','Shipped','Abort','Aborted','Canceled','Terminated') group by to_char(date_trunc('day', finishdatetime), 'YYYY/MM/DD');	
13	[Q-6] Query the database to verify that you have a number that represents the interval that will remove the fewest records at a time from "OmRequest," not including requests in a terminal state	select now() - (select min(finishdatetime) from OmRequest where requestStatus not in ('Failed','Done','Cancelled','Expired','Shipped','Abort','Aborted','Canceled','Terminated')) as interval;	The result of the Q-6 will be used in S-2 for configuration.
14	S-2 [OM Configuration]		#comment
15	In the OMS GUI, look for "OM Configuration" in the sidebar, and click on it to reveal "Cleanup Parms."	Change ARCHIVE_REQUEST_AGE to the value returned in Q-3. Change ARCHIVE_INCOMPLETE_REQUEST_AGE to the value returned in Q-6.	
16	S-3 [Cleanup Old Data - Terminal State]		#comment
17	On f5oml01v, as cmshared, run perl EcOmCleanupOldData.pl OPS	This will clean all terminal-state requests older than ARCHIVE_REQUEST_AGE.	
18	V-1 [Verify Terminal State Cleanup]		#comment
19	In the following query, replace "X" with the result from Q-3: select count(RequestId) from OmRequest where finishdatetime < (now() - (SELECT '1 day'::interval * X)) and RequestId not in (select RequestId from OmBundlingOrder where coalesce(FinishDateTime,ExpirationDate) <= ExpirationDate);	Execute; the result should be zero; any other result may indicate bad data, or a database error.	
20	S-4 [Cleanup Old Data - Force All]		#comment
21	On f5oml01v, as cmshared, run perl EcOmCleanupOldData.pl OPS 1	This will clean all requests older than ARCHIVE_INCOMPLETE_REQUEST_AGE.	

#	Action	Expected Result	Notes
22	V-2 [Verify Force All Cleanup]		#comment
23	In the following query, replace "X" with the result from Q-6: select count(RequestId) from OmRequest where finishdatetime < (now() - (SELECT '1 day'::interval * X)) and RequestId not in (select RequestId from OmBundlingOrder where coalesce(FinishDateTime,ExpirationDate) <= ExpirationDate);	Execute; the result should be zero; any other result may indicate bad data, or a database error.	
24	V-3 [Verify Logs]		#comment
25	Verify the existence of: /usr/ecs/<mode>/CUSTOM/logs/EcOmCleanupOldData.log	The text should resemble the standard output from each run, with the addition of a timestamp.	
26	Check for errors.		

TEST DATA:

EXPECTED RESULTS:

11.15 EcOmDbCleanArchive.pl (ECS-ECSTC-396)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure OMS configuration is current and correct (config files, properties files, database settings, etc.); Have cmshared access on x5oml01v.		
3	Ensure the following file exists: /usr/ecs/<MODE>/CUSTOM/dbms/postgres/OMS/scripts/EcOmDbCleanArchive.pl For reference, read: /usr/ecs/<MODE>/CUSTOM/dbms/postgres/OMS/EcOmCleanArchive.sp		
4	Ensure a PostgreSQL prompt is available and connected to the ecs database. /tools/postgres/9.2/bin/psql -U $USERNAME -h f4dbl03 -d ecs select public.set_search_path('OMS', $MODE);<br	Should list the schemas installed in the mode. Should include 'oms' for OPS or 'oms_<MODE>'.	"readonly" is an acceptable username.

#	Action	Expected Result	Notes
	/>show search_path;		
5	Setup		#comment
6	S-1 [Verify Current Data] In PostgreSQL, ensure that test data exists in the table "OmOrder_R," under the schema "oms" or "oms_<MODE>."		#comment Because test data is not always readily available, the following steps are forumulated to remove as little data from the pool as possible.
7	[Q-1] Query the database to verify that there exists more than 1 entry in the "OmOrder_R" table (query includes count of the records created on each day).	select count(orderid), to_char(date_trunc('day', timeoflastupdate), 'YYYY/MM/DD') from OmOrder_R group by to_char(date_trunc('day', timeoflastupdate), 'YYYY/MM/DD') order by to_char(date_trunc('day', timeoflastupdate), 'YYYY/MM/DD');	
8	[Q-2] Query the database to verify that the table "OmOrder_R" has entries with the range of "timeoflastupdate" field spanning more than 1 day.	select count(distinct to_char(date_trunc('day', timeoflastupdate), 'YYYY/MM/DD')) from OmOrder_R;	
9	[Q-3] Query the database to verify that you have a date that will allow the script to remove the fewest records at a time from "OmOrder_R."	select min(b.unique_day) from (select distinct to_char(date_trunc('day', timeOfLastUpdate), 'YYYY/MM/DD') as unique_day from OmOrder_R) b where unique_day not in (select min(c.unique_day) from (select distinct to_char(date_trunc('day', timeOfLastUpdate), 'YYYY/MM/DD') as unique_day from OmOrder_R) c);	The result of the Q-3 will be used in S-2.
10	S-2 [Clean Archive]		#comment
11	On f5oml01v, as cmshared, replace RETAIN_DATE with the value returned in Q-3 and execute: perl EcOmDbCleanArchive.pl OPS RETAIN_DATE	This will clean all archived requests older than RETAIN_DATE, the second paramter.	
12	V-1 [Verify Archive Cleanup]		#comment
13	In the following query, replace "X" with the result from Q-3:<br	Execute; the result should be zero; any	

#	Action	Expected Result	Notes
	/>select count(orderid) from OmOrder_R where timeoflastupdate < X;	other result may indicate bad data, or a database error.	
14	V-2 [Verify Logs]		#comment
15	Verify the existence of: /usr/ecs/<mode>/CUSTOM/logs/EcOmDbCleanArchive.log.timestamp The timestamp is formatted as follows: 'YYMMDDHHMMSS' (Two digits each of: year, month, day, hour, minute, and second; ls | grep EcOmDbCleanArchive should display all such logs.)	The text should resemble the standard output from each run exactly.	
16	Check for errors.		

TEST DATA:

EXPECTED RESULTS:

11.16 OMS Modified

11.16.1 [PVC] Test Case 4 - SCP Request for a Configured Destination (ECS-ECSTC-310)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	SCP Request for a Configured Destination		#comment
2	Submit an SCP request for multiple granules using the Spatial Subscription GUI, EWOC client, or the acquire utility.		
3	Ensure the scp destination is configured in the OMS GUI.		
4	Verify the request shows up on the OMS GUI Distribution Requests Page and the information is correct.		
5	Verify the “Request Status” column proceeds from Queued to Shipped.		
6	Verify the IsOrderOnly flag is appropriately changed from NULL to B then back to NULL for public granules in the order or from H to Y then back to H for hidden granules in the order.		
7	Verify the appropriate granule files are transferred to the destination.		

#	Action	Expected Result	Notes
8	Verify the Distribution Notice is left in the destination directory and emailed to the end user.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.2 Test Case 1 - HTTPS Request (ECS-ECSTC-307)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Https Request		#comment
2	Submit an Https request for multiple granules, using the Spatial Subscription GUI, EWOC client, or the acquire utility.		
3	Verify the request shows up on the OMS GUI Distribution Requests Page and the information is correct.		
4	Verify the “Request Status” column proceeds from Queued to Shipped.		
5	Verify the IsOrderOnly flag is appropriately changed from NULL to B then back to NULL for public granules in the order or from H to Y then back to H for hidden granules in the order.		
6	Verify the appropriate granule files are available for download from the ftp server and that other request directories are not visible.		
7	Verify DN emails are sent to the user specified in the request.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.3 Test Case 2 - FtpPush Request for a Configured Destination (ECS-ECSTC-308)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	FtpPush Request for a Configured Destination		#comment
2	Submit an FtpPush request for multiple granules, using the Spatial Subscription GUI, EWOC client, or the acquire utility.		
3	Ensure the ftp destination is configured in the OMS GUI.		
4	Verify the request shows up on the OMS GUI Distribution Requests Page and the information is correct.		
5	Verify the “Request Status” column proceeds from Queued to Shipped.		
6	Verify the IsOrderOnly flag is appropriately changed from NULL to B then back to NULL for public granules in the order or from H to Y then back to H for hidden granules in the order.		
7	Verify the appropriate granule files are transferred to the destination.		
8	Verify DN emails are sent to the user specified in the request.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.4 [OBSOLETE] Test Case 3 - FtpPush Request for an Un-Configured Destination (ECS-ECSTC-309)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	FtpPush Request for an Un-Configured Destination		#comment
2	Submit an FtpPush request for multiple granules, using the Spatial Subscription GUI, EWOC client, or the acquire utility.		
3	Ensure the ftp destination is NOT configured in the OMS GUI.		
4	Verify the request shows up on the OMS GUI Distribution Requests Page and the information is correct.		
5	Verify the “Request Status” column proceeds from Queued to Shipped.		
6	Verify the IsOrderOnly flag is appropriately changed from NULL to B then back to NULL for public granules in the order or from H to Y then back to H for hidden granules in the order.		
7	Verify the appropriate granule files are transferred to the destination.		
8	Verify DN emails are sent to the user specified in the request.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.5 Test Case 6 - Suspend, Resume / Cancel, Retry (ECS-ECSTC-312)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Suspend, Resume / Cancel, Retry		#comment
2	Submit a request for HEG Processing		Not supported. HEG processing not ported to Postgres - not tested
3	Submit another request containing Browse data.		
4	Submit another request containing Bulk Browse data.		Not supported - not tested
5	Submit another request containing Non-ECS data.		

#	Action	Expected Result	Notes
6	For each submitted request, verify that the order can be		#comment
7	Suspended and Resumed		
8	Cancelled Individually and Resubmitted (Individually)		
9	Bulk Cancelled and Bulk Resubmitted		
10	Suspended while in Queued state, resumed and shipped		
11	Suspended while in Staging state, resumed and shipped (for all except Bulk Browse)		
12	Cancelled and resubmitted		
13	Resubmitted from an Operator Intervention state and shipped without the OM server coring		
14	Resubmitted after it's shipped and ensure that it goes the correct route (i.e. S4)		
15	When resubmitting orders, ensure the sending of email options works		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.6 Test Case 9 - Ftp Push/SCP Destination Manual Retry Suspension (ECS-ECSTC-315)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ftp Push/SCP Destination Manual Retry Suspension		#comment
2	Use the OMS GUI to set the Max. Operations to 2 for a Push Destination.		
3	Submit multiple Push requests with an invalid password to this Destination.		
4	Verify that at least 2 requests enter Operator Intervention.		
5	Verify that the destination is suspended, and an operator alert is queued.		
6	Verify that the email is sent to the configured address for operator alerts, identifying the nature of the alert and the ftp destination.		

#	Action	Expected Result	Notes
7	Use the OMS GUI Destination monitor to resume the destination.		
8	Close one of the interventions, and modify the ftp push parameters, so the correct password is used.		
9	Verify the request is successfully shipped.		
10	Repeat this test, using a destination that is not configured in the OMS GUI.		
11	Repeat this test using SCP media type. Note: an SCP destination must be configured		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.7 [PVC] Test Case 15 - Checksum during Data Distribution : CK_7F_01, Criterion 30 (ECS-ECSTC-321)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Checksum during Data Distribution : CK_7F_01, Criterion 30		#comment
2	Configure all media types to checksum files that have not been checksummed in the last 1 day.		
3	Configure at least one ftp-push and at least one scp destination to disable checksum verification.		
4	Modify the checksum value of a QA, PH, and DAP granule in the AIM Inventory Database.		
5	Order granules that do not reside in the Data Pool for each media type.		OBE
6	For ftp-push and scp order at least two granules that reside in the Data Pool with last checksum times in the past for destinations that are disabled and for destinations that are enabled.		
7	In addition order three granules from the Archive for each ancillary type (QA, PH, DAP), one that has a checksum value, and one that does not, and one that		

#	Action	Expected Result	Notes
	has a modified value.		
8	Verify that the last checksum time is close to the current time for all granules that did not use the disabled push/scp destinations whose values had not been modified.		
9	Verify that an operator intervention was generated for all files whose checksum value was modified.		
10	Verify that the checksum origin in the DataPool of files coming from the Archive match the checksum origin in AIM		OBE
11	Verify that the checksum values in the DataPool of the files coming from the Archive match the checksum value in AIM.		OBE
12	Repeat test with a different minimum checksum time and verify that the last checksum time is adhered to.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

11.16.8 OMS GUI Config Parameters (ECS-ECSTC-444)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Change a configuration parameter in the OMS GUI under OM Configuration All		
2	Verify that the order of the configuration parameters don't change and that the relevant database entry is modified.		

TEST DATA:

EXPECTED RESULTS:

11.16.9 Ordering newly published replacement granules (ECS-ECSTC-496)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use a hidden granule with a versionnumber > 0.		
2	Order the granule - ftp pull		
3	Publish the granule so that the isorderonly moves from Y to B.		
4	Set the omactionqueue row for the clean up action on this request (actiontype = 'PL') to have an enqueuetime = now().		
5	Bounce OMS Servers.		
6	Verify there are no links left over in the orderdata directory for this request.		

TEST DATA:

EXPECTED RESULTS:

11.16.10 Resubmit request with failed granule(s) (ECS-ECSTC-562)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	The final decision is to reset the granule status for the granules that are not failed by operator i.e. if the granule's explanationcode in OmRequestGranule is not 126, its status should be reset. otherwise, the granule should be skipped.		#comment

#	Action	Expected Result	Notes
2	Single granule request case: start out from a successfully shipped request.resubmit it, make sure it resubmit successfully. a) Fail the granule in the single granule request b) Update the explanationcode of the failed granule to a value other than 126. Refresh, Resubmit the request.	a) Verify the explanationcode is set to 126. Try resubmit the request. It should fail because the granule should be skipped. b) Verify That granule should not be skipped.	
3	Multiple granule request case: start out from a successfully submitted request.resubmit it, make sure it resubmits successfully. a) Fail some granules in the Multiple granule request, resubmit the request. b) Resubmit the request again, update the explanationcode of some of the failed granules(not all) to a value other than 126. Refresh, submit	a) Verify the explanationcode for these granules arae set to 126 and not for the rest of the granules in the request. And after resubmit the request, only those granules that are not failed by the operator should succeed. b) Verify that the failed granules with updated explanationcode should also succeed.	

TEST DATA:

EXPECTED RESULTS:

11.16.11 Partitioning (ECS-ECSTC-636)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Under media configuration, ensure that the FtpPush configuration for ParitionGranuleLimit is set to 1		
2	Construct an order of 2 or more granules and submit as an FtpPush request		
3	After the request ships, resubmit the request via the OMS gui and select the Partition option		
4	Verify that the request is split into separate requests, 1 request per granule in		

#	Action	Expected Result	Notes
	the original request.		

TEST DATA:

EXPECTED RESULTS:

11.16.12 Request Priority (ECS-ECSTC-637)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure Order Manager Server is stopped		
2	Construct 30 requests with a significant number of granules and granule size and submit them via the acquire script, each with their priority set to Normal		
3	In the OMS gui, change the request priorities to be 6 of each: LOW, NORMAL, HIGH, VHIGH, XPRESS		
4	Verify the finsihdatetime for each request is sequentially ordered from the XPRESS priority request to the Low priority request respectively. Doesn't need to be exact as some of the requests might be processed too fast for prioritizing to take place. Increase the amount of requests and their size if results are inconclusive.		

TEST DATA:

EXPECTED RESULTS:

11.16.13 OMS GUI: Request Management (ECS-ECSTC-638)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit 2 FtpPush requests with each an incorrect password.		
2	Open the OMS gui to the Open Interventions Page and verify there is an entry on the list for the newly submitted requests. It may take a few minutes for the requests to process to the point where their status is set to Operator Intervention. Make sure the filter is set up to allow for these requests to be displayed.		
3	On the Open Interventions Page, bulk submit one of the broken FtpPush requests using the check box to submit it and make sure to update the FtpPush parameters to the correct values. This may require entering the Request detail page via the Distribution Request page for this request.		
4	On the Open Interventions Page, bulk fail one of the broken FtpPush requests using the check box to select it for failure.		
5	Verify both requests show up on the Completed Interventions page with the correct details and that the links therein are not broken.		
6	Verify both requests show up on the FtpPush/SCP Request page with the correct details and that the links therein are not broken.		
7	Ensure there are External Processors configured under OM Configuration -> External Processing page.		
8	Submit a request(any Media Type) so that it will be processed by one of the configured External Processors		
9	Verify that this request show sup on the Processing Service Request page filtering on the External Processor		
10	On the Distribution Request Page, select each shipped request submitted thus far using the check box beside the request and order id links, Bulk resubmit.		
11	Verify each request is resubmitted and ships		
12	On the Distribution Request Page, select each shipped request submitted thus far using the check box beside the request and order id links, Bulk resubmit. Once submitted, reselect these requests via the check box and Bulk cancel		
13	Verify each request is canceled.		
14	Configure a Data Access Service via OM Configuration->DataAccess Processing		
15	Submit a request such that it will be processed via the recently configured DataAccess Service		

#	Action	Expected Result	Notes
16	Verify this request ships and that the relevant processing instructions are included.		
17	Verify all requests so far show up on the Staging Request page		

TEST DATA:

EXPECTED RESULTS:

11.16.14 OMS Startup Performance (ECS-ECSTC-657)

DESCRIPTION:

When OMS Starts up it has to load a large number of requests into memory and begin processing them. In the case of FTP Pull Cleanup actions LaRC was experiencing very long wait times (2-3 hour) before OMS would ship it's first request. This test is to ensure that OMS begins shipping requests within an acceptable amount of time after start up.

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the cleanup actions will not expire while setting up this test. From OMS GUI Page, click on OM Configuration link From the Media Configuration Page set Pull Gran Dpl Time (days) [...] = 3		
2	Using the OMS scli to submit 100,000 FtpPull order requests. See the load_request.csh script below. cat load_requests.csh #!/bin/csh set i=0 while ($i < 100000) /usr/ecs/OPS/CUSTOM/utilities/acquire OPS -p /home/hdinh/acquire/ops/par_pull -f /home/hdinh/acquire/ops/geoid -t hdinh$i @ i++ end Note:geoid contains SC:MIB2GEOP.002:397655 		
3	3) After 100,000 requests have shipped, bring down OMS server.		
4	Using the OMS scli to submit two FtpPull order requests. 		

#	Action	Expected Result	Notes
	/usr/ecs/OPS/CUSTOM/utilities/acquire OPS -p /home/hdinh/acquire/ops/par_pull -f /home/hdinh/acquire/ops/geoid_1 -t tag1 /usr/ecs/OPS/CUSTOM/utilities/acquire OPS -p /home/hdinh/acquire/ops/par_pull -f /home/hdinh/acquire/ops/geoid_2 -t tag2		
5	Check to make sure the two requests in step 4 above are in Queued. SELECT DISTINCT omG.Requestid, a.ActionType as "ActType", OmG.GranType ||':'|| EsdtType ||':'|| omG.ECS_GranuleId as "GEOID", a.enqueueTime::varchar(19), r.creationdate::varchar(19), r.finishdatetime::varchar(19),r.requeststatus as "ReqStatus", a.CompletionTime::varchar(19) as "a.CompletionTime" FROM OmRequestGranule omG JOIN AmGranule g ON omG.ECS_GranuleId = g.GranuleId JOIN OmRequest r ON omG.RequestId = r.RequestId JOIN OmActionQueue a ON r.requestid = a.requestid WHERE g.granuleid in (398623,398625) ; 	requestid | ActType | GEOID | enqueuetime | creationdate | finishdatetime | ReqStatus | a.CompletionTime ----------+----------+----------+----------+----------+----------+----------+---------- 0300201544 | V | SC:MOD10C1.005:398623 | 2014-10-15 13:18:36 | 2014-10-15 13:18:36 | 2014-10-15 13:18:36 | Queued | 0300201545 | V | SC:MOD10C1.005:398625 | 2014-10-15 13:18:46 | 2014-10-15 13:18:46 | 2014-10-15 13:18:46 | Queued | (2 rows)	
6	Using sql command below to check OMS database to make sure 100,000 requests are shipped and two requests are in Queued. select r.requeststatus , count(*) from OmRequest r, OmActionQueue o where r.requestid = o.requestId and o.completiontime is null group by r.requeststatus; 	requeststatus | count ----------+-------- Shipped | 100000 Queued | 2 (2 rows)	
7	Update the OmActionQueue::EnqueueTime so that OMS will have to work on the cleanup requests while starting up. run the sql command below to update OmActionQueue:EnqueueTime update OmActionQueue set enqueuetime = now() - '2 day'::interval - '18	UPDATE 100000	

<table>
<tr><th>#</th><th>Action</th><th>Expected Result</th><th>Notes</th></tr>
<tr><td></td><td>hours'::interval

where ActionType = 'PL' and completiontime is null

and enqueueTime > '10/13/2014';
</td><td></td><td></td></tr>
<tr><td>8</td><td>Bring up the OMS server.

Note: use linux date command to mark the time to bring up the OMS server.

For example.

f5dpl01v{cmshared}(current_test)97: date</td><td>Wed Oct 15 13:45:30 EDT 2014

write down the timestamp to verify later in step 9 below.
</td><td></td></tr>
<tr><td>9</td><td>Re-run the query in step 5 above to make sure the Queued requests are promoting to ship.
</td><td>requestid | ActType | GEOID | enqueuetime | creationdate | finishdatetime | ReqStatus | a.CompletionTime

-----------------+----------+---------------------------------+---------------------------+---------------------------+---------------------------+-----------+------------------

0300201544 | PL | SC:MOD10C1.005:398623 | 2014-10-18 13:53:22 | 2014-10-15 13:18:36 | 2014-10-15 13:53:22 | Shipped |

0300201545 | PL | SC:MOD10C1.005:398625 | 2014-10-18 13:53:22 | 2014-10-15 13:18:46 | 2014-10-15 13:53:22 | Shipped |

(2 rows)

Now the OmRequest:finishdatetime is 2014-10-15 13:53:22 compare with the time stamp in step 8 above date Wed Oct 15 13:45:30 EDT 2014</td><td></td></tr>
</table>

TEST DATA:

EXPECTED RESULTS:

11.17 Test Case 23 - OMS GUI Distribution Request Filter (ECS-ECSTC-3884)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit a multiple requests via Spatial Subscription GUI, EWOC client, and the EcOmSrCliDriverStart utility. Include FtpPull, FtpPush, SCP requests.		
2	Configure the Distribution request filter for start time so that the Distribution request page only shows the requests of this test.		
3	In turn, select a single Order Source in the Distribution request page filter, verify that the correct request is returned upon submittal.		
4	Verify that under the Current Filters list for Order Source on the Distribution page, the correct filters are displayed.		
5	Verify that by selecting All, every request submitted is returned on the Distribution request page. Repeat for the Status, MediaType, and OrderType select boxes.		

TEST DATA:

EXPECTED RESULTS:

12 SMAP

12.1 SMAP Modified

12.1.1 Ingest XPath Validation (ECS-ECSTC-364)

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: 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/9.3/bin/psql -U <user> -d ecs -h f4dbl03 -p xxxx 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 (The test data could be out-of-date, always use the latest test data available.)		
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 =		

#	Action	Expected Result	Notes
	<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 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		

12.1.2 Install ISO ESDT (ECS-ECSTC-363)

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	FC	S-AIM-00100, S-

			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.		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.<br		#comment

#	Action	Expected Result	Notes
	/>Prepare a collection level XPATH file, granule level XPATH file, and an ISO-compliant series (collection) level metadata file.		
6	Use the prepared test data in /ecs/formal/ESDT/Sm		
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 /ecs/formal/ESDT/Sm/* /usr/ecs/<MODE>/CUSTOM/data/ESS		
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		

#	Action	Expected Result	Notes
	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 /ecs/formal/ESDT/Sm/*<C1_ESDT>*.desc *<C1_ESDT>*.desc diff /ecs/formal/ESDT/Sm*<C1_ESDT>*.dataset.xpath *<C1_ESDT>*.dataset.xpath diff /ecs/formal/ESDT/Sm*<C1_ESDT>*.series.xpath *<C1_ESDT>*.series.xpath diff /ecs/formal/ESDT/Sm*<C1_ESDT>*.series.xml *<C1_ESDT>*.series.xml		
26	V-3 Verify by inspection that the XPATH files and the series level metadata file were correctly associated with the collection.		#comment
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	Critccr 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

12.1.3 ISO Ingest Attribute Type Conversion (ECS-ECSTC-365)

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: 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. (The xpath files could be out-of-date, always update to the latest available.)		
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 (The test data could be out-of-date, always use the latest test data available.)		#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		

#	Action	Expected Result	Notes
	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
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/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/'		
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/gmd: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	Critccr 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.		

12.1.4 ODL metadata files for ECS Distribution (ECS-ECSTC-366)

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@f5eil01v.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.		

#	Action	Expected Result	Notes
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';		
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@f5eil01v.edn.ecs.nasa.gov		
19	Configure an OMSCLI acquire script geoid file: SC:<SHORTNAME>.<VERSIONID>:<GRANULEID>		

#	Action	Expected Result	Notes
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.
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.		

12.1.5 XML metadata files for ISO Distribution (ECS-ECSTC-367)

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		

#	Action	Expected Result	Notes
	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@f5eil01v.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 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.
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 f5eil01v > 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.		

12.1.6 QA Update Failure Verification (ECS-ECSTC-368)

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 on a granule for a collection whose metadata model type is ISO.	EC	S-AIM-00190

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Pre-Conditions		#comment
2	Ensure the test SMAP granule has been ingested into the public data pool. (Always use the latest ESDT and test data)		#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.		

#	Action	Expected Result	Notes
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>		
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&g		

#	Action	Expected Result	Notes
	t;<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): /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		

#	Action	Expected Result	Notes
	/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.		

13 SSS

13.1 Test Case 1 - Enter Subscriptions via SSS GUI (ECS-ECSTC-327)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Enter Subscriptions via SSS GUI]		#comment
2	Use the Spatial Subscription Server GUI to enter unqualified subscriptions for the following data types: Email Only: MOD03 (latest public version) Email/FtpPush: MI1B2E, MOD02HKM (latest public version) Email/FtpPull: MIL2ASAE, AST08 (latest public version)		
3	Enter qualified subscriptions for Data Pool Insert: Email/Data Pool Insert: NISE, AST_L1B (latest public version)		
4	For at least one of the above subscriptions, specify that the email notification text should include only the qualifying metadata.		
5	For at least one other of the above subscriptions, specify that the email notification text should include all metadata.		

TEST DATA:
see above

EXPECTED RESULTS:

13.2 Test Case 3 - Unqualified Subscription with Email/FtpPush (ECS-ECSTC-329)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Unqualified Subscription with Email/FtpPush]		#comment
2	Ingest an MI1B2E granule and a MOD02HKM granule.		
3	Verify that the subscription notification email was sent to the correct address for each granule.		
4	Using the Order Manager GUI and a listing of the push directory, verify that the granules were correctly distributed using FtpPush.		

TEST DATA:
current versions of MI1B2E and MOD02HKM

EXPECTED RESULTS:

13.3 Test Case 5 - Qualified Subscription for Data Pool Insert (ECS-ECSTC-331)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Qualified Subscription for Data Pool Insert]		#comment
2	Ingest a NISE granule and an AST_L1B granule that match the qualifications on the respective subscriptions placed in Test Case 1.		
3	Verify that the subscription notification was sent to the correct address for each granule.		
4	Verify that the granules were correctly inserted into the public Data Pool.		

TEST DATA:
current versions of NISE and AST_L1B

EXPECTED RESULTS:

13.4 Test Case 6 - Qualifying Metadata in Email Notification (ECS-ECSTC-332)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Qualifying Metadata in Email Notification]		#comment
2	For the subscription(s) where only qualifying metadata were requested in the email notification, verify that all qualifying metadata were present in the email text, and that only qualifying metadata were present.		
3	Verify format with SSS email format in ICD.		

TEST DATA:
see Test Case 1

EXPECTED RESULTS:

13.5 Test Case 7 - All Metadata in Email Notification (ECS-ECSTC-333)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[All Metadata in Email Notification]		#comment
2	For the subscription(s) where all metadata were requested in the email notification, verify that all metadata were present in the email text.		
3	Verify format with SSS email format in ICD.		

TEST DATA:
see Test Case 1

EXPECTED RESULTS:

13.6 Test Case 8 - Spatially Qualified Subscriptions with Spatial Search Type = GPolygon : NS_SY_01, Criterion 20 (ECS-ECSTC-334)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Spatially Qualified Subscriptions with Spatial Search Type = GPolygon : NS_SY_01, Criterion 20]		#comment
2	Using the NSBRV GUI with existing event and action driver log files, place a subscription on a valid ECS event for an ESDT whose SpatialSearchType is GPolygon.		
3	Qualify the subscription spatially.		
4	Specify an email notification action, using the default 'to' address.		
5	Specify that only the qualifying metadata attributes should be included in the email text.		
6	Insert into ECS three granules that match the event of the subscription just placed. The first granule should fall completely within the spatial qualification area. The second granule should intersect the spatial qualification area, but not fall completely within it. The third granule should not intersect the spatial qualification area.		
7	Verify the email notification is sent to the ‘to’ address for the first and second granules.		
8	Verify no email notification is sent to the 'to' address for the third granule.		
9	Verify the email notification text only includes names and values for metadata attributes associated with subscription qualifiers.		
10	Verify log entries are appended to the existing event and action log files.		
11	Verify all required information is included in the log files. (See L4 Requirements S-SSS-00840, S-SSS-00850.)		
12	S-SSS-00840 The NSBRV CI shall log the following information: a. Receipt of event notification from SDSRV b. Start of event processing c. Completion of event		#comment

#	Action	Expected Result	Notes
	processing d. Start of action processing e. Completion of action processing f. Completion of metadata extraction		
13	S-SSS-00850 The NSBRV CI shall include the following information in each log entry, as applicable: a. type of entry b. date and time when logged (at least to the millisecond) c. subscription id d. action Id e. Event Id f. Userid associated with the subscription g. Type of error and error details (for logged errors)		#comment

TEST DATA:
Any current ASDC data type whose Spatial Search Type is GPolygon (e.g., MOD02HKM, MOD03)

EXPECTED RESULTS:

13.7 Test Case 9 - Spatially Qualified Subscriptions with Spatial Search Type = Bounding Rectangle : NS_SY_01 criterion 30 (ECS-ECSTC-335)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Spatially Qualified Subscriptions with Spatial Search Type = Bounding Rectangle : NS_SY_01 criterion 30]		#comment
2	Using the NSBRV GUI with existing event and action driver log files, place a subscription on a valid ECS event for an ESDT whose SpatialSearchType is BoundingRectangle.		
3	Qualify the subscription spatially.		
4	Specify an email notification action, using the default 'to' address.		
5	Insert three granules into ECS which match the event of the subscription just placed. The first granule should fall completely within the spatial qualification area. The second granule should intersect the spatial qualification area, but not fall completely within it. The third granule should not intersect the spatial qualification area		

#	Action	Expected Result	Notes
6	Verify the email notification is sent to the ‘to’ address for the first and second granules.		
7	Verify no email notification is sent to the 'to' address for the third granule.		
8	Verify log entries are appended to the existing event and action log files.		
9	Verify all required information is included in the log files. (See L4 Requirements S-SSS-00840, S-SSS-00850.)		
10	S-SSS-00840 The NSBRV CI shall log the following information: a. Receipt of event notification from SDSRV b. Start of event processing c. Completion of event processing d. Start of action processing e. Completion of action processing f. Completion of metadata extraction		#comment
11	S-SSS-00850 The NSBRV CI shall include the following information in each log entry, as applicable: a. type of entry b. date and time when logged (at least to the millisecond) c. subscription id d. action Id e. Event Id f. Userid associated with the subscription g. Type of error and error details (for logged errors)		#comment

TEST DATA:
Any current ASDC data type whose Spatial Search Type is Bounding Rectangle (e.g., MIL3SAE, MIL3SAL)

EXPECTED RESULTS:

13.8 Test Case 10 - Spatially Qualified Subscriptions with Spatial Search Type = Orbit : NS_SY_01, criterion 40 (ECS-ECSTC-336)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Spatially Qualified Subscriptions with Spatial Search Type = Orbit : NS_SY_01, criterion 40]		#comment
2	Using the NSBRV GUI with existing event and action driver log files, place a		

#	Action	Expected Result	Notes
	subscription on a valid ECS event for a MISR ESDT whose SpatialSearchType is Orbit.		
3	Qualify the subscription spatially.		
4	Specify an email notification action, using the default 'to' address.		
5	Insert three granules into ECS which match the event of the subscription just placed. The first granule should fall completely within the spatial qualification area. The second granule should intersect the spatial qualification area, but not fall completely within it. The third granule should not intersect the spatial qualification area.		
6	Verify the email notification is sent to the ‘to’ address for the first and second granules.		
7	Verify no email notification is sent to the 'to' address for the third granule.		
8	Verify metadata values for Path, StartBlock and EndBlock are listed in the email notification text.		
9	Verify log entries are appended to the existing event and action log files.		
10	Verify all required information is included in the log files. (See L4 Requirements S-SSS-00840, S-SSS-00850.)		
11	S-SSS-00840 The NSBRV CI shall log the following information: a. Receipt of event notification from SDSRV b. Start of event processing c. Completion of event processing d. Start of action processing e. Completion of action processing f. Completion of metadata extraction		#comment
12	S-SSS-00850 The NSBRV CI shall include the following information in each log entry, as applicable: a. type of entry b. date and time when logged (at least to the millisecond) c. subscription id d. action Id e. Event Id f. Userid associated with the subscription g. Type of error and error details (for logged errors)		#comment

TEST DATA:

Any current ASDC data type whose Spatial Search Type is Orbit (e.g., MIL2ASAE or MIL2ASAF data types).

EXPECTED RESULTS:

13.9 Test Case 11 - Combinations of Qualifiers : NS_SY_01, criterion 90 (ECS-ECSTC-337)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Combinations of Qualifiers : NS_SY_01, criterion 90]		#comment
2	Using the NSBRV GUI, place a subscription on a valid ECS event.		
3	Qualify the subscription with 5 string and 5 range qualifiers. Of the 10 qualifiers, at least one should be a core metadata attribute, at least one should be a PSA, and at least one should be a measured parameter. Of the 5 range qualifiers, at least one should be of type integer, at least one of type float, and at least one of type datetime.		
4	Associate an email action and an ftppush distribution action with the subscription.		
5	Verify the operator is not permitted to qualify the subscription using PSAs not supported by the ESDT that is the target of the subscription.		
6	Verify the operator is not permitted to qualify the subscription using measured parameters not supported by the ESDT that is the target of the subscription.		
7	Insert a granule into ECS which matches all of the qualifications for the subscription just placed.		
8	Insert a second granule into ECS which matches some but not all of the qualifications.		
9	Verify the email notification is sent to the 'to' address for the first granule.		
10	Verify the ftppush action is processed to completion for the first granule.		
11	Verify no actions are processed for the second granule.		
12	Verify all required information is included in the log files.		

TEST DATA:

Any current data type which supports the number of qualifiers required in the test

EXPECTED RESULTS:

13.10 Test Case 12 - Placing Bundling Order Subscription : OD_S3_03, criterion 10 (ECS-ECSTC-338)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Placing Bundling Order Subscription : OD_S3_03, criterion 10]		#comment
2	Use the NSBRV GUI to create a bundling order for each type of media.		
3	Verify all optional information can be entered or omitted.		
4	Verify if optional information is omitted, the correct defaults are provided where requirements specify such defaults.		
5	Verify all required information must be entered.		
6	Verify the bundling orders are stored in the database.		
7	Verify each bundling order receives a unique identification that is displayed to the operator		
8	Verify MSS order tracking information is created for the bundling order and the order source and status are set correctly		
9	Verify it is possible to enter subscriptions and pick a bundling order as their distribution action.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

13.11 Test Case 13 - Executing Bundling Order Subscription : OD_S3_03, criterion 110, steps a, b, c, i, and m (ECS-ECSTC-339)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Executing Bundling Order Subscription : OD_S3_03, criterion 110, steps a, b, c, i, and m]		#comment
2	Ensure that the database contains a bundling order for each type of media, as well as several (at least two) subscriptions for each.		
3	Trigger events for which the subscriptions qualify, such that for each subscription, at least two distribution actions are submitted to the Order Management Service, and such that for each bundling order, at least two bundles are completed and one more is started.		
4	Ensure that at least two of the subscriptions for some bundling order overlap, i.e., both subscriptions qualify for some of the triggered events.		
5	Define the bundling orders such that some bundles complete because of the imposed granule limit and others complete because of the size limit.		
6	Ensure that several of the triggered events correspond to the subscriptions that do not reference a bundling order.		
7	Verify the correct distribution actions are submitted to Order Management.		
8	Verify as they are processed by Order Management, the granules are added to a bundle for the associated bundling order.		
9	Verify even if multiple subscriptions for a single bundling order qualify for some granule, the granule is added to the bundle corresponding to that order only once.		
10	Verify the expected number of data distribution requests is generated as bundles complete.		
11	Verify actions triggered for subscriptions that are not bundled lead to unbundled data distribution requests and are accompanied by the correct MSS order tracking information (including correct order source).		

TEST DATA:
Any current data type

EXPECTED RESULTS:

13.12 Test Case 14 - Subscription Expiration : NS_SY_01 criterion 210 (ECS-ECSTC-340)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Subscription Expiration : NS_SY_01 criterion 210]		#comment
2	Use the NSBRV GUI to place an unqualified subscription with an expiration date of tomorrow and an email notification action.		
3	Insert a granule into ECS that qualifies for the subscription just placed.		
4	Verify the email notification is sent to the 'to' address.		
5	Verify all required information is included in the log files.		
6	Allow 1 day to elapse.		
7	Insert into ECS a granule that qualifies for the subscription placed the previous day (in step 1).		
8	Verify the email notification action is not executed.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

13.13 Test Case 15 - XDAAC (ECS-ECSTC-341)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[XDAAC] is a process of ingesting into one test mode and distributing to another mode. An email is sent and parsed into a PDR. The data is pushed via FTP to a directory on a particular host.		#comment

#	Action	Expected Result	Notes
2	Note: The host names below are examples. Use appropriate names during the test.		#comment
3	Note: Ensure directories are configured according to these examples: IngestPollingDirectory = /usr/ecs/Mode/CUSTOM/data/INS/local/IngestPollingDirectory/ EmailDirectory = /usr/ecs/Mode/CUSTOM/data/INS/local/InEmailGWServerPollingDirectory/ FailedDirectory =/usr/ecs/Mode/CUSTOM/data/INS/local/InEmailGWServerFailedDirectory/ EcInEmailGWServer.CFG.rgy		#comment
4	Use the SSS GUI to add a DAAC_2_DAAC subscription: User=PVC_Mode? Status=Inactive Start Date=Apr 19 2004 12:00PM (Example) Expiration Date=Apr 19 2005 12:00PM (Example) Short Name=GDAS0ZFH Version=001 Event Type=INSERT -- Acquire Data associated with Subscription -- User Profile=PVC_Mode? User String=DAAC_To_DAAC Ingest Checkout for Mode? Priority=NORMAL Notify Type=MAIL Email Address=EcInEmailGWServer_Mode?@Host?.EDF.ecs.nasa.gov Media Format=FILEFORMAT Media Type=FtpPush FTP Information User=cmshared Password=******* Host= p0icg01.pvc.ecs.nasa.gov (example) Directory=/usr/ecs/TS2/CUSTOM/icl/p0icg01/data/ins_push		
5	Configure DDIST polling PollingDirectory = /usr/ecs/TS2/CUSTOM/data/INS/local/IngestPollingDirectory HostName = p2ins02 (example)		
6	Ingest GDAS data Copy Host1:/usr/ecs/Mode/CUSTOM/icl/p0icg01/data/reg_data/edc/noaa/gdas.PDR to p0acg05:/usr/ecs/$md/CUSTOM/data/INS/pollGSFC-V0		
7	Check ftp push data p0icg01:/usr/ecs/TS2/CUSTOM/icl/p0icg01/data/ins_push		
8	Ensure a new PDR is created in p2ins02:/usr/ecs/Mode/CUSTOM/data/INS/local/IngestPollingDirectory		

TEST DATA:

EXPECTED RESULTS:

13.14 SSS Modified

13.14.1 Test Case 2 - Unqualified Subscription with Email (ECS-ECSTC-328)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Unqualified Subscription with Email]		#comment
2	Ingest a MOD03 granule.		Jon Pals said the "ECS systems no longer archive MOD03 data. So, the SSS regression test needs to be modified to use data that ECS systems do have. Why don't you substitute a MOD10A1.005 granule for the MOD03 granule?"
3	Verify that the subscription notification email is sent to the correct address for MOD03.		

TEST DATA:
current version of MOD03

EXPECTED RESULTS:

13.14.2 Test Case 4 - Unqualified Subscription with Email/FtpPull (ECS-ECSTC-330)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Unqualified Subscription with Email/FtpPull]		#comment
2	Ingest a MIL2ASAE granule and a AST08 granule.		Jon Pals said " we don't have any AST08 granules. They are created upon demand now. " That TAST08 ESDT is an old test ESDT. We don't use it any more as far as I know. (11:40:19 AM) Mr. Jon: Yes, you could use an AST_L1B granule instead. Yes, the regression test needs to be updated.
3	Verify that the subscription notification was sent to the correct address for each granule.		
4	Using the Order Manager GUI and the Pulldir listing, verify that the granules were correctly distributed using FtpPull.		

TEST DATA:
current versions of MIL2ASAE and AST08

EXPECTED RESULTS:

13.14.3 Validate SSS GUI (ECS-ECSTC-446)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Enter upto 50 characters in the userid text box when creating a new subscription		
2	Verify that the GUI allows entry of 50 characters and a new subscription can		

#	Action	Expected Result	Notes
	be created successfully		

TEST DATA:

EXPECTED RESULTS:

13.14.4 Deleting a failed subscription in SSS GUI (ECS-ECSTC-447)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Create a regular SSS insert subscription		
2	Shutdown the Spatial Drivers: EcNbActionDriver , EcNbDeleteRequestDriver , EcNbRecoverDriver , EcNbSubscribedEventDriver		
3	Ingest a granule to trigger the subscription event		
4	Cause the subscription to fail : update EcNbActionQueue set deleterequest = 'Y';		
5	Check the GUI to make sure the failed subscription is displayed on the List Failed Actions tab		
6	Use the SSS gui to remove the failed subscription		
7	Verify that there are no errors displayed on the GUI and verify there are no errors in the GUI log file		

TEST DATA:

EXPECTED RESULTS:

14 DUPLICATE GRANULES

14.1 Duplicate Granule Detection and Replacement - Identical LocalGranuleIDs (ECS-ECSTC-474)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	#120 S-1 *[Duplicate Granule Detection and Replacement - Identical LocalGranuleIDs] Identify or configure two collections to use rule "a." (Identical LocalGranuleIDs) of the duplicate granule rules from requirement S-DPL-32022. Configure these collections to publish granules upon ingest. Note: It is acceptable to use the Ingest Processing Service's "LAB_TEST" setting to prevent the data files from being renamed to match their LocalGranuleIDs . +MOP03N.005 and g3acld.003 are identified to have DuplicateGranRuleNo = 2 in the AmCollection table. This agrees with the duplicate granule rule's ESDT spreadsheet. Make sure the following 2 queries return 2, if not, set them to Identical LocalGranuleIDs rule in the Ingest GUI. Select DuplicateGranRuleNo From AmCollection Where ShortName = "MOP03N" And VersionId = 5 Select DuplicateGranRuleNo From AmCollection Where ShortName = "g3acld" And VersionId = 3 Treat MOP03N.005 as C1, g3acld.003 as C2. Log into Ingest GUI. Click on Configuration/Datatypes, Select MOP03N.005 and g3acld.003, choose YES for "Publish in public DPL", click on "Apply Changes" Make sure the GUI shows "Public In Data Pool" for both C1 and C2. Set the labtest = Y in the EcDlInProcessingService.CFG. Bounce Ingest and Dpad		

#	Action	Expected Result	Notes
2	#120 S-2 *For each collection, ingest two granules with different LocalGranuleIDs but the same acquisition date. +Ingest 2 granules C1G1, C1G2, using the 2 PDRs in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_2/ MOP03N.005. Ingest 2 granules C2G1, C2G2, using the 2 PDRs in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_2/ g3acld.003. Continue to verification step V-1		
3	#120 S-3 *For each collection, ingest one granule with the same LocalGranuleID as a granule from step S-2 but with a different file name. These granules should replace the corresponding granules from step S-2. +Ingest 1 granule C1G3, using the PDR in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_3/ MOP03N.005. Ingest 1 granule C2G3, using the PDR in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_3/ g3acld.003. C1G3 should replace C1G1 (the one with the same LocalGranuleID as C1G3) C2G3 should replace C2G1 (the one with the same LocalGranuleID as C2G3) Continue to verification step V-2		
4	#120 S-4 *For each collection, ingest one granule with the same LocalGranuleID as a granule from step S-3 but with an earlier ProductionDateTime value and a different file name. These granules should be considered duplicates and not replace the corresponding granules from step S-3. +Ingest 1 granule C1G4, using the PDR in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_4/ MOP03N.005. Ingest 1 granule C2G4, using the PDR in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_4/ g3acld.003. C1G4 should be the duplicate for C1G3 C2G4 should be the duplicate for C2G3 Continue to verification step V-3		
5	#120 S-5 *For each collection, ingest one granule with the same LocalGranuleID and same file name as a granule from step S-3. These granules should replace the corresponding granules from step S-3. +Ingest 1 granule C1G5, using the PDR in /sotestdata/DROP_801/DP_81_04/Criteria/120/120_5/ MOP03N.005. Ingest 1 granule C2G5, using the PDR in		

#	Action	Expected Result	Notes
	/sotestdata/DROP_801/DP_81_04/Criteria/120/120_5/ g3acld.003. C1G5 should replace C1G3 C2G5 should replace C2G3 due to more recent RegistrationTime Note, since they have the same file names, and the file collision rule is always applied first, the duplication detection rule number in AmGranuleReplacement table should be 1, instead of 2 (Identical LocalGranuleIDs) Continue to verification step V-6		
6	#120 V-1 *Verify that the granules from step S-2 were successfully ingested and published. +Log into Ingest GUI; Click on Request Status; Verify that the 4 requests containing the 4 granules C1G1, C1G2, C2G1 and C2G2 respectively have successful status. Select IsOrderOnly, RegistrationTime, GranuleId From AmGranule Where GranuleId in (C1G1, C1G2, C2G1, C2G2) Verify that the RegistrationTime for all 4 granules are not NULL. Verify that IsOrderOnly is NULL for all 4 granules. Continue to setup step S-3		
7	#120 V-2 *Verify that the granules from step S-3 were successfully ingested and published and that the granules from step S-2 that they replaced are successfully unpublished. +Log into Ingest GUI; Click on Request Status; Verify that the 2 requests containing the 2 granules C1G3 and C2G3 respectively have successful status. Select IsOrderOnly, RegistrationTime, GranuleId From AmGranule Where GranuleId in (C1G1, C1G3, C2G1, C2G3) Verify that the RegistrationTime for C1G3 and C2G3 are not NULL. Verify that IsOrderOnly is NULL for C1G3, C2G3 and "H" for C1G1 and C2G1. Continue to verification step V-4		
8	#120 V-3 *Verify that the granules from step S-4 were successfully ingested but not published and that the corresponding granules from step S-3 were not		

#	Action	Expected Result	Notes
	unpublished. +Log into Ingest GUI; Click on Request Status; Verify that the 2 requests containing the 2 granules C1G4 and C2G4 respectively have successful status. Select IsOrderOnly, RegistrationTime, GranuleId From AmGranule Where GranuleId in (C1G3, C1G4, C2G3, C2G4) Verify that the RegistrationTime for C1G4 and C2G4 are not NULL. Verify that IsOrderOnly is NULL for C1G3, C2G3 and "H" for C1G4 and C2G4. Continue to verification step V-5		
9	#120 V-4 *Verify that the replacement granules from step S-3 and the granules that they replaced are recorded in the AIM database along with the rule (identical LocalGranuleIDs) used to detect the granule duplication. +Verify that C1G3 replaces C1G1 and C2G3 replaces C2G1 and they are recorded in AmGranuleReplacment table as: Rep Dup RuleNo C1G3 C1G1 2 C2G3 C2G1 2 Continue to setup step S-4		
10	#120 V-5 *Verify that the duplicate granules from step S-4 are recorded in the AIM database along with the granules which they duplicate and the rule (identical LocalGranuleIDs) used to detect the granule duplication. +Verify that C1G4 is the duplicate of C1G3, C2G4 is the duplicate of C2G3 and they are recorded in the AmGranuleReplacement table as: Rep Dup RuleNo C1G3 C1G4 2 C2G3 C2G4 2 Continue to setup step S-5		
11	#120 V-6 *Verify that the granules from step S-5 were successfully ingested and published and that the granules from step S-3 that they replaced were successfully unpublished. +Log into Ingest GUI; Click on Request Status; Verify that the 2 requests containing the 2 granules C1G5 and C2G5 respectively have successful status.		

#	Action	Expected Result	Notes
	Select IsOrderOnly, RegistrationTime, GranuleId From AmGranule Where GranuleId in (C1G3, C1G5, C2G3, C2G5) Verify that the RegistrationTime for C1G5 and C2G5 are not NULL. Verify that IsOrderOnly is NULL for C1G5, C2G5 and "H" for C1G3 and C2G3		
12	#120 V-7 *Verify that the replacement granules from step S-5 and the granules that they replaced are recorded in the AIM database along with the rule (identical LocalGranuleIDs) used to detect the granule duplication. +Verify that C1G5 replaces C1G3, C2G5 replaces C2G3 and they are recorded in the AmGranuleReplacement table as: Rep Dup RuleNo C1G5 C1G3 2 C2G5 C2G3 2 Note, since they have the same file names, and the file collision rule is always applied first, the duplication detection rule number in AmGranuleReplacement table should be 1(filename collision rule), instead of 2 (Identical LocalGranuleIDs rule)		
13			

TEST DATA:

EXPECTED RESULTS:

15 DBDM

15.1 pgcopy (ECS-ECSTC-500)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Create a wrapper script using your favorite shell which will call pgcopy.		

#	Action	Expected Result	Notes
2	Configure the wrapper script to invoke an error by selecting data from a non-existent database table.		
3	Run wrapper script.		

TEST DATA:

EXPECTED RESULTS:

16 DPL

16.1 Test Case 3 - Publish with Theme (ECS-ECSTC-204)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Publish with Theme]		#comment
2	Ingest ICEBRIDGE data. First verify that the ThemeID value is not in the themepsaname column of AmCollection table for the collection so that the theme association would not happen upon ingest.		For example IRPAR2.001
3	Run the Publish Utility. Use the –theme option to associate these granules with an existing theme. (Use the Data Pool Maintenance GUI Manage Themes tab to see the list of existing themes.)		
4	Verify that the granules were successfully inserted into the Data Pool by checking the status of the insert actions in the Data Pool database (DlInsertActionQueue).		
5	Verify that the granules were inserted into the AmGranule table.		
6	Verify, using Unix cd and ls commands, that the files for the granules were inserted into the appropriate Data Pool directories. select fs.absoluteFileSystemPath + c.GroupId + df.DirectoryPath + df.OnlineFileName from DlFileSystems fs join AmCollection c on fs.fileSystemLabel = c.FileSystemLabel join AmGranule		

#	Action	Expected Result	Notes
	g on c.CollectionId = g.CollectionId join AmDataFile df on g.GranuleId = df.GranuleId where g.GranuleId in (<GranuleIds>)		
7	Verify that the granules are associated with the specified theme, i.e., that appropriate rows for the granules have been inserted in the DlGranuleThemeXref table.		

TEST DATA:
Any nonECS data type

EXPECTED RESULTS:

16.2 Test Case 4 - [Data Pool Maintenance GUI – Check Batch Insert (ECS-ECSTC-3883)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Use the Data Pool Maintenance GUI to check the status of the batch insert using the Batch Summary tab and also using the List Insert Queue tab (filter by batch label).		
2	Verify that the GUI correctly reports the status in the DlInsertActionQueue.		

TEST DATA:

EXPECTED RESULTS:

16.3 [OBSOLETE] - Test Case 8 - Update Granule Utility OBSOLETE ("4.8.5 Update Granule Deleted. Not applicable for Release 7.23.") (ECS-ECSTC-206)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Update Granule Utility]		#comment
2	Use the Update Granule utility (EcDlUpdateGranule.pl) to update the retention priority and expiration date of nonECS granules.		
3	Verify using isql that the retention priority and expiration date for the granules was updated properly in the DlGranuleExpirationPriority table.		

TEST DATA:
nonECS data types

EXPECTED RESULTS:

16.4 [OBSOLETE - PVC] - Test Case 11 - Firewall FTP Access Statistics Utility (ECS-ECSTC-207)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Firewall FTP Access Statistics Utility]		#comment
2	Use the Firewall FTP Access Statistics Utility (EcDlRollupWuFtpLogs.pl) to collect ftp access statistics for the time period during which Test Case 6 was executed.		
3	Verify that the information regarding the download of the file was collected from the Data Pool firewall ftp log (/var/log/xferlog) and stored in the Data Pool database (DlGranuleAccess table).		

TEST DATA:
see Test Case 6

EXPECTED RESULTS:

16.5 [PARTIAL OBSOLETE] -Test Case 13 - Inventory Validation (Orphan, Phantom and Link Checking) - [PVC] (ECS-ECSTC-208)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	This criterion was derived from criterion 240 in ticket DP_7F_01.		#comment
2	S-1 Perform orphan, phantom and link checking on a Data Pool that includes at least two of each of the following cases: a. Orphan ECS XML files, both public and non-public b. Orphan ECS granule files, both public and non-public c. Public orphan granule and metadata files with hidden links d. DELETED e. Orphan browse granules files in the Data Pool unreferenced by a science granule where no entry for the Browse exists in the AIM Inventory Catalog. (Include both orphan MISR and MODIS browse granules.) f. DELETED g. DELETED h. Phantom science granules without any files in the online archive, both public and non-public i. Phantom science granules with science files but lacking XML files in the online archive, both public and non-public j. Phantom science granules with XML files but lacking science files, both public and non-public k. Phantom browse granules referenced by public science granules but lacking their browse files in the Online Archive l. Invalid browse links that do not point to an existing browse file m. Missing browse links (i.e., browse links that are implied by browse cross-references but are not present on disk) n. Invalid hidden links that do not point to an existing public file o. Orphaned files whose age is less than the maximum orphan age p. Public Science Granules with associated Public QA granules where the links to the QA granules are missing in the Data Pool. q. Public Science Granules with associated Public PH granules where the links to the PH granules are missing in the Data Pool. r. Public Science Granules with associated Public QA granules where the links to the QA granules exist but the underlying QA granule files are missing from the Data Pool / Online Archive. s. Public Science Granules with associated Public PH granules where the links		#comment

#	Action	Expected Result	Notes
	to the PH granules exist but the underlying PH granule files are missing. t. Public granules in hidden directories u. Hidden granules in a public directory Perform the test while there are concurrent Ingest and Distribution activities, with at least twenty granules ingested and twenty granules being newly ordered and distributed during the test.		
3	Execute EcDlCleanupFilesOnDisk.pl as mentioned in S-4 and save off the output files and log files. This will be used later to verify V-5.		
4	To create orphans, use touch –d ’yymmdd hh:mm’ <file> to alter the file modification time to make the file appear older/younger than the orphan age limit.		#comment
5	For the following data setup record the granule/browse/ecs ids, shortname, version id, DPL file system, DPL group, file path, and filename for each granule.		#comment
6	a. Ingest the granules in test data S1a. Remove the XML file entry in AmDataFile using SQL. Perform the touch command on the science and XML files in the file system to be at least 7 days old.		
7	b. Ingest the granules in test data S1b. Remove the science granule entry in AmDataFile using SQL. Perform the touch command on the science and xml files in the file system to be at least 7 days old.		
8	c. Ingest the granules in test data S1c. Place an order for the granules to create the hidden links in the file system. Remove the science granule and XML file entries in AmDataFile using SQL. Perform the touch command on the science and XML files in the file system to be at least 7 days old.		
9	e. Ingest the granules in test data S1e. Remove the association between the science and browse granules from AmGranuleBrowseXref table using SQL.		
10	f. DELETED		#comment
11	g. DELETED.		#comment
12	h. Ingest the granules in test data S1h. Remove the science and XML file for these granules from the file system.		
13	i. Ingest the granules in test data S1i. Remove the XML file for these granules from the file system.		
14	j. Ingest the granules in test data S1j. Remove the science file for these granules from the file system.		
15	k. Ingest the granules in test data S1k. Record the browse granules dpl ids and filename. Remove the browse file for these granules from the		

#	Action	Expected Result	Notes
	file system.		
16	l. Ingest the granules in test data S1l. Move the file that the browse links are pointing at to another name causing the browse links to point to something invalid.		
17	m. Ingest the granules in test data S1m. Remove the browse link in the file system.		
18	n. Ingest the granules in test data S1n. Place an order for these granules so hidden links are created. Move the file that the hidden links are pointing at to another name causing the hidden links to point to nothing.		
19	o. Ingest the granules in test data S1o. Remove the granule's entries in AmDataFile. Perform the touch command on the browse files in the file system to be younger than 3 days.		
20	p. Ingest the granules in test data S1p and but then delete the associated QA granules files.		
21	q. Ingest the granules in test data S1q with associated PH granules. Use SQL to delete the links to the PH files from the file system.		
22	r. Ingest the granules in test data S1r with associated QA granules. Delete the associated QA granule files.		
23	S. Ingest the granules in test data S1s with associated PH granules. Delete the PH granules files.		
24	t. Ingest granules in test data S1t. Change the IsOrdered attribute in AmGranule table to H.		
25	u. Locate a hidden granule in the AmGranule table and record its GranuleId. Using the GranuleId and AmDataFile locate the granule path. Change the granule path to the complementary public location.		
26	S-2 Use a maximum orphan age of seven days.		#comment
27	Use the –maxFileAge 7, see S-4.		
28	S-3 Specify the location of the output report file(s).		#comment
29	Use the –outputDir 240, see S-4		
30	S-4 Perform the test while there are concurrent Ingest and Distribution activities, with at least twenty granules ingested and twenty granules being newly ordered and distributed during the test.		#comment
31	After the data setup is complete, ingest and prepare an order (but do not yet submit) for 20 granules. Also prepare to ingest 20 more granules. In quick succession, initiate: Place an order for 20 granules from OMS Ingest 20 granules Run EcDlCleanupFilesOnDisk.pl <mode> -maxFileAge 7 –outputDir		

#	Action	Expected Result	Notes
	240 Run EcDlLinkCheck.ksh /datapool/<mode>/user/ /usr/ecs/<mode>/CUSTOM/data/DPL/Validation/240 /brokenLinks		
32	V-1 Verify that the validation logs the orphans in S-1a through S-1f, that all of the orphans have an age >= the specified maximum orphan age, that the nature of the orphan is correctly identified, and that the pathname of the orphan is included.		#comment
33	vi the EcDlCleanupFilesOnDisk.log and verify that the orphans generated in S-1a though S-1g are logged.		
34	V-2 Verify that the validation logs the phantoms in S-1h through S-1k and that the nature of the phantom is correctly identified.		#comment
35	vi the EcDlCleanupFilesOnDisk.log and verify that the phantoms generated in S-1h through S-1k are logged.		
36	V-3 Verify that the missing browse links referenced in S-1m are correctly identified and the missing link information (including the Pathname) is logged.		#comment
37	vi the EcDlCleanupFilesOnDisk.log and verify that the missing browse links generated in S-1m are logged.		
38	V-4 Verify that the validation logs the invalid links in S-1l, S-1n, and S-1p through S-1 s and that the nature of the incorrect link is correctly identified, and that the pathname of the link is included.		#comment
39	vi the brokenLinks file and verify that the invalid links generated in S1l and S1-n are logged.		
40	V-5 Verify that granules located in the wrong directory (S-1t and S-1u) are logged.		#comment
41	vi EcDlCleanupFilesOnDisk.log and verify that the granules in the wrong directory (S-1t and S-1u) are logged.		
42	V-6 Verify that the granules that are missing the order links (S-1v) are logged.		#comment
43	vi EcDlCleanupFilesOnDisk.log and verify that the the granules that are missing the order links (S-1v) are logged.		
44	V-7 DELETED		#comment
45	V-8 Verify that the orphans, phantoms and incorrect links were not removed.		#comment
46	For orphans, go to the file system and verify that the files exist. For phantoms, go to the file system and verify that the files are not there but the entries remain in the database. For broken links, go to the file system and verify the links are indeed broken or doesn’t exist.		

#	Action	Expected Result	Notes
47	V-9 Verify that the log includes no orphans, phantoms, or link errors other than those specified in V-1 through V-3. diff the outputs from the previous run to the current run and verify that the output files are the same.		#comment
48	V-10 Verify that the inventory validation creates output files reporting the discrepancies in the specified location.		#comment
49	Verify that an output report was generated in the /usr/ecs/${MODE}/CUSTOM/data/DPL/Validation/240 directory from the orphan and phantom checker and that the brokenLinks file exists for the link checker.		
50	V-11 Verify that the output report(s) include all the discrepancies that were logged and no others.		#comment
51	Open the report indicated in V-7 and verify that the report generated contains the orphans, phantoms, and broken links that were logged and no other.		
52	V-12 Verify that the validation exits with an exit code indicating that errors occurred.		#comment
53	Verify that the two utilities exits with an exit code of 2 indicating that orphan, phantom, and broken links were found.		

TEST DATA:

Crit		**sub**	**#**	**Public**	**ESDT**	**Browse**	**Ordered**	**Notes**	**Detect**	**Repair**[N1]
110	S1	a	2	Y	MOD29P1D			DPL Orphan XML files	OP Chk	Publish
110	S1	a	2	N	MYD29P1N			DPL Orphan XML files	OP Chk	Publish
110	S1	b	2	Y	MOD29P1D			DPL Orphan granule files (include MODIS)	OP Chk	Publish
110	S1	b	2	N	MYD29P1N			DPL Orphan granule files	OP Chk	Publish
110	S1	c	2	Y	MOD29P1D		Y	DPL Orphan granule & metadata w/ hidden links	OP Chk	Publish
110	S1	e	2 2 2	Y	MOD29P1D Browse MB2LME MISBR	Y		Orphan browse granules (XRef entry missing, AmCollection entries OK)	OP Chk	Publish

					2 browse (MISBR)					
110	S1	e	2	Y	MB2LME	Y		Orphan browse granules (XRef entry missing, AmCollection entries OK)	OP Chk	Publish
110	S1	f	2 2	Y	MOD29P1D Browse 2 shared linkage	shared		Remove the link between the public granule and the browse. Orphan browse referenced by the non-public granule but not the public granule.	OP Chk	Publish
			2						OP Chk	Publish
110	S1	f	2	Y	MB2LME	shared		Remove the link between the public granule and the MISBR. Orphan browse referenced by the non-public granule but not the public granule.	OP Chk	Publish
110	S1	f	2	N	MISBR				OP Chk	Publish
					2 browse					
					MISBR 2 browse(MISBR)					
110	S1	h	2	Y	MOD29P1D	N		Phantom science granules (all files missing)	OP Chk	Publish
110	S1	h	2	N	MYD29P1N	N		Phantom science granules (all files missing)	OP Chk	Publish
110	S1	i	2	Y	MOD29P1D	N		Phantom: XML file missing (Science file present)	OP Chk	Publish
110	S1	i	2	N	MYD29P1N	N		Phantom: XML file missing (Science file present)	OP Chk	Publish
110	S1	j	2	Y	MOD29P1D	N		Phantom: Science file missing (XML file present)	OP Chk	Publish
110	S1	j	2	N	MYD29P1N	N		Phantom: Science file missing (XML file present)	OP Chk	Publish
110	S1	k	2	Y	MOD29P1D Browse	Y		Phantom: Browse file missing (Science file present)	OP Chk	Publish
110	S1	l	2	Y	MOD29P1D Browse	Y		Link to browse file modified to point to a non-existent file	Lnk Chk	Publish
110	S1	m	2	Y	MYD29P1D	Y		Browse linkage file missing	Lnk Chk	Publish
110	S1	n	2	Y	MOD29P1D	N	Y	Modify the linkage (to the public file) in the .orderdata to be invalid	Lnk	Publish

									Chk	
110	S1	o	2	Y	MOD29P1D	N		Create an orphan granule w/ files LESS than the minimum age	n/a	n/r
110	S1	p	2 2	Y	MOD29P1D QA	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_P		
110	S1	q	2	Y	MOD29P1D PH	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_Q		
110	S1	r	2 2	Y	MOD29P1D QA	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_R		
110	S1	s	2 2	y	MOD29P1D PH	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_S		
110	S1	t	2	y	MOD29P1D	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_T		
110	S1	u	2	n	MOD29P1D	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_U		
110	S1	v	2	y	MOD29P1D	n/r	Y	/sotestdata/DROP_801/DP_81_01/Criteria/110/110_V		
			2		Plus 2 replacement			/sotestdata/DROP_801/DP_81_01/Criteria/110/110_W		
110_4	S4		40		MOD29P1D			20 granules for an Order and 20 granules for background Ingest /sotestdata/DROP_801/DP_81_01/Criteria/110/110_4		

[N1]Insert the utility and parameter to be used.

EXPECTED RESULTS:

16.6 Test Case 14 - Repair â€“ DP_7F_01, Criterion 300 partially (ECS-ECSTC-209)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Repair – DP_7F_01, Criterion 300 partially]		#comment
2	Perform repairs of the discrepancies listed test case 13.		
3	Verify that the orphans identified mentioned in test case 13 have been repaired, that is, that the files are either no longer in the Data Pool or that there are now granule entries which reference these files. (For 13-b, use DPL CleanupFileOnDisk to repair)		

#	Action	Expected Result	Notes
4	Verify that the phantoms mentioned in test case 13 have been repaired, i.e., that all files and links for these granules are now present in the Data Pool inventory and in the public or hidden Data Pool location.		
5	Verify that the invalid links mentioned in test case 13 have been repaired, i.e., that they have been removed or replaced.		
6	Verify that the success or failure of the repairs can be verified either via the Data Pool Maintenance GUI or via exit codes returned by the repair utilities.		

TEST DATA:
Test data in test case 13

EXPECTED RESULTS:

16.7 [OBSOLETE] - Test Case 15 - Most Recent Inserts Utility (ECS-ECSTC-210)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Most Recent Inserts Utility]		#comment
2	Run the “Most Recent Inserts Utility” (EcDlMostRecentInsert.pl).		
3	Inspect the files generated by the utility, at the top level of the Data Pool directory structure (DPRecentInserts_<YYYYMMDD>), and at each collection level (DPRecentInserts_<Shortname>_<VersionId>_<YYYYMMDD>), randomly checking the paths of a few granules, and verify that the information written by the utility is correct.		

TEST DATA:
n/a

EXPECTED RESULTS:

16.8 [OBSOLETE] - Test Case 16 - Remap Collection to New Group, DP_7F_01, Criterion 2020 (ECS-ECSTC-211)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Remap Collection to New Group, DP_7F_01, Criterion 2020]		#comment
2	Take DPL Ingest and DPAD down.		
3	Choose a populated collection from a group with more than one collection assigned to it. The collection must have at least one Most Recent Data Pool Inserts, collection-level file in its collection-level directory.		
4	Use the Data Pool Maintenance GUI to turn off the insertEnabledFlag for this collection. Use the Collection-to-Group Remapping utility to re-assign this collection to a new group for which a directory does not already exist in the Data Pool file system.		
5	Verify that the utility accepts the required command line parameters. Verify that the DlDimensionGroupESDT table has been updated correctly to reflect the new collection-to-group mapping.		
6	Verify that the DlCollection table has been updated correctly to reflect the new collection-to-group mapping.		
7	Verify that the DlFactGroupESDT table has been updated correctly for all granules in the collection.		
8	Verify that the new parent group directory is created on the Data Pool file system.		
9	Verify that the collection directory is accessible from the new parent group directory.		
10	Verify that the Most Recent Data Pool Inserts file(s) in the collection level directory is (are) accessible via the <new parent group directory>/<collection directory> path.		
11	Verify that the correct information is written to the utility log file.		
12	Find a granule that resides in the Data Pool that has an associated browse file and that will not expire before the test is complete.		
13	Bookmark the Data Pool URLs that are obtained through mechanisms other		

#	Action	Expected Result	Notes
	than EDG for this science granule, its associated metadata file, and its associated browse file. Remap the collection to which this granule belongs to a different group.		
14	After the remapping operation has completed successfully, verify that the science granule, its associated metadata, and its associated browse can be retrieved successfully using the bookmarked URLs.		
15	Attempt to start the Collection-to-Group Remapping utility while another copy of the utility is running in the same mode.		
16	Verify that the second copy exits and returns the appropriate error messages both on the command line and in its log.		
17	Attempt a drill down operation on a collection which is being remapped.		
18	Verify that either the drill down is prevented, or that no drill down errors occurs during the remapping operation, and that after the remapping is completed, granules in the collection can be downloaded from a drill down results page.		
19	Attempt to run the Cleanup utility to clean up granules that were removed from AIM, i.e., for which EcDsDeletionCleanup was run for the collection that is being remapped.		
20	Verify that the granules in the collection are eventually cleaned up correctly from the database and from disk.		
21	Perform collection remapping while Data Pool Ingest and OMS are operating.		
22	Choose a collection that is large enough such that at least ten (10) granules can be ingested and ten (10) public granules can be distributed via FTP Pull during the remapping operation; and attempt to ingest and distribute at least this number of granules from the collection during the remapping operation.		
23	Verify that the remapping operation completes successfully.		
24	Verify that if any ingest or distribution errors occur, they all result in operator interventions and one can recover from them (i.e., complete the corresponding ingest request respectively order successfully) by closing the intervention after the collection remapping operation completed		

TEST DATA:

use a collection with only a few granules, but for which there are pending orders, i.e., with granules residing in the hidden directory for the collection

EXPECTED RESULTS:

16.9 Test Case 17 - GUI Security (ECS-ECSTC-212)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[GUI Security]		#comment
2	For the DPM GUI, verify that on startup the operator is asked to login and is allowed the use of “full access” features only after successful login.		

TEST DATA:
n/a

EXPECTED RESULTS:

16.10 [PVC] - Test Case 18 - S4 order (ECS-ECSTC-213)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[S4 order]		#comment
2	Place an order through OMS for a granule that is not already in the Data Pool and the collection is not configured public.		
3	Verify that the granule is stored in the hidden Data Pool, that the “order only” flag is set correctly for the granule in the DlGranules table, and that the granule does not appear on a Data Pool drill down search results using the Data Pool Web Access GUI.		

TEST DATA:
n/a

EXPECTED RESULTS:

16.11 Test Case 20 - Hidden Scrambler Utility (ECS-ECSTC-214)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Hidden Scrambler Utility]		#comment
2	With Data Pool inserts shut down, use the Data Pool Hidden Scrambler utility with the –shortname/-versionid command line parameters to generate a new hidden directory name for a single collection, where there are pending orders for granules in that collection.		
3	Verify that the new hidden directory is created, that all files for pending orders for the collection are moved to the new hidden directory, that all FTPPull links for existing orders referencing the old hidden directory now point to the new hidden directory, and that the old hidden directory is removed.		

TEST DATA:
Any current data type which is already in the DlCollections table with Data Pool inserts enabled.

EXPECTED RESULTS:

16.12 Test Case 21 - Create Collections on DPL Maintenance GUI (ECS-ECSTC-215)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Create Collections on DPL Maintenance GUI]		#comment
2	Using the Data Pool Maintenance GUI, define a new collection group, as well as two new collections for that group.		
3	Verify the following:		
4	Corresponding secret directory names are created and saved in the Data Pool inventory database.		
5	A subsequent order that references at least one granule in each of the collections that is not yet in the Data Pool stages the granules successfully and the corresponding hidden directories have been created		

TEST DATA:
Any current data types that are not already in the DlCollections table in the Data Pool

EXPECTED RESULTS:

16.13 [OBSOLETE] - Test Case 22 - Detection of Illegal Access to Hidden Directories: OD_S5_06, Criterion 90 (ECS-ECSTC-216)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Detection of Illegal Access to Hidden Directories: OD_S5_06, Criterion 90]		#comment
2	Note the complete path for two order-only granules each in two different collections.		
3	Use anonymous ftp to download those granules from outside the firewall several times such that the total number of accesses to each collection directory are not the same.		
4	Configure an alert e-mail address for the firewall ftp rollup script (i.e., NOTIFICATION_EMAIL_FOR_HIDDEN_BREECH in		

#	Action	Expected Result	Notes
	EcDlFwWuFtpRollup.CFG on p4ftl01).		
5	After the firewall log containing the anonymous ftp accesses has been downloaded to the Data Pool host, run the firewall ftp rollup script.		
6	Verify the following:		
7	An e-mail is sent to the alert address.		
8	The email lists each accessed collection, and for each, the correct external address, the correct number of access, and the correct time of the first and last access.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.14 Test Case 23 - DPL Maintenance GUI (ECS-ECSTC-217)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[DPL Maintenance GUI]		#comment
2	Open the DPL Maintenance GUI in a web browser.		
3	Verify that a. A collection can be added		
4	b. A collection can be deleted 		
5	c. A collection can be updated 		
6	d. A collection group can be added 		
7	e. A collection group can be updated 		
8	f. A theme can be added 		
9	g. A theme can be deleted 		
10	h. A theme can be updated		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.15 Test Case 24 - Move Collection Utility (ECS-ECSTC-218)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Move Collection Utility]		#comment
2	Use the Move Collection Utility to move a collection to a new file system.		
3	Ensure that granules are inserted into the collection in the DataPool at 10 granules per minute.		
4	Ensure there are OMS ftp pull orders being executed at the same time as well as Ingest requests against the collection.		
5	Verify the public directories for the collection were moved and contain all the granules that existed in the original collection, plus the newly inserted ones, plus the one that was made public.		
6	Verify the hidden directories for the collection were moved, a link to the new location was left behind, and the moved directories contain all the granules they contained originally minus the one granule that become public, plus the granules that were newly inserted.		
7	Verify that only a link to the collection in the target collection remains in the collection directory in the source.		
8	The OMS granules that were ordered during the move can be pulled successfully after the move completed.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.16 Test Case 26 - Publishing Hidden Science Granule with Browse (ECS-ECSTC-219)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Publishing Hidden Science Granule with Browse]		#comment
2	Use Publish Utility to publish an existing science granule in the hidden Data Pool. The science granule has an associated browse.		
3	For each science granule, verify that the granule file and associated xml files are stored in the public Data Pool.		
4	For the science granule, verify that the browse linkage information is included in the Data Pool xml file for the science granule.		
5	For each browse granule, verify that the jpeg versions of the associated browse granule(s) are stored in the Data Pool in the public directory structure.		
6	[Regression] Verify that the start and completion of the publishing operations are logged.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.17 Test Case 27 - Backfill : DP_72_01, Criterion 170 (ECS-ECSTC-220)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Backfill : DP_72_01, Criterion 170]		#comment
2	Place a (or find an existing) qualified Data Pool insert subscription for a collection which is configured to be archived but NOT to be inserted into the		

#	Action	Expected Result	Notes
	public Data Pool and that is enabled for HEG processing		
3	Submit a PDR for a granule which satisfies the subscription qualifications. The PDR should include an associated Browse granule. The granule should be a real granule.		
4	Verify that the granule is inserted into the public Data Pool directories, and that all information in the Inventory database that should be present for public granules is populated (i.e., band information).		
5	Verify that the HEG Band extraction took no more than five seconds.		
6	Verify that the associated Browse granule files are also in the public Data Pool.		
7	Verify that the tape archive was not accessed, i.e., no transfer from archive and no checksumming took place.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.18 Test Case 28 - Granule Replacement ON, Collection Configured For Public Data Pool Insert : DP_72_01, Criterion 180 (ECS-ECSTC-221)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Granule Replacement ON, Collection Configured For Public Data Pool Insert : DP_72_01, Criterion 180]		#comment
2	Submit a PDR for a granule in a collection that is configured for Granule Replacement, and is configured for archiving and for public Data Pool insert on Data Pool Ingest. After the Ingest request completes successfully, submit a second PDR for another granule in this collection which is a replacement for the first (i.e. has the same acquisition date and either the same LocalGranuleId or the same RangeBeginningTime).		

#	Action	Expected Result	Notes
3	[NOTE: This criterion requires DPL Ingest capabilities as specified in DP_S6_01.]		#comment
4	Verify that the second granule replaces the first granule in the public Data Pool directories and that the first granule is now in the hidden Data Pool.		
5	Verify that both granules are archived successfully.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.19 Test Case 29 - Granule Replacement ON, Older Granule Does not Replace Newer Granule (ECS-ECSTC-222)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Granule Replacement ON, Older Granule Does not Replace Newer Granule]		#comment
2	Submit a PDR for a granule in a collection that is configured for Granule Replacement, and is configured for archiving and for hidden Data Pool insert on Data Pool Ingest.		
3	After the Ingest request completes successfully, modify the collection to enable public Data Pool insert, then submit a second PDR for another granule in this collection which is a replacement for the first (i.e. has the same acquisition date and either the same LocalGranuleId or the same RangeBeginningTime).		
4	After the second Ingest request completes successfully, use the Publish Utility to publish the first ingested granule.		
5	Verify that the second granule stays in public Data Pool, the second granule stays in the hidden Data Pool, and the publication request failed.		
6	Verify that both granules are archived successfully.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.20 Test Case 30 - Granule Replacement OFF, Collection Configured For Public Data Pool Insert : DP_72_01, Criterion 185 with modification (ECS-ECSTC-223)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Granule Replacement OFF, Collection Configured For Public Data Pool Insert : DP_72_01, Criterion 185 with modification]		#comment
2	Submit a PDR for a granule in a collection that is NOT configured for Granule Replacement, but is configured for public Data Pool insert on Data Pool Ingest.		
3	After the Ingest request completes successfully, submit a second PDR for another granule in this collection which is a replacement for the first (i.e. has the same acquisition date and either the same LocalGranuleId or the same RangeBeginningTime).		
4	After the second Ingest request completes, submit a third PDR for another granule in this collection which is a replacement for the first (i.e. has the same acquisition date and either the same LocalGranuleId or the same RangeBeginningTime).		
5	Verify that the first granule is stored in the public Data Pool, and the second and the third granules fail publication but is inserted into the hidden DataPool. Verify the suffix of the second and the third granules are consecutive numbers.		
6	Verify that the PAN is sent for three granules, and that neither PAN reports an ingest error		
7	Verify that three granules are archived successfully, that the second and the third granules are stored in the Data Pool hidden directory, and that each		

#	Action	Expected Result	Notes
	granule has unique filenames.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.21 Test Case 32 - Publish After Validation, DP_7F_01, Criterion 130 (ECS-ECSTC-225)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Publish After Validation, DP_7F_01, Criterion 130]		#comment
2	Perform a Data Pool inventory validation that reports at least four (4) granules from a collection configured for publishing during ingest as missing from the public Data Pool and pick up the output of that run with a publishing run.		
3	Verify that the granules are indeed published, i.e., the granules are no longer flagged as not public in the Data Pool inventory and their files are in the correct public Data Pool directory and no longer in the hidden Data Pool area.		
4	Verify that the start and completion of the publishing run are logged, as well as the publishing attempt/success for each of the granules reported by the Data Pool inventory validation run.		
5	Verify that the next automatic BMGT export will include the URL insertions of the granules that were published.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.22 [OBSOLETE] - Test Case 33 - On-Line Archive Bulk Repair, DP_7F_01, Criterion 330 (ECS-ECSTC-226)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[On-Line Archive Bulk Repair, DP_7F_01, Criterion 330]		#comment
2	Perform a bulk repair using the output from an inventory validation utility that discovered at least 500 granules in the AIM inventory that are missing from the Data Pool (i.e., their Data Pool inventory entries and files are both missing). Ensure that the granules include at least one non-science granule of each type (DAP, PH, QA) in the repair. The granules shall be distributed over at least three (3) tapes. Between 90% and 95% of these granules shall belong to collections configured for publication during ingest; the rest shall not belong to such collections. At least one of the granules shall be a non-science granule.		
3	Identify the hosts to be used for checksumming operations.		
4	Specify limits for the number of concurrent checksumming operations, tape volumes accessed concurrently, and concurrent tape read operations for the same tape; all these limits shall be greater than one.		
5	Provide the command line parameters respectively configuration settings that are needed to control the number of Data Pool publications (maxnumconactions) .		
6	Provide a concurrent ingest workload of at least ten granules per minute, all of which shall belong to collections being published.		
7	Verify that all granules were repaired, i.e., are in the Data Pool inventory.		
8	Verify that the granules that belong to collections configured for publication during ingest are in public Data Pool directories and are public in the Data Pool inventory.		
9	Verify that the granules that do not belong to collections configured for publication during ingest are in hidden Data Pool directories and are identified as non-public in the Data Pool inventory.		
10	Verify that utility parallelized its operation; including access to multiple tape volumes concurrently, multiple concurrent checksumming operations, and		

#	Action	Expected Result	Notes
	multiple concurrent read operations; and that it did so within the constraints imposed by the limits specified in a. to h..		
11	Verify that utility caused parallel Data Pool publishing operations, but that its publishing activity is constrained in accordance with settings chosen in a. to h..		
12	Verify that the successful repair can be verified by an operator via the Data Pool Maintenance GUI or an exit code returned by the utility.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.23 Test Case 34 - Un-publish Granules that were Reported by Data Pool Inventory Validation, DP_7F_01, Criterion 90 (ECS-ECSTC-227)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Un-publish Granules that were Reported by Data Pool Inventory Validation, DP_7F_01, Criterion 90]		#comment
2	Perform a Data Pool inventory validation that reports at least four granules as being public that are not eligible to be public as per S-DPL-08180f(listed below) and pick up the output of that run with an un-publishing run.		
3	Verify that the granules are indeed un-published, i.e., the granules are no longer flagged as public in the Data Pool inventory and their files are in the correct hidden Data Pool directory and no longer in the public Data Pool area.		
4	Verify that the start and completion of the un-publishing run are logged, as well as the un-publishing attempt/success for each of the granules reported by the Data Pool inventory validation run.		
5	Verify that the un-publishing run exits with an exit code indicating success.		
6	Verify that the next automatic BMGT export will include the URL deletions		

#	Action	Expected Result	Notes
	of the granules that were un-published.		
7	S-DPL-08180f - The DPL CI shall not make granules public that are currently flagged as logically deleted (i.e., have a non-NULL deleteEffectiveDate), deleted from archive (i.e., DeleteFromArchive set to ‘Y’), or hidden from normal users (i.e., DeleteFromArchive set to ‘H’) in the AIM CI inventory.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.24 [OBSOLETE] - Test Case 35 - Non-ECS insert: DP_S3_02, Criterion 110 (ECS-ECSTC-228)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	The only use case that we currently have for non-ECS granules is at the ASDC DAAC, who put MISR campaign data into their Data Pool as non-ECS granules. These non-ECS granules have no browse, so we don't need to support or test non-ECS browse. These non-ECS granules have either bounding coordinates or no coordinates, so we don't need to support any other spatial types for non-ECS granules. These non-ECS granules have a RangeDateTime attributes, so we don't need to support SingleDateTime attributes. These non-ECS granules have no day night flag, so we don't need to support day night flags.		#comment
2	As cmshared, create a working directory, such as /tools/common/test/8.2/regression/dpl/893.		
3	Copy all of the files from the /sotestdata/SynergyVI/DP_S3_02/Criteria/100 directory to the working directory. The metadata data files have the following: MISR_AEROSOL...xml has a LocalGranuleID (external identifier), no spatial attributes, no day night flag, and no measured		

#	Action	Expected Result	Notes
	parameters MISR_AM1_CGAS...xml has a LocalGranuleID, global coordinates, no day night flag, and 1 measured parameter MISR_AM1_CGLS...xml has no LocalGranuleID, global coordinates, no day night flag, and 1 measured parameter MISR_ELLIPSOID...xml has a LocalGranuleID, no spatial attributes, no day night flag, and no measured parameters MISR_TERRAIN...xml has no LocalGranuleID, no spatial attributes, no day night flag, and no measured parameters		
4	Change the paths in the Criteria_100.BIR file to match the working directory.		
5	Use the Publish Utility to insert the selected granules into the public data pool. Specify a valid batch label, retention priority, retention period, and dispatch priority.		
6	Verify the insert actions are queued correctly and contain the correct contents. Their execution can be monitored using the Data Pool Monitoring GUI, and batch label and dispatch priority can be displayed. XML file and path name can be displayed by the Data Pool Monitoring GUI.		
7	Verify the granule and metadata files are inserted correctly into the data pool.		
8	Verify the granule and metadata file pathnames are recorded correctly in the data pool inventory.		
9	Verify the publish log contains the required entries.		
10	Verify the source files have been removed from the source directory (the working directory where the files were copied earlier).		

TEST DATA:
Non-ECS data types

EXPECTED RESULTS:

16.25 [PVC] - Test Case 36 - Run DPCV against DPL, Provide a List of Granule IDs : CK_7F_01, Criterion 540 (ECS-ECSTC-229)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Run DPCV against DPL, Provide a List of Granule IDs : CK_7F_01, Criterion 540]		#comment
2	Select several granules (at least 10) from Data Pool that belong to at least two ESDTs.		
3	Manually alter the checksum values in the Data Pool database for at least 2 granule files.		
4	Set the Last Verification time to null for 2 other granules.		
5	Set the Checksum Verification Status to “failed” for 2 granules.		
6	Run DPCV against Data Pool providing the list of granule IDs as input.		
7	Verify that the DPCV run completed successfully.		
8	Verify that DPCV performed checksum verification for all data files that belong to the list of granules in S-540-1.		
9	Verify that DPCV checksum verification was successful for those files whose checksum values were not altered in S-540-1. Verify the following in DPL database: Checksum time was updated correctly for each affected file Checksum origin was updated as “DPCV” for each affected file that had a null last checksum verification time. Checksum verification status was set to a success status.		
10	Verify that DPCV failed checksum verification for those files whose checksums have been altered as described in S-540-2. Verify the following in DPL database: Checksum verification status was set to a failure status.		
11	Verify that DPCV logs an error message for each file that failed checksum verification, and that the error message includes information specified in the L4 requirement S-DPL-49180.		
12	Verify that the Last Verification time and status for the granules modified in S-540-3 was populated.		
13	Verify that the verification status was set to “success” for the granules modified in S-540-4.		
14	Verify that DPCV logs an error message for each file that failed checksum verification, and that the error message includes information specified in the L4 requirement S-DPL-49180 and S-DPL-49190.		

TEST DATA:
any current collection group

EXPECTED RESULTS:

16.26 Test Case 37 -Run DPCV for a Single ESDT, with Granule Insert Date Range with Checksum Failures: CK_7F_01, Criterion 510 (ECS-ECSTC-230)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Run DPCV for a Single ESDT, with Granule Insert Date Range with Checksum Failures: CK_7F_01, Criterion 510].		#comment
2	Select an ESDT in Data Pool that has at least 30 granules.		
3	Note the earliest insert date and latest insert date of the granules selected. Choose a date range that will contain some, but not all of the granules.		
4	Manually alter the checksum values in the Data Pool database for at least 2 of science files belonging to granules whose insert time lies within the chosen date range.		
5	Run DPCV against Data Pool for the ESDT selected above, specifying the insert date range determined in S-510-1		
6	Verify that DPCV run completed successfully.		
7	Verify that DPCV performed checksum verification for only those data files selected in S-510-1 whose granule insert times fall within the desired granule insert time range.		
8	Verify that DPCV checksum verification was successful for those files whose checksum values were not altered in S-510-1. Verify the following in DPL database: Checksum time was updated correctly for each affected file Checksum origin was updated as “DPCV” for each affected file that had a null last checksum verification time. Checksum verification status was set to a success status for files with checksums that were not altered.		

#	Action	Expected Result	Notes
9	Verify that DPCV failed checksum verification for those files whose checksums have been altered as described in S-510-1. Verify the following in DPL database:		
10	Checksum verification status was set to a failure status.		
11	Verify that DPCV logs all the information specified S-DPL-49180 and S-DPL-49190.		

TEST DATA:
any current collection group

EXPECTED RESULTS:

16.27 [PARTIAL OBSOLETE] - Test Case 39 -QA/PH DPL Ingest, Insert:DP_7G_01, Criterion 100 - [PVC - Web Access portion is Obsolete] (ECS-ECSTC-231)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[QA/PH DPL Ingest, Insert and Web Access:DP_7G_01, Criterion 100]		#comment
2	Configure QA and PH collections to be public (i.e., have their granules queued for publishing as part of ingest processing).		
3	Use the DPM GUI to enable at least two public AMSR collections for distribution of QA, PH and Browse.		
4	Ensure that there is at least one public collection X (non AMSR) that is not enabled in that manner but does have Browse.		
5	For each type of associated granule, ensure that there is at least one granule (granule set Y) in each AMSR collection that does not have that associated granule (but does offer the others).		
6	Ensure that there is at least one granule (granule set Z) in each AMSR collection that does not have any associated granules.		
7	For each public AMSR collection, ingest at least two science granules		

#	Action	Expected Result	Notes
	(granule set W) with their QA, PH and Browse.		
8	Cause the Science and Browse granules to be published before the associated QA and associated PH.		
9	For each public AMSR collection, ingest at least two science granules (granule set W) with their QA, PH and Browse.		
10	Cause the Science and Browse granules to be published before the associated QA and associated PH.		
11	Verify that the QA, PH, and Browse granules are published.		
12	Verify that the directories into which the science granules were placed contain symbolic links pointing to the correct QA, PH, and Browse granules and that these links are correctly named, as per S-DPL-10220.		
13	steps related to Web Access starting from step 12 are removed for 8.2		#comment

TEST DATA:
any current collection group

EXPECTED RESULTS:

16.28 Test Case 40 - QA/PH/Browse DPL Replacement (ECS-ECSTC-232)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[QA/PH/Browse DPL Replacement]		#comment
2	Configure QA and PH collections to be public (i.e., have their granules queued for publishing as part of ingest processing). Use the DPM GUI to enable at least two public AMSR collections for distribution of QA, PH, and Browse. For one of these collections (C1), Browse granules shall be related to one science granule. For the other collection (C2), Browse granules shall be related to many science granules. Ensure that granule replacement is turned on for the AMSR and the QA and PH collections.		

#	Action	Expected Result	Notes
3	Ingest at least one AMSR granule for collections C1, together with its PH, QA, and Browse (granule set G2). Ensure that there is at least one granule (granule set G1) from this AMSR collection in the public Data Pool for a previous date, together with its QA, PH and Browse (also from a previous date). These granule sets will be used to test science granule replacement (together with QA, PH and Browse).		
4	Ingest at least one AMSR granule for collections C1, together with its PH, QA, and Browse (granule set G3). Ensure that there is at least one granule (granule set G4) from this AMSR collection in the public Data Pool for a previous date, together with its QA, PH and Browse (also from a previous date). These granule sets will be used to test browse granule replacement.		
5	Ingest at least one set of granules for collection C2 that share a Browse, together with the Browse and their QA and PH granules (granule set G5). Ensure that there is at least one other set of such granules from that collection in the public Data Pool for a previous date, together with their QA, PH and Browse (granule set G6). These granule sets will be used to test science granule replacement (together with QA, PH and Browse).		
6	Ingest at least one other set of granules for collection C2 that share a Browse, together with the Browse and their QA and PH granules (granule set G7). Ensure that there is at least one other set of such granules from that collection in the public Data Pool for a previous date, together with their QA, PH and Browse (granule set G8). These granule sets will be used to test Browse granule replacement.		
7	Ingest at least one other set of granules for collection C2 that share a Browse, together with the Browse and their QA and PH granules, but omit one of the science granules of this set that would be linked with that Browse granule, as well (granule set G9). Ensure that there is at least one other set of such granules from that collection in the public Data Pool for a previous date, together with their QA, PH and Browse (granule set G10). These granule sets will be used to test addition of a science granule (with its QA, PH, and Browse) to an existing set of science granules, thereby replacing their Browse.		
8	After the ingest and publishing operations complete, verify that the symbolic links for the PH, QA, and Browse files are created, point to the correct files, and have the correct names.		
9	Ingest a replacement granule for each newly ingested AMSR granule in granule set G2 together with its QA, PH, and Browse (i.e., on the same		

#	Action	Expected Result	Notes
	day). Verify that the new PH, QA, and Browse granules and their files are in the public Data Pool.		
10	Verify that the symbolic links for the new PH, QA, and Browse granules are present, point to the files of the replacement granules, and have the correct names.		
11	Verify that the previous versions of the AMSR granules and their PH and QA granules were unpublished and reside in the hidden Data Pool.		
12	Verify that the previous version of the Browse granule was removed from the Data Pool.		
13	Verify that the Data Pool inventory contains a Browse cross reference between the new science granule and its Browse.		
14	Verify that the AIM inventory contains the cross references between the new science granule and the new QA, PH, and Browse granules.		
15	Ingest a replacement granule for a granule in each AMSR granule in granule set G1 together with its QA, PH, and Browse. Verify that the new PH, QA, and Browse granules and their files are in the public Data Pool.		
16	Verify that the symbolic links for the new QA, PH, and Browse granules are present, point to the correct files of the replacement granules, and have the correct names.		
17	Verify that the previous versions of the AMSR granules are unpublished, but that the previous versions of the QA and PH granules remain public.		
18	Verify that the previous version of the Browse granule was removed from the Data Pool.		
19	Verify that the Data Pool inventory contains a Browse cross reference between the new science granule and its Browse.		
20	Verify that the AIM inventory contains the cross references between the new science granule and the new QA, PH, and Browse granules.		
21	Ingest a replacement granule for each of the Browse granules used by the science granules in granule set G3. Verify that the new Browse granules and their files are in the public Data Pool.		
22	Verify that the symbolic links for the new Browse granules are present, point to the correct files of the replacement Browse, and have the correct names.		
23	Verify that the previous version of each Browse granules was removed from the Data Pool.		
24	Verify that the Data Pool inventory contains a Browse cross reference between each science granule and its new Browse.		
25	Verify that the AIM inventory contains the cross references between the		

#	Action	Expected Result	Notes
	science granules and their new Browse granules and no cross-references between these science granules and their old Browse granules.		
26	Ingest a replacement granule for each of the Browse granules used by the science granules granule set G4. Verify that the new Browse granules and their files are in the public Data Pool.		
27	Verify that the symbolic links for the new Browse granules are present, point to the correct files of the replacement Browse, and have the correct names.		
28	Verify that the previous version of each Browse granules was removed from the Data Pool.		
29	Verify that the Data Pool inventory contains a Browse cross reference between each science granule and its new Browse.		
30	Verify that the AIM inventory contains the cross references between the science granules and their new Browse granules and no cross-references between these science granules and their old Browse granules.		
31	Ingest a replacement granule for one of the newly ingested AMSR granules in each granule set G5 together with its QA, PH, and Browse (i.e., on the same day). Verify that the new PH, QA, and Browse granules and their files are in the public Data Pool.		
32	Verify that the symbolic links for the new PH, QA, and Browse granules are present, point to the files of their replacement granules, and have the correct names.		
33	Verify that the previous versions of the ingested AMSR granules and their PH and QA granules were unpublished and reside in the hidden Data Pool.		
34	Verify that each previous version of the ingest Browse granules was removed from the Data Pool.		
35	Verify that the Data Pool inventory contains a Browse cross reference between all of the science granules in each set G5 and their Browse.		
36	Verify that the AIM inventory contains the cross references between each new science granule and the corresponding new QA and PH Browse granules.		
37	Verify that the AIM inventory replaced the cross references between all of the science granules in each set G5 with cross references to their corresponding newly ingested Browse.		
38	Verify that the AIM inventory still contains the cross references between the replaced science granules and their corresponding QA, PH and old Browse granules.		
39	Ingest a replacement granule for one of the AMSR granule in each granule set		

#	Action	Expected Result	Notes
	G6 together with its QA, PH, and Browse. Verify that the new PH, QA, and Browse granules and their files are in the public Data Pool.		
40	Verify that the symbolic links for the new PH, QA, and Browse granules are present, point to the files of their replacement granules, and have the correct names.		
41	Verify that the previous versions of the AMSR granules are unpublished, but that the previous versions of the QA and PH granules remain public.		
42	Verify that the previous versions of the AMSR granules are unpublished, but that the previous versions of the QA and PH granules remain public.		
43	Verify that the Data Pool inventory contains a Browse cross reference between all of the science granules in each set G6 and their Browse.		
44	Verify that the AIM inventory contains the cross references between each new science granule and the corresponding new QA and PH Browse granules.		
45	Verify that the AIM inventory replaced the cross references between all of the science granules in each set G6 with cross references to their corresponding newly ingested Browse.		
46	Verify that the AIM inventory still contains the cross references between the replaced science granules and their corresponding QA, PH and old Browse granules.		
47	Ingest a replacement granule for each of the Browse granules used by the science granules in granule set G7. Verify that the new Browse granules and their files are in the public Data Pool.		
48	Verify that the symbolic links for the new Browse granules are present, point to the correct files of the replacement Browse, and have the correct names.		
49	Verify that the previous version of each Browse granules was removed from the Data Pool.		
50	Verify that the Data Pool inventory contains a Browse cross reference between the science granules in each granule set G7 and their respective new Browse.		
51	Verify that the AIM inventory contains the cross references between the science granules and their new Browse granules and no cross-references between these science granules and their old Browse granules.		
52	Ingest a replacement granule for each of the Browse granules used by the science granules in granule set G8. Verify that the new Browse granules and their files are in the public Data Pool.		
53	Verify that the symbolic links for the new Browse granules are present, point		

#	Action	Expected Result	Notes
	to the correct files of the replacement Browse, and have the correct names.		
54	Verify that the previous version of each Browse granules was removed from the Data Pool.		
55	Verify that the Data Pool inventory contains a Browse cross reference between the science granules in each granule set G8 and their respective new Browse.		
56	Verify that the AIM inventory contains the cross references between the science granules and their new Browse granules and no cross-references between these science granules and their old Browse granules.		
57	For each of the granule sets G9, ingest the granule that was originally omitted from the set, together with is QA, PH and a new Browse. Verify that the added science granules, their QA and PH granules, and their Browse granules are in the public Data Pool.		
58	Verify that the symbolic links for the new PH, QA, and Browse granules are present, point to the files of their replacement granules, and have the correct names.		
59	Verify that each previous version of the ingest Browse granules was removed from the Data Pool.		
60	Verify that the Data Pool inventory contains a Browse cross reference between all of the science granules in each set G9 and their Browse.		
61	Verify that the AIM inventory contains the cross references between each new science granule and the corresponding new QA and PH granules.		
62	Verify that the AIM inventory replaced the cross references between all of the science granules in each set G9 with cross references to their newly ingested Browse.		
63	Verify that each previous version of the ingest Browse granules was removed from the Data Pool.		
64	Verify that the Data Pool inventory contains a Browse cross reference between all of the science granules in each set G10 and their Browse.		
65	Verify that the AIM inventory contains the cross references between each new science granule and the corresponding new QA and PH granules.		
66	Verify that the AIM inventory replaced the cross references between all of the science granules in each set G10 with cross references to their newly ingested Browse.		
67	Ensure that the inventory changes in the verification steps (but not the ingest operation in the setup steps) are performed in a single automatic BMGT export cycle. Cause that export to occur and verify that it includes		

#	Action	Expected Result	Notes
	the URLs for the replacement AMSR granules, including their PH and QA and Browse URLs; includes the URL deletions for all the replaced AMSR science granules; and that the QA and PH and Browse URLs are correct and are labeled in accordance with S-BGT-31940.		
68	Perform an inventory validation and verify that it does not report invalid links, orphans, or phantoms associated with any of the science, QA, and PH granules used for this test (use a maximum orphan age of zero).		

TEST DATA:
See CCU test plan

EXPECTED RESULTS:

16.29 Test Case 41 - QA/PH publishing and unpublishing (ECS-ECSTC-233)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[QA/PH publishing and unpublishing] Choose an AMSR science granule with associated QA, PH, and Browse.		#comment
2	Unpublish the science granule. Verify that the symbolic links to the QA, PH, and Browse are removed.		
3	Verify that the Browse granule stays in public Data Pool.		
4	Publish the science granule.		
5	Verify that the symbolic links to the QA, PH, and Browse are re-established, point to the correct files, and have the correct names.		
6	Unpublish the QA and PH granule. Verify that the QA and PH granules are no longer public and that their files reside in the hidden Data Pool.		
7	Verify that the symbolic links to the QA and PH files were removed.		
8	Ensure that the inventory changes in the above steps occur in a single automatic BMGT export cycle. Cause the automatic export to occur. Verify that the net effect of the export will be the removal of the QA and PH URLs		

#	Action	Expected Result	Notes
	from ECHO, but not that of the other granule URLs.		
9	Perform an inventory validation and verify that it does not report invalid links, orphans, or phantoms associated with the science, QA, and PH granules used for this test (use a maximum orphan age of zero).		
10	Submit an ECHO order that includes the granule whose QA and PH granules were unpublished and request distribution of the associated QA and PH (the ECHO order can be submitted via an EWOC test driver). Verify that the order completes successfully and the granule including its QA and PH granules are distributed.		
11	Publish the QA and PH granules. Verify that the QA and PH granules are now public and that their files reside in the public Data Pool.		
12	Ensure that the inventory changes in the above step occur in a single automatic BMGT export cycle. Cause the automatic export to occur.		
13	Verify that the net effect of the export will be the addition of the QA and PH URLs to ECHO.		
14	Verify that the symbolic links to the QA and PH files were re-established, point to the correct files, and have the correct names.		
15	Unpublish the Browse granule. Verify that the Browse granule was removed from Data Pool.		
16	Verify that the symbolic links to the Browse files were removed.		
17	Publish the Browse granule. Verify that the Browse granule is now public and that its files reside in the public Data Pool.		
18	Verify that the symbolic links to the Browse files were re-established, point to the correct files, and have the correct names.		

TEST DATA:
See CCU test plan

EXPECTED RESULTS:

16.30 [OBSOLETE] - Test Case 42 - On-Line Archive Repair including QA and PH links (ECS-ECSTC-234)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[On-Line Archive Repair including QA and PH links]		#comment
2	Turn at least ten public AMSR granules that have associated QA, PH, and Browse granules into phantoms, i.e., their Data Pool (and AIM) inventory entries are intact, but their science and XML files and their symbolic links are missing from the On-Line Archive. At least one of the AMSR granules shall be referenced by an order. Remove its hidden links.(S1)		
3	Also turn at least two public QA and two PH granules into phantoms, i.e., their Data Pool (and AIM) inventory entries are intact, but neither their files nor the symbolic links that point to them are still in the On-Line Archive, but their AMSR granules are present and public, and are not among the phantoms listed in S-1. At least one of the QA and one of the PH granules shall be referenced by an order. Remove their hidden links, as well.(S2)		
4	Submit an on-line archive repair for the AMSR granules in S-1 and the QA/PH granules in S-2 via the command line, specifying the granules to be repaired in an input file; and request restoration of links that may be missing for granules whose files are being restored.(S3)		
5	Verify that all AMSR science granules were repaired, i.e., their files and symbolic links are now in the correct public Data Pool locations, have the original file respectively, link names, and that the links point to the same QA, PH, and Browse files as before.		
6	Verify that all QA and PH granules were repaired, i.e., their files are now in the correct public Data Pool locations and the symbolic links that point to the files have been restored in the correct locations, as well.		
7	Verify that the hidden links referenced in S-1 and S-2 were restored.		

TEST DATA:

EXPECTED RESULTS:

16.31 [PVC] - Test Case 43 - Move AMSR Collection (ECS-ECSTC-235)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Move AMSR Collection]		#comment
2	Move at least two AMSR collection each having at least 5,000 granules with QA, PH, and Browse associations. Each AMSR collection shall be distributed over at least ten (10) directories. At least one of the AMSR collections shall be coresident with the QA and/or PH collection on the same file system and at least one shall be on a different file system. Unless each DAAC has reserved a separate file system for Browse, at least one of the AMSR collections shall be coresident with the Browse collection on the same file system and at least one shall be on a different file system. Ensure that the move is performed during a single BMGT export cycle.		
3	Verify that the collections are moved to the new file system, including their QA, PH, and Browse links.		
4	For each collection, verify for at least five granules in five different directories that their Browse, QA, and PH links are correct.		
5	Perform an inventory validation and verify that it does not report invalid links, orphans, or phantoms associated with any of the science granules in the moved collections, nor for any QA and PH granules referenced by the granules in these collections (use a maximum orphan age of zero).		
6	Cause the automatic BMGT export for the cycle.		
7	Verify that the export packages will not cause changes to the QA and PH URLs in the ECHO inventory, i.e., does not export QA and PH URL updates, or if it does, the exported QA/PH URLs are not different from the QA/PH URLs prior to the move.		

TEST DATA:

EXPECTED RESULTS:

16.32 [PVC] - Test Case 46 - Link Checking Restart After Fault, DP_7F_01, Criterion 1022 (ECS-ECSTC-236)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Link Checking Restart After Fault, DP_7F_01, Criterion 1022]		#comment
2	Check the links in a Data Pool path containing at least 360,000 links and of which at least ten (10) are invalid links.		
3	Perform the link check specified above, but cause it to terminate due to a fault after 20-25% of its run time and then restart it.		
4	The test can be executed in the absence of concurrent workload if this is necessary to achieve the required throughput.		
5	Verify that the discrepancies are logged and reported correctly and only once.		
6	Verify that the check completes in less than X hours, where X is the number of symbolic links in the hidden Data Pool path selected for the test divided by 180,000, plus a 10% allowance for the recovery (i.e., the sum of the run times of the interrupted and the restarted run is less than X).		

TEST DATA:

EXPECTED RESULTS:

16.33 Test Case 47 - Fault Recovery â€“ Collection to Group Remapping utility (ECS-ECSTC-237)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault Recovery – Collection to Group Remapping utility]		#comment
2	Turn off the InsertEnabledAllowPublishFlagflag for the collections to be used in this test.		

#	Action	Expected Result	Notes
3	Run the Collection to Group Remapping utility (EcDlRemap.pl), remapping a collection with enough granules to allow the utility to be killed while in progress.		
4	Then run the utility again, mapping a different collection to a new group.		
5	Verify that when the utility is run the second time, the log indicates that a recovery of the first run took place.		
6	Verify that both collections are correctly remapped to the specified new groups (check the DlCollection table, the DlFile table, and the Data Pool directories).		

TEST DATA:
Any current data type

EXPECTED RESULTS:

16.34 [OBSOLETE] - Test Case 48 - Fault Recovery â€“ Most Recent Inserts Utility (ECS-ECSTC-238)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault Recovery – Most Recent Inserts Utility]		#comment
2	Run the Most Recent Inserts utility, and kill the utility process while it is running.		
3	Rerun the utility for the same reporting date, and let it run to completion.		
4	Verify that only one set of files is created at the top level and at each collection level for the date which the utility was reporting, and that the contents of these files are correct.		

TEST DATA:
n/a

EXPECTED RESULTS:

16.35 Test Case 49 - Fault Recovery â€“ Hidden Scrambler Utility (ECS-ECSTC-239)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault Recovery – Hidden Scrambler Utility]		#comment
2	Run the Hidden Scrambler utility with the –collgroup option (check with the lab lead before choosing a collection group). There should be outstanding orders for granules from collections in this collection group.		
3	Kill the Hidden Scrambler utility AFTER the “Updating DlCollections with new orderOnly SNDir Names” message appears in the log, but before the run completes.		
4	Restart the utility with the same –collgroup option.		
5	Verify that the utility log indicates a recovery is in process.		
6	At the end of the second run of the utility, verify that new hidden directories have been generated for all collections in the group, that all hidden files for the collection group have been copied to the new hidden directories, that all FTPPull links for granules in these collections now point to the new hidden directories, and that all old hidden directories for collections in the group have been removed.		

TEST DATA:
any current collection group

EXPECTED RESULTS:

16.36 [PVC] - Test Case 50 - Fault Recovery - Data Pool Un-publishing â€“ Restart after Fault, DP_7F_01, Criterion 1032 (ECS-ECSTC-240)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault Recovery - Data Pool Un-publishing – Restart after Fault, DP_7F_01, Criterion 1032]		#comment
2	Process the logical deletion of at least 50,000 science granules from the AIM inventory, as communicate by the AIM CI via the corresponding AIM-DPL interface.		
3	Each of these granules shall be related with a Browse granule that is not referenced by any other science granules.		
4	The collection shall be public and all of the science granule shall be public.		
5	Use Unpublish Utility to unpublish the above granules and cause the utility to terminate due to a fault after 20-25% of its run time and then restart it.		
6	The test can be executed in the absence of concurrent workload if this is necessary to achieve the required throughput, but do not run this test immediately after that for criterion 1030 (Data Pool Un-publishing Performance) to limit database cache effects.		
7	Verify that the science granules are un-published.		
8	Verify that the Browse granules are removed from the public Data Pool.		
9	Verify that the utility completes in X hours or less, where X is the number of science granules included in the test divided by 50,000.		

TEST DATA:

EXPECTED RESULTS:

16.37 [PVC] - Test Case 51 - Fault Recovery - Bulk On-Line Archive Repair â€“ Restart after Fault, DP_7F_01 Criterion 1052 (ECS-ECSTC-241)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault Recovery - Bulk On-Line Archive Repair – Restart after Fault, DP_7F_01 Criterion 1052]		#Run the EcDlRestoreOlaFromTapeStart utility
2	Perform a bulk repair that requires restoring at least 150 GB and at least 7,000 granules that reside on at least 2 and no more than 5 different LTO tapes.		
3	The granules shall not reside in the archive cache.		
4	Preferably, the tapes used for the test will not already reside in drives at the start of the test; for any tapes that are already loaded, no tape mount allowance shall be applied (see V-3.)		
5	The Data Pool inventory entries for these granules shall be intact.		
6	Between 90 and 95% of the granules shall be public, the remainder shall not be public. Identify the hosts to be used for checksumming operations.		
7	Specify limits for the number of concurrent checksumming operations, tape volumes accessed concurrently, and concurrent tape read operations for the same tape; all these limits shall be greater than one.		
8	The test can be executed in the absence of concurrent workload if this is necessary to achieve the required throughput.		
9	Perform a bulk repair as specified above. Cause it to terminate due to a fault after 20-25% of its run time and then restart it.		
10	The test can be executed in the absence of concurrent workload if this is necessary to achieve the required throughput.		
11	Verify that all granules were repaired.		
12	Verify that the repair completes in X seconds or less, where X is the total volume that was restored divided by 150MB, plus a 10% allowance for recovery (i.e., the sum of the run times of the interrupted and the restarted run is less than X), plus the allowances for tape loading and media information look-up as specified in criterion 1050 step V-3: Verify that the repair completes in X seconds or less, where X is the total volume that was restored		

#	Action	Expected Result	Notes
	divided by 150 MB, plus one minute for each tape that did not reside in a tape drive at the start of the test (allowance for tape loading) and half a minute for every 1,000 granules (allowance for media information look-up).		

TEST DATA:

EXPECTED RESULTS:

16.38 Test Case 52 - Fault Recovery - Data Pool Move Collection Utility when the Move of the Collection is Interrupted and then Restarted (ECS-ECSTC-242)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Fault Recovery - Data Pool Move Collection Utility when the Move of the Collection is Interrupted and then Restarted]		#comment
2	After running the Move Collection about ½ the elapsed time of test case 26, type a control-c at the command line to interrupt the Move Collection. A Linux prompt is returned.		
3	Restart the MoveCollection utility again using the same arguments.		
4	View the Move Collection log and find out if the interrupt was done before or after the fileSystemLabel update in DlCollection. If it is found, the interrupt is after otherwise the interrupt was before. In either case a BEFORE or AFTER recovery menu is displayed. a. Test deferring a move		
5	Verify the move collection script exits. b.Test completing a move in recovery		
6	Verify the Web Access Gui will not allow the user to drill down any further than the data set (collection level) page. Note the move collection message under the data set description on the data set page. Verify that the data set name is text and not a hyperlink.		
7	Verify the recovery run is completed. Verify that new hidden directories have		

#	Action	Expected Result	Notes
	been generated for all collections in the group, that all hidden files for the collection group have been copied to the new hidden directories, that all FTPPull links for granules in these collections now point to the new hidden directories, and that all old hidden directories for collections in the group have been removed.		

TEST DATA:

EXPECTED RESULTS:

16.39 Test Case 53 - DPL XML Check Utility (ECS-ECSTC-243)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[DPL XML Check Utility]		#comment
2	Note: Refer to the DPL XML Check Utility 609 document for detailed instructions.		
3	Usage of EcDlXcu.pl:		
4	EcDlXcu.pl <MODE> [-days <NUMBER OF DAYS>] [-percent <PERCENT 1-100>] (-ESDT <ESDTS ex. MOD29.005> [-startdate <STARTDATE ex: Jan 1 2008> [-enddate <ENDDATE ex: Dec 25 2008>]] | -granuleid <GRANULEIDS> | -file <FILENAME>) [-outputDir <DIRECTORY>]		
5	Run XCU to check the well-formedness of all the xml files or only the specified xml files (by using the optional parameters) in the Data Pool Online Archive of the mode.		
6	Verify that the utility reports the validation results and list the xml files which are not well-formed.		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.40 DP_81_01_TP055 DPL Cleanup with input file (ECS-ECSTC-342)

DESCRIPTION:
DPL Cleanup with input file

PRECONDITIONS:
ingest granules

STEPS:

#	Action	Expected Result	Notes
1	[DPL Cleanup with input file] Create 2 input files for the Data Pool Cleanup utility containing granules that correspond to the following conditions: a. At least 2 hidden ECS sciences granule in a directory which contains no other granules b. At least 2 public ECS science granules that are the last remaining granules in their directories c. At least 2 public ECS granules that are in directories containing other granules d. At least 2 public ECS granules containing browse links in the Data Pool And one file containing: e. f. At least 2 non-ECS granules that are the last 2 granules within a directory g. At least 2 non-ECS granules that are not the last 2 granules within a directory. h. At least 2 ECS granules that are part of a current order. One file should contain the ECS granule cases and the other should contain the non-ECS granule cases. Run a Data Pool cleanup utility to process each input file.	a) The science are stored its own directory corresponding to the RangeBeginningDate column in AmGranule table use the Unpublish utility to unpublish granules a. b) The science files of step b and c should be stored in same directory corresponding to the RangeBeginningDate column in AmGranule table c) ingest granules. d) Ingest granules then use the sql select g.ShortName, g.VersionId, convert(varchar(10), g.GranuleId) granuleId, g.IsOrderOnly, convert(varchar(10), b.BrowseId) BrowseId from AmGranule g, AmBrowseDataFile b, AmBrowseGranuleXref x where g.GranuleId = x.GranuleId and x.BrowseId = b.BrowseId and g.GranuleId in (granuleid in step d) e)	

#	Action	Expected Result	Notes
		Ingest granules and order those granules. Run a Data Pool cleanup utility to process each input file Create an input file calls inputfile from GranuleId in table above. EcDlCleanupGranules.pl <MODE> -file inputfile 	
2	Verify that all granule files and browse links, except those that were on order (S1-e), were removed from the Data Pool disks.	Go to the public directory to make sure the granule files and browse links were removed except those that were on order (S1-e). /datapool/<MODE>/user/FileSystemLabel/GroupId/SubTye/RangeBeginingDate/ /datapool/DEV02/user/FS2/MOLT/MOD14.005/2011.01.01/	
3	Verify that the directories related to the removed ECS granules were not removed from the Data Pool	Make sure that directories related to the removed ECS granules were not removed from the Data Pool. See V-1 directory.	
4	Verify the AIM Inventory Catalog entries for all of the removed ECS granules now indicate the granules are in the “hidden” Data Pool and that the “Warehouse metadata” for the public granules was removed. (Note: these ECS granules now become “phantoms”)	Select ShortName, VersionId, convert(varchar(10), GranuleId) granuleId, IsOrderOnly, PublishTime, IsOrderOnly from AmGranule where GranuleId in (GranuleIds in step b,c, d, e, g, and h) make sure IsOrderOnly is “H” go to the public directory to make sure there are no files, these files should be in hidden directories.	
5	Verify that the files that were removed were logged and that the granules that were skipped because they are on order (S-1e) was logged. For granules that were skipped the total number of granules as well as their total size should be logged.	Open the EcDlCleanupGranules.log to make sure that the files that were removed were logged and that the granules that were skipped because they are on order (S-1e) was logged.	
6	Verify that the total number of files that were cleaned up is logged and is correct.	Verify the 2 non-ECS granules (S1-f) were removed along with the	

#	Action	Expected Result	Notes
		directory.	
7	Verify the 2 non-ECS granules (S1-g) were removed but the directory is still present and contains the non-ECS granule(s) that were present prior to running the cleanup utility and were not part of the cleanup request.	Verify the 2 step on-ECS granules (S1-g) were removed but the directory is still present and contains the non-ECS granule(s) that were present prior to running the cleanup utility and were not part of the clean up request.	

TEST DATA:

EXPECTED RESULTS:

16.41 DP_81_01_TP056 DPL Cleanup on non-ECS granules (ECS-ECSTC-343)

DESCRIPTION:
DPL Cleanup on non-ECS granules

PRECONDITIONS:
Ingest non-ecs granules.

STEPS:

#	Action	Expected Result	Notes
1	[DPL Cleanup on non-ECS granules] Ensure that the Data Pool contains the following granule conditions: a. At least 2 non-ECS granules whose expiration time is prior to 8 hours ago from the previous day b. At least 2 non-ECS granules whose expiration dates are in the future. c. At least 2 non-ECS granules whose expiration time is between midnight of the previous day minus 8 hours and midnight of the previous day. Run a Data Pool cleanup utility with options to remove expired granules with an offset of 8 hours. 	Ingest the pdrs in the test requirement above. EcDlBatchInsertPublishUtilityStart.pl <MODE> -nonecs -file filename Note: expirationDate is Apr 6 2011 1:15:44:683PM" a) Select * from DlGranuleExpirationPriority Where granuleId in(<granuleId in a> EcDlUpdateGranule.pl DEV02 <MODE> -grnid 3002000306 GranuleId -exp "2011/05/16" -ret 225<br	

#	Action	Expected Result	Notes
		/>Or use sql command to update update DlGranuleExpirationPriority set expirationDate = “Apr 5 2011 3:00:44:683PM " b) update DlGranuleExpirationPriority set expirationDate = “Apr 6 2011 20:00:44:683PM " Select * from DlGranuleExpirationPriority Where granuleId in(<granuleId in b>) Make sure the expiration dates are in the future. c) update DlGranuleExpirationPriority set expirationDate = “Apr 5 2011 20:00:44:683PM " EcDlCleanupGranules.pl <mode > -offset 8 	
2	Verify that all expired non-ECS granules that qualify to be removed (S1-a) were deleted from the Data Pool disks and that the DataPool metadata (including the WebAccess Warehouse table entries) and AIM Inventory Catalog entries were removed.	Select * from AmGranule Where GranuleId = <GranuleId in S1-a> Make sure zero rows return. select g.ShortName, g.VersionId, g.granuleId from DlGranulesAmGranule g, DlFactDayNight fn, DlFactGroupESDT fa where g.granuleId = fn.granuleId and g.granuleId = fa.granuleId and g.granuleId = < GranuleId in S1-a > group by g.ShortName, g.VersionId, g.granuleId make sure zero rows return 	
3	Verify that the non-ECS granules that did not qualify for deletion (S1-a, S1-c) where not removed from the Data Pool disks or AIM Inventory Catalog.	Select * from AmGranule Where GranuleId = <GranuleId in S1-a> Make sure zero rows return. select	

#	Action	Expected Result	Notes
		g.ShortName, g.VersionId, g.granuleId from AmGranule g, DlFactDayNight fn, DlFactGroupESDT fa where g.GranuleId = fn.granuleId and g.GranuleId = fa.granuleId and g.GranuleId = < GranuleId in S1-a > group by g.ShortName, g.VersionId, g.granuleId make there are rows return 	
4	Verify that the non-ECS granules that did not qualify for deletion (S1-a, S1-c) where not removed from the Data Pool disks or AIM Inventory Catalog.	Select * from AmGranule Where GranuleId = <GranuleId in S1-a> Make sure zero rows return. select g.ShortName, g.VersionId, g.granuleId from AmGranule g, DlFactDayNight fn, DlFactGroupESDT fa where g.GranuleId = fn.granuleId and g.GranuleId = fa.granuleId and g.GranuleId = < GranuleId in S1-a > group by g.ShortName, g.VersionId, g.granuleId make there are rows return 	
5	Verify that the files that were removed were logged.	Open the EcDlCleanupGranule.log to verify that the files that were removed were logged.	
6	Verify that the total number of files that were cleaned up is logged and is correct.	Open the EcDlCleanupGranule.log to verify that the total number of files that were cleaned up is logged and is correct in S1-a.	
7	Verify that the total amount of disk space that was cleaned up is logged and is correct.	Open the EcDlCleanupGranule.log to verify that the total amount of disk space that was cleaned up is logged and is correct. E.g Deleted 6 file occupying 12820346 bytes. 	

TEST DATA:

EXPECTED RESULTS:

16.42 DPL Modified

16.42.1 Data Pool Maintenance GUI - Check Batch Insert Status (ECS-ECSTC-205)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Data Pool Maintenance GUI – Check Batch Insert Status]		#comment
2	Use the Data Pool Maintenance GUI to check the status of the batch insert (by Publish Utility) using the Batch Summary tab and also using the List Insert Queue tab (filter by batch label).		
3	Verify that the GUI correctly reports the status in the DlInsertActionQueue.		

TEST DATA:
Any current data type

EXPECTED RESULTS:

16.42.2 Publish Utility, DP_7F_01, Criterion 500 (ECS-ECSTC-224)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Publish Utility, DP_7F_01, Criterion 500]		#comment
2	Use Publish Utility to insert at least two granules for each of the following		

#	Action	Expected Result	Notes
	conditions; all granules shall be on tapes that are currently resident in the tape silo unless explicitly noted otherwise: a. granule is not in the AIM inventory. b. granule is flagged deleted from archive in the AIM inventory (DeleteFromArchive set to ‘Y’). c. granule is hidden in the AIM inventory (DeleteFromArchive set to ‘H’) d. granule is flagged as logically deleted in the AIM inventory (deleteEffectiveDate is not NULL). e. granule is not flagged as logically deleted, deleted from archive, or hidden in the AIM inventory and belongs to a collection configured for publishing during ingest. f. granule is not flagged as logically deleted, deleted from archive, or hidden in the AIM inventory and does not belong to a collection configured for publishing during ingest. g. granule is not flagged as logically deleted, deleted from archive, or hidden in the AIM inventory, belongs to a collection configured for granule replacement and publishing during ingest, and will trigger the granule replacement logic but is not eligible to replace the existing public granule. h. granule is not flagged as logically deleted, deleted from archive, or hidden in the AIM inventory, belongs to a collection configured for granule replacement and publishing during ingest, and will trigger the granule replacement logic and is eligible to replace the existing public granule. j. granule belongs to a collection that resides on a file system that is currently flagged as unavailable.(This is obsolete because all the ECS granules are in hidden DPL since 8.1) l. granule is flagged as golden in the AIM inventory (DeleteFromArchive set to ‘G’) and belongs to a collection configured for granule publication during ingest. n. hidden browse granule is flagged as logically deleted in the AIM inventory (deleteEffectiveDate is not NULL) o. hidden browse granule is not flagged as logically deleted		
3	Verify that the Data Pool Publish Utility rejects the granules referenced in a, b,c,d, f		
4	Verify that the Data Pool Publish Utility logs the rejected granules with the reason for the rejection.		
5	Verify that the Data Pool Publish Utility queued insert actions with the Data Pool Insert Service for the granules referenced in e, g, h, l, n,o		
6	Verify that the granules referenced in e, h, l, o were inserted into the public Data Pool.		
7	Verify that the granules referenced in g, n remain in the hidden Data Pool.		
8	Verify that the Data Pool database contains the correct status detail for each of the insert actions. 		

#	Action	Expected Result	Notes
9	Make the Data Pool file system available into which the granule referenced in j needs to be inserted, and verify that the insert of the granule into the Data Pool completes successfully. 		

TEST DATA:
Any current data types

EXPECTED RESULTS:

16.42.3 DataPool Maintenance GUI - QualitySummary URL should allow secure URL (ECS-ECSTC-495)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	DataPool Maintenance GUI - update the QualitySummary URL field using secure URL		
2	Using the DPL Maintenance GUI to update a collection's QualitySummary URL field		
3	Verify that a secure URL starting with https:// is allowed.		

TEST DATA:

EXPECTED RESULTS:

16.42.4 Inventory Validation(Orphan, Phantom and Link Checking) (CleanupFilesOnDisk(2)) (NCR 8051022, 8052031, 8052056) (ECS-ECSTC-663)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	NCR8051022: Identify the following granules: a. public SC g1(isorderonly is null), with a file missing on disk (remove the file) b. hidden SC g2(isorderonly = 'H'), with a file missing on disk. c. public on order SC g3(isorderonly = 'B'), with a public file missing on disk d. public browse b1(isorderonly is null), with a file missing on disk e. hidden browse b2(isorderonly = 'H'), with a file missing on disk f. public SC g4(isorderonly is null), with a hidden browse b3(unpublish the browse if necessary). g. public on order SC g5(isorderonly = 'B'), with a hidden orderlink missing on disk		
2	Run the cleanupFilesOnDisk with a new -rmTran option. a. This will sleep 30 seconds after all the collections are processed and query the database one last time to see if the phantom granules identified have changed the state. if so, don't report. If not, continue to step b. b. get all the associated public browse links from db one more time, if the new files returned are in the identified phantom browselinks at the end of step a, or the new files don't exit on disk, then they will be the phantom browse links that will be reported. c. Any transient phantoms not reported will be logged with the keyward "Transient phantom removed".		
3	When you see "Sleeping 30 seconds to eliminate transient phantoms before querying database one last time prior to reporting" on the screen at step 2 a. Finish running the following within the 30 second window: update AmGranule set isorderonly = 'H' where granuleid = ; update AmGranule set isorderonly = null where granuleid = ; update AmGranule set isorderonly = null where granuleid = ; update AmBrowse set isorderonly = 'H' where browseid = ; update AmBrowse set isorderonly = null where browseid = ;		

#	Action	Expected Result	Notes
	update AmBrowse set isorderonly = null where browseid = ; This will change all the granule's states in step 1, except for step g, to simulate the transient situation.		
4	Verify that the g1,g2, g3,g4, b1,b2 are all logged as transient phantoms and not reported in the output files.		
5	Verify that the g1,g2, g3,g4, b1,b2 are all logged as transient phantoms and not reported in the output files.		
6	NCR8052031: run Mkcfg for ESIR and RQS and make sure the STITCH_DIRECTORY and DEFAULT_OUTPUT_DIR paths are equal to //stitch and //glas respectively. They should not include the "temp" dir in the path.		
7	NCR8052056: Run the EcDlCleanupFilesOnDisk.pl utility with the -fix option.		
8	Verify in the EcDlCleanupFilesOnDisk.log that there exists a line: Executing : find /datapool//user/pub/ -maxdepth 1 -name "DPRecentInserts_*" -type f -ctime +7 -delete -printf "%p %s\n" If the line is missing, verify that the directory /datapoo//user/pub/ directory doesn't exist in the box.		

TEST DATA:

EXPECTED RESULTS:

16.43 [OBSOLETE] - Test Case 1 - Qualified Subscription Insert (ECS-ECSTC-905)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Qualified Subscription Insert] Insert a granule with browse into ECS for which there is an existing qualified Data Pool insert subscription.		

#	Action	Expected Result	Notes
2	Verify that the granule and its browse file were successfully inserted into the Data Pool by checking the status of the insert action in the Data Pool database (DlInsertActionQueue).		
3	Verify, using isql, that the granule was inserted into the DlGranules table and the browse granule was inserted into the DlBrowse table.		
4	Verify, using UNIX ‘cd’ and ‘ls’ commands, that the files for the granule were inserted into the appropriate Data Pool directory and that the browse files were inserted into the appropriate Data Pool directory, and that a link to the browse file is present in the science file directory.		

TEST DATA:

EXPECTED RESULTS:

16.44 [OBSOLETE] - Test Case 6 - Download Granules via Web (ECS-ECSTC-906)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[Download Granules via Web] Verify that anonymous ftp is configured for the mode in which you are testing.		
2	From outside of the firewall, use the Data Access Testbed to download the granules inserted in Test Cases 1 and 3.		
3	Verify that the granules are correctly downloaded to the specified download directory.		

TEST DATA:

EXPECTED RESULTS:

16.45 [OBSOLETE] - Test Case 19 - HEG Order from Web Access (ECS-ECSTC-915)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	[HEG Order from Web Access] Insert an AST_L1B granule into the Data Pool, using the Batch Insert utility.		
2	Use the Data Pool Web Access GUI to drill down for this granule.		
3	Perform a HEG order for this granule.		
4	Verify that the band subsetting icon appears for this granule on the HEG order page (although you do not need to select it).		
5	Verify that the HEG order completes successfully, and that the HEG output files are placed in the /datapool//user//.orderdata/OUTPUTS/HEGOUT.001/HEG/. directory.		
6	Verify that the order status appears on the OMS GUI.		

TEST DATA:

EXPECTED RESULTS:

17 LONG SHORTNAME AND POSTGRES 9.3 (ALL UTILITIES)

17.1 ESDT Installation (ECS-ECSTC-541)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Install the test collection ESDT. Place the ESDT descriptor files under /usr/ecs/OPS/CUSTOM/data/ESS and select Install	Verify that the ESDT was successfully installed in the ESDT Maintenance GUI	

#	Action	Expected Result	Notes
2	Add the collection to a collection group in the DataPool Maintenance GUI	Verify that the collection is successfully added in the DPM GUI	
3	On the Ingest GUI, add volume group for the collection	Verify that the amcollection table has the new collection and group	
4	Configure the collection to be public on Ingest		
5	Restart Ingest and ActionDriver in the mode		
6	Run EcBmConfigureCollection.pl to add the collection to the bmgt table or update the bg_collection_configuration manually with the collection information from amcollection	Verify that the bg_collection_configuration table has an entry for the collection added	

TEST DATA:

EXPECTED RESULTS:

17.2 Ingest a Granule (ECS-ECSTC-530)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that you have a provider configured for the test collection and that it is configured correctly (SIPs or NON-SIPS)		
2	Ensure a polling location is defined for the provider		
3	Place the PDR file in the polling location for the provider for the test collection	The granule should be ingested successfully on Ingest GUI.	
4	Verify the granule is stored correctly in the AIM database.	select registrationtime, archivetime from AmGranule where granuleId = < granuleId>; Both columns should contain non null values.	
5	Verify the granule files are stored in the Archive.	The granule should be stored under /stornext/snfs1/<mode>/*/	
6	Verify the granule files are stored int the Data Pool.	select ProcGetGrFiles to find the	

#	Action	Expected Result	Notes
		datapool location of the granule. cd to the directory verify that the file should be there.	

TEST DATA:
1 granule with a file size greater than 2 GB.

EXPECTED RESULTS:

17.3 BMGT Nominal Exports (ECS-ECSTC-545)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure a test granule is ingested for the collection and enabled for BMGT export		
2	Do a manual export for the test collection. ./EcBmBMGTManualStart MODE -metc -c snvi	Verify that the collection exported successfully	
3	Do a manual export of the granule. ./EcBmBMGTManualExport MODE -metg -c snvi	Verify that the granules for the collection exported successfully	
4	Do an automatic export. Update the daynightflag of the test granule.	Verify that the granule is exported for the collection	
5	Do a longform verification. ./EcBmBMGTManualStart MODE --long -metc -c snvi	Verify that the verification export is successful	
6			

TEST DATA:

EXPECTED RESULTS:

17.4 Order a granule (ECS-ECSTC-531)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Order a granule from an ESDT with a long short name by an FTP Pull request using an OMSCLI client		
2	Verify that the order is queued and shipped successfully		
3	Order a granule with the long shortname ESDT by an FTP Push request using an OMSCLI client		
4	Verify that the order is queued adn shipped successfully		
5	Order a granule with the long shortname ESDT by an SCP request usign an OMSCLI client		
6	Verify that the order is shipped successfully		
7	Repeat steps 1-6 with a EWOC client		
8	Repeat steps 1-6 with the SSS subscription GUI		

TEST DATA:

EXPECTED RESULTS:

17.5 Unpublish a granule (ECS-ECSTC-533)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Locate a granule for the test ESDT that is public. select granuleid, shortname, versionid, isorderonly,getdatafile(granuleid) from amgranule where length(shortname) > 8 and isorderonly is null;	Verify that the isorderonly flag is null in amgranule table	

#	Action	Expected Result	Notes
2	Ensure that the granule files are in the public datapool. Record the file size in public datapool. select getdatafile(granuleid);		
3	Unpublish the granule. EcDlUnpublishStart.pl -mode -g granuleid	Verify that there is no error in the Unpublish utility log.	
4	Check the amgranule table	Verify that the IsOrderOnly is 'H' for the granule.	
5	Check the granule files in the hidden datapool. select getdatafile(granuleid);	cd to the hidden directory, verify the granule file is there and it's the same size as the one in the public and the one in the archive (under /stornext)	

TEST DATA:

EXPECTED RESULTS:

17.6 Publish a granule (ECS-ECSTC-534)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Identify a granule that is hidden for the test collection	Verify that the isorderonly flag is 'H' in amgranule	
2	Ensure that the granule is in hidden datapool. Record the filesize and location.		
3	Publish the granule. EcDlPublishUtilityStart MODE -ecs -g granuleid	Verify that is no error in the PublishUtility log	
4	Check the DlInsertActionQueue	Verify that the status in DlInsertActionQueue is COMPLETE for the granule;	
5	Check the amgranule table	Verify that the isorderonly flag is set to null	
6	Check the public datapool for the granule. select ProcGetGrFiles(shortname,	Verify that the file has the same size	

#	Action	Expected Result	Notes
	granuleid) to get the public data pool location	as the one in the hidden and the one in the archive (under /stornext)	

TEST DATA:

EXPECTED RESULTS:

17.7 Duplicate Granule (ECS-ECSTC-542)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ingest a granule g1a for the test collection	Verify that the ingest is successful	
2	Turn on dupgran switch on		
3	Ingest the same PDR for granule g1b	Verify that the ingest is successful	
4	Check the amgranule table for both the old g1a and new granule g1b	Verify that the old granule is replaced	
5	Check the amgranulereplacement table	Verify that the replacement is recorded in the amgranulereplacement table	
6	Turn off dup gran switch		
7	Ingest a different granule g2a		
8	Ingest this granule again g2b		
9	Check the amgranule and amgranulereplacement tables for granules g2a and g2b	Verify that the granule is not replaced. There is no entry in the amgranulereplacement table	
10	Run the script EcDsAmIdentifyDuplicateGranules.pl		
11	Check the amgranulereplacement table	Verify that the granule g2a is replaced by g2b	
12	Duplicate Granule Reporting		

TEST DATA:

EXPECTED RESULTS:

17.8 RestoreOLAFromTape (ECS-ECSTC-535)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure a test granule is ingested for the test collection	Verify the location of the file in the datapool	
2	Modify the granule file in the datapool, Run EcDlRestoreOlaFromTapeStart to restore it from archive. EcDlRestoreOlaFromTapeStart MODE -file granuleidsfile -contents granuleids -norecovery	Verify that there are no errors in the RestoreOLAFromTape log.	
3	Check the files for the granule in the datapool.	Verify that the files in the datapool are same as the file in archive under /stornext.	

TEST DATA:

EXPECTED RESULTS:

17.9 RestoreTapeFromOLA (ECS-ECSTC-536)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure you have a test granule ingested for the test collection		
2	Modify the granule file in the archive. Run EcDlRestoreTapeFromOlaStart	Verify that there are no errors in the	

#	Action	Expected Result	Notes
	to restore it from archive. EcDlRestoreTapeFromOlaStart MODE -file granuleidfile -contents granuleids -norecovery	the RestoreTapeFromOla logs	
3	Check the files in the archive and datapool.	Verify that the files are the same as the ones in datapool.	

TEST DATA:

EXPECTED RESULTS:

17.10 DPCV (ECS-ECSTC-537)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that a granule is ingested for the test collection	Verify that the test granule exists in the datapool	
2	Use DPCV to checksum the granule: EcDlDpcvStart MODE -file granuleidfile	Verify that ther is no error on the command line.	
3	Check DPCV log.	Verify that there is no error in the DPCV log.	

TEST DATA:

EXPECTED RESULTS:

17.11 ACVU (ECS-ECSTC-539)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that a granule is ingested for the test collection	Verify that the granule files exist in the datapool	
2	Use ACVU to checksum the granule: EcDsAmAcvu.pl MODE -granuleid 62863	Verify that there is no error on the command line.	
3	Check the ACVU log	Verify that there is no error in the log.	

TEST DATA:

EXPECTED RESULTS:

17.12 Data Access Nominal Test (ECS-ECSTC-540)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that a granule is ingested for the test collection and has been exported to echo		#comment
2	Ensure that the data access webservices are up and running in the mode		
3	Run GdalMatrix to configure formats for the test collection. ./EcDlDaGdalColFmtCfgStart --mode OPS -p DBPORT -s DBHOST --singleCollection snvi	Verify that there are no errors in the matrix.log	
4	On the DataAccess GUI. run the configuration for the dataObjects. http://f5dpl01v:22500/DataAccessGui/config/datasetService/dataObjects/GDAL/snvi	Verify that there are no errors in the DA_Gui.log	
5	On the DataAccessGUI, Enable the collection for GDAL Service and HEG Service and update service for the collection		
6	On the ESI Inventory Drilldown, check GranuleInfo and DataObjects for the test granule	Verify that the granule information is displayed without errors	

#	Action	Expected Result	Notes
7	Order the granule with HEG service	Verify that the request is successful	
8	Order the granule with GDAL service	Verify that the request is successful	

TEST DATA:

EXPECTED RESULTS:

17.13 Move Collection (ECS-ECSTC-546)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test granule for the collection is ingested in the mode	Verify that the granule files exist in the datapool directory	
2	Run the move collection utility to move the collection from one filesystem to another. ./EcDlMoveCollection.pl MODE -shortname SHORTNAME -versionid VERSIONID -sourcefs FS1 -targetfs FS2 -verbose	Verify that there are no errors reported on the command line	
3	Check the logs	Verify that there are no errors in the MoveCollectionUtility log	
4	Check the datapool filesystem for the files for the test collection	Verify that all the files for the collection and its granules were successfully moved from FS1 to FS2	

TEST DATA:

EXPECTED RESULTS:

17.14 Remap Collection Group (ECS-ECSTC-547)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Update the Insert Enabled to OFF in the DPM GUI for the testcollection. or update allowpublishflag in amcollection to a 'N' in the database		
2	Run the remap collection group utility. ./EcDlRemap.pl MODE -esdt SOURCE_COLLECTION_NAME -version SOURCE_COLLECTION_VERSION -oldgrp SOURCE_COLLECTION_GROUP -newgrp DESTINATION_GROUP	Verify that the datapool directory has the collection moved to the directory of the destination group	

TEST DATA:

EXPECTED RESULTS:

17.15 QAUU (ECS-ECSTC-548)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that a granule is ingested for the test collection		
2	Create a QA request file to update the Science QA flags for the ESDT based on the format instructions specified in the 609		
3	Run the QAUU utility. ./EcDsAmQAUUStart MODE -file QAREQUESTFILE	Check the log file to verify that the request was successful	
4	Check the metdata exported for the QA flags for the collection	Verify that the QA flags are updated as requested in the request file	
5	Create a QA request file to update the Operational QA flags for the ESDT		
6	Run the QAUU utility with the new request file. ./EcDsAmQAUUStart	Check the log file to verify that the	

#	Action	Expected Result	Notes
	MODE -file QAREQUESTFILE2	request was successful	
7	Check the metadata exported for the Operational QA flags for the collection	Verify that the QA flags are updated as requested in the request file	
8	Repeat steps 2-7 for a request specifying the localgranuleid		
9	Repeat steps 2-7 for a request specifying the GranuleUR		

TEST DATA:

EXPECTED RESULTS:

17.16 XCU (ECS-ECSTC-549)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test granules are installed for the collection		
2	Run XML Check Utility for the test collection. EcDlXcu.pl MODE -ESDT snvi	Verify that there are no errors in the XCU log file	

TEST DATA:

EXPECTED RESULTS:

17.17 XRU (ECS-ECSTC-550)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the test granule for the collection is ingested		
2	Create the replacement xml file as specified in the 609		
3	Run the XML replacement utility. ./EcDsAmXruStart MODE -xmlfile INPUTFILE	Verify that there are no errors displayed after the run and no errors in the log file	
4	Check the exported granule xml file	Verify that the changes in the xml file for the granule were successfully exported.	

TEST DATA:

EXPECTED RESULTS:

17.18 EMS long shortname End to End test (ECS-ECSTC-551)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Install an ESDT with a ShortName longer than 8 characters		
2	Ingest 6 granules for the ESDT that was installed.		
3	Order granule 1 via scp through OMS		
4	Order granule 2 via FtpPush through OMS		
5	Order granule 3 via FtpPull through OMS		
6	Order granule 4 via HTTP through DataAccess		
7	Run EcDlBulkDelete with -dfa flag to DFA granule 5		
8	Use FTP to download granule 6 from DataPool		
9	Run EcDlWuFtpRollup.pl		
10	Run EcDlWuHttpRollup.pl		
11	Run EcDbEMSdataExtractor.pl	Verify the ESDT shows up in the relevant EMS Ingest and Search	

#	Action	Expected Result	Notes
		Expression reports. Verify granules 1-6 show up in the EMS Archive report. Verify granules 1-3 show up in the relevant EMS OMS Dist reports. Verify that granules 4 and 6 show up in the relevant EMS DPL reports. Verify granule 5 shows up in the EMS DFA report.	

TEST DATA:

EXPECTED RESULTS:

17.19 Map Granule Generation (ECS-ECSTC-552)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the granules are ingested for the test collection		
2	Run the map generation utility to generate the map files for the test granule. ./EcAmInsertMapGenerationRequest.pl -mode MODE -g GRANULEID	Verify that there are no errors in the log and the map output granules are generated in the output directory	

TEST DATA:

EXPECTED RESULTS:

17.20 Cleanup Files on Disk (ECS-ECSTC-553)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Run the CleanupFilesOnDisk for the test collection group. ./EcDlCleanupFilesOnDisk MODE -collgroup TESTCOLLECTIONGROUP	Verify the log to check the results of the run.	
2	Check the output files under /usr/ecs/MODE/CUSTOM/temp/DPL		
3	Run the utility with the fix option. /EcDlCleanupFilesOnDisk MODE -collgroup TESTCOLLECTIONGROUP -fix	Verify the log for any errors in the run	

TEST DATA:

EXPECTED RESULTS:

17.21 Cleanup Granules (ECS-ECSTC-554)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Run ./EcDlCleanupGranules.pl MODE -grans testgranuleid		Verify that there are no errors in the log

TEST DATA:

EXPECTED RESULTS:

17.22 Inventory Validation Tool (ECS-ECSTC-557)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the granules are ingested for the test collection		
2	Run the Inventory Validation tool. ./EcDlInventoryValidationTool.pl MODE	Verify that the output files for all the discrepancies are created in the default output directory defined in VALIDATION_OUTPUT_DIR in the config file	
3	Turn the dup gran switch off.		
4	Ingest the same granule with long short name a few times.	Verify the later ingested granule is replacing the earlier ones by moving the older granules into the hidden datapool.	
5	Unpublish the latest granule.	Verify that the granule is unpublished	
6	Run IVT tool again. ./EcDlInventoryValidationTool.pl MODE	Verify the last granule is identified in the output directory in the InventoryDiscrp_should_be_public_granuleids_RepairByPublish...	
7	Turn the dup gran switch on		
8	Repeat steps 4-6		

TEST DATA:

EXPECTED RESULTS:

17.23 Datapool Hidden Scrambler Utility (ECS-ECSTC-559)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure test granules are ingested for test collection		
2	Run the Hidden Scrambler Utility for the test collection . ./EcDlHiddenScrambler.pl MODE -shortname SHORTNAME -versionid VERSIONID	Verify that the test collection was renamed and reencrypted with no errors	

TEST DATA:

EXPECTED RESULTS:

17.24 Search and Delete Utilities (ECS-ECSTC-560)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure that the granule for the test collection is ingested		
2	./EcDsBulkSearch.pl -mode MODE -name SHORTNAME -version VERSIONID -geoidfile GEOIDS_FILE	Verify that all the granules in the test collection with the geoids are listed in the output file	
3	Edit the GEOIDS_FILE to include the just the test granule to delete. Run the Bulk Delete utility. ./EcDsBulkDelete.pl -mode MODE -dfa -geoidfile GEOIDS_FILE	Verify that the granule to be deleted is marked for deletion. deletefromarchive in amgranule should not be null	
4	Undelete the granule. /EcDsBulkUnDelete.pl -mode MODE -dfa -geoidfile GEOIDS_FILE	Verify that the granule marked for deletion is undeleted. deletefromarchive in amgranule should be null	
5	Logically delete the test granule. ./EcDsBulkDelete.pl -mode MODE -physical -geoidfile GEOIDS_FILE	Verify that the granule is marked for deletion. Verify that the deleteeffective date for the test granule	

#	Action	Expected Result	Notes
		is not null in amgranule	
6	Unpublish the test granule ./EcDlUnPublishUtility.pl DEV08 -ecs -g granule_to_unpublish	Verify that the isorderonly flag in amgranule is set to 'H' in amgranule	
7	./EcDsDeletionCleanup.pl -mode DEV08	Verify that the granule does not exist in the amgranule table	

TEST DATA:

EXPECTED RESULTS:

17.25 Delete Collection (ECS-ECSTC-561)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Delete the granules in the test collection		
2	Remove the collection from the collection group in Datapool Maintenance GUI		
3	Remove the volume group for the collection from the Ingest GUI		
4	Delete the collection from the ESDT Maintence GUI	Verify that the collection was successfully deleted	

TEST DATA:

EXPECTED RESULTS:

18 BIG GRANULES

18.1 Ingest a Large Granule (ECS-ECSTC-347)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ingest a granule whose science file is larger than 2 GB. /sotestdata/DROP_802/NCR8050675/MOD09CMG.005 contains a big granule PDR. copy it to a pulling directory, make sure ingest processing and pulling are running.	The big granule should be ingested successfully on Ingest GUI.	
2	Verify the granule is stored correctly in the AIM database.	select registrationtime, archivetime from AmGranule where granuleId = < granuleId>; Both columns should contain non null values.	
3	Verify the granule files are stored in the Archive.	The granule should be stored under /stornext/snfs1/<mode>/*/, with a file name like :SC:MOD09CMG.005:62863:1.HDF	
4	Verify the granule files are stored int the Data Pool.	select ProcGetGrFiles to find the datapool location of the granule. cd to the directory and the big file should be there.	

TEST DATA:
1 granule with a file size greater than 2 GB.

EXPECTED RESULTS:

18.2 Order a large granule (ECS-ECSTC-349)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	2) Order the granule using Pull, Push, and SCP Verify the distributed granule has the same checksum as the ingested granule.		#comment
2	Find or ingest a big granule with a file size > 2GB, in a mode that's configured in the config above. Or simply copy a big file to one of the subdirectories.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user. 	
3	Locate the string 5 box where EWOC is running. Order the big granule ingested through submitting the following EWOC client requests: 1. ftp push Example: cd /home/sxu/TESTINGS/HowTo/EWOC Client_dev09 PlainPushRequest_dev09 2. ftp pull Example: cd /home/sxu/TESTINGS/HowTo/EWOC Client_dev09 PlainPullRequest_dev09 3. scp Example: cd /home/sxu/TESTINGS/HowTo/EWOC Client_dev09 PlainScpRequest_dev09 Note: for scp, we have to configure the scp destination on OMS GUI and use the directory configured in the request. 	All the 3 orders are submitted successfully on the commandline.	
4	Verify ftp push is successful and the distributed granule has the same checksum as the ingested granule.	Login to OMS GUI, verify the ftp push request is "shipped". cd to the ftp push directory, verify the big granule file(s) are there. run md5sum on the big file and verify the result is the same as the one on the ingested file in datapool.	
5	Verify ftp pull is successful and the distributed granule has the same checksum as the ingested granule.	Login to OMS GUI, verify the ftp pull request is "shipped". Verify DN, distribution notification is sent to the email configured in the request. Follow the instruction in the DN and get the big file. run md5sum on the big file and verify the result is the same as the one on the ingested file in datapool.	
6	Verify Scp is successful and the distributed granule has the same checksum as the ingested granule.	Login to OMS GUI, verify the Scp request is "shipped". cd to the Scp directory, verify the big	

#	Action	Expected Result	Notes
		granule file(s) are there. run md5sum on the big file and verify the result is the same as the one on the ingested file in datapool.	

TEST DATA:

EXPECTED RESULTS:

18.3 RestoreOLAFromTape (ECS-ECSTC-350)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	3) RestoreOLAFromTape Verify granule in the DataPool is the same as what was ingested		#comment
2	Find or ingest a big granule with file size > 2GB.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
3	Modify the big file in the datapool, run EcDlRestoreOlaFromTapeStart to restore it from archive EcDlRestoreOlaFromTapeStart DEV09 -file ./sxuidfile -contents granuleids -norecovery	No error in the log	
4	Verify the files in datapool are successfully restored.	They are the same as the ones in archive, under /stornext.	

TEST DATA:

EXPECTED RESULTS:

18.4 RestoreTapeFromOLA (ECS-ECSTC-351)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	4) RestoreTapeFromOLA Verify granule in the Archive is the same as what was ingested		#comment
2	Find or ingest a big granule with file size > 2GB.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
3	Modify the big file in the archive, run EcDlRestoreTapeFromOlaStart to restore it from archive EcDlRestoreTapeFromOlaStart DEV09 -file ./sxuidfile -contents granuleids -norecovery	No error in the log	
4	Verify the files in archive are successfully restored.	They are the same as the ones in datapool.	

TEST DATA:

EXPECTED RESULTS:

18.5 Unpublish a large granule (ECS-ECSTC-352)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	5) Unpublish Verify the granule file is moved to the hidden DataPool and has not changed		#comment
2	Find or ingest a big granule with file size > 2GB.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	

#	Action	Expected Result	Notes
3	Make sure the granule is in public datapool (publish it first if it's in hidden datapool already) Record the file size in public datapool. Unpublish the granule by running, for example, EcDlUnpublishUtilityStart DEV09 -ecs -g 62863	There is no error in the Unpublishing utility log; IsOrderOnly is 'H' for the granule.	
4	Verify the granule has not changed.	select ProcGetGrFiles('<shortname>', <granuleid>); cd to the hidden directory, verify the granule file is there and it's the same size as the one in the public and the one in the archive (under /stornext)	

TEST DATA:

EXPECTED RESULTS:

18.6 Publish a large granule (ECS-ECSTC-353)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	6) Publish Verify the granule is in the public data pool and has not changed		#comment
2	Find or ingest a big granule with file size > 2GB.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
3	Make sure the granule is in hidden datapool (unpublish it first if it's in public datapool already) Record the file size in hidden datapool. Publish the granule by running, for example, EcDlPublishUtilityStart DEV09 -ecs -g 62863	There is no error in the publishing utility log; The status in DlInsertActionQueue is COMPLETE for the granule; IsOrderOnly is null for the granule.	
4	Verify the granule has not changed.	select	

#	Action	Expected Result	Notes
		ProcGetGrFiles('<shortname>', <granuleid>); cd to the public directory, verify the granule file is there and it's the same size as the one in the hidden and the one in the archive (under /stornext)	

TEST DATA:

EXPECTED RESULTS:

18.7 Use DPCV to checksum a large granule (ECS-ECSTC-354)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	7) use DPCV to checksum the granule verify there were no errors in the DPCV log		#comment
2	Find or ingest a big granule with a file size > 2GB, in a mode that's configured in the config above. Or simply copy a big file to one of the subdirectories.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
3	Use DPCV to checksum the granule: EcDlDpcvStart DEV09 -file sxuidfile	The is no error on the command line.	
4	Verify there's no error in the DPCV log.	There is no error in the log. For example, /usr/ecs/DEV09/CUSTOM/logs/EcDlDpcv.log.8490 contains no error.	

TEST DATA:

EXPECTED RESULTS:

18.8 Use CVS to checksum a large granule (ECS-ECSTC-355)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	8) use CVS to checksum the granule verify there were no errors in the CVS log		#comment
2	Find or ingest a big granule with a file size > 2GB, in a mode that's configured in the config above. Or simply copy a big file to one of the subdirectories.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
3	Use CVS to checksum the granule: EcDlCVSStart DEV09 -- starts the server EcDlInsertChecksumRequest.pl -mode DEV09 -g 62863 --insert the request. The server will pick up the request.	The is no error on the command line.	
4	Verify there's no error in the CVS log.	There is no error in the log. For example, /usr/ecs/DEV09/CUSTOM/logs/EcDlChecksumServer.ALOG contains no error.	

TEST DATA:

EXPECTED RESULTS:

18.9 Use ACVU to checksum a large granule on tape (ECS-ECSTC-356)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	9) use ACVU to checksum the granule on tape verify there were no errors in the ACVU log		#comment
2	Find or ingest a big granule with a file size > 2GB, in a mode that's configured in the config above. Or simply copy a big file to one of the subdirectories.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
3	Use ACVU to checksum the granule: EcDsAmAcvu.pl DEV09 -granuleid 62863	The is no error on the command line.	
4	Verify there's no error in the ACVU log.	There is no error in the log. For example, /usr/ecs/DEV09/CUSTOM/logs/EcDsAmAcvu.log contains no error.	

TEST DATA:

EXPECTED RESULTS:

18.10 Download a large granule using wu-ftp (with alternative compression methods) (ECS-ECSTC-357)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	10) download granule using wu-ftp (with alternative compression methods) verify that the decompressed file checksum matches what was ingested		#comment
2	Locate string 5 box where wu-ftp config resides: /etc/ftpd/ftpaccess Check the config file to find out which mode is configured. 	For example, you'll see sth similar to this: class all real,guest,anonymous !i4apl02.hitc.com anonymous-root /datapool/DEV02/user all 	
3	Find or ingest a big granule with a file size > 2GB, in a mode that's configured in the config above. Or simply copy a big file to one of the	A file with size > 2GB can be found in a subdirectory under	

#	Action	Expected Result	Notes
	subdirectories.	/datapool/<mode>/user.	
4	On the command line, use wget to download the big file using the following compression formats: .tar, .gz, .tgz, .zip and .z For example: wget ftp://f5eil01v/FS2/MOLT/MOD29.004/2006.07.03.tar wget ftp://f5eil01v/FS2/MOLT/MOD29.004/2006.07.03/MOD09CMG.A2011001.255.2011003094303.hdf.gz(.tgz, .zip and .z) . 	All the wget commands should run successfully. the .tar, .gz, .tgz, .zip or .z file should appear in the directory.	
5	Verify all the decompressed file checksum matches what was ingested.	Decompress the compressed file using tar -xvf *.tar; gunzip *.gz; gunzip *.tgz, tar -xvf *.tar; unzip *.zip and gzip -d *.z Run md5sum to get the checksum of the decompressed big file and verify it's the same as the one on the ingested file in datapool.	

TEST DATA:

EXPECTED RESULTS:

18.11 Download a large granule using apache (with alternative compression methods) (ECS-ECSTC-358)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	11) download granule using apache (with alternative compression methods) verify that the decompressed file checksum matches what was ingested		#comment
2	Locate string 5 box where apache config resides: /usr/ecs/OPS/COTS/apache/conf/httpd.conf Check the config file to find out the port number and mode associations. 	For example, you'll see sth similar to this: <VirtualHost p5eil01.pvc.ecs.nasa.gov:8081 [FD00:0:0:111::123]:8081> DocumentRoot /datapool/OPS/user 	

#	Action	Expected Result	Notes
3	Find or ingest a big granule with a file size > 2GB, in a mode that's configured in the config above. Or simply copy a big file to one of the subdirectories.	A file with size > 2GB can be found in a subdirectory under /datapool/<mode>/user.	
4	On the command line, use wget to download the big file using the following compression formats: .tar, .gz, .tgz, .zip and .z For example: wget http://p5eil01.pvc.ecs.nasa.gov:8081/FS2/MOLT/MODMGGAN.001/2000.09.03.tar wget http://p5eil01.pvc.ecs.nasa.gov:8081/FS2/MOLT/MODMGGAN.001/2000.09.03/MOD09CMG.A2011001.255.2011003094303.hdf.gz(.tgz, .zip and .z) . 	All the wget commands should run successfully. the .tar, .gz, .tgz, .zip or .z file should appear in the directory.	
5	Verify all the decompressed file checksum matches what was ingested.	Decompress the compressed file using tar -xvf *.tar; gunzip *.gz; gunzip *.tgz, tar -xvf *.tar; unzip *.zip and gzip -d *.z Run md5sum to get the checksum of the decompressed big file and verify it's the same as the one on the ingested file in datapool.	

TEST DATA:

EXPECTED RESULTS:

18.12 Subset a large granule using ESI (ECS-ECSTC-359)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	12) subset granule using ESI verify subsetting tool Not subsetting tool available yet for this test.		#comment

TEST DATA:

EXPECTED RESULTS:

19 PUBLISH ALL BROWSE

19.1 DPL Cleanup

19.1.1 Public Browse and MODIS unpublish (ECS-ECSTC-369)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure a MODIS collection to have granules published automatically by Ingest (PublishByDefault = Y).		
2	Ingest a Science granule along with a Browse granule for the MODIS collection (make sure this is not a duplicate Science granule).		
3	Verify the science granule and its browse are published.		
4	Verify there is a link to the browse granule in the directory containing the science granule.		
5	Verify JPEG images are created and stored in the public browse directory.		
6	Use the DataPool Unpublish service to unpublish the Science granule.		
7	Verify the science granule is unpublished.		
8	Verify the symbolic link to the public browse granule is removed from the science granule directory.		
9	Verify the browse granule remains public (IsOrderOnly is null).		
10	Verify the browse JPEG files still exist in the public DataPool.		

TEST DATA:

EXPECTED RESULTS:

19.1.2 Public Browse and Unpublish of deleted MODIS Browse (ECS-ECSTC-370)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Create or identify a Public MODIS granule with a Public Browse granule.		
2	Use the AIM Granule Deletion Service (EcDsBulkDelete.pl) to mark the MODIS granule as deleted. Use the option to delete the associated granules as well.		
3	Verify the MODIS granule and the associated browse granule are marked as deleted (DeleteEffectiveDate is set to current time).		
4	Run the DataPool Unpublish Service, using the option to pick up events from AIM.		
5	Verify the science granule is unpublished.		
6	Verify the browse granule is unpublished.		
7	Verify the link in the science granule directory that points to the browse is removed.		

TEST DATA:

EXPECTED RESULTS:

19.1.3 Public Browse and publishing of undeleted MODIS Science and Browse (ECS-ECSTC-371)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure the MODIS collection to automatically publish granules in Ingest.		
2	Verify the AmCollection PublishByDefaultFlag has the value 'Y'.		

#	Action	Expected Result	Notes
3	Follow the steps in the test case: Public Browse and Unpublish of deleted MODIS Browse.		
4	Verify the deleted science granules are unpublished.		
5	Verify the deleted browse are unpublished.		
6	Use the AIM Granule Deletion service (EcDsBulkUndelete.pl) to "un-delete" the science granule. NOTE: Do not use the option to skip associated granules.		
7	Verify the science granule is no longer marked as deleted.		
8	Verify the browse granule is no longer marked as deleted.		
9	Use the Data Pool Publish Utility with the option to pick up events in AIM to publish the "undeleted" MODIS science and browse granule.		
10	Verify the science granule is now marked as public in the Inventory Catalog (AmGranule::IsOrderOnly in the aim schema).		
11	Verify the browse granule is now marked as public in the Inventory Catalog.		
12	Verify the science granule directory contains a link the the public browse JPEG file.		

TEST DATA:

EXPECTED RESULTS:

19.1.4 Public Browse and Browse only publishing of undeleted MODIS Science and Browse (ECS-ECSTC-372)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure the MODIS collection to NOT automatically publish granules in Ingest.		
2	Verify AmCollection::PublishByDefaultFlag is 'N'.		
3	Delete a MODIS granule with a public Browse granule.		
4	Verify the AmGranule::DeleteEffectiveDate is set to the current date.		

#	Action	Expected Result	Notes
5	Verify the AmBrowse::DeleteEffectiveDate is set to the current date.		
6	Use the DataPool Unpublish Utility to pick up the AIM events and unpublish the deleted Browse granule. NOTE: If the MODIS Science granule was public it will also be unpublished.		
7	Verify the AmBrowse::IsOrderOnly is no longer public (it should be "H").		
8	Verify the JPEG files associated with the Browse granule are no longer recorded in the AmBrowseOnlineFile table.		
9	Verify the JPEG files are no longer in the Data Pool file system.		
10	Use the AIM Granule Deletion Service (EcDsBulkUndelete.pl) to undelete the MODIS Science and Browse granule.		
11	Verify the Science granule is no longer marked as deleted.		
12	Verify the Browse granule is no longer marked as deleted.		
13	Verify the EcDlActionDriver pick up the AIM events and publish the "undeleted" Browse granule.		
14	Verify the MODIS science granule is still in the hidden Data Pool and the Inventory Catalog indicates it is hidden (AmGranule::IsOrderOnly is "H").		
15	Verify the Browse granule is marked as public in the Inventory Catalog (AmBrowse::IsOrderOnly is NULL).		
16	Verify the Inventory Catalog correctly records the file names and directory for the Browse JPEG files.		
17	Verify the browse JPEF files exist in the public Data Pool.		

TEST DATA:

EXPECTED RESULTS:

19.1.5 Public Browse and DataPool Cleanup orphan and phantom checking (ECS-ECSTC-373)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Identify the list of public Browse granules that are associated with hidden Science granules. The list should include Browse granules associated with MODIS, ASTER, and MISR collections. The list can be created in a database table or file.		
2	Verify the contents of the list matches what is recorded in the Inventory Catalog as public browse. This can most easily be done by creating private tables that copy the contents of the AmBrowse table and the AmBrowseOnlineFile tables.		
3	Run the DataPool cleanup service (EcDlCleanupFilesOnDisk -fix) to remove orphans and phantoms from the file system. NOTE: It maybe necessary to run this utility several times to fix issues in the mode that are not associated with the test. The final run of this utility should be against a mode that satisfies step 1!		
4	Verify the contents of the AmBrowse table were not changed.		
5	Verify the contents of the AmBrowseOnlineFile table were not changed.		
6	Verify the utility did not report any orphans or phantoms.		

TEST DATA:

EXPECTED RESULTS:

19.1.6 Public Browse and Data Pool recovery from publication errors (ECS-ECSTC-374)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Identify a public MODIS Science granule with a public Browse.		
2	Verify that the Science granule is recorded as public in the inventory catalog.		
3	Verify that the Browse granule is recorded as public in the inventory catalog.		
4	Verify that the Science files are present in the public data pool file system.		

#	Action	Expected Result	Notes
5	Verify that the Browse JPEG files are present in the public data pool file system.		
6	Verify that the Science granule directory contains a link to the public Browse JPEG file.		
7	Run the data pool EcDlCleanupGranules.pl utility to cleanup the MODIS science granule.		
8	Verify the Science granule is recorded as hidden in the inventory catalog (both AmGranule and AmDataFile should indicate the granule is hidden)		
9	Verify the Browse granule is recorded as public in the inventory catalog.		
10	Verify the Science files are no longer present in the hidden or public data pool file systems.		
11	Verify the Browse JPEG files are present in the public data pool file system.		
12	Verify the Science granule directory does NOT contain a link to the public Browse JPEG file.		

TEST DATA:

EXPECTED RESULTS:

19.2 IVT

19.2.1 Public Browse and hidden Science with public Browse (ECS-ECSTC-375)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Identify a mode with hidden MODIS, ASTER, and MISR granules that contain links to public Browse granules.		
2	Run the Inventory Validation Utility (EcDlInventoryValidationTool.pl) to verify the mode.		
3	Verify that the tool doesn't report the Browse granules that are public but are associated with hidden science granules.		

TEST DATA:

EXPECTED RESULTS:

19.3 Ingest

19.3.1 Public Browse and ASTER ingest (ECS-ECSTC-344)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure AST_L1A to ingest into the hidden datapool.		
2	Configure AST_L1B to ingest into the public datapool.		
3	Ingest an AST_L1A granule with browse.		
4	Verify the AST_LIA is hidden and the browse is published.		
5	Ingest an AST_L1B granule without browse that has the same the acquistion time of the AST_L1A granule.		
6	Verify the AST_L1B granule is published and that it is linked to the browse of the AST_L1A granule ingested in step 3.		
7	Verify the datapool directory containing the AST_L1B granule also contains a link to the browse.		
8	Unpublish the AST_L1B granule.		
9	Verify the AST_L1B granule is moved to the hidden data pool and the browse link is removed.		
10	Also verify the browse remains public.		

TEST DATA:

EXPECTED RESULTS:

19.3.2 Public Browse and MISRBR ingest (ECS-ECSTC-345)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure MISBR to be published by Ingest.		
2	Configure a MISR Level 1 science collection (from DsMdMisrProcessingCriteria) to not be published by Ingest).		
3	Configure a MISR Level 2 science collection (from DsMdMisrProcessingCriteria) to be published by Ingest).		
4	Ingest a granule from the MISR Level 1 collection.		
5	Ingest a MISBR granule that will match the processing criteria of the MISR Level 1 granule.		
6	Verify the MISBR is published and a Browse granule is not produced from it.		
7	Verify the MISR science granule is associated with the MISR L1 granule ingested in step 3 (dsmdmisrbrowsegranulexref).		
8	Insert a MISR L2 science granule from the collection configured in step 3 that will also match the processing criteria of the MISBR granule.		
9	Verify the MISR Level 2 granule is published, and a browse "jpeg" is produced from the MISBR published in step 5.		
10	Verify the directory containing the MISR L2 granule also contains a link to the Browse JPEG file created.		
11	Verify the MISR L1 granule ingested in step 4 is not linked to the new Browse granule.		
12	Unpublish the MISBR granule ingested in step 5.		
13	Verify the browse image created from the MISBR granule remains public in the data pool and that the link to it remains in the L2 science granule directory.		

TEST DATA:

EXPECTED RESULTS:

19.3.3 Public Browse and MODIS ingest (ECS-ECSTC-346)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Configure a MODIS ESDT to not be published by Ingest.		
2	Ingest a granule for the MODIS collection configured in step 1 along with an associated browse (in the PDR). The browse granule should not be associated with any other Science granules.		
3	Verify the MODIS science granule is in the hidden data pool and the associated browse granule is in the public data pool.		
4	Verify there are no links to the public browse in the MODIS hidden or public directories.		

TEST DATA:

EXPECTED RESULTS:

19.3.4 Public Browse and AMSR ingest (ECS-ECSTC-360)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Ensure none of the test granules are in AIM.		
2	Configure the AMSR Level 2 collection to have ingest publish the granules.		
3	Ingest a day's worth of AMSR Level 2 granules that have daily browse (one browse for the ascending part of the orbit and one browse for the decending part of the orbit). Ingest only the AMSR Level 2 granules. Do not ingest the browse.	The granules should ingest without errors	
4	Wait for all 29 granules to successfully complete processing.		

#	Action	Expected Result	Notes
5	Ingest the 2 daily browse for the AMSR Level 2 granules using "linkage" files containing a lists of the associated Level 2 granules.		
6	Wait for the browse granules to successfully complete processing.		
7	Verify the associated browse are published and that the AMSR Level 2 granule directories in the Data Pool contain links to the associated public browse.		
8	Verify the HDF copy of the 2 browse granules is removed from the hidden Data Pool when the publication is completed.		
9	Verify the inventory database correctly records the location of the public browse files.		
10	Unpublish all the AMSR Level 2 listed in one of the browse linkage files.		
11	Verify the AMSR Level 2 granules that were unpublished are moved from the public Data Pool to the Hidden Data Pool.		
12	Verify the associated symbolic link to the Browse that is associated with these granules is removed from the public Science granule directory in the Data Pool		
13	Verify the associated browse is still cataloged as public in the Inventory database		
14	Verify the browse files are still present in the public Data Pool.		

TEST DATA:

A day's worth of AMSR Level 2 granules that have daily browse:
/sotestdata/DROP_801/DP_81_01/Criteria/327/2/G5 - Contains 15 AMSR science granule PDRs and 1 AMSR browse plus linkage file PDR
/sotestdata/DROP_801/DP_81_01/Criteria/327/2/G6 - Contains 14 AMSR science granule PDRs and 1 AMSR browse plus linkage file PDR
The mode in which you want to ingest these PDRs needs to have the AE_Land.002, Browse.001, PH.001, and QA.001 ESDTs installed and configured. Also, these granules should not already be in the mode otherwise the linkages won't work. So, delete them if they exist.
To ingest these granules, the science PDRs need to be copied into the mode's AMSR_E_SIPS polling directory. The browse PDRs should not be copied into the mode's AMSR_E_SIPS polling directory until every science granule has been successfully ingested.

EXPECTED RESULTS:

19.3.5 Public Browse and AMSR Browse replacement (ECS-ECSTC-361)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Follow the steps in the test case "Public Browse and AMSR ingest" to ingest a set of AMSR granules and 2 daily browse.		
2	Verify all granules are ingested correctly and are published.		
3	Ingest one of the 2 daily browse granules again using the same "linkage" file.		
4	Verify the original browse granule associated with the AMSR Level 2 granules is unpublished and marked as deleted.		
5	Verify the new "replacement" browse granule is in the public data pool.		
6	Verify the symbolic link in the AMSR Level 2 directory for the original browse granule was removed and a new symbolic link was created for the new browse granule.		

TEST DATA:

EXPECTED RESULTS:

19.3.6 Public Browse and MODIS browse replacement (ECS-ECSTC-362)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Follow the steps in the test case "Public Browse and MODIS ingest" to ingest a hidden MODIS granule with a public browse.		
2	Verify the MODIS science granule is hidden and the Browse granule is		

#	Action	Expected Result	Notes
	public.		
3	Publish the MODIS granule ingested in step 1.		
4	Verify the MODIS science granule is public.		
5	Verify there is a link to the public browse in the MODIS science granule directory.		
6	Configure the MODIS collection to automatically publish granules during Ingest.		
7	Ensure the AmCollection column named PublishByDefaultFlag is set to 'Y'.		
8	Ingest the same MODIS science granule and browse ingested in step 1.		
9	Verify the "original" MODIS granule ingested in step 1 is unpublished.		
10	Verify the link to the "original" browse file in the MODIS science granule directory is removed.		
11	Verify the "original" browse granule is still public.		
12	Verify the "replacement" MODIS science granule is now public.		
13	Verify the new browse granule is public.		
14	Verify the MODIS science granule directory contains a link the the new public browse granule.		
15	Restore the PublishByDefaultFlag to its original value.		

TEST DATA:

EXPECTED RESULTS:

20 HTTPS TESTING

20.1 Submit egi/esi requests for ICESat-2 data (ECS-ECSTC-3858)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for ICESat2 granule(s).		
2	Perform a service request on the granule(s) by selecting "Customize".		
3	Perform parameter subsetting. Select the different output format options: NetCDF-3, NetCDF4-CF, No Formatting, Tabular ASCII, Shapefile. Verify the requests complete successfully.		
4	Perform spatial subsetting. Verify the request completes successfully.		
5	Perform temporal subsetting. Verify the request completes successfully.		
6	Monitor the status of the requests on the Data Access GUI.		
7	Verify email notifications were sent for the requests.		
8	Verify the status complete email has the links to the files.		
9	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.2 Submit egi/esi requests for FRI Service (ECS-ECSTC-3859)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for AST_L1T.003 granules.		
2	Perform a service request on the granule(s) by selecting "Customize".		
3	Perform parameter subsetting. Select the different output format options: TIR, VNIR, and BOTH. Verify the requests complete successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests.		
6	Verify the status complete email has the links to the files.		

#	Action	Expected Result	Notes
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.3 Submit egi/esi requests for MOFIS Service (ECS-ECSTC-3860)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for MCD43GF.006 granules.		
2	Perform a service request on the granule(s) by selecting "Customize".		
3	Perform Band subsetting. Verify the request completes successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests.		
6	Verify the status complete email has the links to the files.		
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.4 Submit egi/esi requests for HEG Service (ECS-ECSTC-3861)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for a collection that is configured for HEG service.		
2	Select granule(s) and perform a service request by selecting "Customize" on the edit option page.		
3	Perform parameter subsetting. Select the different output format options. Verify the request completes successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests.		
6	Verify the status complete email has the links to the files.		
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.5 Submit egi/esi requests for GDAL Service (ECS-ECSTC-3862)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for a collection that is configured for GDAL service.		MISR data
2	Select granule(s) and perform a service request by selecting "Customize" on the edit option page.		
3	Perform parameter subsetting. Select the different output format options. Verify the request completes successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests.		
6	Verify the status complete email has the links to the files.		
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.6 Submit order request with no processing (ECS-ECSTC-3863)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for a collection that is configured for a service.		
2	Select granule(s) and perform a order request by selecting "Stage For Delivery" on the edit options page.		
3	Set the Distribution Type to "HTTPS" and the Select Processor to "No Processing".		
4	Select the "Download Data" button. Verify that the request completes successfully.		
5	Monitor the request on the Distribution Request page of the Order Manager GUI. Verify the request shipped.		
6	Verify that a email notification containing the https download links is sent.		There should be 2 Notifications sent. The ECS Order Notification that contains the download links and the cmr notification.
7	Verify that the files can be downloaded using the links.		

TEST DATA:

EXPECTED RESULTS:

20.7 Perform a Direct Download via the Earthdata Search Client (ECS-ECSTC-3864)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for a collection that is configured in Data Access.		
2	Select granule(s) and perform a direct download by selecting "Direct Download" on the edit options page.		
3	Select the Download Data button and verify it returns a page with the button to be able to view the download links.		
4	Click on the "View download links" button and verify it displays a page with the download links with https. Also, verify that the files can be downloaded using the links.		

TEST DATA:

EXPECTED RESULTS:

20.8 Cancelling requests in processing in Data Access via the GUI (ECS-ECSTC-3865)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Submit egi request with several granules.		
2	From the Monitor page of the Data Access GUI, cancel the request while it is in the processing state.		
3	Verify the request is cancelled and no longer continues to process the remaining granules.		
4	Verify that an order email notification is sent with the canceled status.		

TEST DATA:

EXPECTED RESULTS:

20.9 Uploading Service and Order Forms (ECS-ECSTC-3870)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Bring up the Data Access GUI.		
2	Choose a collection under the Collection Configuration tab.		
3	Right click on the collection, select "Upload Processing-Options Form for Collection(s)		
4	For the Name of the ACCOUNT, put earthdata login userid,		
5	Click on "Login" and enter earthdata login password	The three checkboxes on the right under "Accepted" should be checked when complete.	
6	Then click on "Upload Forms" button.	The checkbox for the collection should be checked when complete.	
7	Check the PUMP log on f5dpl01 in /usr/ecs//CUSTOM/data/DPL/DA_GUI. Verify both the order and service forms were uploaded successfully.		

TEST DATA:

EXPECTED RESULTS:

20.10 Configure Dataset-Service Data Objects for a Collection (ECS-ECSTC-3871)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Bring up the Data Access GUI.		
2	From the Collection Configuration tab, right click on the Service under the collection that needs to be configured.		
3	Select "Reconfigure Dataset-Service Data Objects"		
4	Select "hegtool" under "Use Selected Agent:" for the HEG service. Click Submit.	Another tab would open with the results.	
5	Verify there are no errors in the result tab		
6	From the Collection Configuration tab, doubleclick on the service. Click on the down arrow for "Configure Hdf Objects"		
7	Verify the data objects are listed.		

TEST DATA:

EXPECTED RESULTS:

20.11 Submit egi/esi requests for SMAPL1L2 Service (ECS-ECSTC-3872)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for a collection that is configured for SMAPL1L2 service.		
2	Select granule(s) and perform a service request by selecting "Customize" on the edit option page.		
3	Perform parameter subsetting. Select the different output format options. Verify the request completes successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests		

#	Action	Expected Result	Notes
6	Verify the status complete email has the links to the files.		
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.12 Submit egi/esi requests for GLAS Service (ECS-ECSTC-3874)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for a collection that is configured for GLAS service.		
2	Select granule(s) and perform a service request by selecting "Customize" on the edit option page.		
3	Perform parameter subsetting. Select the different output format options. Verify the request completes successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests.		
6	Verify the status complete email has the links to the files.		
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS:

20.13 Submit egi/esi requests for GEDI data (ECS-ECSTC-3885)

DESCRIPTION:

PRECONDITIONS:

STEPS:

#	Action	Expected Result	Notes
1	Using EDSC/Data Access Testbed search for GEDI granule(s).		
2	Perform a service request on the granule(s) by selecting "Customize".		
3	Perform parameter subsetting. Select the No Formatting format option. Verify the requests complete successfully.		
4	Monitor the status of the requests on the Data Access GUI.		
5	Verify email notifications were sent for the requests.		
6	Verify the status complete email has the links to the files.		
7	Verify the files can be downloaded and inspected.		

TEST DATA:

EXPECTED RESULTS: