



NASA ECHO Project

ECHO Data Partner's Guide

**March 2010
Document Version 10.7**

This page is intentionally left blank.

REVISION HISTORY

Date	Version	Brief Description	Author
11/1/07	10.1	Initial version for ECHO 10	Judith Wright
12/3/07	10.2	Added specific information on how to associate browse metadata to a collection/granule already loaded into the database.	Lorena Marsans
1/16/08	10.3	<ul style="list-style-type: none">• Incorporated ECHO Operations' comments• Removed "Authenticators" as they are no longer used	Judith Wright
11/01/09	10.5	Refactored document structure to allow for easier information access. Updated document according to recent schema and functionality changes. Updated images used for spatial	
1/15/10	10.6	Added Getting Started Section & Finalized comments	Matthew Cechini
3/15/10	10.7	Updating ACL & Group Management documentation to reflect new functionality. Added additional information re: SSL order configuration.	ECHO

TABLE OF CONTENTS

CONVENTIONS	XII
CHAPTER 1.BEFORE YOU BEGIN	13
1.1 Tasks That You Will Perform as a Data Partner	13
1.2 Skills You Will Need as a Data Partner	13
1.3 ECHO Concept and Design	14
1.3.1 ECHO as a Spatially Enabled Metadata Search and Order System	15
1.3.2 Security	15
1.4 ECHO Capability and Functionality	15
1.4.1 Benefits to Data Partners	16
1.5 ECHO Systems.....	16
CHAPTER 2.THE BASICS.....	17
2.1 ECHO Web Services	17
2.2 ECHO Globally Unique Identifiers (GUIDS).....	17
2.3 ECHO Entities.....	18
2.3.1 Users.....	18
2.3.2 Groups	18
2.3.3 Roles	18
2.3.4 Conditions	18
2.3.5 Rules	19
2.3.6 Catalog Items.....	19
2.3.7 Orders	19
2.3.8 Order Options.....	21
2.3.9 Extended Services	22
CHAPTER 3.ECHO SPATIAL REPRESENTATIONS	23
3.1 Spatial Overview.....	23
3.1.1 Collection & Granule Spatial Relationships	23
3.2 Geometry Representations	24
3.2.1 Coordinate Systems.....	24
3.2.2 Data Types and Representation	24
3.3 Invalid Spatial Representations.....	34
3.3.1 Polygon Points in Counter-Clockwise Order.....	34
3.3.2 Twisted Polygon.....	35
3.3.3 Hole Crosses over Outer Ring.....	36
3.3.4 Polygon Crosses Antemeridian or Pole	38

3.3.5	Inappropriate Point Density.....	39
3.3.6	Tolerance	41
3.4	Orbit Data	41
3.5	Global Data.....	42
CHAPTER 4	METADATA MODEL	45
4.1	ECHO Metadata Constructs.....	45
4.2	Required & Recommended Elements	45
4.2.1	Collections.....	45
4.2.2	Granules.....	47
4.2.3	Browse	48
4.3	Temporal Data	49
4.3.1	Collections.....	49
4.3.2	Granules.....	50
4.4	Spatial Data	50
4.4.1	Cartesian/Geodetic/Orbital Information	50
4.4.2	Two-Dimensional Coordinate System	50
4.5	Additional Attributes	52
4.5.1	Collections.....	53
4.5.2	Granules.....	54
4.6	Platforms, Instruments, and Sensors	55
4.6.1	Collections.....	55
4.6.2	Granules.....	56
4.7	Measured Parameters	57
4.8	Online Data Access URL and Online Resources URL	59
4.9	Keywords.....	59
4.9.1	Science Keywords	60
4.9.2	Spatial Keywords	60
4.9.3	Temporal Keywords	60
4.10	Access Control.....	61
4.10.1	Visibility Flag	61
4.10.2	Restriction Flag	61
4.11	Browse Images and Browse Image Metadata.....	62
CHAPTER 5	INGEST	63
5.1	Ingest Overview	63
5.1.1	Ingest Processing States	65
5.1.2	Ingest Processing Order	66

5.2	Delivering Data To ECHO For Ingest	66
5.2.1	Data Partner FTP Locations	66
5.2.2	Package File Delivery	67
5.2.3	Single File Delivery	67
5.3	Metadata Generation Tips.....	68
5.3.1	Creating ECHO-Compatible XML Files	68
5.3.2	Dates and Times.....	68
5.3.3	Unique Values.....	68
5.3.4	Metadata Inheritance	68
5.4	New Items vs Replace Items.....	69
5.4.1	Non-Replaceable Fields.....	70
5.5	Partially Adding/Updating/Deleting Items	70
5.5.1	Collections.....	71
5.5.2	Granules.....	71
5.5.3	Browse	72
5.6	Deleting Items.....	72
5.6.1	Ingest Reaper	73
5.6.2	Audit History.....	73
5.7	Reconciliation	74
5.7.1	Metadata Verification	74
5.7.2	Inventory Verification	79
5.8	Ingest Errors	84
5.8.1	Job Errors.....	84
5.8.2	File Errors.....	85
5.8.3	Item Errors	87
5.9	Ingest Reporting	89
5.9.1	Ingest Summary Report.....	89
5.9.2	Ingest Emails.....	92
5.9.3	ECHO Ingest Accounting Tool.....	92
5.10	Ingest Configuration.....	93
5.10.1	General Ingest Configuration	93
5.10.2	Provider Specific Configuration.....	93
CHAPTER 6	DATA MANAGEMENT	94
6.1	Access Control Concepts.....	94
6.1.1	Groups	94
6.1.2	Access Control List (ACL).....	95

6.1.3	Permissions.....	95
6.1.4	Catalog Item Identifiers	96
6.1.5	Provider “Administrators” Group	97
6.1.6	Restriction Flag	97
6.2	Access Control Recommendations	97
6.2.1	Data Mgmt & User Services Groups.....	98
6.2.2	Catalog Item ACLs.....	98
6.3	Reconciliation	98
6.3.1	ECHO Ingest.....	98
6.3.2	Data Management Service	99
CHAPTER 7.FULFILLING ORDERS		100
7.1	Order Fulfillment	100
7.1.1	Provider Policies	100
7.2	Order Workflow.....	102
7.3	Order Information.....	104
7.3.1	ECHO Order vs Provider Order	104
7.3.2	Provider Tracking ID	104
7.3.3	Order Items	104
7.3.4	User Information	105
7.4	Order Options	106
7.4.1	ECHO Forms Specification	106
7.4.2	Option Definitions.....	106
7.4.3	Option Assignments.....	107
CHAPTER 8.GETTING STARTED		109
8.1	Introduction.....	109
8.2	ECHO Environments.....	109
8.3	ECHO Provider Identification	109
8.3.1	Data Center ID	109
8.3.2	Provider Administrators.....	110
8.4	ECHO Ingest Setup	110
8.4.1	Ingest Notifications.....	110
8.4.2	Ingest Summary Reports	110
8.4.3	Packaging Method	110
8.4.4	Delivery Directory Structure	110
8.4.5	Input Detection Frequency & Threshold	111
8.4.6	No input warning threshold	111

8.4.7	Out of Sequence Notification Interval	111
8.5	ECHO Metadata	111
8.5.1	Metadata Data Model.....	111
8.5.2	Spatial Representations.....	111
8.5.3	Special Considerations	112
8.6	Data Management	112
8.6.1	Provider Object Access.....	112
8.6.2	Catalog Item Access.....	112
8.6.3	Provider Administrators.....	112
8.6.4	Groups	113
8.7	Ordering.....	113
8.7.1	Ordering or Not	113
8.7.2	Order Options.....	113
8.8	WIST 113	
8.8.1	Button Mapping.....	113
8.8.2	Calendar.....	114
8.8.3	Additional Attributes	117
APPENDIX A. ACRONYMS USED IN ECHO		120
APPENDIX B. FUNCTIONAL BREAKDOWN BY USER/ROLE TYPE.....		122
APPENDIX C. ECHO PATH URIS		127
C.1	Format	127
C.2	Behavior	127
APPENDIX D. ECHO ERROR HANDLING.....		128
APPENDIX E. INGEST SCHEMAS AND DTDS		130
APPENDIX F. BEST PRACTICES.....		131
F.1	Tips for Preparing Your Metadata	131
F.2	Factors That Affect Ingest Rates.....	131
F.3	Common Errors	131
APPENDIX G. INGEST SUMMARY REPORT		132
APPENDIX H. INDEX.....		134

LIST OF TABLES

Table 1. Description of Data Partner-Related ECHO Web Services.....	17
Table 2. Supported Spatial Data Types.....	23
Table 3. Valid XPath Values for the Qualified Tags	Error! Bookmark not defined.
Table 4. Browse Elements.....	Error! Bookmark not defined.
Table 5. ECHO Path URI Behavior	127

LIST OF FIGURES

Figure 1. ECHO System Context.....	14
Figure 2: ECHO Order Structure	19
Figure 3. The General Order Workflow	21
Figure 4. Orbit Swath.....	42
Figure 5. Ingest Overview.....	65
Figure 6 - WIST Main Search Page.....	114
Figure 7 - Scrolling calendar alerts in WIST	115
Figure 8 - WIST calendar page	116
Figure 9 - Sample data center calendar	117
Figure 10 - WIST attribute searching.....	118

LIST OF CODE LISTINGS

Code Listing 1. Geometry Example	24
Code Listing 2. Single Point Example	25
Code Listing 3. Multiple Points	25
Code Listing 4. Single Line Example	25
Code Listing 5. Multiple Line Example	26
Code Listing 6. Interpreted in Cartesian and Geodetic Systems	26
Code Listing 7. Adding Density	27
Code Listing 8. Single Polygon.....	29
Code Listing 9. Densified Geodetic Polygon	30
Code Listing 10. Single Polygon with a Hole.....	32
Code Listing 11. Bounding Rectangle	33
Code Listing 12. Polygon with Points in Counter-Clockwise Order.....	34
Code Listing 13. Twisted Polygon	35
Code Listing 14. Hole Crosses over the Outer Ring	37
Code Listing 15. Polygon Crosses International Dateline	38
Code Listing 16. Incorrect Density.....	39
Code Listing 17. Correct Density.....	40
Code Listing 18. Add orbit data to Granule Metadata	42
Code Listing 19. Add orbit parameters to Collection Metadata	42
Code Listing 20. Add global data to Granule Metadata.....	43
Code Listing 21. Add global data to Collection Metadata	43
Code Listing 22. Expression of Temporal Information (RangeDateTime).....	50
Code Listing 23. Expression of Temporal Information (RangeDateTime).....	50
Code Listing 24. Collection Two-Dimensional Coordinate System Coordinates	52
Code Listing 25. Granule Two-Dimensional Coordinate System Coordinates.....	52
Code Listing 26. Collection-Level Additional Attributes.....	53
Code Listing 27. Granule level additional attributes	54
Code Listing 28. Full Platform/Instrument/Sensor Description.....	56
Code Listing 29. Sources and Sensors	57
Code Listing 30. Measured Parameters	58
Code Listing 31. Online Access and Resource URLs	59
Code Listing 32. Sample Science Keyword	60
Code Listing 33. Sample Spatial Keyword	60
Code Listing 34. Sample Temporal Keyword	61
Code Listing 35. Sample Browse Insert (ECHO Hosted)	62

List Of Code Listings

Code Listing 36. Sample Browse Insert (Externally Hosted).....	62
Code Listing 37. Sample Granule Insert with Browse Associations.....	63
Code Listing 38. Manifest File Example	67
Code Listing 39. Metadata Update Example	69
Code Listing 40. Collection Deletes.....	72
Code Listing 41. Granule Deletes.....	73
Code Listing 42. Browse Deletes	73
Code Listing 43. Collection Metadata Verification Sample	74
Code Listing 44. Granule Metadata Verification Sample.....	75
Code Listing 45. METADATA_MISMATCH field-level error message	75
Code Listing 46. METADATA_MISMATCH object-level error message	76
Code Listing 47. METADATA_MISMATCH list field-level error message.....	76
Code Listing 48. METADATA_MISMATCH list object-level item missing error message	76
Code Listing 49. METADATA_MISMATCH list object-level extra item error message.....	77
Code Listing 50. COLLECTION_MISSING error message	77
Code Listing 51. GRANULE_MISSING error message.....	77
Code Listing 52. Successful Verification Processing Totals	78
Code Listing 53. Job Error Message	84
Code Listing 54. Missing Manifest File Error	84
Code Listing 55. FULL_SCHEMA Error Message.....	85
Code Listing 56. STRUCTURAL_SCHEMA Error Message	85
Code Listing 57. SCHEMA_VALIDATION_ERROR Error Message	86
Code Listing 58. FILE_TYPE_INDETERMINABLE Error Message	86
Code Listing 59. Referenced Granule Does Not Exist	87
Code Listing 60. Input Data Older than the Current Record	87
Code Listing 61. Referenced Browse Does Not Exist	88
Code Listing 62. Collection Referenced by Input Granule(s) Does Not Exist	88
Code Listing 63. Invalid Spatial Coverage Area Data	88
Code Listing 64. Data Integrity Error Message	89
Code Listing 65. Example of the Ingest Summary Report Overview Section	90
Code Listing 66. Example of the Ingest Summary Report Details Section	92
Code Listing 67. ECHO 8.0 Metadata Update DTD	Error! Bookmark not defined.
Code Listing 68: Removing All Online Data URLs for a Granule without Updating the Last Update for Granules Gr1 and Gr2	Error! Bookmark not defined.
Code Listing 69: Deleting a Specific Online Data URL	Error! Bookmark not defined.
Code Listing 70: Insert a New Online Resource	Error! Bookmark not defined.
Code Listing 71. DTD Excerpt for Collection Metadata.....	Error! Bookmark not defined.
Code Listing 72. DTD Excerpt for Granule Metadata	Error! Bookmark not defined.

Code Listing 73. Partial Metadata Insert	Error! Bookmark not defined.
Code Listing 74. Partial Metadata Delete	Error! Bookmark not defined.
Code Listing 75. Partial Granule Update	Error! Bookmark not defined.
Code Listing 76. Browse DTD	Error! Bookmark not defined.
Code Listing 77. Sample Insert Browse XML File	Error! Bookmark not defined.
Code Listing 78. Update Browse Example: testbrowseu.xml.....	Error! Bookmark not defined.
Code Listing 79. Delete Browse Example: testbrowseu.xml.....	Error! Bookmark not defined.
Code Listing 80: Catching Exceptions from ECHO	128

CONVENTIONS

All references to time are in Universal Time Coordinated (UTC).

Words in **bold** text are key words or concepts.

Programming examples (code listings) use a fixed width font (courier new) and have blue upper/lower lines separating them from the rest of the text.

Comments (denoted by // within examples)

Best practices or warnings appear in italicized, boxed text.

CHAPTER 1. BEFORE YOU BEGIN

The NASA-developed Earth Observing System (EOS) Clearinghouse (ECHO) is a spatial and temporal metadata registry that enables the science community to more easily use and exchange NASA's data and services. ECHO's main objective is to enable broader use of NASA's EOS data. It allows users to more efficiently search and access data and services and increases the potential for interoperability with new tools and services. The value of these resources increases as the potential to exchange and interoperate increases. ECHO has been working with other organizations to provide their Earth science metadata alongside NASA's for users to search and access. ECHO stores metadata from a variety of science disciplines and domains, including Climate Variability and Change, Carbon Cycle and Ecosystems, Earth Surface and Interior, Atmospheric Composition, Weather, and Water and Energy Cycle.

NASA's Earth science data has already proven itself to be extremely useful in understanding the planet Earth as an integrated system. To help science communities that need data from multiple organizations and multiple disciplines, ECHO provides a uniform view of NASA's data. It allows users to more efficiently search and access data and services and increases the potential for interoperability with new tools and services. For examples of how NASA's Earth science data is helping scientists understand the complexities of our Earth, visit Sensing our Planet and Other Featured Research Articles at <http://nasadaacs.eos.nasa.gov/articles/index.html>.

ECHO was designed to increase access to NASA Earth science data and services by providing a system with a machine-to-machine interface, that is, an Application Programming Interface (API). This API facilitates the discovery, online access, and order brokering for a Data Partner's data holdings. ECHO Data Partners retain complete control over what metadata are represented in ECHO by means of inserting new metadata, modifying existing metadata, removing old metadata, and controlling access to their metadata. ECHO Client Partners develop client applications that access the ECHO API and take advantages of the services made available. These clients, such as WIST (<https://wist.echo.nasa.gov>) allow end users to discover data which has been registered in ECHO's holdings and can be custom made to meet the needs of a general user audience, or a specific science application.

1.1 TASKS THAT YOU WILL PERFORM AS A DATA PARTNER

Usually performed in the order shown below:

- Generating metadata compliant with ECHO's xml format □ Chapter 34
- Ingesting your metadata into ECHO □ 5.95
- Managing holdings within ECHO □ **Error!
Reference
source not
found.6**
 - Access Control
 - Reconciliation
- Fulfilling Orders □ Chapter 7
 - Creating and managing order definitions
 - Supporting order fulfillment for ECHO submitted orders

1.2 SKILLS YOU WILL NEED AS A DATA PARTNER

Since ECHO uses platform-independent web service definitions for its API, there are no requirements for a client programming language. All examples in this document are in snippets of Java code; however, the code samples provided could be translated to any web service capable language.

As an ECHO Data Partner, you need to be familiar with basic software development and Service Oriented Architecture (SOA) concepts such as:

- XML and XML Schema (XSD)
- Web Service Definition Language (WSDL)
- Service-based Application Programmer's Interface (API)

1.3 ECHO CONCEPT AND DESIGN

NASA's Earth Science Data and Information System (ESDIS) has built ECHO based on Extensible Markup Language (XML) and Web Service technologies. ECHO interfaces with different clients and users through its series of Application Program Interfaces (APIs). ECHO is an open system with published APIs available to the ECHO Development and User community.

Internally, ECHO specifies APIs and provides middleware components, including data and service search and access functions, in a layered architecture. The figure below depicts the ECHO system context in relation to its public APIs.

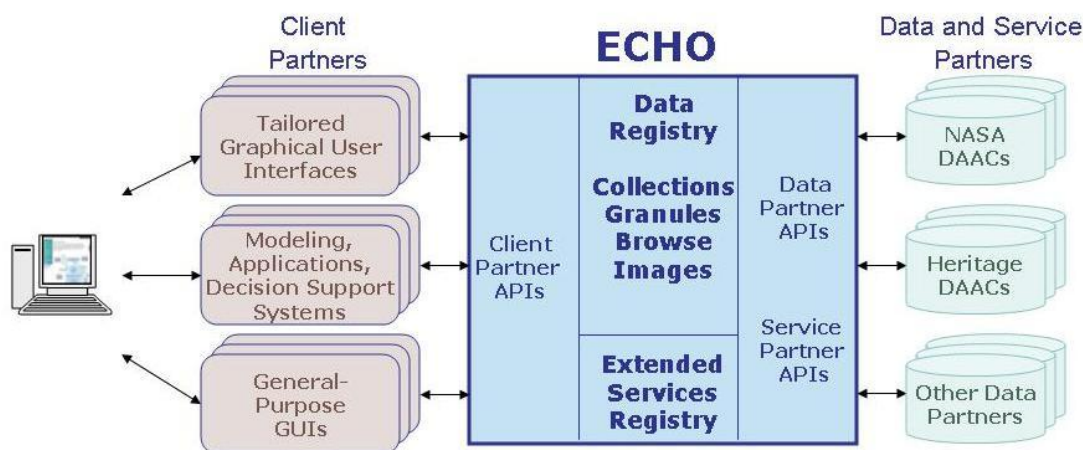


Figure 1. ECHO System Context

ECHO allows Data Partners to cache copies of their metadata within it. Data Partners have complete control over what metadata ECHO represents on their behalf. You, as a Data Partner, can insert new data, modify existing data, and remove old data.

All ECHO metadata is stored in an Oracle database with spatial extensions. The metadata model is derived primarily from that used by the Earth Observing System Data and Information System (EOSDIS) Core System (ECS). For more details about the ECHO model, refer to the ECHO Ingest documentation available here: <http://www.echo.nasa.gov/ingest/schemas/operations/docs/>.

Key features of the ECHO architecture are:

- *Ease of Partner Participation* – Designed to be low-cost and minimally intrusive, ECHO offers a set of standard ways for partners to interface with the system and a metadata exchange approach that accommodates existing partners and technology.
- *Open System/Published APIs* – To accommodate independent ECHO clients, ECHO uses an open system approach and publishes domain APIs. These APIs are independent of the underlying transport protocols used. ECHO communicates using WS-I Basic Profile v1.0 compliant web services. Documentation regarding the ECHO API can be viewed here: <http://api.echo.nasa.gov/echo/apis.html>.

Interactions with ECHO may involve user interactions in real time or may be machine to machine.

- *Evolutionary Development* – The ECHO system is being developed incrementally to allow for insight and feedback during the development cycle. Industry trends are followed and the use of commercial, off-the-shelf (COTS) products is optimized.

1.3.1 ECHO as a Spatially Enabled Metadata Search and Order System

ECHO allows Data Partners to define the spatial extent of a granule or a collection with different spatial constructs (for example: point and polygon). These spatial extents may be in either the Geodetic or Cartesian coordinate systems. Orbital data may also be provided to describe a collection or granules spatial extents. A Client Partner can then construct a search using a point, a line, or a polygon (or multiple polygon) spatial type, and ECHO responds with data whose spatial region intersects the described region.

ECHO provides services for interacting with its **Catalog** of metadata. Queries can be performed in a number of ways; result contents can be specified, and the resulting data sets can be incrementally accessed so that large return sets can be handled gracefully. ECHO also supports constructing, submitting, and tracking orders for the data that the metadata represents. ECHO supports both an embedding of a Uniform Resource Locator (URL) within the metadata for accessing the data (which the client simply accesses via Hypertext Transfer Protocol [HTTP]) or File Transfer Protocol (FTP), and a more complicated order process in which quotes and order options are accommodated.

1.3.2 Security

The ECHO system requires Secure Sockets Layer (SSL)-based communication from Client Applications to the ECHO API and supports, but does not require, secure communication from ECHO to a Data Partner's order fulfillment service. Internally, the ECHO system is protected through a layer of software and hardware control mechanisms to preserve the integrity of ECHO's holdings. When configuring order fulfillment, Data Partners are strongly encouraged to utilize SSL communications (See Chapter 7 for more information)

1.4 ECHO CAPABILITY AND FUNCTIONALITY

ECHO provides an infrastructure that allows various communities to share tools, services, and metadata. As a metadata clearinghouse, it supports many data access paradigms such as navigation and discovery. As an order broker, ECHO forwards orders for data discovered through the metadata query process to the appropriate Data Partners for order fulfillment. As a service broker, ECHO decentralizes end user functionality and supports interoperability of distributed functions.

Although this Guide focuses on the needs of Data Partners, ECHO supports the following different, nonexclusive types of Partners:

- *Data Partners* – Organizations that supply metadata representing their data holdings to the ECHO database
- *Client Partners* – Organizations that participate by developing software applications to access the Earth science metadata in the ECHO database
- *Service Partners* – Organizations that participate by advertising their Earth science-related services to the user community via ECHO, which maintains service descriptions in a Service Catalog (either special services, or services that are available as an option on a selected set of granules/collections) and support the user in ordering those services.
- *Extended Service Partners* – Organizations that participate by providing a central location for registration, classification, and maintenance of Earth science services, interfaces, GUIs, and advertisements

ECHO addresses science user needs through a set of well-defined and open interfaces upon which the user community can build its own client applications. In this way, ECHO supports extendable, flexible user interfaces, allowing industry and the science community to drive the progress of available Earth science applications. For more complete information about client applications, refer to the companion piece to this Guide, the *ECHO Client Partner's Guide*.

The ECHO approach allows users to build their own user interfaces to ECHO, rather than being limited to the data search and order system provided by NASA. For Data Partners, ECHO offloads the burden of providing the

system resources required for searching and gives users the flexibility to support community-specific services and functionality. ECHO's interoperability features allow all participants to benefit from the distributed development of functions, again reducing dependence on NASA resources.

1.4.1 Benefits to Data Partners

ECHO'S open system provides Earth science data and services to a large, diverse pool of users, enabling scientific community interaction and collaboration. ECHO benefits Data Partners in the following ways.

- Makes data resources available to a wide ranges of potential users
- Virtual "co-location" with other data sources and services
- Common data language
- Enable loosely coupled application solutions
- Metadata and order subscription services
- Enhanced data control mechanisms

1.5 ECHO SYSTEMS

There are three ECHO Systems that you, as a Data Partner, have access to. A short description of each system is given below. For additional information, refer to the ECHO Website (<http://www.echo.nasa.gov>) and select *ECHO System* on the side navigation.

ECHO Operations. The ECHO Operational system is a production system which serves the needs of the Earth Science community. The Data Holdings within this system include those which ECHO's Data Partners have made available for use by the Earth Science community.

ECHO Partner Test. The ECHO Partner Test system is a test system which serves the needs of the ECHO Data, Client, and Service partners. The Data Holdings within this system include those which ECHO's Data Partners have made available for their own testing purposes

ECHO Testbed. The ECHO Alpha Testbed system has been established in order to facilitate an exchange of ideas and provide early insight into the upcoming capabilities offered by ECHO.

CHAPTER 2. THE BASICS

This chapter describes the basic terms and concepts used in subsequent discussions of the Web Services API and the ingest process.

2.1 ECHO WEB SERVICES

To access a particular service through the ECHO Web Services API, refer to the resources section of each ECHO system's information page on the ECHO website (<http://www.echo.nasa.gov/> >> ECHO Systems).

The table below shows each ECHO service and a brief description of its capabilities. To access the Web Service Description Language (WSDL) document that describes a service, attach the suffix .wsdl from the API page following this format: `http://<mode>.echo.nasa.gov/echo-wsdl/v10/<Service Endpoint>.wsdl`. Service Endpoints are found in Table 1.

Table 1. Description of Data Partner-Related ECHO Web Services

Service Name	Description	Service Endpoint
Authentication	Facilitates ECHO token generation..	AuthenticationService
Catalog	Facilitates data discovery.	CatalogService
Data Management	Facilitates access control management , order option management, reconciliation, and holdings analysis.	DataManagementService
Extended Services	Facilitates creating, retrieving, updating, and deleting web services within ECHO, including interfaces, implementations, advertisements, and GUIs .	Extended ServicesService
Event Notification	Facilitates the management of metadata event subscriptions.	EventNotificationService
Group Management	Facilitates user group management used by ECHO's access control capability.	GroupManagementService
Order Management	Facilitates the creation, submission, and retrieval of user submitted metadata orders.	OrderManagementService
Order Processing	Facilitates order fulfillment communication from Data Partners to ECHO.	OrderProcessingService
Provider	Facilitates provider registration, information management, and policy management.	ProviderService
Subscription	Facilitates the creation and management of metadata subscriptions.	SubscriptionService
Taxonomy	Facilitates management of data taxonomies useful for classifying services associated with ECHO's data holdings.	TaxonomyService
User	Facilitates user account creation and maintenance.	UserService

2.2 ECHO GLOBALLY UNIQUE IDENTIFIERS (GUIDS)

An ECHO Globally Unique Identifier (GUID) is a mostly random number with a large number of unique keys that is assigned to an item by ECHO. A GUID is normally a 16-byte (128-bit) number in hexadecimal form.

ECHO uses GUIDs to identify items such as users, providers, contacts, orders, etc. Client applications use GUIDs to find and operate on items using the ECHO API.

2.3 ECHO ENTITIES

This section describes several high-level concepts that help you understand the ECHO system. The following is a selected list of entities.

2.3.1 Users

The most basic entity in the ECHO system is a **user**. Each user is identified by a unique user name. There are two types of users: **registered users** and **guests**. Registered users can save information they plan to use in their next session. Guests have the ability to do many of the things registered users can do, but they cannot count on persistent access to information across sessions in addition to other limitations.

To see a detailed breakdown of functionality by user type and role type within ECHO, see Appendix B, Functional Breakdown By User/Role Type.

2.3.2 Groups

The term **groups** refers to an aggregating mechanism in ECHO that allows Data Partners to associate a Group name with a given set of users. When a group is created, the group's owner specifies one or more ECHO users to be the group's manager(s). Group managers can be then added and removed after creation by other group managers. Group managers add members to their associated group. After becoming a member of a group, an ECHO user can be granted access to restricted metadata via the Data Management Service—refer to Chapter 6 for details.

2.3.3 Roles

ECHO regulates access privileges based on the concept of user roles. User roles are a way to grant a user access to the system. These roles facilitate greater flexibility with operation-level authorization and allow certain users the ability to have more than one role without having more than one account in the system.

As a Data Partner, you will be granted a role of **provider role**. You may have one or more provider roles, each of which is associated with one provider in the ECHO system. Your provider roles allow you to access and update information about the providers with which they are associated. For example, if you have a provider role for Oak Ridge National Laboratory (ORNL), then you can use the UpdateProviders operation of the ProviderService to update the contact information for ORNL.

When accessing the ECHO API, ECHO will automatically allow you to take advantage of your provider role privileges if you have only one provider role associated with your user account. If you have more than one provider role, the ECHO client you are using to access the ECHO API must explicitly request that your generated token be associated with your privileged role.

Note: WIST does not currently allow an ECHO Data Partner to request a specific provider role. PUMP and the EIAT both facilitate privileged provider access based upon your chosen provider.

To see a detailed breakdown of functionality by user type and role type within ECHO, see Appendix B, Functional Breakdown By User/Role Type.

2.3.4 Conditions

Conditions represent a partial equation to be evaluated as part of the Access Control (ACL) honoring system. The type of the condition defines the evaluation process. Temporal Conditions use a date range, so that a date associated with a granule can be compared against the date range to check for applicability of the condition. The primary use of conditions is to facilitate reuse among data rules. The same temporal condition can be used by both a restriction and a permission to control access to metadata. To extend a time range, you only have to change one **TemporalCondition** as opposed to changing multiple data rules. Another type of condition is **RestrictionFlag**, which can be used to restrict access to collections or granules based on the value of a **RestrictionFlag** metadata field. For example, this might be used to control access based on a granule's science quality.

2.3.5 Rules

Rules include conditions and provide a complete evaluation. Rules define which specific data is to be controlled, as well as the condition to use for evaluating whether the data should be controlled. Rules also contain a comparator, which is a key part of rule evaluation. Lastly, rules contain data including **ActionType** (describes which actions the rule applies to), and in the case of a permission, a **GroupName** (describes which Group the permission applies to). **Restrictions** (one type of a rule) apply to the global ECHO population, and **permissions** (the other type of a rule) apply to a specific group. Refer to Chapter 6 for details about using rules for data management.

2.3.6 Catalog Items

A **catalog item** is any metadata item (granule or collection) that is available for discovery and online access or ordering from the ECHO system. To find the desired catalog items to order, a query is submitted through the ECHO API's catalog service. The results of a query may return several granules or collections, which are uniquely identified by an ECHO **catalog item GUID** (CatalogItemId, which is an assigned XML metadata tag).

The catalog items that are included in the results for a user's submitted query are filtered based upon access control **rules** which have been defined by the Data Partner. Data Partner's may also override access control **rules** by setting a collection or granule's visibility or orderability in the provided metadata. Catalog items which have an associated online accessible URL will be made available for immediate access by ECHO Client applications.

2.3.7 Orders

An **order** is a collection of **catalog** items that a user would like to order from a Data Partner. Each catalog item in the order is associated with a quantity and order options, if order options are required for that item. Within ECHO, a user creates an order and then adds, deletes, and updates each item in the order before submitting the order to ECHO.

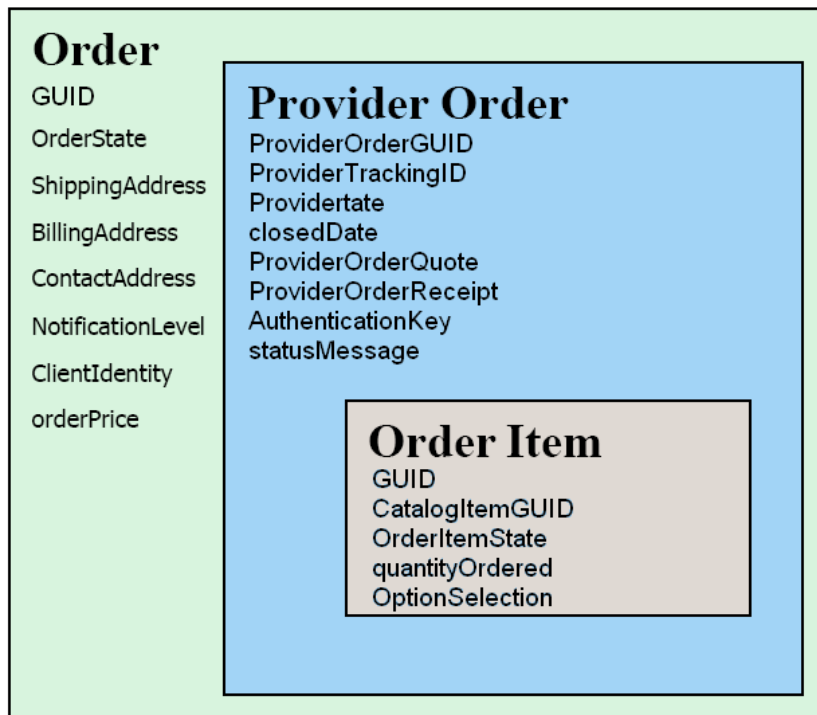


Figure 2: ECHO Order Structure

The contents of a user's order can span many providers. ECHO will organize catalog items associated with an order into what is called a **provider order**. The provider order contains the listing of catalog items that are associated with each provider. Since an order may contain orders from multiple providers, an order can consist of

one or more provider orders. Each provider order can consist of one or more catalog items that belong to the same provider. To identify a specific provider order, you need the GUID of the order that includes that provider order and the GUID of the data provider associated with that provider order. Both of these GUIDs can be obtained through the ECHO API. When a full order is submitted, ECHO splits the user's order into separate provider orders and submits the appropriate provider order to each Data Provider.

The **OrderManagementService** allows users to create and change orders, provider orders, or individual catalog items. However, once the **SubmitOrder** operation is executed for an order within the **OrderManagementService**, the user can no longer execute any further changes on that order. A registered user may look at the current and historical status of any of their submitted orders.

Once a provider order is submitted to the appropriate provider, the status of that order can be changed in two ways:

- The Data Provider can send an immediate response, whether they will or will not accept the order, to an order submission.
- The Data Provider can wait and asynchronously use the **OrderProcessingService** to change the status of an order after they have had time to process the order.

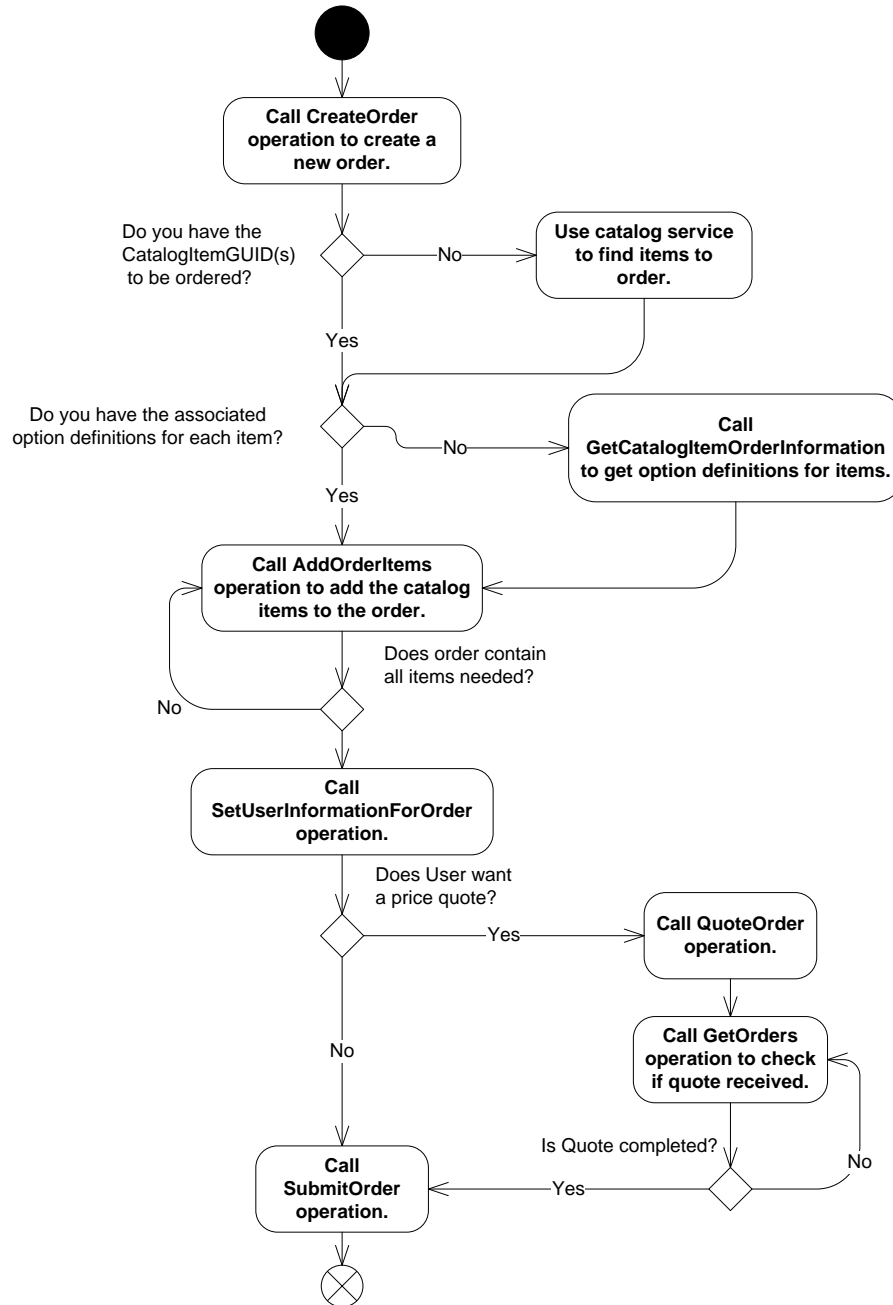


Figure 3. The General Order Workflow

2.3.8 Order Options

Data Partners often require additional information in order to fulfill a user's order request, or would like to make available additional processing options (e.g. parameter subsetting) for specific catalog items in an order. ECHO allows providers to create **order options** to describe the structure of the data which is to be requested by an ECHO client as well as how to display the order form to the user. An **order option** contains a unique name, description, XML order option form, scope, and sort key. The **scope** of an order option is always PROVIDER for order options that have been created by a data provider. There are also order options which have a scope of SYSTEM which are globally accessible order options available to all providers for inclusion in their order options.

Order options cannot be deleted from the ECHO system if there are orders in ECHO that utilize the order option. Providers wishing to replace an order option may choose to 'deprecate' an order option, which will allow existing orders to be fulfilled or reviewed, but not allow new orders to utilize the order option. For more information on how to create order options, see Chapter 7.

2.3.8.1 Option Definition

Option definitions are an association between an **order option** and a dataset within ECHO's holdings. Multiple option definitions may exist for a single ECHO dataset. Data providers may choose to filter the order options that are displayed to a user for a specific collection based on metadata values which are found in each catalog item. For example, a provider may wish to have a specific order option displayed to a user if the quality flag in their metadata has a value of 5.

2.3.8.2 Option Selection

An **OptionSelection** is a subset of the order option which contains the relevant order information that the provider has requested from the user. It is an ECHO client's responsibility to correctly display and order option and return a valid option selection for a specific order option.

2.3.9 Extended Services

An ECHO Extended Service is a functional capability offered by the community which can be used in coordination with the ECHO system. The services are "Extended" because they are not native to ECHO. ECHO Data Services can be many different things, from dedicated machines that transform or reduce the size of data, to innovative algorithms for discovering data that is relevant to particular research topics, to offerings by organizations to gather in data. ECHO is designed to support any type of related data service, enabling more innovative approaches to meeting the EOS mission. ECHO provides interfaces and mechanisms that allow organizations to publish their services and correlate their service to the data types in the clearinghouse.

An ECHO Service Partner is an individual or organization that participates with ECHO by providing access to functions that create Earth Science Data or information. ECHO advertises these services as well as brokers these services, as the Service Partner requests.

CHAPTER 3. ECHO SPATIAL REPRESENTATIONS

This chapter describes the terms and concepts relevant to ECHO's spatial model. Examples are provided for each supported spatial representation included discussions regarding known spatial topics.

3.1 SPATIAL OVERVIEW

Spatial metadata refers to the area of the Earth that a collection or granule covers. The spatial coverage area for a granule should be within the spatial coverage area of its primary collection. ECHO Ingest will not perform validation for this relationship. To ensure that spatial searches will return the correct results, you must prepare the spatial metadata according to the detailed guidelines discussed below.

The ECHO system accepts spatial data represented in the Cartesian and Geodetic coordinate systems. ECHO also accepts spatial information representing orbital data. Refer to the table below for supported spatial data types and guidelines for limitations to granularity. You should choose a coordinate system based on the size and projection of the original data of the spatial area covered. You may not combine spatial types for granules within the same collection as the representations are mutually exclusive.

The ECHO implementation of the Cartesian coordinate system accepts spatial data types of Point, Line, Bounding Rectangle and Polygon. The ECHO implementation of the Geodetic With coordinate system (World Geodetic System 84) accepts spatial data types of Point, Line, Bounding Rectangle, and Polygon.

Table 2. Supported Spatial Data Types

Spatial Data Type	Cartesian	Geodetic	Orbit	Guidelines and Restrictions
Point	✓	✓	✓	
Bounding Rectangle	✓	✓	✗	ECHO stores bounding rectangle data as a polygon with four vertices.
Line	✓	✓	✗	A line may not have consecutive vertices with the same latitude and longitude. A line must be less than one half the circumference of the Earth in the Geodetic coordinate system.
Polygon	✓	✓	✗	A polygon's vertices must be stored in order of vertex connection. Provide the vertices in clockwise order. No consecutive vertices may have the same latitude and longitude, that is, no repeating points. Also, polygonal lines may not cross each other within the polygon. No polygon should cover more than half the Earth in the Geodetic coordinate system.

The ECHO system will not manipulate any of the spatial input metadata. You are responsible for the correctness and integrity of your spatial metadata. To prepare your metadata such that ECHO can support the correct searching for your data, please observe the information in the following sections.

3.1.1 Collection & Granule Spatial Relationships

The following items are some guidelines that should be considered when generating spatial metadata within ECHO collection and granule records.

- Each collection may specify only one coordinate system for its spatial coverage.
- Each collection's coordinate system is independent of all other collections.
- Each collection's coordinate system is independent of its granule spatial representation (i.e. a collection's spatial extent may be expressed in the Cartesian geometry, but have all of its granules specify their spatial extents in the Geodetic geometry).
- A collection specifies its granules' spatial representation, which cannot be overridden by a granule.

- A collection with an orbital granule spatial representation must also specify specific orbit parameters in order to facilitate granule discovery via spatial constraints.
- Ingest for a metadata record will fail if any spatial metadata input is invalid according to the associate rules with the utilized coordinate system.

3.2 GEOMETRY REPRESENTATIONS

Spatial data are most commonly described as geometry such as a polygon or line. They are stored as spatial objects to record shape, spatial locations of corner points and spatial coordinate system used (e.g. Cartesian or Geodetic). It is important to ensure that the correct coordinate system is specified. The same set of coordinates will result in different spatial areas which affects the discovery of data and the representation of that data to users via ECHO clients.

3.2.1 Coordinate Systems

3.2.1.1 Cartesian Coordinate System

The Cartesian coordinate system is a flattened coordinate system with longitude ranged from -180 to 180 degrees and latitude ranged from -90 to 90 degrees. The projected map is flattened and open along Antemeridian with the North Pole and South Pole as top and bottom line respectively. Please be aware of the following Cartesian coordinate system constraints:

- Any single spatial area may not cross the Antemeridian and/or Poles.
- Two vertices will be connected with a straight line.

3.2.1.2 Geodetic Coordinate System

The Geodetic coordinate system is defined in angular (latitude and longitude) and is defined relative to spherical polar coordinate and Earth Geodetic datum. The ECHO implementation of the Geodetic coordinate system follows OGC standards, which are defined at <http://www.opengeospatial.org/>. The Geodetic coordinate ECHO chose to support is World Geodetic System 84 (WGS 84). Please be aware of the following Geodetic coordinate system constraints:

- The ECHO Geodetic model uses the great circle distance to connect two vertices to construct a polygon area or line. If there is not enough density (that is, the number of points) for a set of vertices, then the line or the polygon area might be misinterpreted or the metadata might be considered invalid.
- Any single spatial area may cross the Antemeridian and/or Poles
- Any single spatial area may not cover more than one half of the earth.

3.2.2 Data Types and Representation

3.2.2.1 Geometry

Spatial data in Cartesian or Geodetic coordinate systems are specified within a `<Geometry>` tag.

Code Listing 1. Geometry Example

```
<Spatial>
...
<HorizontalSpatialDomain>
...
  <Geometry>
...
  </Geometry>
```

```
...
  </HorizontalSpatialDomain>
</Spatial>
```

Within a <Geometry> tag points, lines, bounding rectangles, and/or polygons can be included to define the spatial extent of your data.

3.2.2.2 Point

ECHO can receive and store spatial data representing one or more points. ECHO also supports searching for spatial data representing one or more points. In the XML metadata, follow the syntax shown in the following code sample to define a spatial extent as one or more points:

Code Listing 2. Single Point Example

```
<Geometry>
  <Point>
    <PointLongitude>-123.948</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
  </Point>
</Geometry>
```

Code Listing 3. Multiple Points

```
<Geometry>
  <Point>
    <PointLongitude>-123.948</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
  </Point>
  <Point>
    <PointLongitude>-133.546</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
  </Point>
</Geometry>
```

3.2.2.3 Line

ECHO can receive and store spatial data representing one or more lines. ECHO also supports searching for spatial data representing one or more lines. In the XML metadata, follow the syntax shown in the following code sample to define a spatial extent as one or more lines:

Code Listing 4. Single Line Example

```
<Geometry>
  <Line>
    <Point>
      <PointLongitude>-123.948</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
```

```
        <PointLongitude>-133.546</PointLongitude>
        <PointLatitude>45.0664</PointLatitude>
    </Point>
</Line>
</Geometry>
```

Code Listing 5. Multiple Line Example

```
<Geometry>
  <Line>
    <Point>
      <PointLongitude>-123.948</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-133.546</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
  </Line>
  <Line>
    <Point>
      <PointLongitude>-123.948</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-133.546</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-143.546</PointLongitude>
      <PointLatitude>40.0664</PointLatitude>
    </Point>
  </Line>
</Geometry>
```

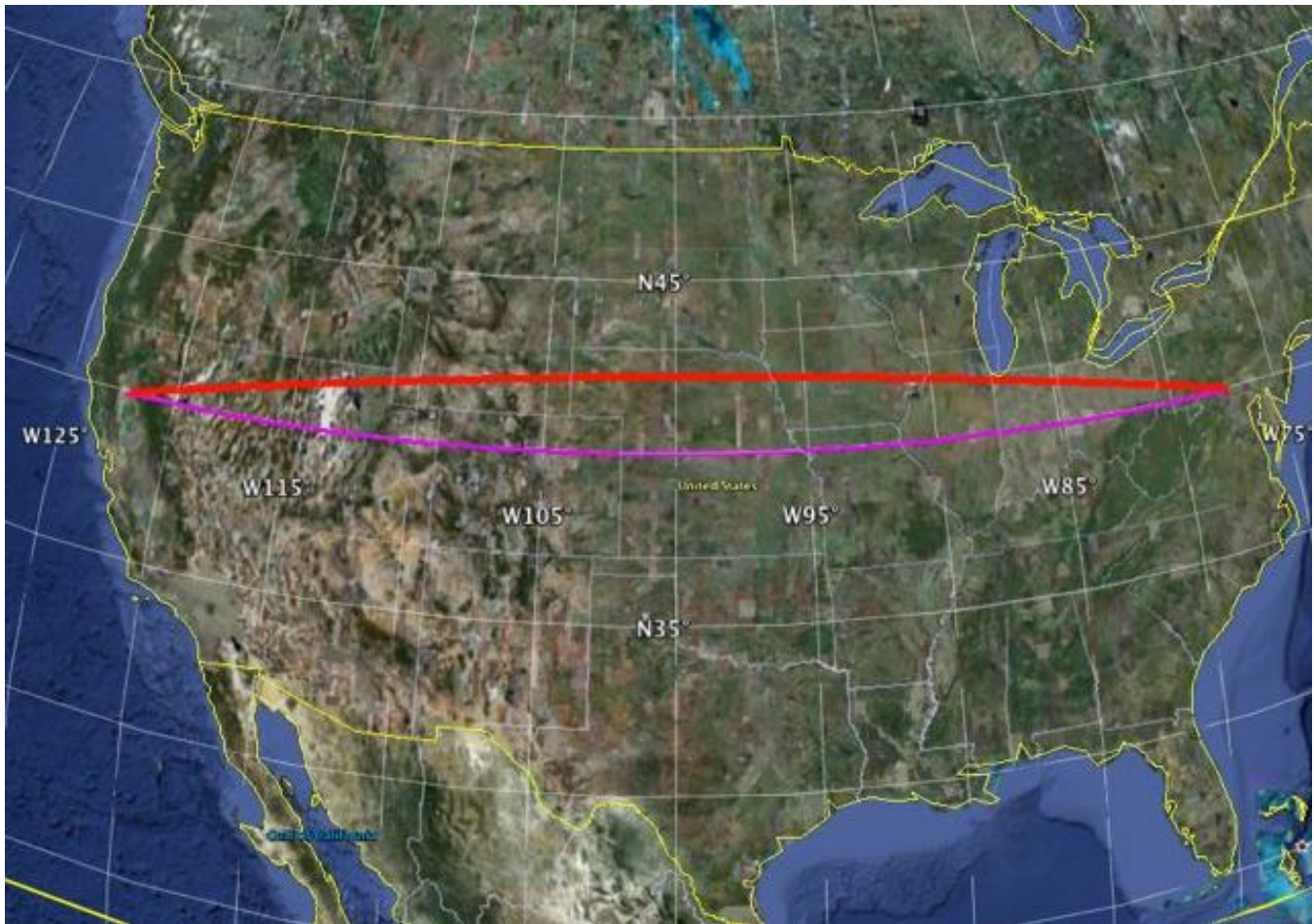
Due the differences in how the Cartesian and Geodetic coordinate systems represent spatial data, the same line will represent a noticeably different path on the Earth's surface depending on the chosen coordinate system. The following line segment is shown in its Cartesian and Geodetic representation.

Code Listing 6. Interpreted in Cartesian and Geodetic Systems

```
<Geometry>
  <Line>
    <Point>
      <PointLongitude>-122.1</PointLongitude>
      <PointLatitude>39.98</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-77.35</PointLongitude>
      <PointLatitude>39.98</PointLatitude>
    </Point>
  </Line>
</Geometry>
```

```
</Line>  
</Geometry>
```

The figure below shows the line in the Cartesian and Geodetic coordinate systems. The Cartesian line is drawn in purple. The Geodetic line is drawn in red.



If it is desired that the Geodetic line follow the same path as the Cartesian line more points can be giving the line more density. The more points that are added, the closer the Geodetic Line will follow the Cartesian path. The following code expression shows an example of adding more points in the Geodetic coordinate system.

Code Listing 7. Adding Density

```
<Geometry>  
  <Line>  
    <Point>  
      <PointLongitude>-122.1</PointLongitude>  
      <PointLatitude>39.98</PointLatitude>  
    </Point>  
    <Point>  
      <PointLongitude>-110</PointLongitude>
```



```
        <PointLatitude>39.98</PointLatitude>
    </Point>
    <Point>
        <PointLongitude>-100</PointLongitude>
        <PointLatitude>39.98</PointLatitude>
    </Point>
    <Point>
        <PointLongitude>-90</PointLongitude>
        <PointLatitude>39.98</PointLatitude>
    </Point>
    <Point>
        <PointLongitude>-80</PointLongitude>
        <PointLatitude>39.98</PointLatitude>
    </Point>
    <Point>
        <PointLongitude>-77.35</PointLongitude>
        <PointLatitude>39.98</PointLatitude>
    </Point>
</Line>
</Geometry>
```



3.2.2.4 Polygon

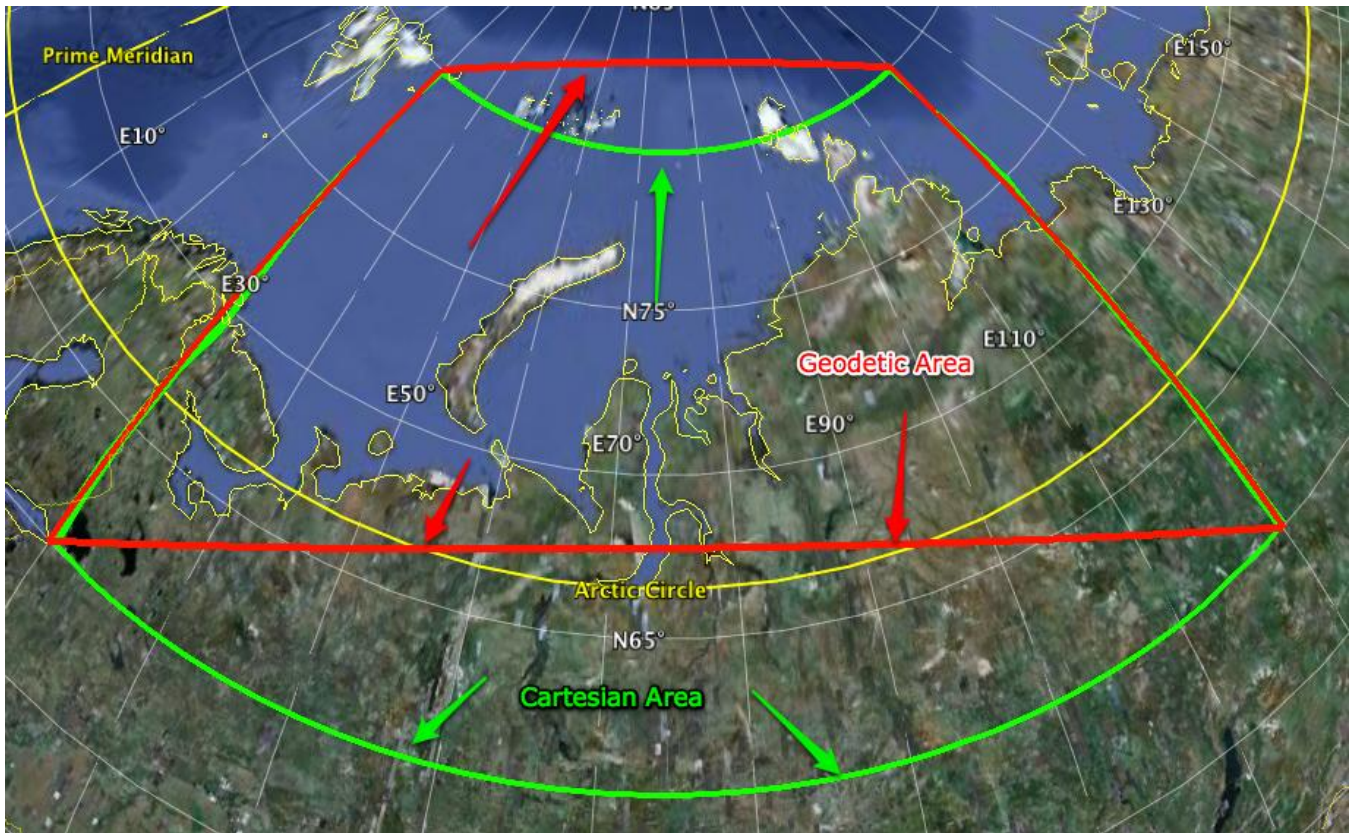
ECHO can receive and store spatial data representing one or more polygons with or without an areas of exclusion. ECHO also supports searching for spatial data representing any of these polygons. In the XML metadata, follow the syntax shown in the following code sample to define a spatial extent as a polygon:

Code Listing 8. Single Polygon

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>120</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>30</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>30</PointLongitude>
        <PointLatitude>80</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>120</PointLongitude>
        <PointLatitude>80</PointLatitude>
      </Point>
    </Boundary>
  </GPolygon>
</Geometry>
```

A single polygon can have multiple holes, each represented by a single outer ring surrounding the area within it. In the Cartesian coordinate system, straight lines connect the points of the ring in the order in which they are listed, which must always be in clockwise order. In the Geodetic coordinate system, the points are connected using a great circle arc according to the shortest distance between two points. Remember that polygonal coverage cannot span more than half the earth and may not cross the Antemeridian and/or poles in the Cartesian coordinate system. A polygon cannot represent more than one half of the earth in the Geodetic coordinate system.

The previously included XML polygon example is shown in the following figures as it would be represented in both the Cartesian and Geodetic coordinate systems. The red outline shows the Geodetic polygon. The green outline shows the Cartesian polygon.



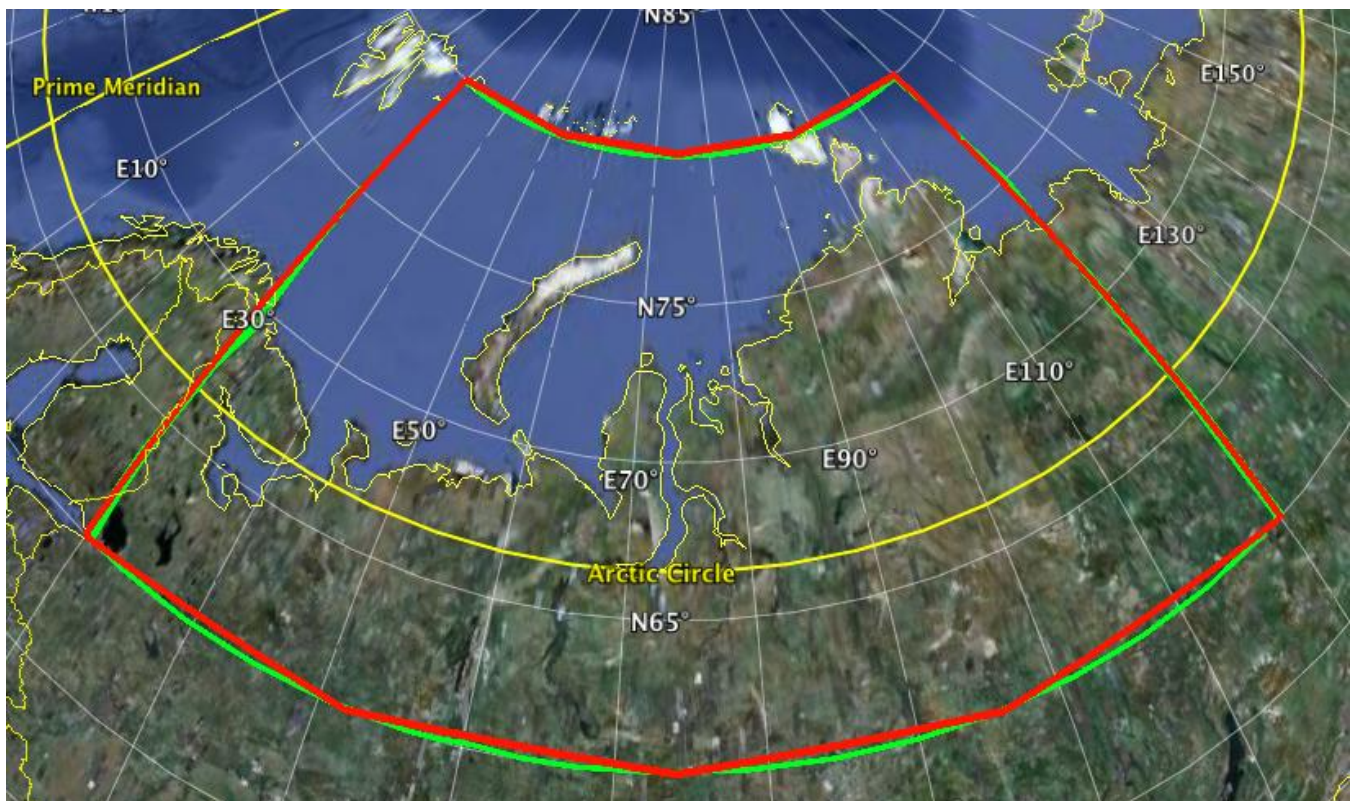
Applying a similar process of densification, as describe previously in conjunction with lines, the polygon could be densified into the following XML metadata and would appear as is shown below.

Code Listing 9. Densified Geodetic Polygon

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>120</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>96</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>74</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>52</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>30</PointLongitude>
```



```
<PointLatitude>60</PointLatitude>  
</Point>  
<Point>  
  <PointLongitude>30</PointLongitude>  
  <PointLatitude>80</PointLatitude>  
</Point>  
<Point>  
  <PointLongitude>52</PointLongitude>  
  <PointLatitude>80</PointLatitude>  
</Point>  
<Point>  
  <PointLongitude>74</PointLongitude>  
  <PointLatitude>80</PointLatitude>  
</Point>  
<Point>  
  <PointLongitude>96</PointLongitude>  
  <PointLatitude>80</PointLatitude>  
</Point>  
<Point>  
  <PointLongitude>120</PointLongitude>  
  <PointLatitude>80</PointLatitude>  
</Point>  
</Boundary>  
</GPolygon>  
</Geometry>
```



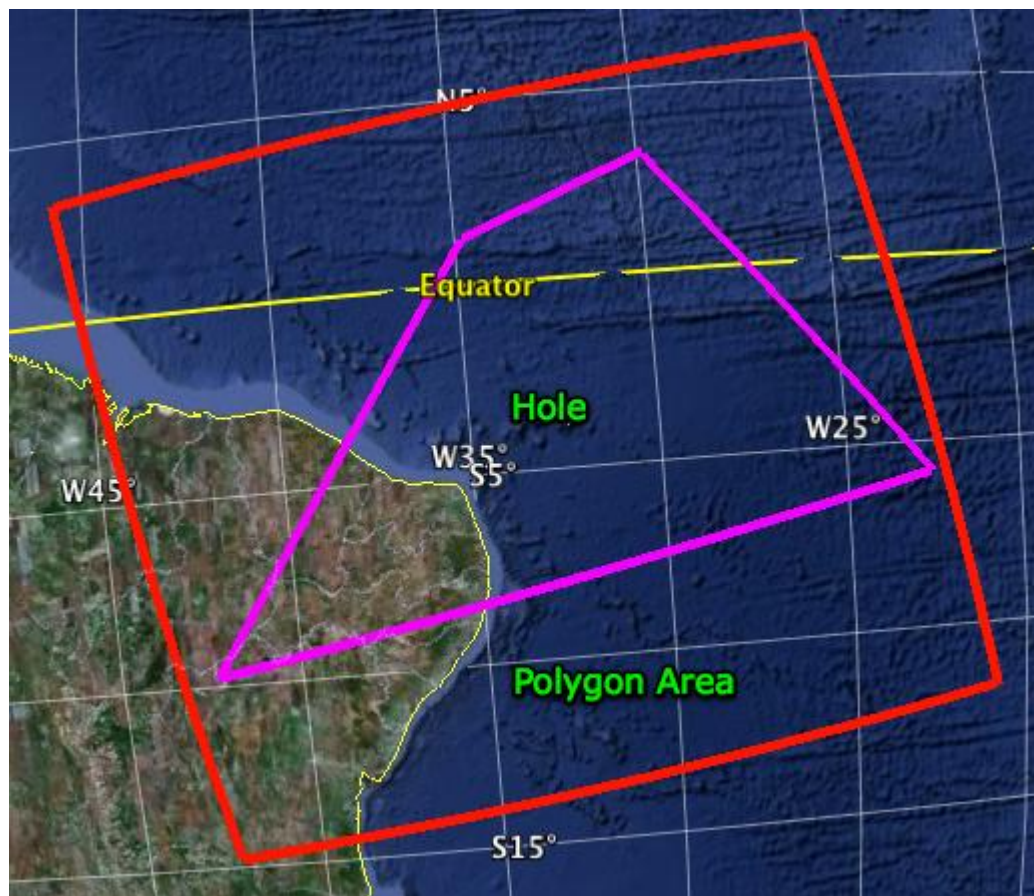
The following sample metadata shows a single polygon with a 'hole' in its spatial coverage. The figure below shows how this polygon will be represented in the Geodetic coordinate system.

Code Listing 10. Single Polygon with a Hole

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>-20.9342</PointLongitude>
        <PointLatitude>-11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-42.3067</PointLongitude>
        <PointLatitude>-14.7732</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-45.7985</PointLongitude>
        <PointLatitude>3.198</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-24.8982</PointLongitude>
        <PointLatitude>6.1665</PointLatitude>
      </Point>
    </Boundary>
    <ExclusiveZone>
      <Boundary>
        <Point>
          <PointLongitude>-22.9342</PointLongitude>
          <PointLatitude>-5.9045</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-42.3067</PointLongitude>
          <PointLatitude>-9.7732</PointLatitude>
        </Point>

        <Point>
          <PointLongitude>-34.7985</PointLongitude>
          <PointLatitude>1.198</PointLatitude>
        </Point>

        <Point>
          <PointLongitude>-29.8982</PointLongitude>
          <PointLatitude>3.1665</PointLatitude>
        </Point>
      </Boundary>
    </ExclusiveZone>
  </GPolygon>
</Geometry>
```



While a single polygon with a hole can have only one outer ring that represents the area surrounded within, it can have multiple inner rings that represent holes. All the rules, restrictions and discussions for the outer ring in both coordinate systems apply to inner rings as well. An inner ring should be completely contained within the outer ring.

3.2.2.5 Bounding Rectangle

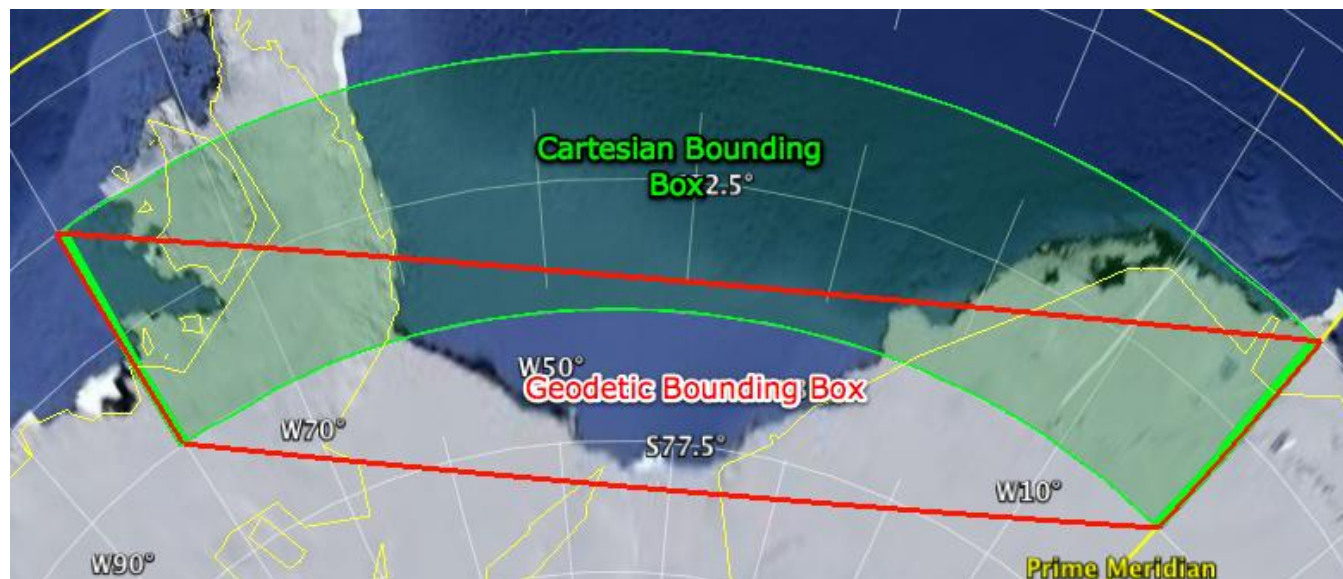
In the Cartesian and Geodetic coordinate system, ECHO is capable of receiving, storing and supporting the search on spatial data representing one or more bounding rectangles. In the XML metadata, follow the syntax shown in the following code sample to define a spatial extent as a bounding rectangle:

:

Code Listing 11. Bounding Rectangle

```
<Geometry>
  <BoundingRectangle>
    <WestBoundingCoordinate>-80</WestBoundingCoordinate>
    <NorthBoundingCoordinate>-70</NorthBoundingCoordinate>
    <EastBoundingCoordinate>0</EastBoundingCoordinate>
    <SouthBoundingCoordinate>-75</SouthBoundingCoordinate>
  </BoundingRectangle>
</Geometry>
```


The figure below represents the spatial area covered when applying the code shown above. The Geodetic Bounding Rectangle is shown in red and the Cartesian Bounding Rectangle is shown in yellow.



ECHO stores a bounding rectangle as a four-pointed polygon, subject to the specifications and constraints described for the polygon

3.3 INVALID SPATIAL REPRESENTATIONS

The following sections outline specific instances where ECHO will consider a spatial area invalid.

3.3.1 Polygon Points in Counter-Clockwise Order

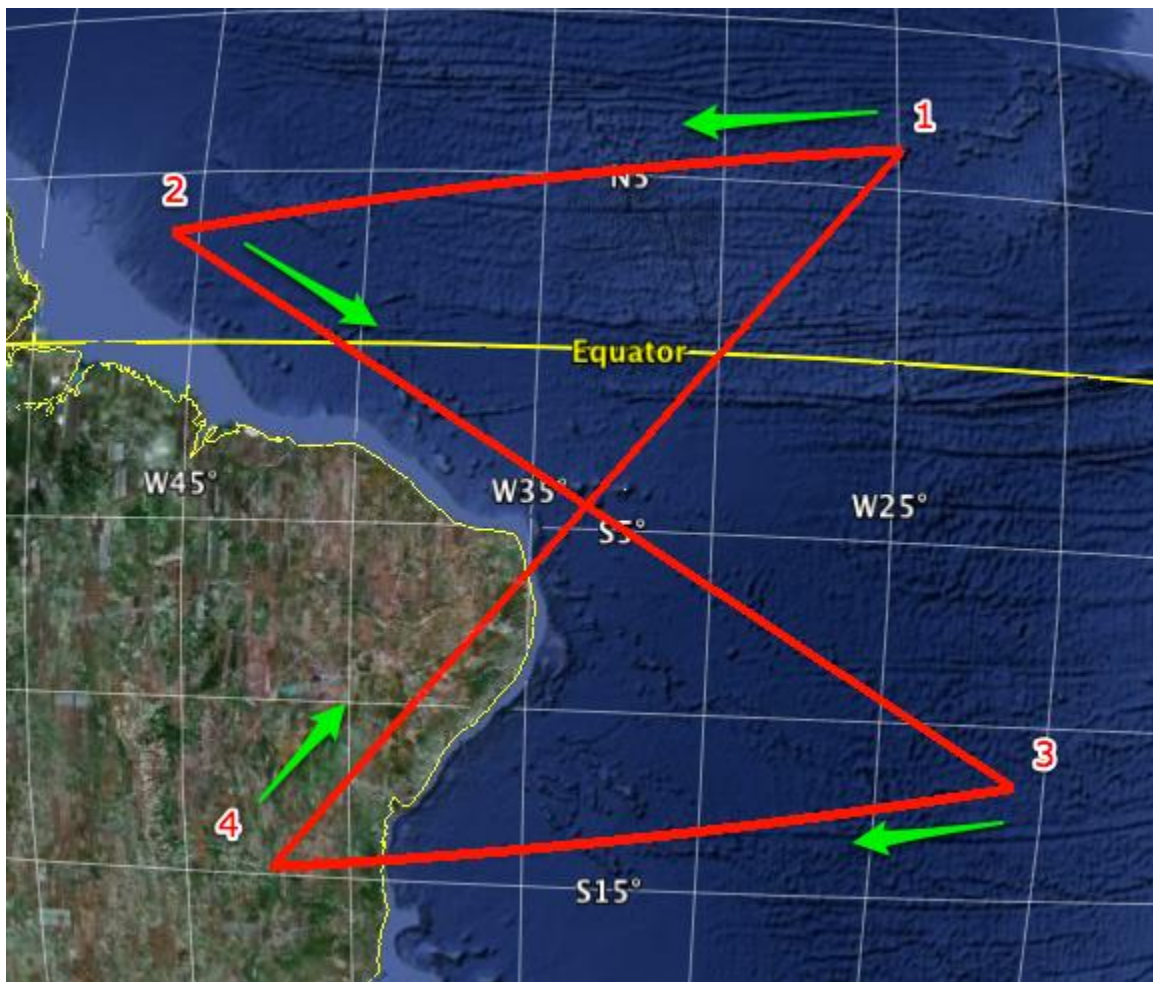
This spatial area expression is invalid in the Cartesian coordinate system. However, this same expression may be considered valid in the Geodetic coordinate system if the inversion does not cause the coverage to be more than one half of the Earth. Although ECHO may accept this polygon, the coverage will be interpreted very differently. The following sample shows a polygon with points in reversed, counter-clockwise, order.

Code Listing 12. Polygon with Points in Counter-Clockwise Order

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>170</PointLongitude>
        <PointLatitude>30</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-170</PointLongitude>
        <PointLatitude>30</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-170</PointLongitude>
        <PointLatitude>-30</PointLatitude>
      </Point>
    </Boundary>
  </GPolygon>
</Geometry>
```

```
<Point>  
  <PointLongitude>170</PointLongitude>  
  <PointLatitude>-30</PointLatitude>  
</Point>  
</Boundary>  
</GPolygon>  
</Geometry>
```

3.3.2 Twisted Polygon



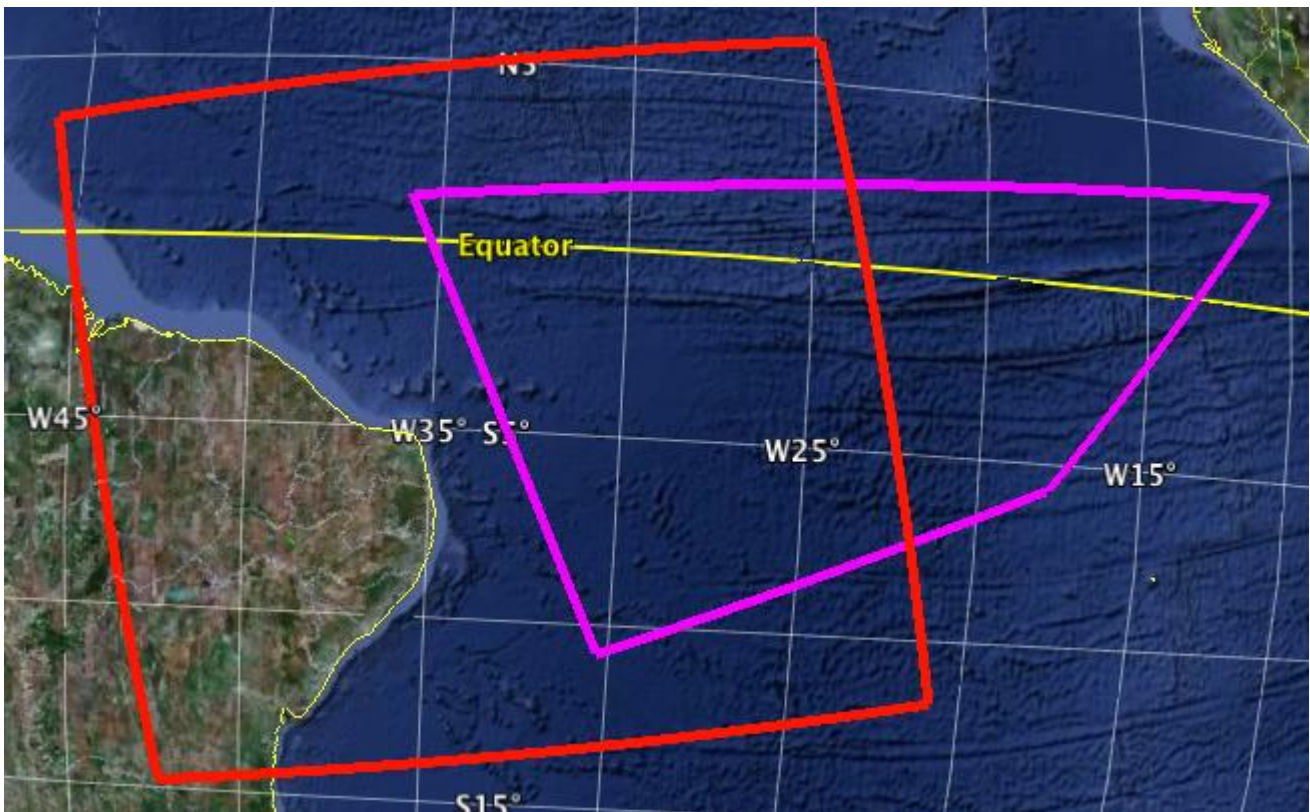
In the case where the points in a polygon cause the line segments to cross, ECHO will reject this as invalid spatial data in both the Cartesian and the Geodetic coordinate systems. The figure above shows how this is represented on the Earth and the following code sample shows the same invalid area.

Code Listing 13. Twisted Polygon

```
<Geometry>  
  <GPolygon>
```

```
<Boundary>  
  <Point>  
    <PointLongitude>-20.9342</PointLongitude>  
    <PointLatitude>-11.7045</PointLatitude>  
  </Point>  
  <Point>  
    <PointLongitude>-42.3067</PointLongitude>  
    <PointLatitude>-14.7732</PointLatitude>  
  </Point>  
  <Point>  
    <PointLongitude>-24.8982</PointLongitude>  
    <PointLatitude>6.1665</PointLatitude>  
  </Point>  
  <Point>  
    <PointLongitude>-45.7985</PointLongitude>  
    <PointLatitude>3.198</PointLatitude>  
  </Point>  
</Boundary>  
</GPolygon>  
</Geometry>
```

3.3.3 Hole Crosses over Outer Ring

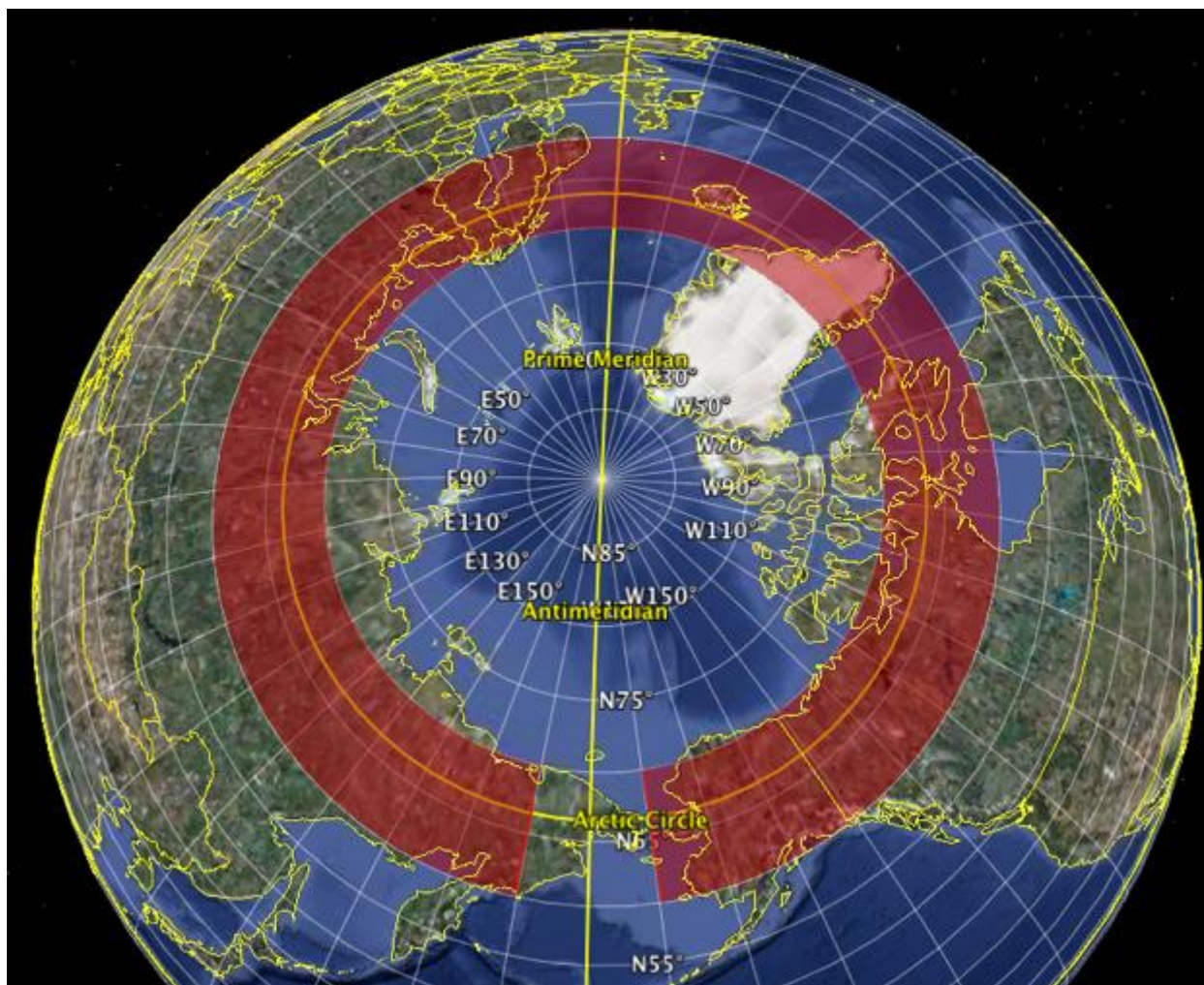


In the case where the an exclusion area (hole) intersects with the bounding polygon, ECHO will reject this as invalid spatial data in both the Cartesian and the Geodetic coordinate systems. The figure above shows how this is represented on the Earth and the following code sample shows the same invalid area.

Code Listing 14. Hole Crosses over the Outer Ring

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>-20.9342</PointLongitude>
        <PointLatitude>-11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-42.3067</PointLongitude>
        <PointLatitude>-14.7732</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-45.7985</PointLongitude>
        <PointLatitude>3.198</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-24.8982</PointLongitude>
        <PointLatitude>6.1665</PointLatitude>
      </Point>
    </Boundary>
    <ExclusiveZone>
      <Boundary>
        <Point>
          <PointLongitude>-17.9342</PointLongitude>
          <PointLatitude>-5.7045</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-30.3067</PointLongitude>
          <PointLatitude>-10.7732</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-35.7985</PointLongitude>
          <PointLatitude>1.198</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-10.8982</PointLongitude>
          <PointLatitude>3.1665</PointLatitude>
        </Point>
      </Boundary>
    </ExclusiveZone>
  </GPolygon>
</Geometry>
```

3.3.4 Polygon Crosses Antemeridian or Pole



ECHO will not allow a Cartesian polygon to cross the international dateline or the poles. The lines will be connected the long way around the earth as shown above. The figure above shows how this is represented on the Earth and the following code sample shows the same invalid area. To correct this situation, the polygon should be split into two or more spatial areas that are split at the Antemeridian and/or poles as needed.

Code Listing 15. Polygon Crosses International Dateline

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>170</PointLongitude>
        <PointLatitude>70</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>170</PointLongitude>
        <PointLatitude>60</PointLatitude>
      </Point>
      <Point>
```



```
        <PointLongitude>-170</PointLongitude>
        <PointLatitude>60</PointLatitude>
    </Point>
    <Point>
        <PointLongitude>-170</PointLongitude>
        <PointLatitude>70</PointLatitude>
    </Point>
</Boundary>
</GPolygon>
</Geometry>
```

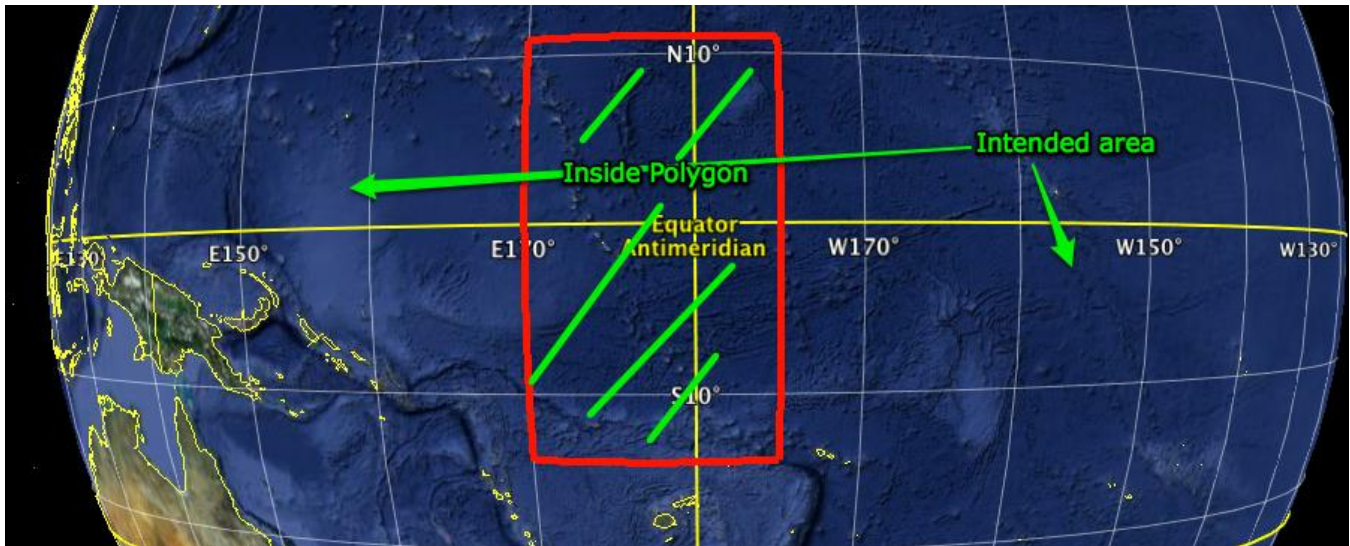
3.3.5 Inappropriate Point Density

Due to the nature of how spatial data is represented in the Geodetic coordinate system, it is possible that ECHO will validate a low density polygon, even though the spatial area does not represent what is expected. As an example, consider the following sample polygon metadata.

Code Listing 16. Incorrect Density

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>170.9342</PointLongitude>
        <PointLatitude>11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-175.3067</PointLongitude>
        <PointLatitude>11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-175.3067</PointLongitude>
        <PointLatitude>-13.198</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>170.9342</PointLongitude>
        <PointLatitude>-13.198</PointLatitude>
      </Point>
    </Boundary>
  </GPolygon>
</Geometry>
```

The expression above is valid spatial data in the Geodetic coordinate system. However, the spatial coverage area represented will be as shown below:



The area shown outside the left and right of the red polygon may have been expected, but the area inside the red polygon is what will be used for spatial comparison. To represent this spatial coverage correctly, you must increase the point density by adding extra points. The sample below shows one way you might express these additional points, to represent this spatial coverage area correctly.

Code Listing 17. Correct Density

```
<Geometry>
  <GPolygon>
    <Boundary>
      <Point>
        <PointLongitude>170.9342</PointLongitude>
        <PointLatitude>11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>0.0</PointLongitude>
        <PointLatitude>11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-175.3067</PointLongitude>
        <PointLatitude>11.7045</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>-175.3067</PointLongitude>
        <PointLatitude>-13.198</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>0.0</PointLongitude>
        <PointLatitude>-13.198</PointLatitude>
      </Point>
      <Point>
        <PointLongitude>170.9342</PointLongitude>
        <PointLatitude>-13.198</PointLatitude>
      </Point>
    </Boundary>
  </GPolygon>
</Geometry>
```

```
</GPolygon>  
</Geometry>
```

3.3.6 Tolerance

ECHO makes use of resolution parameter settings to associate a level of precision with spatial data. ECHO uses these parameters as evaluation parameters when validating spatial data input. Cartesian tolerance is specified as fractions of a degree and Geodetic tolerance is specified in meters.. If the Cartesian tolerance is 0.05 for both latitude and longitude, and if the distance between two points is less than 0.05 degrees for both longitude and latitude, then those two points are considered the same point. In this situation, the spatial expression is invalid because ECHO spatial constructs require each point to have a unique spatial location.

ECHO defaults for tolerance is:

- Cartesian Tolerance: .0001 degrees
- Geodetic Tolerance: 5 centimeters

3.4 ORBIT DATA

Orbit searching is by far the most accurate way to search for level 0-2 orbital swath data. Unfortunately orbital mechanics is a quite difficult field, and the most well known orbit model, the NORAD Propagator, is quite complex. The NORAD Propagator is designed to work with a wide range of possible orbits, from circular to extremely elliptical, and consequently requires quite a bit of information about the orbit to model it well.

To facilitate earth science, the orbits of satellites gathering earth science data are quite restricted compared to the variety of orbits the NORAD Propagator is designed to work with. Generally, the earth science community would like global coverage, with a constant field of view, at the same time every day. For this reason, most earth science satellites are in a sun-synchronous, near-polar orbit. Even missions that are not interested in global coverage, e.g., the Tropical Rainfall Measuring Mission (TRMM), are still interested in having a constant field of view so the coverage of the sensor is at a constant resolution. For this reason, ALL earth science satellites are in circular orbits.

The Backtrack Orbit Search Algorithm, designed and developed by Ross Swick, exploits this fact to simplify the orbit model by modeling an orbit as a great circle under which the Earth rotates. This reduces the number of orbital elements required for the model from 22 to three. Moreover, the NORAD Propagator is designed to predict future orbits based on current status, and consequently must be reinitialized periodically to correct for cumulative error as the model spins forward. As the name implies Backtrack spins the orbit backwards, and in practice spins backwards at most one orbit, so there is no cumulative error.

For more information on Backtrack, please see <http://geospatialmethods.org/bosa/>.

Note: Orbit granules may not be ingested unless the parent collections have orbit parameters defined.

3.4.1.1 Backtrack Orbit Model

Three parameters to define an orbit:

- a. Instrument swath width (in kilometers)
- b. Satellite declination or inclination (in degrees)
- c. Satellite period (in minutes)

3.4.1.2 Orbit Data Representation

Three parameters to represent orbit data:

- d. Equatorial crossing longitude (in degrees)
- e. Start circular latitude (or start latitude and start direction)
- f. End circular latitude (or end latitude and end direction)

3.4.1.3 How Data Providers Configure Orbit Data

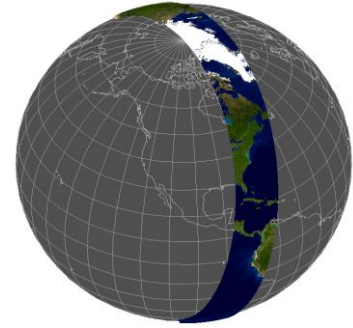


Figure 4. Orbit Swath

Code Listing 18. Add orbit data to Granule Metadata

```
<Spatial>
  <HorizontalSpatialDomain>
    <Orbit>
      <AscendingCrossing>160.14462465545338</AscendingCrossing>
      <StartLat>69.021242</StartLat>
      <StartDirection>D</StartDirection>
      <EndLat>-68.995831</EndLat>
      <EndDirection>A</EndDirection>
    </Orbit>
  </HorizontalSpatialDomain>
</Spatial>
```

Code Listing 19. Add orbit parameters to Collection Metadata

```
<Spatial>
  <SpatialCoverageType>Horizontal</SpatialCoverageType>
  <OrbitParameters>
    <SwathWidth>400</SwathWidth>
    <Period>98.88</Period>
    <InclinationAngle>98.2</InclinationAngle>
  </OrbitParameters>
  <GranuleSpatialRepresentation>ORBIT</GranuleSpatialRepresentation>
</Spatial>
```

3.5 GLOBAL DATA

ECHO does not support an explicit 'global' designation for a collection's or granule's spatial representation. Instead, a Cartesian bounding box with corners covering the entire earth will be interpreted as "global" within ECHO. Due to the usage of a Cartesian coordinate system, the collections and granules will be spatially searchable and be discovered with all spatial areas. ECHO will also facilitate "global only" searching which will only discover metadata items with the spatial geometry shown below.

Code Listing 20. Add global data to Granule Metadata

```
<Spatial>
  <HorizontalSpatialDomain>
    <Geometry>
      <BoundingRectangle>
        <WestBoundingCoordinate>
          -180
        </WestBoundingCoordinate>
        <NorthBoundingCoordinate>
          90
        </NorthBoundingCoordinate>
        <EastBoundingCoordinate>
          180
        </EastBoundingCoordinate>
        <SouthBoundingCoordinate>
          -90
        </SouthBoundingCoordinate>
      </BoundingRectangle>
    </Geometry>
  </HorizontalSpatialDomain>
</Spatial>
```

Code Listing 21. Add global data to Collection Metadata

```
<Spatial>
  <SpatialCoverageType>Horizontal</SpatialCoverageType>
  <HorizontalSpatialDomain>
    <Geometry>
      <CoordinateSystem>CARTESIAN</CoordinateSystem>
      <BoundingRectangle>
        <WestBoundingCoordinate>
          -180
        </WestBoundingCoordinate>
        <NorthBoundingCoordinate>
          90
        </NorthBoundingCoordinate>
        <EastBoundingCoordinate>
          180
        </EastBoundingCoordinate>
        <SouthBoundingCoordinate>
          -90
        </SouthBoundingCoordinate>
      </BoundingRectangle>
    </Geometry>
  </HorizontalSpatialDomain>
  <GranuleSpatialRepresentation>CARTESIAN</GranuleSpatialRepresentation>
</Spatial>
```

CHAPTER 4. METADATA MODEL

4.1 ECHO METADATA CONSTRUCTS

The following three metadata constructs are utilized by the ECHO system:

- **Collection** - A grouping of science data that all come from the same source, such as a modeling group or institution. Collections have information that is common across all the granules they contain and a template for describing additional attributes not already part of the metadata model.
- **Granule** - The smallest aggregation of data that can be independently managed (described, inventoried, and retrieved). Granules have their own metadata model and support values associated with the additional attributes defined by the owning collection.
- **Browse** - An image which provides a high-level view of the associated granule or collection metadata item. Browse images are not spatially enabled, but are very useful during data discovery and cross-referencing to other granules or collections.

A collection within ECHO may contain zero or more granules, however granules cannot exist without being associated to a collection. Browse images may be associated with either collections or granules. Browse images may exist in ECHO without an associated collection or browse, but cannot be discovered in this state. Data Partners should pay close attention to ensure browse image associations are managed closely to eliminate orphaned browse.

4.2 REQUIRED & RECOMMENDED ELEMENTS

This section categorizes a subset of all collection elements into the following perspectives:

- **Required Fields** – The metadata elements that must be present in order to pass the base XML Schema validation.
- **Recommended Fields** – The metadata elements that we recommend to facilitate searching and data usability by the science community.

The metadata name and description of each metadata field are listed in tabular form, along with the data type and relevant constraints, where applicable. Only the top-level elements of complex metadata elements are listed. Where this occurs, the requisite child element information is inherently required to correctly represent the parent metadata element.

For a full description of the ECHO data model, refer to the schema and documentation referenced on the ECHO Website (<http://www.echo.nasa.gov/ingest/schemas/operations/docs/>).

4.2.1 Collections

4.2.1.1 Required Elements

Name	Description
ShortName	This attribute identifies the short name associated with the collection. This is the official reference name used in identifying the contents of the data collection. All characters must be in upper case.
VersionId	This attribute specifies the version identifier of the data collection
InsertTime	This attribute specifies the insert date/time the collection entered the data provider's database. This date is provided by the data provider.
LastUpdate	This attribute specifies the most recent date/time an update occurred in the data provider's database. This date is provided by the data provider.

LongName	This attribute will identify the long name associated with the collection. This is the reference name used in describing the scientific contents of the data collection.
DataSetId	This attribute specifies a unique name for the collection. This information is computed by ECHO according to the data provider's policy.
Description	This attribute identifies the major emphasis of the content of the collection.
Orderable	This attribute indicates whether this collection is orderable.
Visible	This attribute indicates whether the collection is visible. Visibility is a basic access control mechanism that bypasses all ACL rules. If a collection is not visible, only users with the owning provider role will be able to see the item. No other users will be able to see the item—no matter what ACL permissions are in place. If group based permissions are needed, use the Restriction Flag field instead of visibility. Visibility is more commonly set at the collection level than the granule level. If a collection is not visible, none of the granules in the collection will be visible.

4.2.1.2 Recommended Elements

Name	Description
ProcessingLevelId	The processing level class contains the level identifier and level description of the collection.
Price	This attribute specifies the price for ordering the collection
SpatialKeywords	This attribute specifies a word or phrase that serves to summarize the spatial regions covered by the collection. It may be repeated if several regions are covered. This often occurs when a collection is described as covering some large region, and several smaller sub regions within that region.
TemporalKeywords	This attribute specifies a word or phrase that serves to summarize the temporal characteristics referenced in the collection.
Temporal	This attribute contains records that describe the basis of the time system used for a specific collection.
Contact	This attribute contains the basic characteristics for a person or an organization type of contact. These contacts may provide information about a Collection, Delivered Algorithm Package, PGE, or Data Originator. System and user profile contact information is held elsewhere.
ScienceKeywords	This attribute holds a cross reference between collections and science keywords. The Science Keyword list is managed by the Global Master Change Directory (GCMD).
Platform	This attribute describes the relevant platforms associated with the acquisition of the collection or granule. Platform types include Spacecraft, Aircraft, Vessel, Buoy, Platform, Station, Network, Human etc.
Instrument	This entity registers the device used to measure or record data, including direct human observation. In cases where instruments have a single sensor or the instrument and sensor are used synonymously (e.g., AVHRR), both the Instrument and Sensor should be recorded. The Sensor information is represented by other entities.
Sensor	This entity holds the referential information for the collection source/sensor configuration including sensor parameter settings such as technique, etc.
Campaigns	This entity contains attributes describing the scientific endeavor(s) to which the collection is associated. Scientific endeavors include campaigns, projects, interdisciplinary science investigations, missions, field experiments, etc.
TwoDCoordinateSystem	This entity defines the two dimensional coordinate systems for the collection. The two dimensional coordinate system information is an alternative way to express spatial coverage. Granules in the collection that specify two dimensional coordinate data must conform to one of the systems defined by the collection.

Name	Description
OnlineAccessURL	This entity stores the online URL(s) for the granule, if there any. The URL either provides the site the user can obtain granule data or gives further instructions for obtaining the granule data.
OnlineResource	This entity records the documentation information of the collection including documentation type and documentation URL where applicable. These resources may include additional subsetting or processing services available for the granule.
AssociatedDIFs	This entity specifies the collection's Directory Interchange Format (DIF) identifier. The DIF identifier is used to uniquely identify a provider's data set. ECHO coordinates this list with the GCMD portal to facilitate the discovery of collections through GCMD. Providers may specify their own format, but the following is recommended: <ShortName>_V<Version_ID> (ex: MOD14_V005). Notice that the version id is zero padded to be three digits long.
Spatial	This entity contains the collection's spatial coverage information.
ArchiveCenter	Center where the collection is archived.
AdditionalAttributes	This entity stores the product specific attributes (i.e. attributes used to describe the unique characteristics of the collection which extend beyond those defined in this model).
AssociatedBrowseImages	List of browse images associated with the collection.

4.2.2 Granules

4.2.2.1 Required Elements

Name	Description
GranuleUR	The Universal Reference (UR) ID of the granule referred to by the data provider. This ID is unique per data provider.
InsertTime	This attribute is the date/time this granule entered the data provider's database.
LastUpdate	This attribute is the date/time the data provider last updated the granule information in the data provider's database
Collection	This attribute holds the reference information for a granule to a collection. The Granule references the collection either by collection short name and collection version or by collection dataset ID.
Orderable	This attribute indicates whether the granule is orderable.

4.2.2.2 Recommended Elements

Name	Description
DataGranule	This entity stores the basic descriptive characteristics associated with a granule.
PGEVersionClass	This entity stores basic descriptive characteristics related to the Program Generation Executable (PGE) associated with a granule.
Temporal	This entity contains records that describe the basis of the time system used for a specific collection.
Spatial	This entity contains the granule's spatial coverage information.

Name	Description
MeasuredParameters	<p>This attribute specifies the names of the geophysical parameters expressed in the data as well as associated quality flags and quality status.</p> <p>The quality status contains measures of quality for the granule. The parameters used to set these measures are not preset and will be determined by the data producer. Each set of measures can occur many times either for the granule as a whole or for individual parameters.</p> <p>The quality flags contain the science, operational, and automatic quality flags that indicate the overall quality assurance levels of specific parameter values within a granule.</p>
Platform	<p>This attribute describes the relevant platforms associated with the acquisition of the collection or granule. Platform types include Spacecraft, Aircraft, Vessel, Buoy, Platform, Station, Network, Human, etc.</p>
Instrument	<p>This attribute registers the device used to measure or record data, including direct human observation. In cases where instruments have a single sensor or the instrument and sensor are used synonymously (e.g., AVHRR) both the Instrument and Sensor should be recorded. The Sensor information is represented by other entities.</p>
Sensor	<p>This attribute holds the referential information for the granule source/sensor configuration including sensor parameter settings such as technique, etc.</p>
Campaigns	<p>This entity contains attributes describing the scientific endeavor(s) to which the granules is associated. Scientific endeavors include campaigns, projects, interdisciplinary science investigations, missions, field experiments, etc.</p>
DataFormat	<p>This attribute contains the file format of the raw data (such as HDF) for this granule</p>
TwoDCoordinateSystem	<p>This entity stores the two dimensional coordinate system information for the granule. The two dimensional coordinate system information is an alternative way to express granule's spatial coverage based on a certain two dimensional coordinate system defined by the providers.</p>
Price	<p>This attribute specifies the price of the granule data when ordered.</p>
OnlineAccessURL	<p>This entity stores the online URL(s) for the granule, if any. The URL either provides the site the user can obtain granule data or gives further instructions for obtaining the granule data.</p>
OnlineResource	<p>This entity records the documentation information of the granule including documentation type and documentation URL where applicable.</p> <p>These resources may include additional subsetting or processing services available for the granule.</p>
CloudCover	<p>A percentage value indicating how much of the area of a granule (the ECS data unit) has been obscured by clouds. It is worth noting that there are many different measures of cloud cover within the ECS data holdings and that the cloud cover parameter that is represented in the DataPool is specific to Data Set.</p>
AssociatedBrowseImages	<p>The list of associated browse images to this granule</p>
AdditionalAttributes	<p>This entity stores the Product Specific Attributes with value a granule associates. The attribute name and type must exist in the parent collection for this granule.</p>

4.2.3 Browse

4.2.3.1 Required Elements

Name	Description
ProviderBrowseId	<p>The unique id for the browse image given by the provider. This should be unique per provider.</p>
FileName*	<p>The name of the delivered browse image</p>
FileSize*	<p>The size of the delivered browse image in bytes. A browse image delivered that is a different size than indicated will be rejected.</p>

Name	Description
FileURL**	The URL for this browse image. Include file URL when a browse image is not hosted by ECHO, e.g. a browse image is hosted by the provider.

* Providers must provide the browse file name and size if ECHO will be hosting the browse image.

** Providers must provide the browse URL if the browse image will be hosted external to ECHO.

4.2.3.2 Recommended Elements

Name	Description
InsertTime	This attribute specifies the date/time this browse image was created in the data provider's database.
LastUpdate	This attribute specifies the last time the browse image was modified in the provider's database.

4.3 TEMPORAL DATA

Temporal metadata refer to the date and time associated with the data represented within a collection or granule. This is also commonly referred to as "acquisition date and time." Temporal data is an essential search criteria for collections and granules within ECHO. Temporal information is not a required element for collections or granules. However, if a granule specifies temporal information, the collection must also. ECHO Ingest will validate that a granule's temporal data falls within the range of its owning collection's temporal data. If a collection specifies an open ended temporal range, ECHO will accept all granules that provide temporal data that is subsequent to the collection's starting date.

4.3.1 Collections

A collection may be associated with one or more of the following three types of temporal expressions:

- **Single Date Time** – A single date and time.
- **Range Date Time** – A date and time range specified by a beginning and ending date and time. An ending date is not required in order to designate an on-going collection.
- **Periodic Date Time** – A repeating date and time range specified by a beginning and ending date and time with a period cycle and duration.

A collection may provide more than one instance of a single type of temporal expression. For instance, there may be two ranges associated with the collection signifying that there was a period of time for which no data was collected. ECHO will facilitate discovery of all temporal information, however it will also calculate an internal beginning and ending date/time based upon the least or greatest (respectively) value provided in the list of temporal expressions.

These expressions are used to provide the necessary information in order to uniquely identify the temporal range for which data may be discovered within the collection. In addition to these items, you may also specify the following items providing users with additional information regarding the temporal specifications of the collection:

- **Time Type** – The time system which time values found in the temporal ranges represent. (e.g. UTC)
- **Date Type** – The type of date represented by the value in the temporal ranges represent. (e.g. Gregorian)
- **Temporal Range Type** – Designates how the temporal coverage is specified. (e.g. Continuous Range)
- **Precision of Seconds** – The precision of seconds used in the measurement
- **Ends At Present** – Boolean flag denoting that a data collection which covers, temporally, a discontinuous range, currently ends at the present date. This way, the granules, which comprise the data collection, that are continuously being added to inventory need not update the data collection metadata for each one.

Code Listing 22. Expression of Temporal Information (RangeDateTime)

```
<Temporal>
  <TimeType>UTC</TimeType>
  <DateType>Gregorian</DateType>
  <TemporalRangeType>Continuous Range</TemporalRangeType>
  <PrecisionofSeconds>1</PrecisionofSeconds>
  <EndsatPresentFlag>Y</EndsatPresentFlag>
  <RangeDateTime>
    <BeginningDateTime>1998-01-01T00:00:00.0Z</BeginningDateTime>
  </RangeDateTime>
</Temporal>
```

4.3.2 Granules

A granule may be associated with one of the following two types of temporal expressions:

- **Single Date Time** – A single date and time.
- **Range Date Time** – A date and time range specified by a beginning and ending date and time. An ending date is not required in order to designate an on-going granule.

A granule may provide more than one instance of a single type of temporal expression. For instance, there may be two ranges associated with the granule signifying that there was a period of time for which no data was collected. ECHO will facilitate discovery of all temporal information, however it will also calculate an internal beginning and ending date/time based upon the least or greatest (respectively) value provided in the list of temporal expressions.

These expressions are used to provide the necessary information in order to uniquely identify the temporal range for which data in the granule represents.

Code Listing 23. Expression of Temporal Information (RangeDateTime)

```
<Temporal>
  <RangeDateTime>
    <BeginningDateTime>1998-01-01T00:00:00.0Z</BeginningDateTime>
    <EndingDateTime>1998-01-01T00:00:00.0Z</EndingDateTime>
  </RangeDateTime>
</Temporal>
```

4.4 SPATIAL DATA

4.4.1 Cartesian/Geodetic/Orbital Information

For a full description of ECHO spatial representations, please refer to section 3.0 of this document.

4.4.2 Two-Dimensional Coordinate System

In order to facilitate discovery based upon an additional mechanism, a two-dimensional coordinate system may be used. Examples of two-dimensional coordinate system values are path/row for Worldwide Reference System

(WRS) data and Moderate Resolution Imaging Spectroradiometer (MODIS) tiles IDs. A collection must define all two dimensional coordinate systems that granules in that collection may use. ECHO Ingest will validate that a granule's TwoDCoordinateSystem element references a valid coordinate system defined in the collection. If the collection's definition for the coordinate system defines minimum and maximum values for the system axes, ECHO Ingest will validate the granule's metadata against these constraints.

Note: Due to manual configuration needs for to facilitate WIST discovery of Two-dimensional coordinate systems, please coordinate with ECHO Operations to ensure searching will be enabled.

4.4.2.1 Coordinate System Axis Labels

The ECHO data model does not allow providers to configure axis labels for each TwoDCoordinateSystem. This was chosen to avoid issues where the same coordinate system in two collections contains different labels for the same axis. This would be a significant issue for ECHO Client Developers. ECHO has chosen to identify the following unique coordinate systems, with the axis labels included below. This listing of labels is recommended for ECHO Client Partners and is used in WIST. If additional coordinate systems are utilized, Data Partners should coordinate with ECHO Operations to ensure that the labeling recommendations are correct.

- **WRS-1**
 - **X Axis** – “Path”
 - **Y Axis** – “Row”
- **WRS-2**
 - **X Axis** – “Path”
 - **Y Axis** – “Row”
- **MISR**
 - **X Axis** – “Path”
 - **Y Axis** – “Block”
- **MODIS Tile**
 - **X Axis** – “Horizontal Tile #”
 - **Y Axis** – “Vertical Tile #”
- **Calipso**
 - **X Axis** – “Orbit”
 - **Y Axis** – “Path”

4.4.2.2 Collections

The Two-Dimensional Coordinate System information that exists at the collection level and contains the following information:

- **TwoDCoordinateSystemName** – The identifying name of the coordinate system
- **Start/End Coordinate1** – The ‘X’ axis coordinate minimum and maximum range values
- **Start/End Coordinate2** – The ‘Y’ axis coordinate minimum and maximum range values

The minimum and maximum range values are not required for both the 'X' and 'Y' axes. This allows for finite, infinite (in both directions), or no range validation for a specific coordinate system axis. The following example provides a finite range for Coordinate1 and infinite ending for Coordinate 2.

Code Listing 24. Collection Two-Dimensional Coordinate System Coordinates

```
<TwoDCoordinateSystem>
  <TwoDCoordinateSystemName>WRS2</TwoDCoordinateSystemName>
  <Coordinate1>
    <MinimumValue>0</MinimumValue>
    <MaximumValue>38</MaximumValue>
  </Coordinate1>
  <Coordinate2>
    <MinimumValue>0</MinimumValue>
  </Coordinate2>
</TwoDCoordinateSystem>
```

4.4.2.3 Granules

The Two-Dimensional Coordinate System information that exists at the granule level and contains the following information:

- **Start/End Coordinate1** – The 'X' axis start and end (not required) coordinate values
- **Start/End Coordinate2** – The 'Y' axis start and end (not required) coordinate values
- **TwoDCoordinateSystemName** – The identifying name of the coordinate system

Code Listing 25. Granule Two-Dimensional Coordinate System Coordinates

```
<TwoDCoordinateSystem>
  <StartCoordinate1>21</StartCoordinate1>
  <StartCoordinate2>29</StartCoordinate2>
  <EndCoordinate2>33</EndCoordinate2>
  <TwoDCoordinateSystemName>WRS2</TwoDCoordinateSystemName>
</TwoDCoordinateSystem>
```

4.5 ADDITIONAL ATTRIBUTES

Additional attributes are parameters, also known as Provider-Specific Attributes (PSAs), which further describe the data represented in each granule within a collection. These values are important search criteria for the granules. Example attributes include values for cloud cover, MODIS Tile grid coordinates, and elevation information. All additional attribute definitions are included in the collection metadata. A collection may also specify a value, to be understood as the value for all granules. Granules reference defined additional attributes and supply a value that is associated to that granule. Granules may not define a new additional attribute that is not defined by the collection.

The following table shows the supported additional attribute data types that within ECHO. The "XML Type" table lists the XML schema type that will be used to validate any attribute values specified in the collection of granules. The "Range Validation" column specifies whether range values are allowed at the collection level, and, if present, whether granules additional attribute values will be validated against the supplied ranges. The "**_STRING" types

allow providers to specify date, time, and datetime attribute values without the associated data validation performed for the pure date and time types.

Attribute Type	XML Type	Range Validation
STRING	string	No
FLOAT	float	Yes
INT	int	Yes
BOOLEAN	boolean	No
DATE	date	Yes
TIME	time	Yes
DATETIME	dateTime	Yes
DATE_STRING	string	No
TIME_STRING	string	No
DATETIME_STRING	string	No

4.5.1 Collections

An additional attribute, as defined within collection metadata, must include the following information:

- **Name** – Unique name of the additional attribute.
- **Data Type** – Data type of the additional attribute, chosen from the supported list of types.
- **Description** – A textual description of the attribute to help end users understand the purpose and data represented by the attribute.

Your additional attribute definition may also contain the following elements providing additional information:

- **MeasurementResolution** – Identifies the smallest unit increment to which the parameter value is measured.
- **ParameterRangeBegin** – Minimum value of all attribute values that will be provided by the collection or granules.
- **ParameterRangeEnd** – Maximum value of all attribute values that will be provided by the collection or granules.
- **ParameterUnitsOfMeasure** – Unit of measure for parameter (e.g. AVHRR: unit of geophysical parameter-units of geophysical parameter.)
- **ParameterValueAccuracy** – Estimate of the accuracy of the assignment of attribute value. This can be specified in percent or the unit with which the parameter is measured.
- **ValueAccuracyExplanation** – Defines the method used for determining the parameter value accuracy.

As was discussed previously, ECHO Ingest will perform value validation based upon the attribute's data type and range values. See the table in the previous section for more information regarding validation.

Code Listing 26. Collection-Level Additional Attributes

```
<AdditionalAttributes>
```

```
<AdditionalAttribute>
  <Name>PROCESSVERSION</Name>
  <DataType>STRING</DataType>
  <Description>Version of the software generating the product</Description>
</AdditionalAttribute>
<AdditionalAttribute>
  <Name>HORIZONTALTILENUMBER</Name>
  <DataType>INT</DataType>
  <Description>
    Horizontal grid tile number, increasing from left to right.
  </Description>
  <ParameterRangeBegin>1</ParameterRangeBegin>
  <ParameterRangeEnd>100</ParameterRangeEnd>
</AdditionalAttribute>
</AdditionalAttributes>
```

4.5.2 Granules

An additional attribute, as defined within granule metadata, must include the following information:

- **Name** – Unique name of the additional attribute as defined by the collection.
- **Values** – One or more values for the additional attribute.

As was discussed previously, ECHO Ingest will validate that an additional attribute with the same name is defined within the granule's collection. Also, Ingest may perform value validation based upon the attribute's data type and range values. See the table in the previous section for more information regarding validation. If more than one value is provided, ECHO will preserve the order of the values when queried through the ECHO API.

Code Listing 27. Granule level additional attributes

```
<AdditionalAttributes>
  <AdditionalAttribute>
    <Name>VERTICALTILENUMBER</Name>
    <Values>
      <Value>12</Value>
    </Values>
  </AdditionalAttribute>
  <AdditionalAttribute>
    <Name>TileID</Name>
    <Values>
      <Value>51013012</Value>
      <Value>51013013</Value>
    </Values>
  </AdditionalAttribute>
</AdditionalAttributes>
```

4.6 PLATFORMS, INSTRUMENTS, AND SENSORS

ECHO adopts a layered representation of platforms, instruments (also known as sources), and sensors in collection and granule metadata. This layered representation is as follows:

Platform->* Instrument->* Sensor

All possible platforms, instruments, and sensors that may exist within the granules must be defined with the collection metadata. A collection could be associated with zero (0) or more platforms; each platform could contain zero (0) or more instruments, and each instrument could contain zero (0) or more sensors. Each item is uniquely identified by that item's **ShortName** element value. Granules reference defined platform/instrument/sensor combinations associated to that granule. Granules may not define a new platform/instrument/sensor combination that is not defined by the collection.

The platform, instrument, and sensor must comply with the GCMD standard, located at <http://gcmd.gsfc.nasa.gov/Resources/valids/index.html>.

4.6.1 Collections

A platform has the following information that is included in the collection definition:

- **ShortName** – Unique name of the platform within the collection.
- **LongName** – The expanded or long name of the platform associated with an instrument. (Required)
- **Type** – The most relevant platform type. (Required)
- **Characteristics** – The characteristics of platform specific attributes. The characteristic names must be unique on this platform; however the names do not have to be unique across platforms.

An instrument has the following information that is included in the collection definition:

- **ShortName** – Unique name of the platform within the collection.
- **LongName** – The expanded name of the primary sensory instrument.
- **Technique** - Technique applied for this instrument in the configuration.
- **NumberOfSensors** - Number of sensors used on the instrument when acquire the granule data.
- **Characteristics** - The characteristics of this instrument expressed as custom attributes. The characteristic names must be unique on this instrument; however the names do not have to be unique across instruments.
- **OperationModes** - The operation mode applied on the instrument when acquire the granule data.

A sensor has the following information that is included in the collection definition:

- **ShortName** – Unique name of the sensor within the collection.
- **LongName** – The expanded name of the sensor.
- **Technique** - Technique applied for this sensor in the configuration.
- **Characteristics** - The characteristics of this sensor expressed as custom attributes. The characteristic names must be unique on this instrument; however the names do not have to be unique across sensor.

Code Listing 28. Full Platform/Instrument/Sensor Description

```
<Platform>
  <ShortName>Terra</ShortName>
  <LongName>
    First EOS Polar Orbiting Satellite, 10:30 AM Descending Equator Crossing
  </LongName>
  <Type>Spacecraft</Type>
  <Characteristics>
    <Characteristic>
      <Name>EquatorCrossingTime</Name>
      <Description>
        Local time of the equator crossing and direction
        (ascending or descending)
      </Description>
      <DataType>varchar</DataType>
      <Unit>Local Mean Time</Unit>
      <Value>10:30, descending</Value>
    </Characteristic>
  </Characteristics>
  <Instruments>
    <Instrument>
      <ShortName>MODIS</ShortName>
      <LongName>
        Moderate-Resolution Imaging Spectroradiometer
      </LongName>
      <Technique>Imaging Spectroradiometry</Technique>
      <NumberOfSensors>2</NumberOfSensors>
      <Characteristics />
      <Sensors>
        <Sensor>
          <ShortName>MODIS</ShortName>
          <LongName>Cross-track Scanning Radiometer</LongName>
          <Technique>Radiometry</Technique>
          <Characteristics />
        </Sensor>
      </Sensors>
    </Instrument>
  </Instruments>
</Platform>
```

4.6.2 Granules

A platform has the following information that is included in the granule metadata:

- **ShortName** – Unique name of the platform within the collection.

An instrument has the following information that is included in the granule metadata:

- **ShortName** – Unique name of the platform within the collection.

- **Characteristics** - The characteristics of this instrument expressed as custom attributes. The characteristic names must be unique on this instrument; however the names do not have to be unique across instruments.
- **OperationModes** - The operation mode applied on the instrument when acquire the granule data.

A sensor has the following information that is included in the granule metadata:

- **ShortName** – Unique name of the sensor within the collection.
- **Characteristics** - The characteristics of this sensor expressed as custom attributes. The characteristic names must be unique on this instrument; however the names do not have to be unique across sensor.

As was discussed previously, ECHO Ingest will validate that an additional attribute with the same name is defined within the granule's collection. Also, Ingest may perform value validation based upon the attribute's data type and range values. See the table in the previous section for more information regarding validation. If more than one value is provided, ECHO will preserve the order of the values when queried through the ECHO API.

Code Listing 29. Sources and Sensors

```
<Platform>
  <ShortName>Terra</ShortName>
  <Instruments>
    <Instrument>
      <ShortName>MODIS</ShortName>
      <Sensors>
        <Sensor>
          <ShortName>MODIS</ShortName>
        </Sensor>
      </Sensors>
    </Instrument>
  </Instruments>
</Platform>
```

4.7 MEASURED PARAMETERS

Measured parameters are associated only at the granule level only and are important search criteria for granules. For some providers, the value of certain measured parameters determines the visibility of the granule.

Measured parameters contain the name of the geophysical parameter expressed in the data as well as associated quality flags and quality status. The quality status contains measures of quality for the granule. The parameters used to set these measures are not preset and will be determined by the data producer. Each set of measures can occur many times either for the granule as a whole or for individual parameters. The quality flags contain the science, operational and automatic quality flags that indicate the overall quality assurance levels of specific parameter values within a granule.

A measured parameter is uniquely identified by its **ParameterName** element, and has the following information:

- **QAStats** – The name of the geophysical parameter expressed in the data as well as associated quality flags and quality status.
 - **QAPercentMissingData** - Granule level % missing data. This attribute can be repeated for individual parameters within a granule.
 - **QAPercentOutOfBoundsData** – Granule level % out of bounds data. This attribute can be repeated for individual parameters within a granule.

- **QAPercentInterpolatedData** – Granule level % interpolated data. This attribute can be repeated for individual parameters within a granule.
- **QAPercentCloudCover** – This attribute is used to characterize the cloud cover amount of a granule. This attribute may be repeated for individual parameters within a granule. (Note - there may be more than one way to define a cloud or it's effects within a product containing several parameters; i.e. this attribute may be parameter specific)
- **QAFlags** – The name of the geophysical parameter expressed in the data as well as associated quality flags and quality status.
 - **AutomaticQualityFlag** – The granule level flag applying generally to the granule and specifically to parameters the granule level. When applied to parameter, the flag refers to the quality of that parameter for the granule (as applicable). The parameters determining whether the flag is set are defined by the developer and documented in the Quality Flag Explanation.
 - **AutomaticQualityFlagExplanation** – A text explanation of the criteria used to set automatic quality flag, including thresholds or other criteria.
 - **OperationalQualityFlag** – The granule level flag applying both generally to a granule and specifically to parameters at the granule level. When applied to parameter, the flag refers to the quality of that parameter for the granule (as applicable). The parameters determining whether the flag is set are defined by the developers and documented in the Operational Quality Flag Explanation.
 - **OperationalQualityFlagExplanation** – A text explanation of the criteria used to set operational quality flag; including thresholds or other criteria.
 - **ScienceQualityFlag** – Granule level flag applying to a granule, and specifically to parameters. When applied to parameter, the flag refers to the quality of that parameter for the granule (as applicable). The parameters determining whether the flag is set are defined by the developers and documented in the Science Quality Flag Explanation.
 - **ScienceQualityFlagExplanation** – A text explanation of the criteria used to set science quality flag; including thresholds or other criteria.

Code Listing 30. Measured Parameters

```
<MeasuredParameters>
  <MeasuredParameter>
    <ParameterName>Snow_Cover_Daily_Tile</ParameterName>
    <QAStats>
      <QAPercentMissingData>0</QAPercentMissingData>
      <QAPercentCloudCover>65</QAPercentCloudCover>
    </QAStats>
    <QAFlags>
      <AutomaticQualityFlag>Passed</AutomaticQualityFlag>
      <AutomaticQualityFlagExplanation>
        No automatic quality assessment done in the PGE
      </AutomaticQualityFlagExplanation>
      <OperationalQualityFlag>Passed</OperationalQualityFlag>
      <OperationalQualityFlagExplanation>
        Passed
      </OperationalQualityFlagExplanation>
      <ScienceQualityFlag>Not Investigated</ScienceQualityFlag>
      <ScienceQualityFlagExplanation>
        See http://landweb.nascom.nasa.gov/cgi-bin/QA\_WWW/qaFlagPage.cgi?sat=terra for the product Science Quality status.
      </ScienceQualityFlagExplanation>
    </QAFlags>
  </MeasuredParameter>
```

```
</MeasuredParameters>
```

4.8 ONLINE DATA ACCESS URL AND ONLINE RESOURCES URL

For some granules or collections, the raw data are made available online via FTP or web URL. ECHO stores this online access information for directly accessible granule and collection data differently from information covering other aspects of granule and collection data. Directly accessible data require the <OnlineAccessURLs> tag and include the URL to that data. Use OnlineAccessURLs only for the actual data. Online access URLs specified in collection and granule metadata include the following elements:

- **URL** – URL for the online data. (Required)
- **URLDescription** – Description of the data available via the supplied URL.
- **MimeType** – The mime type of the online data.

Any other online information covering aspects of the data, such as guides, product listings, validation information, etc., should be listed in an <OnlineResources>, along with the URLs to that information. Online resource URLs specified in collection and granule metadata include the following elements:

- **URL** – URL for the online resource. (Required)
- **URLDescription** – Description of the resource available via the supplied URL.
- **Type** - The type of the resource such as 'Collection Guide' or 'Campaign Guide' etc. This value should be a short phrase that can be used in an ECHO client for displaying the URL. (Required)
- **MimeType** – The mime type of the online resource.

Code Listing 31. Online Access and Resource URLs

```
<OnlineAccessURLs>
  <OnlineAccessURL>
    <URL>ftp://daac.nasa.gov/granule_1234.zip</URL>
    <URLDescription>Compressed data granule</URLDescription>
    <MimeType>application/zip</MimeType>
  </OnlineAccessURL>
</OnlineAccessURLs>

<OnlineResources>
  <OnlineResource>
    <URL>http://daac.nasa.gov/products/product_A.html</URL>
    <URLDescription>Main product overview page.</URLDescription>
    <Type>Product Overview</Type>
    <MimeType>text/html</MimeType>
  </OnlineResource>
</OnlineResources>
```

4.9 KEYWORDS

ECHO supports three kinds of keyword associations for collections: science keywords, spatial keywords, and temporal keywords. Science keyword and spatial keyword values should come from the Global Change Master Directory (GCMD) keywords standard located at <http://gcmd.gsfc.nasa.gov/Resources/valids/index.html>.

Currently, the ECHO structure for Science Keywords matches the GCMD standard, however the Spatial Keywords do not. There is no published Temporal Keywords list managed by GCMD.

4.9.1 Science Keywords

The science keyword metadata element fully implements the GCMD keyword hierarchy, which contains the following fields. There are no associated granule metadata elements for the science keyword.

- **Category** – Keyword used to describe the general category of the collection.
- **Topic** – Keyword used to describe the general topic of the collection.
- **Term** – Keyword used to describe the science parameter area.
- **Variable Level 1** – Keyword containing the first level science keyword variable.
- **Variable Level 2** – Keyword containing the second level science keyword variable.
- **Variable Level 3** – Keyword containing the third level science keyword variable.
- **Detailed Variable**– Keyword containing a free form field for further keyword specification.

Code Listing 32. Sample Science Keyword

```
<ScienceKeywords>
  <ScienceKeyword>
    <CategoryKeyword>EARTH SCIENCE</CategoryKeyword>
    <TopicKeyword>ATMOSPHERE</TopicKeyword>
    <TermKeyword>ATMOSPHERIC CHEMISTRY</TermKeyword>
    <VariableLevel1Keyword>
      <Value>OXYGEN COMPOUNDS</Value>
    </VariableLevel1Keyword>
  </ScienceKeyword>
</ScienceKeywords>
```

4.9.2 Spatial Keywords

The spatial keyword metadata element does not currently implement the updated GCMD keyword hierarchy. The ECHO spatial keyword contains a single value, which can be understood to be the 'Location' field in the GCMD managed list of keywords. There are no associated granule metadata elements for the science keyword.

Code Listing 33. Sample Spatial Keyword

```
<SpatialKeywords>
  <Keyword>GEOGRAPHIC REGION</Keyword>
</SpatialKeywords>
```

4.9.3 Temporal Keywords

The temporal keyword metadata element does not associated with a GCMD managed keyword list. The ECHO spatial keyword contains a single value. There are no associated granule metadata elements for the science keyword.

Code Listing 34. Sample Temporal Keyword

```
<TemporalKeywords>  
  <Keyword>UTC</Keyword>  
</TemporalKeywords>
```

4.10 ACCESS CONTROL

Collection and granule metadata may utilize two access control elements in order to facilitate a data center's data management needs. These two control mechanisms are:

- **Visibility Flag** – A Boolean value designating whether a collection or granule is visible or not.
- **Restriction Flag** – A numeric value which can be used to facilitate a data center specific set of access control levels.

For a further description regarding access control mechanisms and recommended usage, see section 6.

4.10.1 Visibility Flag

ECHO will restrict collections and collections from being viewed or ordered by anyone except users bearing the provider role by verifying a catalog item's Visibility Flag. A collection marked as not visible will restrict all associated granules. A collection marked as visible will automatically make all of its granules visible, unless further access control mechanisms are used. For additional information regarding usage of the Visibility Flag, reference section 6.1.5 of this document.

```
<Visible>>true</Visible>
```

4.10.2 Restriction Flag

The restriction flag is a numerical value indicating access constraints on a collection or granule. ECHO references this value, in combination with your own data access rule, to restrict public access to a collection or granule. You may use any range of integers for the RestrictionFlag value and assign your own meanings to these numbers. When designing your restriction flag scheme, consider the following:

- a. Granules inherit the value of their collection unless specified in the granule metadata
- b. If the collection has no value for the Restriction Flag, then there will be no default value for granules within the collection.

For example:

- You decide to use RestrictionFlag for a data quality summary, with a range of values from 0 to 10, with 0 indicating unknown quality, 1 indicating poor quality, and 10 indicating excellent quality,
- You have established a data access rule that restricts access to granules with a RestrictionFlag value of less than or equal to 5
- This means that Guest users will only be allowed to view granules with a RestrictionFlag value of 6 or higher.

Based on this example, the granule below will not be restricted from any user.

```
<RestrictionFlag>7</RestrictionFlag>
```

```
<RestrictionComment>default for collection</RestrictionComment>
```

4.11 BROWSE IMAGES AND BROWSE IMAGE METADATA

Browse images provide a high-level view of granule or collection data. Providers may choose to host their own browse images, or request that the ECHO system make available the browse imagery. This decision should be coordinated with ECHO Operations to ensure that your data availability needs are met, and so that ECHO can ensure that sufficient storage will be available for your browse images.

If you are making your browse imagery externally available, then your browse metadata will contain an externally visible URL to the browse file. If you are requesting ECHO to host your browse images, you should send both browse image files and a browse image metadata XML file to the ECHO system. The ECHO system will allocate the storage for browse image files, build a browse image URL, and associate the browse image URL to its item record in the ECHO database.

When ECHO processes a browse image ingest, all the browse image files must also be included. In addition to checking for the existence of the browse image files, ECHO also verifies the actual browse image file size against the file size indicated for that file in the browse image metadata input XML file. If any browse image file indicated in the browse image metadata input XML file does not exist, or if any file size does not match the indicated size, then the browse image files and browse image metadata input XML file will not be processed. For externally hosted browse, no validation of the URL or file size is performed.

ECHO supports many-to-many referencing between collections and browse images or granules and browse images. After processing, ECHO places the browse image files online. Using the Data Partner's unique identifier of the browse image, associations can then be made to collections or granules. This information is then made available to the end user.

Code Listing 35. Sample Browse Insert (ECHO Hosted)

```
<?xml version="1.0" encoding="UTF-8"?>
<BrowseMetaDataFile>
  <BrowseImages>
    <BrowseImage>
      <ProviderBrowseId>BR:036015124</ProviderBrowseId>
      <FileName>
        MOBCTT.A2004351.0425.005.2007036015124.jpg
      </FileName>
      <FileSize>558403</FileSize>
    </BrowseImage>
  </BrowseImages>
</BrowseMetaDataFile>
```

ECHO Ingest will load the sample browse and insert into the ECHO database, where it will remain until a browse image delete is submitted.

Code Listing 36. Sample Browse Insert (Externally Hosted)

```
<?xml version="1.0" encoding="UTF-8"?>
<BrowseMetaDataFile>
  <BrowseImages>
    <BrowseImage>
      <ProviderBrowseId>BR:036015124</ProviderBrowseId>
      <FileURL>
        http://daac.nasa.gov/MODIS/2009/03/02/BR:036015124.jpg
      </FileURL>
    </BrowseImage>
  </BrowseImages>
</BrowseMetaDataFile>
```

```
</FileURL>
</BrowseImage>
</BrowseImages>
</BrowseMetaDataFile>
```

Code Listing 37. Sample Granule Insert with Browse Associations

```
<?xml version="1.0" encoding="UTF-8"?>
<GranuleMetaDataFile>
  <Granules>
    <Granule>
      <GranuleUR>GR:2343124</GranuleUR>
      <InsertTime>2007-01-05T10:30:30.156Z</InsertTime>
      <LastUpdate>2007-01-05T12:30:30.550Z</LastUpdate>
      <Collection>
        <ShortName>MOD06_L2</ShortName>
        <VersionId>5</VersionId>
      </Collection>
      <Orderable>true</Orderable>
      <AssociatedBrowseImages>
        <ProviderBrowseId>
          BR:036015124
        </ProviderBrowseId>
      </AssociatedBrowseImages>
    </Granule>
  </Granules>
</GranuleMetaDataFile>
```

Once ingested, a browse image can be associated with any number of collections or granules. The order of processing followed by ECHO Ingest would allow both the browse image insert and the granule insert above to be placed in the same package and be ingested successfully.

CHAPTER 5. INGEST

5.1 INGEST OVERVIEW

Ingest refers to the process of inserting, deleting, updating, or validating metadata in the ECHO database and affects only that Data Partner's specific metadata. Data Partners may generate a full metadata record that will be processed as an insert or full replacement. Partners may generate metadata to update or delete specific fields within a metadata record or the entire metadata record from ECHO. Partners may also generate reconciliation metadata to perform a full metadata verification or inventory verification.

Note that deleting a collection will cascade to delete all associated granules. However, a cascaded or explicit granule deletion will not cascade to delete associated browse records.

The ECHO Ingest process directly accepts and processes metadata input files conforming to the ECHO 10 format and it is the responsibility of the Data Partner to ensure their metadata conforms to this specification. Unless supported by previous versions of Ingest, metadata input files conforming to other formats require that you perform a conversion before submitting the files to the ECHO ingest process. Contact the ECHO Operations team for more information.

The main tasks for the metadata ingest process are loading metadata and updating the ECHO database. Although there are slight variations from one Data Partner to another in the transmission process and interaction

with ECHO Operations, the process includes the following steps. For full details, please refer to the additional information found in this section.

- a. Data Partners send metadata xml files and browse image files to a configured FTP location on the ECHO system.
- b. ECHO Ingest regularly monitors each Data Partner's ingest ftp directory for new or updated metadata based upon an Operationally configured interval. When Ingest detects metadata which has been transmitted and quiesced for a configured amount of time, Ingest will create a new Ingest job and remove the files from the ftp directory.
 - 1) If the single-file delivery mechanism is used, received files will be aggregated into a single job.
 - 2) If the package delivery mechanism is used, each package will become a unique Ingest job.
- c. ECHO ingest will process jobs based upon the order received for jobs created from the single-file delivery mechanism or packages which do not specify a sequence number. If packages specify a sequence, Ingest will process jobs in the specified order.
- d. ECHO Ingest will validate the input file against the ECHO 10 Format and business rules. If this validation fails, the ingest process will reject the input file if it cannot be parsed or reject only the invalid metadata items found in an input file. Whenever the ingest process rejects an input file or metadata item, it will record an error and include the error in the Ingest Detail Report.
- e. Once metadata items in an input file have been successfully validated, the ECHO ingest process will load the metadata into the ECHO database, making the metadata available for public search. To ensure proper metadata processing, only one ingest job may be in this 'loading' state at a time. The ingest process will send an end of job email to a preconfigured provider contact email address. This email will contain information regarding successful and unsuccessful metadata actions. An xml Ingest Summary Report file is also placed in an ftp accessible location for the Data Partner to retrieve.

The following diagram shows the full flow of ingested data beginning at Data Partner ftp submission through report creation and email.

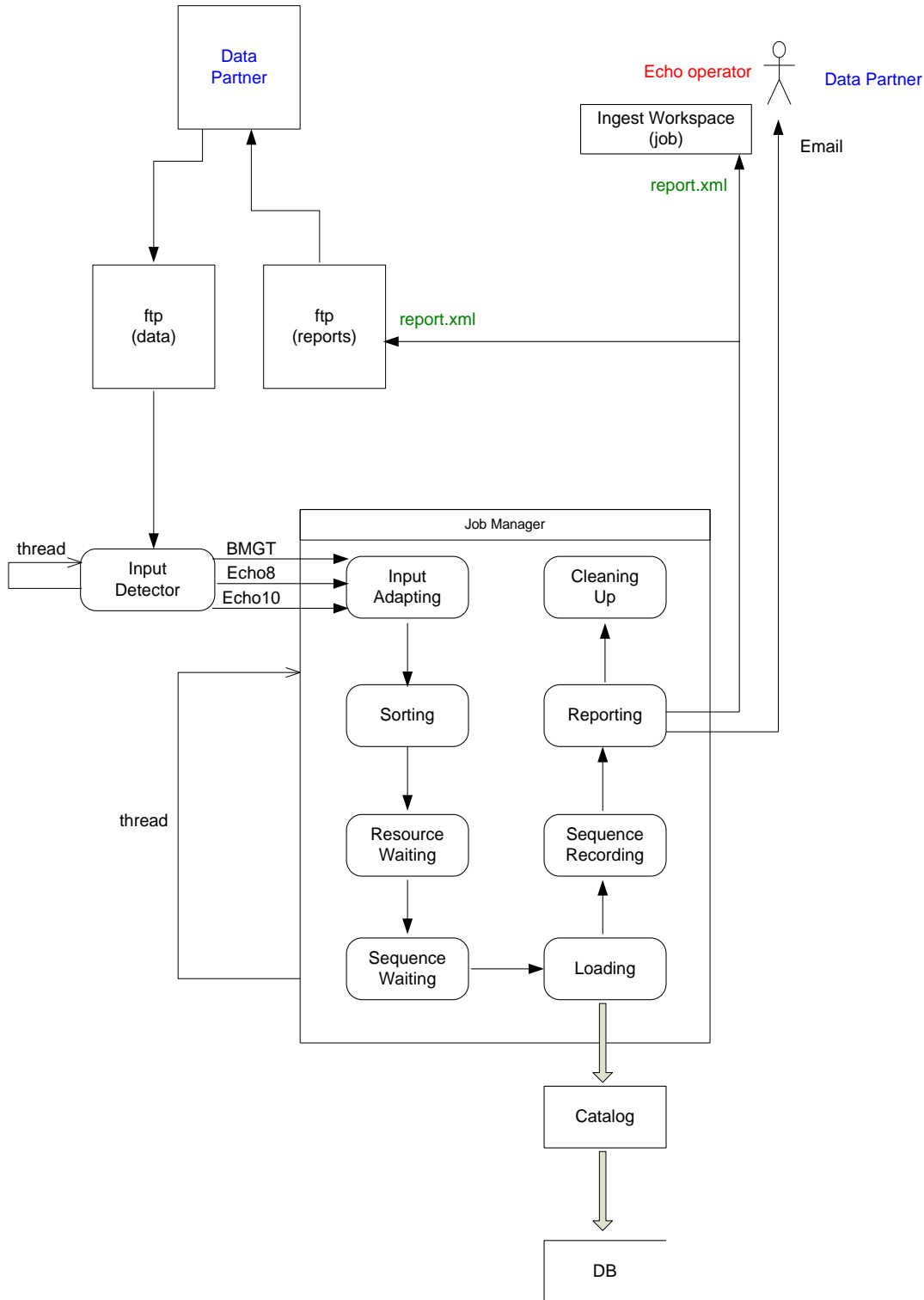


Figure 5. Ingest Overview

5.1.1 Ingest Processing States

During the Ingest process, an ingest job will pass through numerous internal states. These states are described below. Data Partners will not be notified as a job passes through each state. However, through the ECHO Ingest Accounting Tool, a Data Partner may view the current status of a job.

1. RESOURCE_WAITING - Indicates the job is currently waiting for further resources to arrive from the provider. Resources currently include browse image files.
2. SEQUENCE_WAITING - Indicates the job is waiting on a sequence number earlier in the list to arrive and be processed.
3. INPUT_ADAPTING - Indicates the job is having some of its files adapted from a provider specific format.
4. SORTING - Indicates the job is having its files sorted by item type and action.
5. LOAD_WAITING - Indicates the job is waiting for another job from the same provider to finish loading.
6. LOADING - Indicates the job's metadata files are currently being validated and loaded into the database.
7. REPORTING - Indicates a report is being generated for the job.
8. CLEANING_UP - Indicates that cleanup is occurring on the job.
9. COMPLETED - Indicates the job has been completed.

5.1.2 Ingest Processing Order

In order to ensure metadata records are processed correctly and consistently, ECHO Ingest will sort received metadata files and process actions in the following order:

- a. Browse inserts/replacements
- b. Collection inserts/replacements
- c. Collection partial deletes
- d. Collection partial updates
- e. Collection deletes
- f. Granule inserts/replacements
- g. Granule partial deletes
- h. Granule partial updates
- i. Granule deletes
- j. Browse deletes
- k. Collection metadata verifications
- l. Granule metadata verifications
- m. Collection inventory verifications
- n. Granule inventory verifications
- o. Browse inventory verifications

5.2 DELIVERING DATA TO ECHO FOR INGEST

5.2.1 Data Partner FTP Locations

The ECHO Operations team configures an ftp directory for each Data Partner utilizing a specific username and password. Data Partners have permissions to write data to this input directory, but cannot remove files. If file removal is needed, the ECHO Operations team (echo@echo.nasa.gov) should be contacted. Note that ECHO Ingest will support filenames up to 255 characters long.

Note: Due to security constraints, ftp access to ECHO resources is restricted based upon the originating IP address. Initial configuration or subsequent changes in test or operational systems will require Data Partners to coordinate with ECHO Operations to ensure the proper provisions are made.

5.2.2 Package File Delivery

Package delivery is the preferred Data Partner delivery mechanism. Data Partners deliver a package that is a ZIP archive of metadata files. The archive contains metadata files of any action and type as well as a manifest file that provides a list of the files in the package and an optional sequence number. Packages have the benefit of compression, which greatly reduces the amount of data delivered. They also support optional sequencing that allows you to indicate the order that the packages should be executed in—which is important in the event that packages arrive out of order and have dependencies on other packages.

ECHO Ingest guarantees that a single package will map to a single job with all the files in the package included in the job. Browse binary files are still delivered externally to the archive due to the size and number of the files involved.

If package sequencing is being used by a provider, packages with a sequence number of '-1' will be processed on a first come first serve basis, outside of the regularly sequenced packages.

Code Listing 38. Manifest File Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Manifest packageName="1101" sequenceNumber="1">
  <Files>
    <File name="CollectionInsertFile.XML"/>
    <File name="GranuleInsertFile.XML"/>
  </Files>
</Manifest>
```

Note: The manifest file for package delivery must be named "manifest.xml"...all lowercase letters.

For more information, refer to the package manifest schema and documentation located on the ECHO website (<http://www.echo.nasa.gov/ingest/schemas/operations/docs/index.html>)

5.2.3 Single File Delivery

Single file delivery is a legacy delivery mechanism. This mechanism allows you to deliver uncompressed metadata files one at a time. ECHO Ingest will monitor the delivered files for completion based on arrival date. Once a file is determined to be complete, a job will be constructed and the file added.

ECHO Ingest will attempt to put as many files into a single job as possible, but cannot guarantee that all of the delivered files will be in a single job because some files may not be considered complete when the input scan is performed. It is therefore important for you to deliver the files serially in the order that they desire execution. Once a job is created, ECHO Ingest will generate an internal sequence number to ensure that individual jobs are executed in the order they were created.

5.3 METADATA GENERATION TIPS

5.3.1 Creating ECHO-Compatible XML Files

The ECHO Ingest process directly accepts and processes metadata input files conforming to the ECHO 10 schema. Metadata input files conforming to other DTDs require that you perform an XML-to-XML conversion before submitting the files to the ECHO ingest process.

When the ECHO ingest process detects potential input files in your FTP input directory, it will then examine and validate each file, checking to see if the first line of text contains the <!xml....> declaration. If this line is not present, Ingest will reject the input file as invalid.

5.3.2 Dates and Times

Providers utilizing the ECHO 10.0 XML schema format must conform to the [XML Schema Specification](#). Unless otherwise specified in the provided date/time value, ECHO will process all date and time fields in the UTC timezone.

5.3.3 Unique Values

ECHO requires uniqueness in specific metadata fields in order to uniquely identify metadata items within a Data Partner's holdings. These fields are validated for uniqueness during ingest and used during updates, replacements, and deletions in order to uniquely identify a target record. The fields which are validated for uniqueness are found below.

- **(Collection) DatasetID** – This field used as the primary identifier of a collection.
- **(Collection) ShortName and VersionID** – The combination of these two fields must be unique.
- **(Collection) LongName** – This field must be unique.
- **(Granule) Granule UR** – This field is used as the primary identifier of a granule.
- **(Browse) ProviderBrowseID** – This field is used as the primary identifier of a browse image.

5.3.4 Metadata Inheritance

Specific metadata fields have been designated to require an enforced 'inheritance' validation between collections and associated granules. This means that values for these fields which are specified at the collection level include a superset of all values which may be specified at the granule level. For example, if a collection specifies Campaigns A, B, and C, all associated granules may have any subset of these campaigns. However, if a granule specifies a Campaign D, it will be rejected. Collection and granule updates will always perform this validation check to ensure the inheritance is valid. Affected fields include the following:

- Campaign
- Platform
- Instrument
- Sensor
- Additional Attributes

5.4 NEW ITEMS VS REPLACE ITEMS

ECHO ingest requires a complete set of collection, granule or browse metadata for both inserts and replacements. When processing collection, granule, or browse metadata, the ECHO system first checks to see if the item already exists. Item existence is based upon the primary unique identifier for each item type, as described in section 5.3.2. If the item already exists in the system, ECHO will replace the metadata associated with the item with the new metadata. Otherwise, ECHO will consider the metadata item to be an insert and load the new item into the ECHO database. The ECHO system applies the same principles and validation when dealing with the insertion or replace of the collection, granule, or browse item when processed against XML metadata input.

When replacing collection, granule, or a browse item, only the version with the most recent last update time will be stored in the ECHO database. ECHO will reject any collection, granule, or browse item received that has a last update date pre-dating the records already in the database. If there are duplicated items in one input file the same rule applies, the most recent item will be ingested into ECHO and earlier versions of the item may be reported as rejected, depending on the order of items processed within the file. If the last update time of the item is the same as the one currently in the database, the replacement will be applied. Due to backward compatibility with previous versions of Ingest, the last update time threshold is +/- a half a second.

Code Listing 39. Metadata Update Example

```
<GranulePartialAdd>
  <Targets>
    <Target>
      <GranuleUR>GR:115628</GranuleUR>
    </Target>
  </Targets>
  <Fields>
    <Field>
      <BrowseImage>BR:036015124</BrowseImage>
    </Field>
  </Fields>
</GranulePartialAdd>
```

5.4.1 Non-Replaceable Fields

The following information describes situations where ECHO will reject a processed replacement upon dependent metadata changes.

5.4.1.1 Dataset ID

When processing a collection replacement, ECHO Ingest will fail to replace a record if the new collection attempts to update the dataset ID. This is because the primary identifier of a collection within ECHO, Ingest currently does not support changing the value.

5.4.1.2 Granule UR

When processing a granule replacement, ECHO Ingest will fail to replace a record if the new granule attempts to update the Granule UR (provider granule ID). This is because the primary identifier of a granule within ECHO, Ingest currently does not support changing the value.

5.4.1.3 Granule Spatial Representation

When processing a collection replacement, ECHO Ingest will fail to replace a record if the new collection attempts to update the granule spatial representation. ECHO Ingest uses this value to determine which internal geometry tables to store granule spatial information. Switching this spatial representation at the collection level would need to cascade to a reprocessing of all of the granule spatial information. Currently, Ingest does not support changing this value. To make a change, the collection would need to be deleted from ECHO and then re-ingested, along with all of the associated granules. Special case handling can be facilitated by ECHO Operations.

5.4.1.4 Temporal Range

When processing a collection replacement, ECHO Ingest will fail to replace a record if the new collection's temporal range does not include all temporal ranges of its associated granules. Conversely, ECHO Ingest will fail to replace a granule record if the new granule's temporal range is outside of its collection's temporal range.

5.4.1.5 Collection / Granule Inheritance

When processing a collection or granule replacement, ECHO Ingest will fail to replace a record if one of the following fields is updated such that the required referential inheritance is broken. For instance, if a new additional attribute is added to a granule record that is not found in the collection, the granule replacement will be rejected. Similarly, if an existing additional attribute is removed from a collection record, but a granule exists in that collection with the attribute, the collection replacement will be rejected.

- Campaign
- Platform
- Instrument
- Sensor
- Additional Attributes
- TwoDCoordinateSystem

5.5 PARTIALLY ADDING/UPDATING/DELETING ITEMS

If you do not want to replace a complete collection or granule record in ECHO, you may specify individual fields to add, update, or remove content. For simplicity, the partial add, update, and delete capability is commonly referred to simply as "partial updates" and will be hereafter. The partial update format requires full metadata only for the particular field that should be updated along with the ID of the collection or granule record that should be updated. When processing partial updates, ECHO Ingest applies the same principles and validation as when processing inserts. The following sections outline the metadata fields that can be partially added, updated, or deleted.

While performing partial adds, updates or deletes, Ingest will affect different results depending on the nature of the item. In the following sections, each metadata field will have one of the following distinctions identifying how Ingest will perform the partial add, update, or delete.

- **Atomic** – This type of item exists once within a metadata record (e.g. Restriction Flag) and is effectively added, replaced, or removed during partial updates.
- **Unique** – This type of item has a uniquely identifying field and more than one unique instance may occur within a metadata record (e.g. Measured Parameter). Each unique instance may be added, updated, or removed during partial updates.
- **Static** – This type of item may have more than one instance within a metadata record (e.g. Browse Associations), but cannot be updated. These items can only be added or removed during partial updates.

5.5.1 Collections

An ECHO Collection record may have the following elements partially added or updated:

- 1) Visibility (Atomic)
- 2) Temporal (Atomic)
- 3) Spatial (Atomic)
- 4) Restriction Flag (Atomic)
- 5) Delete Time (Atomic)
- 6) Browse image associations (Static)

An ECHO Collection record may have the following elements partially deleted:

- 1) Temporal (Atomic)
- 2) Delete Time (Atomic)
- 3) Restriction Flag (Atomic)
- 4) Spatial (Atomic)
- 5) Specific or All Browse image associations (Static)

5.5.2 Granules

An ECHO Granule record may have the following elements partially added or updated:

- 1) Visibility (Atomic)
- 2) Temporal (Atomic)
- 3) Spatial (Atomic)
- 4) Day/Night Flag (Atomic)
- 5) Cloud Cover (Atomic)
- 6) Delete Time (Atomic)
- 7) Restriction Flag (Atomic)
- 8) Online access URL (Unique)

- 9) Online resource URL (Unique*)
- 10) Measured Parameter (Unique)
- 11) Additional Attribute (Unique)
- 12) Browse image associations (Static)

*Online Resource Urls currently support duplicate entries with the same URL.

An ECHO Granule record may have the following elements partially deleted:

- 1) Temporal (Atomic)
- 2) Day/Night Flag (Atomic)
- 3) Delete Time (Atomic)
- 4) Restriction Flag (Atomic)
- 5) Cloud Cover (Atomic)
- 6) Specific **or** All Online access URL(s) (Unique)
- 7) Specific **or** All Online resource URL(s) (Unique)
- 8) Specific **or** All Additional attribute(s) (Unique)
- 9) Measured Parameter (Unique)
- 10) Specific **or** All Browse image association(s) (Static)

5.5.3 Browse

Partial adds/updates are not supported for browse

5.6 DELETING ITEMS

ECHO Ingest will facilitate the deletion of collection, granule, or browse records from within its holdings. ECHO uses only the item's identification to process the deletion of the item and all the metadata associated with this item. ECHO will keep the deleted items' identification and deletion date in the database for metadata history auditing purposes. The only way to re-install the items in the ECHO system is to re-submit the metadata for those items for insertion.

When a collection is deleted, all associated granules will automatically be deleted. When a browse file is deleted, all collection or granule associations to that browse file will be deleted while the referencing item(s) themselves will remain in ECHO. When collections or granules with browse associations are deleted the browse images will remain in ECHO until an explicit browse image delete has been submitted through ingest. If a browse image being deleted includes a file hosted by ECHO, the image file will be deleted, and is not recoverable.

Code Listing 40. Collection Deletes

```
<CollectionMetaDataFile>
  <CollectionDeletes>
    <CollectionDelete>
      <DataSetId>Insert Additional Attributes Not Unique V001</DataSetId>
    </CollectionDelete>
  </CollectionDeletes>
</CollectionMetaDataFile>
```

Code Listing 41. Granule Deletes

```

<GranuleMetaDataFile>
  <GranuleDeletes>
    <GranuleDelete>
      <GranuleUR>SC:MOD021KM.004:19250276</GranuleUR>
    </GranuleDelete>
  </GranuleDeletes>
</GranuleMetaDataFile>

```

Code Listing 42. Browse Deletes

```

<BrowseMetaDataFile>
  <BrowseImageDeletes>
    <BrowseImageDelete>
      <ProviderBrowseId>:BR:Browse.001:19250276</ProviderBrowseId>
    </BrowseImageDelete>
  </BrowseImageDeletes>
</BrowseMetaDataFile>

```

5.6.1 Ingest Reaper

The ECHO Collection, Granule, and Browse metadata schema allow for data partners to update their records with a deletion time metadata element. If the value of this field has been set, ECHO Ingest will automatically detect when the time has expired, and will facilitate deletion of the associated record. This automatic deletion will cascade in the same manner as described above. In order to facilitate the automated deletion, ECHO Ingest is configured with an internal “reaper” which runs on a scheduled basis to clean out items which have passed their deletion time. The frequency which this reaper runs is configured by ECHO Operations. There is no additional notification sent to providers when an item is removed by the reaper. Where possible, Data Partners are encouraged to export explicit deletions instead of setting the deletion time.

5.6.2 Audit History

ECHO Ingest generates an internal audit history recording a subset of information for all metadata items that are deleted during the past 60 days. This history is not available to Data Partners, so any questions regarding item deletions should be coordinated with the ECHO Operations team. The following table shows the metadata fields which are recorded for each type of deletion. Note that the provider time fields are optional because those elements are not a required value in the ECHO metadata model.

Name	Collection	Granule	Browse
Short Name	X	X	
Version ID	X	X	
Long Name	X		
Dataset ID	X	X	
Granule UR		X	
Browse ID			X
Provider Insert Time*	X	X	X
Provider Update Time*	X	X	X
Provider Delete Time*	X	X	X

Name	Collection	Granule	Browse
ECHO Insert Time	X	X	X
ECHO Update Time	X	X	X
ECHO Delete Time	X	X	X

5.7 RECONCILIATION

Through the ECHO 10.0 Ingest Schema, Data Partners may perform two types of metadata reconciliation, *metadata verification* and *inventory verification*. These two verification processes are described below:

- **Metadata Verification** – A full reconciliation of a collection or granule metadata item to include verification of all fields. ECHO will automatically attempt to correct differences within its holdings.
- **Inventory Verification** – A shortened reconciliation mechanism which will allow for the identification of inventory items which are missing from the ECHO holdings or should no longer be held by ECHO. Verified inventory items include collections, granules within a specified collection, or browse records associated with granules within a specified collection.

Both of these methods allow Data Partners to take advantage of Ingest's parallelized data processing and detailed reporting mechanism. The following sections provide a detailed overview of each reconciliation capability.

5.7.1 Metadata Verification

The *Metadata Verification* capability allows an ECHO data partner to perform a full validation on all collection and granule metadata records and fields within the ECHO holdings. For each verified item, ECHO will ensure that every metadata field is correct and will report any inconsistencies discovered. Full verification on browse metadata items is not currently supported due to the initial requirements for this functionality. However, if providers have an interest, this is something that ECHO can consider this for a future release.

In order to utilize this capability, providers must re-export the full collection or granule metadata record, as is generated during normal exports, but with a slightly modified XML structure specifying that the received record should be processed as a verification action. Sample XML blocks for collections and granules are shown below with the new *CollectionVerifications* and *GranuleVerifications* elements. The respective *Collection* and *Granule* elements are repeated as necessary.

Code Listing 43. Collection Metadata Verification Sample

```
<CollectionMetaDataFile>
  <CollectionVerifications>
    <Collection>
      ...
    </Collection>
    ...
  </CollectionVerifications>
</CollectionMetaDataFile>
```

Code Listing 44. Granule Metadata Verification Sample

```
<GranuleMetaDataFile>
  <GranuleVerifications>
    <Granule>
      ...
    </Granule>
    ...
  </GranuleVerifications>
</GranuleMetaDataFile>
```

While verifying a metadata record, ECHO will first determine whether the item exists within ECHO. If it does not, ECHO will treat the item as a metadata insert and attempt to insert the item into its holdings. If the record does exist within ECHO's holdings, then ECHO will perform a detailed comparison of every metadata field and if discrepancies are found ECHO will treat the item as a full metadata replacement and attempt to replace the item in its holdings with the new record. The results of the verification will then be included in the XML ingest report outlining all issues discovered.

This capability should not be used as a means to identify collections or granules which ECHO does not have within its holdings. The *Inventory Verification* capability referenced in the following major section should be used for that purpose. Exports using this *Metadata Verification* can be any subset of a providers holdings.

Date value comparisons will initially be carried out to the milliseconds without any leniency. It may be possible to expand date comparison leniency if needed.

Discrepancies discovered during metadata verification will be reported as item errors within the Ingest Report and notification email. These item errors will use the following error codes which have been added for this new capability.

- **METADATA_MISMATCH** – Reported if a metadata field did not match between the ECHO holdings and provider's verification granule & collection.
- **COLLECTION_MISSING** – Reported if a collection in the verification listing did not exist in ECHO.
- **GRANULE_MISSING** – Reported if a granule in the verification listing did not exist in ECHO.

When ingest attempts to insert or replace a metadata record which was found to be missing or invalid during verification, it is possible that the subsequent action will fail due to invalid metadata in the verification record. The errors that would be reported include all those that are currently reported during normal ingest inserts and updates. ECHO providers should be sure to analyze the results of a verification export to identify why items have failed. Sample ingest report error messages are shown below for the three new error codes.

5.7.1.1 Ingest Item Errors

The following item error will be included in the ingest report if a 'field-level' mismatch is discovered.

Code Listing 45. METADATA_MISMATCH field-level error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
    <Message>
      EchoGranule.DeleteTime mismatch
    </Message>
  </ItemError>
</ItemErrorGroup>
```

```
        Expected: 2009-01-05T11:53:50.550Z
        Actual   : 2010-01-05T11:53:50.550Z
    </Message>
</ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an 'object-level' mismatch is discovered. An XML representation of the object is included in the message. An error message has a max length of 1024, so it is possible that the message will be truncated.

Code Listing 46. METADATA_MISMATCH object-level error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
    <Message>
      EchoGranule.AdditionalAttributes mismatch
      Expected: <AdditionalAttributeRef><Name>Name</Name>
                <Values><Value>1</Value></Values></AdditionalAttributeRef>
      Actual   : [null]
    </Message>
  </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a 'field-level' mismatch for an object in a list is discovered. In this case, the object which has a mismatching field has a unique identifier (additional attributes, online access URLs, etc.).

Code Listing 47. METADATA_MISMATCH list field-level error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
    <Message>
      EchoGranule.OnlineAccessUrls mismatch
      OnlineAccessURL.Description mismatch for: http://provider_url
      Expected: Description Text
      Actual   : Incorrect Value
    </Message>
  </ItemError>
</ItemErrorGroup>
```

The following item errors will be included in the ingest report if a mismatch for objects in a list is discovered. In this case, the objects being compared do not have unique identifiers (online resources, points, etc.) and may produce two messages. One message will indicate that an item was added to one list and another message indicating that an item was deleted from another.

Code Listing 48. METADATA_MISMATCH list object-level item missing error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
    <Message>
```

```
        EchoGranule.OnlineResources mismatch
        Expected:
<OnlineResource><URL>missing_url</URL><Description>123</Description></OnlineResource>
        Actual   : [null]
        </Message>
    </ItemError>
</ItemErrorGroup>
```

Code Listing 49. METADATA_MISMATCH list object-level extra item error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
    <Message>
      EchoGranule.OnlineResources mismatch
      Expected: [null]
      Actual   :
    <OnlineResource><URL>extra_url</URL><Description>123</Description></OnlineResource>
    </Message>
  </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a matching collection record was not found within ECHO during verification.

Code Listing 50. COLLECTION_MISSING error message

```
<ItemErrorGroup errorCode="COLLECTION_MISSING">
  <ItemError itemType="COLLECTION" itemId="COLLECTION_ID" level="WARNING">
    <Message>
      Collection was missing from ECHO. An insert attempt will be made for
      this collection.
    </Message>
  </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a matching granule record was not found within ECHO during verification.

Code Listing 51. GRANULE_MISSING error message

```
<ItemErrorGroup errorCode="GRANULE_MISSING">
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
    <Message>
      Granule was missing from ECHO. An insert attempt will be made for
      this granule.
    </Message>
  </ItemError>
```

```
</ItemErrorGroup>
```

5.7.1.2 Ingest Report Processing Totals

The metadata verification activity will be reflected in the processing totals and item errors reported in the Ingest Report and provider notification email. In order to record the verification actions that are performed, the collection and granule processing totals will have an attribute entitled "verifications." This attribute will include the number of verification items that were processed, including both successful and unsuccessful verifications. The processingTotals attribute will also include an accounting of the verification actions. Sample processing totals which are included in the Ingest report and notification email are shown below for three possible outcomes.

The following processing totals will be included in an ingest job report and provider notification email for a successful verification of 1000 collection and 1000 granule items.

Code Listing 52. Successful Verification Processing Totals

```
<ProcessingTotals>
  <CollectionProcessingTotals processed="1000" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="1000" inventories="0"/>
  <GranuleProcessingTotals processed="1000" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="1000" inventories="0"/>
  <BrowseProcessingTotals processed="0" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="0"/>
</ProcessingTotals>
```

The following processing totals will be included in an ingest job report and provider notification email for a verification of 1000 collection and 1000 granule items where each metadata type had 200 missing items and 200 mismatches. There were no subsequent failures while inserting and replacing these 400 items.

Code Listing 41. Mismatches & Missing Items with No Insertion/Replacement Errors

```
<ProcessingTotals>
  <CollectionProcessingTotals processed="1400" inserted="200" replaced="200"
updated="0" deleted="0" rejected="0" verifications="1000" inventories="0"/>
  <GranuleProcessingTotals processed="1400" inserted="200" replaced="200"
updated="0" deleted="0" rejected="0" verifications="1000" inventories="0"/>
  <BrowseProcessingTotals processed="0" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="0"/>
</ProcessingTotals>
```

The following processing totals will be included in an ingest job report and provider notification email for a verification of 1000 collection and 1000 granule items where each metadata type had 200 missing items and 200 mismatches. There were 100 subsequent failures for each metadata type while inserting the missing 200 items.

Code Listing 42. Mismatches & Missing Items with Insertion/Replacement Errors

```
<ProcessingTotals>
  <CollectionProcessingTotals processed="1400" inserted="100" replaced="200"
updated="0" deleted="0" rejected="100" verifications="1000" inventories="0"/>
```

```

    <GranuleProcessingTotals processed="1400" inserted="100" replaced="200"
updated="0" deleted="0" rejected="100" verifications="1000" inventories="0"/>
    <BrowseProcessingTotals processed="0" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications'0" inventories="0"/>
  </ProcessingTotals>

```

5.7.2 Inventory Verification

The *Inventory Verification* capability allows ECHO data partners to compare a full listing of collection, granule, or browse metadata items between ECHO and their holdings. For each metadata item type, ECHO will perform a two-way comparison for the listing of received items against its holdings. Items which are included in the verification package, but missing in ECHO, will be reported along with items which are in the ECHO holdings, but missing from the verification package. Inventory verification of granules should include a listing of all granules within a single collection. Inventory verification of browse should include a listing of all browse files associated with granules in a specific collection.

In order to utilize this capability, providers must export the full collection, granule, or browse listings in the following XML structure which includes the required elements for uniquely identifying a metadata record. Sample XML blocks for collection, granule, and browse inventory verification are shown below with the new *CollectionInventory*, *GranuleInventories*, and *BrowseInventories* elements. The respective *CollectionRef*, *GranuleInventory*, and *BrowseInventory* elements are repeated as necessary.

Code Listing 42. Collection Inventory Verification XML Sample

```

<CollectionMetaDataFile>
  <CollectionInventory>
    <CollectionReferences>
      <CollectionRef>
        <ShortName></ShortName>
      <VersionId></VersionId>
    </CollectionRef>
    <CollectionRef>
      <DataSetId></DataSetId>
    </CollectionRef>
    ...
  </CollectionReferences >
</CollectionInventory >
</CollectionMetaDataFile>

```

Code Listing 42. Granule Inventory Verification XML Sample

```

<GranuleMetaDataFile>
  <GranuleInventories>
    <GranuleInventory>
      <CollectionRef>...</CollectionRef>
      <GranuleURs>
        <GranuleUR>GRANULE_UR</GranuleUR>
        ...
      </GranuleURs>
    </GranuleInventory>
    ...

```

```
</GranuleInventories>  
</GranuleMetaDataFile>
```

Code Listing 42. Browse Inventory Verification XML Sample

```
<BrowseMetaDataFile>  
  <BrowseInventories>  
    <BrowseInventory>  
      <CollectionRef>...</CollectionRef>  
      <ProviderBrowseIds>  
        <ProviderBrowseId>BROWSE_ID</ProviderBrowseId>  
        ...  
      </ProviderBrowseIds>  
    </BrowseInventory>  
    ...  
  </BrowseInventories>  
</BrowseMetaDataFile>
```

As described previously, while verifying a collection, granule, or browse inventory ECHO will identify discrepancies including items which are missing from or should not exist in the ECHO holdings. No corrective actions will be taken as a part of the verification. Discrepancies discovered during inventory verification will be reported as item errors within the Ingest Report and notification email. These item errors will use the following error codes which have been added for this new capability.

- **COLLECTION_MISSING** – Reported if a collection in the verification listing did not exist in ECHO.
- **COLLECTION_UNEXPECTED** – Reported if ECHO has a collection which is not in the verification listing.
- **GRANULE_MISSING** – Reported if a granule in the verification listing did not exist in ECHO within the specified collection inventory.
- **GRANULE_UNEXPECTED** – Reported if ECHO has a granule within the specified collection which is not in the verification listing.
- **BROWSE_LINK_MISSING** – Reported if a browse record in the verification listing did not exist in ECHO associated to the specified inventory collection or granules within that collection.
- **BROWSE_LINK_UNEXPECTED** – Reported if ECHO has a browse record associated with a granule in the specified collection, or the collection itself, which is not in the verification listing.

Sample ingest report error messages are shown below for the new error codes.

5.7.2.1 Ingest Item Errors

The following item error will be included in the ingest report if a matching collection record was not found within ECHO during inventory comparison.

Code Listing 42. COLLECTION_MISSING Error Message Text

```
<ItemErrorGroup errorCode="COLLECTION_MISSING">  
  <ItemError itemType="COLLECTION" itemId="COLLECTION_ID" level="CRITICAL">  
    <Message>  
      Inventory mismatch: [Collection] V[1] was not found in ECHO.    </Message>  
  </ItemError>  
</ItemErrorGroup>
```

```
</Message>  
</ItemError>  
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an extra collection record was found within ECHO during inventory comparison.

Code Listing 43. COLLECTION_UNEXPECTED Error Message Text

```
<ItemErrorGroup errorCode="COLLECTION_UNEXPECTED">  
  <ItemError itemType="COLLECTION" itemId="COLLECTION_ID" level="CRITICAL">  
    <Message>  
      Inventory mismatch: [Collection] V[1] exists in ECHO, but was not in  
the supplied inventory.  
    </Message>  
  </ItemError>  
</ItemErrorGroup>  
GRANULE_MISSING Error Message Text
```

The following item error will be included in the ingest report if a matching granule record was not found within ECHO during inventory comparison.

Code Listing 43. GRANULE_MISSING Error Message Text

```
<ItemErrorGroup errorCode="GRANULE_MISSING">  
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="CRITICAL">  
    <Message>  
      Inventory mismatch: [Granule UR] does not exist in collection  
[Collection] V[1].  
    </Message>  
  </ItemError>  
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an extra granule record was found within ECHO during inventory comparison.

Code Listing 44. GRANULE_UNEXPECTED Error Message Text

```
<ItemErrorGroup errorCode="GRANULE_UNEXPECTED">  
  <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="CRITICAL">  
    <Message>  
      Inventory mismatch: [Granule UR] exists in [Collection] V[1], but was  
not in the supplied inventory.  
    </Message>  
  </ItemError>  
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a matching browse record was not found within ECHO during inventory comparison.

Code Listing 45. BROWSE_LINK_MISSING Error Message Text

```
<ItemErrorGroup errorCode="BROWSE_LINK_MISSING">
  <ItemError itemType="BROWSE" itemId="BROWSE_ID" level="CRITICAL">
    <Message>
      Inventory mismatch: [Browse ID] is not currently linked to collection
      [Collection] V[1] or any of the collection's granules.
    </Message>
  </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an extra browse record was found within ECHO during inventory comparison.

Code Listing 46. BROWSE_LINK_UNEXPECTED Error Message Text

```
<ItemErrorGroup errorCode="BROWSE_LINK_UNEXPECTED">
  <ItemError itemType="BROWSE" itemId="BROWSE_ID" level="CRITICAL">
    <Message>
      Inventory mismatch: [Browse ID] is currently linked to collection
      [Collection] V[1] or any of the collection's granules, but was not in the
      supplied inventory.
    </Message>
  </ItemError>
</ItemErrorGroup>
```

5.7.2.2 Ingest Report Processing Totals

The *inventory verification* activity will be reflected in the processing totals and item errors reported in the Ingest Report and provider notification email. In order to record the number of inventories reconciled, the collection, granule, and browse processing totals will have a new attribute entitled "inventories." This attribute will include the number of complete inventories items that were processed. This does not include the number of complete inventory items which were compared. For instance, if an ingest file contains 100,000 granules for a single collection, the "inventories" attribute will have a value of 1, not 100,000. ECHO providers should be sure to analyze the results of an inventory reconciliation to identify why items have failed. Sample processing totals which are included in the Ingest report and notification email are shown below for two possible outcomes.

The following processing totals will be included in an ingest job report and provider notification email for a successful verification of a provider's 1000 collections, 3 granule inventories, and 3 browse inventories. Note that the collection processing totals' value for the new *inventories* attribute will always be 1 since there is only one inventory of collections.

Code Listing 47. Successful Verification Processing Totals

```
<ProcessingTotals>
  <CollectionProcessingTotals processed="1" inserted="0" replaced="0"
  updated="0" deleted="0" rejected="0" verifications="0" inventories="1"/>
  <GranuleProcessingTotals processed="3" inserted="0" replaced="0"
  updated="0" deleted="0" rejected="0" verifications="0" inventories="3"/>
```

```
<BrowseProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="3"/>
</ProcessingTotals>
```

The following processing totals will be included in an ingest job report and provider notification email for a verification of a provider's 1000 collections, 3 granule inventories, and 3 browse inventories, where 1 of each of the granule and browse inventories referenced invalid collections.

Code Listing 48. Inventory Verification with Rejections

```
<ProcessingTotals>
  <CollectionProcessingTotals processed="1" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="1"/>
  <GranuleProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="1" verifications="0" inventories="2"/>
  <BrowseProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="1" verifications="0" inventories="2"/>
</ProcessingTotals>
```

5.8 INGEST ERRORS

During Ingest processing, the following errors may be encountered:

- **Job Errors** – Created when ECHO cannot process the job in its entirety.
- **File Errors** – Created when ECHO cannot parse or validate the input file.
- **Item Errors** – Created when ECHO can parse the input file but finds the data invalid.

Any error that is generated will have a level associated with it: CRITICAL or WARNING. In general, critical errors indicate that the job or part of the job was rejected by ECHO Ingest. Errors that are warnings indicate that some abnormality was detected but the job or affected part of the job may have been ingested dependent on the absence of other critical errors. Each of these errors is discussed in more detail below. For more information regarding the Ingest Summary Report which is referenced, please refer to section 5.8.1 of this document. For a full list of possible errors review the online Ingest documentation at: <http://www.echo.nasa.gov/ingest/schemas/operations/docs/>.

5.8.1 Job Errors

Initial scanning of a job by ECHO Ingest may reveal problems that will stop processing for that job immediately. The Ingest Summary Report will contain the necessary information to determine why the job was canceled.

5.8.1.1 DUPLICATE_SEQUENCE_NUMBER Error

If a processed package's sequence number is less than the current sequence number being processed a DUPLICATE_SEQUENCE_NUMBER job error will be generated.

Code Listing 53. Job Error Message

```
<JobErrors>
  <Error errorCode="DUPLICATE_SEQUENCE_NUMBER" level="CRITICAL">
    <Message>
      The sequence number [300258] is less than or equal to the last
      sequence number seen for the provider.
    </Message>
  </Error>
</JobErrors>
```

5.8.1.2 MANIFEST_MISSING Error

If an Ingest package does not include the manifest.xml file either due to non-existence or an incorrect naming, a MANIFEST_MISSING job error will be generated.

Code Listing 54. Missing Manifest File Error

```
<JobErrors>
  <Error errorCode="MANIFEST_MISSING" level="CRITICAL">
    <Message>
      No [manifest.xml] file found in package
    </Message>
  </Error>
```

```
</JobErrors>
```

5.8.2 File Errors

Before ECHO ingests your metadata, it analyzes and validates the metadata against the ECHO Format. If an error is detected, ECHO may create an input file error or an item error.

5.8.2.1 FULL_SCHEMA Error

If an input file does not pass validation against the ECHO Format a FULL_SCHEMA file error will be generated. To allow the ingest of files that may have some valid and some invalid items, FULL_SCHEMA errors are considered warnings. An input file with a FULL_SCHEMA error will be subjected to validation of each individual item against the ECHO Format to determine if any items within the input file are valid.

Code Listing 55. FULL_SCHEMA Error Message

```
<FileErrors>
  <Error errorCode="FULL_SCHEMA" level="WARNING">
    <Message>
      Line:180 Col:140, cvc-minLength-valid: Value '' with length =
      '0' is not facet-valid with respect to minLength '1' for
      type 'GranuleUR'.
    </Message>
  </Error>
</FileErrors>
```

The error message indicates the problem line and column number in the input file. The precise message will vary depending on the type of error found. The sample above indicates that a zero-length GranuleUR was submitted and the ECHO Format specifies that GranuleUR has a minimum length of 1.

5.8.2.2 STRUCTURAL_SCHEMA Error

If an input file is marked with a FULL_SCHEMA error ECHO Ingest will attempt to verify that the input file can be parsed by validating the input file conforms the ECHO Format structurally. If an input file is not structurally valid a STRUCTURAL_SCHEMA error will be generated. STRUCTURAL_SCHEMA errors are critical. If an input file is structurally valid then each metadata item will be subject to validation against the full ECHO Format, in order to ingest any valid metadata items.

Code Listing 56. STRUCTURAL_SCHEMA Error Message

```
<FileErrors>
  <Error errorCode="FULL_SCHEMA" level="WARNING">
    <Message>
      Line:66 Col:22, cvc-complex-type.2.3: Element
      'GranulePartialAdds' cannot have character [children],
      because the type's content type is element-only.
    </Message>
  </Error>
  <Error errorCode="STRUCTURAL_SCHEMA" level="CRITICAL">
    <Message>
      Line:66 Col:22, cvc-complex-type.2.3: Element
      'GranulePartialAdds' cannot have character [children],
      because the type's content type is element-only.
    </Message>
  </Error>
</FileErrors>
```

```
</Message>
</Error>
</FileErrors>
```

In the sample above the input was structurally invalid with a character data present in the GranulePartialAdds tag. Such a file cannot be parsed by ECHO Ingest and so a STRUCTURAL_SCHEMA error is generated and the entire input file is rejected.

5.8.2.3 SCHEMA_VALIDATION_ERROR Error

If an input file was not valid against the full ECHO Format but was structurally valid then each item will be individually validated against the full ECHO Format. A SCHEMA_VALIDATION_ERROR will be generated for any item failing validation and the item will not be ingested. Otherwise, items passing validation will continue normal processing by ECHO Ingest.

Code Listing 57. SCHEMA_VALIDATION_ERROR Error Message

```
<ItemErrorGroups>
  <ItemErrorGroup errorCode="SCHEMA_VALIDATION_ERROR">
    <ItemError itemType="COLLECTION"
      itemId="MLS/Aura L2 Diagnostics, Miscellaneous Grid V001"
      level="CRITICAL">
      <Message>
        Line:14 Col:141, cvc-minLength-valid: Value '' with
        length = '0' is not facet-valid with respect to
        minLength '1' for type 'ShortName'.
      </Message>
    </ItemError>
  </ItemErrorGroup>
</ItemErrorGroups>
```

In the sample above the offending collection could be identified and the dataset ID was included in itemId. The collection was not ingested because the ShortName supplied was an empty string and the ECHO Format requires a minimum ShortName length of 1.

5.8.2.4 FILE_TYPE_INDETERMINABLE Error

If an input file is invalid XML, a FILE_TYPE_INDETERMINABLE error will be generated. ECHO Ingest will also generate a FILE_TYPE_INDETERMINABLE error if an input file is valid XML but ECHO Ingest cannot determine the metadata item type of the file. FILE_TYPE_INDETERMINABLE errors are critical.

Code Listing 58. FILE_TYPE_INDETERMINABLE Error Message

```
<FileErrors>
  <Error errorCode="FILE_TYPE_INDETERMINABLE" level="CRITICAL">
    <Message>
      Could not determine file type from input file
      [EPGMOLT200726120072620101.20070919004641.xml]
    </Message>
  </Error>
</FileErrors>
```

5.8.3 Item Errors

All items valid against the ECHO Format are also subject to data integrity validation according to ECHO business rules. This includes inserts, replacements, and updates to existing items. For each data integrity error discovered, ECHO Ingest will include the type of error, the invalid item ID, and the type of item. All data integrity errors are critical errors. For a comprehensive list of the errors which may occur during Ingest, refer to the ECHO Ingest Summary Report specification available on the ECHO website (<http://www.echo.nasa.gov/ingest/schemas/operations/docs/>).

All data integrity errors will identify the invalid item using itemId and include a detailed message regarding the cause of the error. The precise message will vary depending on the type of error. An overview of some common item errors is given in the following section.

5.8.3.1 GRANULE_NOT_EXISTS Error

If a granule deletion, partial update, or partial deletion references a granule which does not exist in the ECHO holdings, ECHO will return a GRANULE_NOT_EXISTS error and not process the specified record.

Code Listing 59. Referenced Granule Does Not Exist

```
<ItemErrorGroups>
  <ItemErrorGroup errorCode="GRANULE_NOT_EXISTS">
    <ItemError itemType="GRANULE" itemId="SC:g3asspb.004:24162750"
      level="CRITICAL">
      <Message>
        Validation error, SC:g3asspb.004:24162750. Granule does not exist.
      </Message>
    </ItemError>
  </ItemErrorGroup>
</ItemErrorGroups>
```

5.8.3.2 OUT_OF_DATE Error

If the date of a collection or granule in the ECHO database is more recent than the date on the same collection or granule in your input, ECHO will return an OUT_OF_DATE error and not process the specific record.

Code Listing 60. Input Data Older than the Current Record

```
<ItemErrorGroups>
  <ItemErrorGroup errorCode="OUT_OF_DATE">
    <ItemError itemType="COLLECTION" itemId="Collection1 V001" level="CRITICAL">
      <Message>
        Validation error, new last update date Tue Oct 10 13:00:00 EDT 2000 is
        before current last update date Thu Oct 10 13:00:00 EDT 2002
      </Message>
    </ItemError>
  </ItemErrorGroup>
</ItemErrorGroups>
```

5.8.3.3 BROWSE_NOT_EXISTS Error

If a granule or collection browse linking partial update or browse deletion references a browse record which does not exist in the ECHO holdings, ECHO will return a BROWSE_NOT_EXISTS error and not process the specified record.

Code Listing 61. Referenced Browse Does Not Exist

```
<ItemErrorGroups>
  <ItemErrorGroup errorCode="BROWSE_NOT_EXISTS">
    <ItemError itemType="BROWSE" itemId=":SC:MISBR.005:6561341:1.HDF-EOS"
      level="CRITICAL">
      <Message>
        Validation error, SC:MISBR.005:6561341:1.HDF-EOS does not exist.
      </Message>
    </ItemError>
  </ItemErrorGroup>
</ItemErrorGroups>
```

5.8.3.4 COLLECTION_REF_INVALID Error

If a granule insert references a collection record which does not exist in the ECHO holdings, ECHO will return a COLLECTION_REF_INVALID error and not process the specified record.

Code Listing 62. Collection Referenced by Input Granule(s) Does Not Exist

```
<ItemErrorGroups>
  <ItemErrorGroup errorCode="COLLECTION_REF_INVALID">
    <ItemError itemType="GRANULE" itemId="ScInputGranule001" level="CRITICAL">
      <Message>
        Referenced collection by short name [Non-Existant-Collection] and version
        [1] does not exist
      </Message>
    </ItemError>
  </ItemErrorGroup>
</ItemErrorGroups>
```

5.8.3.5 SPATIAL_INVALID Error

If a granule or collection insert or update contain invalid spatial data, ECHO will return a SPATIAL_INVALID error and not process the specified record. For information on spatial rules for avoiding invalid spatial errors, refer to section 3 of this document.

Code Listing 63. Invalid Spatial Coverage Area Data

```
<ItemErrorGroups>
  <ItemErrorGroup errorCode="SPATIAL_INVALID">
    <ItemError itemType="COLLECTION" itemId="MISR Level 2 Aerosol parameters
    V002" level="CRITICAL">
      <Message>
        Validation error, Spatial Validation Error BOUNDING_RECTANGLE #1 [13356
        [Element <1>] [Coordinate <1>][Ring <1>]]
      </Message>
    </ItemError>
  </ItemErrorGroup>
</ItemErrorGroups>
```

```
</ItemErrorGroup>  
</ItemErrorGroups>
```

5.8.3.6 TEMPORAL_INVALID_DATE_RANGE Error

If a granule insert or partial update changes the granule's temporal range to a value which is outside of its collections temporal range, ECHO will return a TEMPORAL_INVALID_DATE_RANGE error and not process the specified record.

Code Listing 64. Data Integrity Error Message

```
<ItemErrorGroup errorCode="TEMPORAL_INVALID_DATE_RANGE">  
  <ItemError itemType="GRANULE"  
    itemId="sample.granule.dat"  
    level="CRITICAL">  
    <Message>  
      The granule's start time [1994-04-09T12:00:00Z] is before  
      the collection's [1994-08-16T12:00:00Z]  
    </Message>  
  </ItemError>  
</ItemErrorGroup>
```

5.9 INGEST REPORTING

5.9.1 Ingest Summary Report

At the completion of each Ingest Job, Ingest will automatically generate an Ingest Summary Report in XML format for the job that details a job's activities as well as any abnormalities discovered in the input file(s). This report will still be created in the event that the job is deleted by an Ingest Operator. The reports are designed to be machine-readable and therefore may not be formatted for human consumption. The reports can be used to perform a post processing analysis of a completed job.

The Ingest Summary Report will be available in the designated provider report directory (located as agreed upon with ECHO Operations). This directory will be accessible from the ftp site for you to download reports and will be archived in the job directory for the Ingest Operator. (Data Partners will not have access to the job directory. If access to a report is needed after it has been removed from the ftp site, ECHO Operations should be contacted.) ECHO Operations will periodically cleanup Ingest reports older than 60 days from the FTP site.

For a full description of the Ingest Detail Report, refer to the schema and documentation referenced on the ECHO Website (<http://www.echo.nasa.gov/ingest/schemas/operations/docs/index.html>).

The Ingest Detail Report is applicable to the following types of items:

- Collection Inserts, Updates, Replacements, Deletions, and Reconciliation
- Granule Inserts, Updates, Replacements, Deletions, and Reconciliation
- Browse Inserts, Replacements, Deletions, and Reconciliation

The Ingest Detail Report consists of two parts: overview and details. Both of these sections are described below. For an example of a full Ingest Summary Report, see Appendix H.

5.9.1.1 Ingest Detail Report Overview

The overview section of the Ingest Detail Report provides a high level summary of what actions were performed while processing the ingest job.

includes the following information:

- Job Start & End Date
- Job Name (If Provided)
- Sequence Number (If Provided)
- Collection, Granule, and Browse Processing Totals including:
 - 1) Total Processed - The number of ingest actions performed including item inserts, updates, and deletions, collection/granule/browse inventory existence checks(+1 per inventory), collection/granule metadata verification (+1 per item).
 - 2) Total Inserted – The number of items inserted. This will include explicit inserts processed and items that are inserted as a part of the correct metadata verification actions.
 - 3) Total Replaced – The number of items replaced. This will include explicit replacements processed and items that are successfully replaced as a part of the correct metadata verification actions.
 - 4) Total Updated – The number of items that were partially updated, including partial adds and partial deletes.
 - 5) Total Deleted – The number of items which were deleted from the ECHO inventory. This number will not reflect items which have had their <DeleteTime> metadata value set for future deletion by the ECHO Ingest reaper.
 - 6) Total Rejected – The number of items for which an insert, replacement, update or deletion was rejected. When performing inventory verification, each inventory that cannot be processed will be listed as a single rejection. When performing metadata verification, each item which cannot be verified, or which fails a subsequent corrective insert or replacement action.
 - 7) Total Inventories – The number of inventories which verification was successfully performed, irrespective of how many items were found to be missing or unexpected. Since there is only 1 collection inventory, this field will report a '1' if collection inventory verification is performed. For granules and browse, this field will contain the number of inventories that were compared. This will **not** include the number of items in each inventory that were compared.
 - 8) Total Verifications – The number of items for which metadata verification was successfully performed, irrespective of whether the items were found to be matching or not.
- Job-level errors encountered during Ingest

Code Listing 65. Example of the Ingest Summary Report Overview Section

```
<Overview sequenceNumber="5734" endDate="2009-05-06T08:11:57.535-04:00"
  startDate="2009-05-06T08:07:51.703-04:00">
  <JobErrors />
  <ProcessingTotals>
    <CollectionProcessingTotals processed="0" rejected="0" deleted="0"
      updated="0" replaced="0" inserted="0" inventories="0"
      verifications="0"/>
    <GranuleProcessingTotals processed="604" rejected="2" deleted="0"
      updated="240" replaced="0" inserted="362" inventories="0"
      verifications="0"/>
    <BrowseProcessingTotals processed="240" rejected="0" deleted="0"
      updated="0" replaced="0" inserted="240" inventories="0">
```

```
        verifications="0"/>  
    </ProcessingTotals>  
</Overview>
```

5.9.1.2 Ingest Detail Report Details

The following details for each file will be included:

- a. File Name (this is the original file name as sent by the Data Provider)
- b. Collection, Granule, and Browse Processing Totals including:
 - 1) Total Processed - The number of ingest actions performed including item inserts, updates, and deletions, collection/granule/browse inventory existence checks(+1 per inventory), collection/granule metadata verification (+1 per item).
 - 2) Total Inserted – The number of items inserted. This will include explicit inserts processed and items that are inserted as a part of the correct metadata verification actions.
 - 3) Total Replaced – The number of items replaced. This will include explicit replacements processed and items that are successfully replaced as a part of the correct metadata verification actions.
 - 4) Total Updated – The number of items that were partially updated, including partial adds and partial deletes.
 - 5) Total Deleted – The number of items which were deleted from the ECHO inventory. This number will not reflect items which have had their <DeleteTime> metadata value set for future deletion by the ECHO Ingest reaper.
 - 6) Total Rejected – The number of items for which an insert, replacement, update or deletion was rejected. When performing inventory verification, each inventory that cannot be processed will be listed as a single rejection. When performing metadata verification, each item which cannot be verified, or which fails a subsequent corrective insert or replacement action.
 - 7) Total Inventories – The number of inventories which verification was successfully performed, irrespective of how many items were found to be missing or unexpected. Since there is only 1 collection inventory, this field will report a '1' if collection inventory verification is performed. For granules and browse, this field will contain the number of inventories that were compared. This will **not** include the number of items in each inventory that were compared.
 - 8) Total Verifications – The number of items for which metadata verification was successfully performed, irrespective of whether the items were found to be matching or not.
- c. File errors encountered during Ingest
- d. Item errors encountered during Ingest including:
 - 1) Error Code
 - 2) List of items associated with each error code including:
 - a) Item ID
 - (1) Dataset ID for collections
 - (2) Granule UR for granules
 - (3) Browse ID for Browse
 - b) Item Type
 - c) Error Level
- e. Rejection Reason

Code Listing 66. Example of the Ingest Summary Report Details Section

```
<Details>
  <MetadataFiles>
    <MetadataFile name="EchoDel.acdisc_main_ops.T20090506101026_0.xml">
      <ProcessingTotals>
        <CollectionProcessingTotals processed="0" rejected="0"
          deleted="0" updated="0" replaced="0" inserted="0" />
        <GranuleProcessingTotals processed="2" rejected="2"
          deleted="0" updated="0" replaced="0" inserted="0" />
        <BrowseProcessingTotals processed="0" rejected="0"
          deleted="0" updated="0" replaced="0" inserted="0" />
      </ProcessingTotals>
      <FileErrors />
      <ItemErrorGroups>
        <ItemErrorGroup errorCode="GRANULE_NOT_EXISTS">
          <ItemError itemType="GRANULE" itemId=" SOR3TSI6.009:sorce_tsi"
            level="CRITICAL">
            <Message>
              Validation error, SOR3TSI6.009:sorce_tsi granule does not exist.
            </Message>
          </ItemError>
        </ItemErrorGroup>
      </ItemErrorGroups>
    </MetadataFile>
  </MetadataFiles>
</Details>
```

5.9.2 Ingest Emails

ECHO Ingest is configured by ECHO Operations to notify one or more Data Partner personnel regarding Ingest processing. The following situations will generate an email to the configured addresses.

- **Ingest Startup/Shutdown** – When ECHO Operations starts up or shuts down Ingest.
- **Job Start** – When an Ingest job is detected by ECHO Ingest and queued for processing.
- **Job Completion** – When an Ingest job completes processing.
- **Manual Provider Pause/Resume** – When ECHO Operations manually pauses or resumes an Ingest provider.
- **Error Condition Provider Pause/Resume** – When ECHO Ingest pauses an ingest provider due to an internal error. ECHO Operations is also notified with the error information for analysis.
- **Out of Sequence** – When an Ingest job has been queued but cannot be processed because a prior sequenced job is expected. ECHO Operations configures this warning threshold in conjunction with each data partner.
- **No Input Received** – When no Ingest jobs have been detected for a configurable amount of time. ECHO Operations configures this warning threshold in conjunction with each data partner.

5.9.3 ECHO Ingest Accounting Tool

Data Partners may use the ECHO Ingest Accounting Tool (EIAT) in order to view the current status of submitted Ingest jobs and summary information for completed jobs. The EIAT can be access via the following URLs. Login

credentials require a username and password for a user which has been assigned 'provider role' for an ECHO provider in the associated mode.

- **Operations**– <http://www.echo.nasa.gov/eiat>
- **Partner Test** – <http://eiat-test.echo.nasa.gov>

5.10 INGEST CONFIGURATION

5.10.1 General Ingest Configuration

ECHO Operations will configure the following items for all ingest activity. Each of these values will be coordinated with data partners and adjusted as necessary to support operational Ingest.

- **Out Of Sequence Threshold** – The maximum amount of time that ingest will wait if receiving a job out of sequence for the other jobs in sequence before notifying Ingest Operators and the Provider contacts.
- **Reaper Interval** – The frequency at which the Ingest internal reaper will run to remove Collections, Granules, and Browse which have been marked with an expired deletion time.
- **Resource Wait Time** – The maximum amount of time that can elapse without receipt of resources for a file (i.e., browse images) prior to Ingest continuing in processing.
- **Input Quiet Time** – The amount of time that a file must have no modifications and be complete, before it is added it to the Job. E.g., file FOO.XML received from GSFC must not have been modified for the past 10 minutes before it is added to the Job.

5.10.2 Provider Specific Configuration

ECHO Operations will configure the following items on a per-provider basis. Each of these values will be coordinated with data partners and adjusted as necessary to support operational Ingest.

- **Provider Processing Threads (Pipelines)** – The number of processing threads dedicated to a provider's metadata ingest.
- **Provider Email Addresses** – Data Partner email addresses that will receive automated Ingest notification emails.
- **Report Directory** – The ftp directory into which Ingest Summary Reports will be placed upon job completion.
- **Input Directory** – The ftp directory from which Ingest will recursively search for metadata files and packages to process.
- **Delivery Mechanism** – The Ingest delivery mechanism (Package or Single-File) which will be detected for job creation.
- **Scan Interval** – The frequency that Ingest will scan the input directory for received Ingest packages or files.
- **No Input Received Threshold** – The maximum amount of time that can elapse without receipt of Ingest input before the Ingest Operator and Provider are notified that no input has been received.

CHAPTER 6. DATA MANAGEMENT

Data Management tasks may be performed through the Provider User Management Program (PUMP) or the ECHO DataManagementService, AccessControlService, and Group2ManagementService APIs if the provider would like to write their own tool. For more information regarding PUMP, refer to the Data Partners > Managing Data section of the ECHO website (<http://www.echo.nasa.gov>). For more information regarding how to use the ACL and Group Management functionality, refer to the “Group Management & ACLs How To” guide (ECHO_Guide_003), the “ECHO ACLs and Roles” (ECHO_OpsCon_013), and the “ECHO Group Management” (ECHO_OpsCon_014) operations concepts available on the ECHO website (http://www.echo.nasa.gov/?goto=controlled_docs).

6.1 ACCESS CONTROL CONCEPTS

The ECHO API facilitates allows Data Partners to control access to two types of data objects within the ECHO system:

- **Provider Objects** – Provider information items which are only managed by members of the Data Partner’s team and ECHO Operations. For example, provider orders, order policies, and provider groups.
- **Catalog Items** – Provider metadata items which are made available for discovery and ordering to end users. This includes the control of collection and granule items, but not explicitly browse.

While interacting with the ECHO access control capability, you will need to understand the following basic concepts. Each item is discussed in greater detail in subsequent sections of this document.

- **Groups** – Provider-defined groups of ECHO registered users that may have specific access to Provider Objects or Catalog Items based upon the group’s permissions. ECHO Operations also manages system level groups for their own access. Virtual system groups (e.g. guest users) are also available for specific purposes.
- **ACL** – An **Access Control List** is associated with each Provider Object or provider-defined set of Catalog Items and lists the permissions granted to a provider or system group.
- **Permissions** – Each Provider Object or Catalog Item has a specific set of permissions (e.g. Create, Read, Update, Delete) that may be granted to a group. Due to the nature of object types, not all objects will have the same available permissions.
- **Catalog Item Identifiers** – Each *Catalog Item* ACL has a unique set of identifiers which are used to determine which catalog items will be affected by the ACL.
- **Provider “Administrators” Group** – Each ECHO Data Partner will have a provider group named “Administrators” which will initially be granted all permissions on all objects and be given the ability to pass on any of their permissions.
- **Restriction Flag** – Metadata values specifying a provider specific restriction value scheme.

6.1.1 Groups

An ECHO Group is identified by the following information:

- **Name** – Unique group name within the Data Partner’s list of groups
- **Description** – Description of the purpose of the group and the users which it contains.
- **Members** – List of ECHO registered users who are a part of the group.
- **Provider** – The provider who owns the group, or possibly the system if it is a system level group.

ECHO groups are associated with an ECHO provider and appear only in the context of that provider. The ECHO API supports retrieving groups for a single provider at a time. Group names are scoped to the owning provider, therefore names only need to be unique within the list of groups owned by the provider, and not across all groups in the system. ECHO Data Partners may create as many groups as are needed in order to fulfill their data access control needs.

ECHO also supports the concept of system level groups which are managed by the ECHO Operations team. These groups allow the ECHO Operations team to manage their own access to Provider Objects and Catalog Items without needing to coordinate with each provider. Although ECHO Operations will have all permissions on all objects, they will continue to communicate changes with ECHO Data Partners to ensure good coordination.

In addition to ECHO Operations' managed system groups, ECHO also has the concept of *virtual* system groups. These groups are "managed" by the ECHO system and include a *Registered Users* group and *Guest Users* group. Due to the nature of these lists, ECHO will dynamically associate a user with one of these virtual groups when assessing permissions. Permissions to **both** of these groups must be independently managed.

The permissions to create new groups and view existing groups are managed by assigning permissions on the *Group Provider Object ACL*. Permissions to update or delete existing groups are managed by assigning permissions on the *Group Management Provider Object ACL*. When creating a new group, an initial group must be specified as the *Initial Management Group*. This group will be given permissions to update and delete the new group, and others may be added later.

6.1.2 Access Control List (ACL)

An Access Control List is responsible for linking groups to a specific *Provider Object* or *Catalog Item* and describing the permissions the group has been granted. As was described previously, *Provider Objects* include such things as provider orders, order policies, and groups. *Catalog Item ACLs* are assigned to a custom static or dynamic listing of collection and granule items. All provider objects and catalog items are **not accessible** by default. An ECHO Data Partner uses ACLs to grant permissions to ECHO groups so that group members will have access to specific objects or sets of catalog items. An ACL may exist without any assigned permissions. For example, an ECHO Data Partner may wish to not assign any permissions to the "Extended Services" *Provider Object*, or they may define a *Catalog Item ACL* which controls access to a specific data set, but choose to not assign any permissions at the present time.

6.1.3 Permissions

The permissions for each *Catalog Item* may be *View* or *Order*, without exception. The permissions for each *Provider Object* may be *Create*, *Read*, *Update*, or *Delete*, however the available permissions differs depending on the nature of that object. For example, the *Provider Policies* object can be granted the *Read* or *Update* permissions, while the *Provider Audit Report* object can only be granted the *Read* permission. Some permissions, such as the ability to read option definitions, are not grantable on some *Provider Objects* because access is open to any user. The following table outlines the grantable permissions for each *Provider Object*.

Table 3. Provider Object Grantable Permissions

Provider Object	Grantable Permissions	Description
Audit Report	Read	Allows the viewing of an audit report for actions associated with a specific provider.
Dataset Information	Read	Allows the usage of the reconciliation GetDatasetInformation() method.
Extended Services (all types)	Create, Update, Delete	Allows the creation, updating, and deletion of extended services.
Ingest Operations	Read, Update	Controls access to who can log into the EIAT (Not used by ECHO Ingest).
Groups	Create, Read	Allows the creation of new groups or viewing of existing provider groups.

Provider Object	Grantable Permissions	Description
Group Management	Update, Delete	Allows the updating or deletion of an existing provider group.
Option Assignments	Create, Read, Delete	Allows the assignment of an option definition to one or more datasets.
Option Definitions	Create, Delete	Allows the creation and deleting of an option definition.
Option Definition Deprecation	Create	Allows the deprecation of an option definition.
Provider Context	Read	Allows a user to act as a provider and perform all permitted provider actions.
Provider Holdings	Read	Allows the viewing of a provider's holdings (dataset & granule count).
Provider Information	Update	Allows provider information to be updated.
Provider Orders	Read	Allows the viewing of all orders associated with a specific provider.
Provider Order Resubmission	Create	Allows the resubmission of a provider's order.
Provider Order Acceptance	Create	Allows the acceptance of a provider's order. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Order Rejection	Create	Allows the rejection of a provider's order. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Order Closure	Create	Allows the closure of a provider's order. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Order Tracking Id	Update	Allows an order to be updated with a provider tracking ID. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Policies	Read, Update, Delete	Allows the editing of provider policies.
User	Read	Allows the viewing, updating, and deletion of an ECHO user.
Authenticator Definition	Create, Delete	Allows the creation, deletion of provider authenticators. (Not currently being used)

6.1.4 Catalog Item Identifiers

Each *Catalog Item* ACL will have a specific set of identifiers which designate the catalog items to which the ACL will apply. There are three main areas of identifiers: Catalog Item Type, Collection Identifiers, and Metadata Filters. Each of these areas are described in this section.

6.1.4.1 Catalog Item Type

When creating a *Catalog Item* ACL, you may choose to have it apply to *Collections*, *Granules*, or *Both*. The catalog item type chosen will determine whether the ACL will apply to collection items, granule items, or both types of items. For instance, if *Collections* is chosen then the ACL will be used by ECHO when granting access to collection items. This is useful if a collection doesn't have granules or if the granules will require different permissions. A catalog item type of *Granules* indicates that an ACL should be used by ECHO when granting access to granule items. If *Both* is chosen, then the ACL will be used by ECHO when granting access to both collection and granule items. This is useful if the collections and granules will have the same permissions.

6.1.4.2 Collection Identifiers

Each *Catalog Item* ACL has a list of collections which is used to identify collections or granules within collections to which the ACL applies. Collections may be identified in the following ways:

- **Selected List** – A static list of collections which must be manually managed.
- **Any Collection** – A dynamic list of all collections in the Data Partner's holdings
- **Collection Pattern Matching** – Patterns may be selected to perform text matching on the Data Set ID, Short Name, and/or Version ID.

6.1.4.3 Metadata Filters

In order to facilitate access control of collections and granules based on metadata fields, a *Catalog Item* ACL may contain collection and granule metadata filters. The filters are based on temporal fields, restriction flag values, or specific granule UR values. Temporal filters may apply to the acquisition, production, ECHO Insert, or ECHO Last Update field and may be described using an intersection, containment, or disjoint comparator. The full listing of filters is included below:

- **Collection Filters**
 - **Temporal Range**
 - **Rolling Temporal Range**
 - **Restriction Flag**
- **Granule Filters**
 - **Temporal Range**
 - **Rolling Temporal Range**
 - **Restriction Flag**
 - **Granule UR Pattern**

6.1.5 Provider “Administrators” Group

Each ECHO Data Partner will be initially configured with a group named “Administrators.” This group will be granted all permissions on all *Provider Object* ACLs and the permissions to manage *Catalog Item* ACLs. This group is initially managed by itself and the system “Administrators” group, which is the ECHO Operations team. The provider “Administrators” group may manage *Provider Object* ACLs and grant permissions to other groups, however the ability to manage and grant permissions may not be given to other provider groups. This is done to provide control over the management of *Provider Object* permissions. The ability to grant management of *Catalog Item* ACLs can be granted to other groups. This is distinctly different than *Provider Object* ACL management and is designed to allow for groups such as a Data Partner's User Services team to manage *Catalog Item* ACLs without coordinating with the provider “Administrators.”

6.1.6 Restriction Flag

The restriction flag is a decimal value which is specified in a Data Partner's collection or granule metadata. Data Partners may configure a *Catalog Item* ACL which utilizes the restriction flag value within a record's metadata. The availability of data can then be changed during ECHO Ingest through the usage of partial updates or full metadata replacement. A granule or collection's restriction flag cannot be updated through the API.

6.2 ACCESS CONTROL RECOMMENDATIONS

The following recommendations outline some suggested access control mechanisms to facilitate a Data Partner's data management needs, along with those of the ECHO Operations team.

6.2.1 Data Mgmt & User Services Groups

As has been described, a provider “Administrators” group will be created and granted all permissions on all *Provider Objects*. It is suggested that membership in this group be limited to those individuals who have need of managing access to the provider. There are two distinct “roles” which may be facilitated by provider groups and those are a “Data Management” group and “User Services” group. The “Data Management” group would be given a subset of *Provider Object* ACL permissions relevant to their job role. The “User Services” group would have a slightly different subset of *Provider Object* ACL permissions, but have the ability to manage *Catalog Item* ACLs.

6.2.2 Catalog Item ACLs

When initially configuring *Catalog Item* ACLs, the ECHO Operations team will define two ACLs:

- **All Collections (No Granules)** – This ACL dynamically applies to all collections within a Data Partner’s holdings, and only affects access to collection metadata. The ECHO Operations team uses this ACL to assign view permissions for the WIST valids process.
- **All Collections and Granules** – This ACL dynamically applies to all granules in all collections within a Data Partner’s holdings, and affects access to both collection and granule metadata. This ACL is useful for assigning full permissions to members of the Data Partner team. The ECHO Operations team uses this ACL to assign *view* permissions for the System “Administrators” group.

The following *Catalog Item* ACL conditions are also suggested in order to facilitate general data management:

- **Public Collections and Granules** – This ACL dynamically applies to all granules within a static listing of public collections. As collections are added to the provider’s holdings, the provider may add the collection to the listing used by this ACL. *View* permissions to this ACL may be granted to the *Registered Users* and *Guest Users* system level groups to allow for data discovery.
- **Orderable Granules** – This ACL dynamically applies to all granules within a static listing of collections within which the granules are orderable. *Order* permissions to this ACL may be granted to the *Registered Users* and *Guest Users* system level groups to allow for order creation and submission.

6.3 RECONCILIATION

Data Partners may perform data reconciliation via two separate methods, via ECHO Ingest processing or the ECHO Data Management Service API. Both of these methods are described in the following sections.

6.3.1 ECHO Ingest

Through the ECHO 10.0 Ingest Schema, Data Partners may perform two types of metadata reconciliation, *metadata verification* and *inventory verification*. These two verification processes are described below:

- **Metadata Verification** – A full reconciliation of a collection or granule metadata item to include verification of all fields. ECHO will automatically attempt to correct differences within its holdings.
- **Inventory Verification** – A shortened reconciliation mechanism which will allow for the identification of inventory items which are missing from the ECHO holdings or should no longer be held by ECHO. Verified inventory items include collections, granules within a specified collection, or browse records associated with granules within a specified collection.

Both of these methods allow Data Partners to take advantage of Ingest’s parallelized data processing and detailed reporting mechanism. For a full explanation of how to utilize these reconciliation capabilities, refer to Section 5.7 of this document.

6.3.2 Data Management Service

This historical method uses the GetDatasetInformation method found on the DataManagementService API. Data Partners use the API to request a subset of metadata to be generated by ECHO for granules within a specific collection filtered by temporal range, online availability, browse availability, and visibility. This method utilizes an optimized internal mechanism to pull the necessary information for each granule matching the request. Due to the large amount of data that may be returned by this method, output is generated in an XML file which is delivered via FTP Push to a specified location.

If temporal ranges are specified, the range type field in the granule must be between the start and stop times. If the range type is acquisition, a range intersection (rather than containment) check will be performed. Although the range types could be repeated, there is no benefit to repeating the same range type.

All of the restriction fields (dataset ID, ranges, online flag, browse flag, and visibility) will be joined together with the Boolean AND when the search is performed. The standard FTP URL format is:

```
ftp://[user ID:password@]host_name[:port]/[path name/][file name]
```

The file name is ignored and a unique name will be generated by ECHO to ensure uniqueness of the file name. ECHO will default to send an email notification to the requesting user when the process has completed. Data Partners may request that ECHO suppress these emails. In this case, the target ftp area will need to be monitored in order to determine when file generation has completed.

CHAPTER 7. FULFILLING ORDERS

The ECHO system acts as an order brokering service between end users and Data Partners. Users may submit orders for collections and granules that have been ingested into ECHO and marked as 'orderable.' ECHO will keep basic metrics and information regarding the order including the orderer's information, order contents, and historical order updates.

Through the means described in Section 6, Data Partners can configure access control for their data to determine which users are allowed to order data. All information in presented in this section will assume that Data Partners have configured their data access mechanisms as needed to support their needs. Data Partners generally use Provider User Management Program (PUMP) or the API itself to configure and track orders submitted to their data center. However, a data-center specific tool may be written to use the ECHO API in order to facilitate order management.

For Order Fulfillment API Documentation, Order Fulfillment Types XML Schema, and the Order Fulfillment Service WSDL refer to the Data Partner Tools page on the ECHO website.

7.1 ORDER FULFILLMENT

Data Partners who would like ECHO to facilitate order brokering services must implement an "Order Fulfillment Service" based on the OrderFulfillmentService API. The API specification and documentation can be found here (<http://api.echo.nasa.gov/echo/apis.html>) in the "Order Fulfillment" section. In short, an "Order Fulfillment Service" implements the necessary methods in order to receive order quotes, submission, and cancellation requests from ECHO. If the service is correctly written according to the specification made public by ECHO, then the provider will successfully receive order workflow information from ECHO. The data partner may then perform whatever internal activities are needed in order to fulfill the order.

While fulfilling an order, the Data Partner may call methods on the ECHO OrderProcessingService API to update, accept, reject, or close existing ECHO orders which have been submitted. The API specification and documentation can also be found here (<http://api.echo.nasa.gov/echo/apis.html>).

7.1.1 Provider Policies

Each ECHO provider has a set of "provider policies" which specify necessary information in order to facilitate and manage orders placed for data within a Data Partner's holdings. These policies are configurable using PUMP or the ProviderService API. It is important that Data Partners correctly configure these policies in order to ensure orders are properly brokered by ECHO. The following topics discuss the policies that Data Partners are to maintain.

7.1.1.1 Retry Information

When an order is submitted through ECHO, it is possible that the Data Partner's Order Fulfillment service will be unavailable due to maintenance or other situations. Were this to occur, ECHO can queue any orders which fail submission to be retried later. Each Data Partner may configure their own retry policies. This policy includes the following:

- **Retry Attempts** – The number of retries ECHO will perform until failing order submission.
- **Retry Wait Time** – The number of seconds ECHO will wait between retries.

It is important to ensure that the retry policies are sufficient to account for Data Partner regular maintenance and possible extended failure situations. ECHO recommends a retry policy that attempts order submission every hour for at least 48 hours. As an example, a policy which has a wait time of 3600 seconds and 48 attempts will ensure that this 48 hour retry policy is met.

7.1.1.2 Supported Order Transactions

ECHO supports the following order actions:

- Quote
- Submission
- Cancellation

A Data Partner can configure their ECHO provider to support any of these order actions. Order submission must be explicitly chosen in order to allow ECHO to accept orders for a provider's data. Data Partners with data that has an associated cost can support quoting to allow users to request an initial quote outlining the cost of their order prior to submission. Data Partners whose Order Fulfillment service allows for orders to be cancelled after submission can choose to support order cancellation through ECHO.

7.1.1.3 Endpoint for Receiving Orders

The ECHO provider policies must be configured with the Data Partner's Order Fulfillment service's "end point," a Uniform Resource Identifier (URI), also known as network address, which ECHO will use for all order communication. The network address is usually either an IP or HTTP address.

It is important to make sure that your network will allow ECHO to communicate with this URL. Please contact ECHO Operations to get the originating IP address information.

7.1.1.4 Order Creation Settings

Data Partners have the capability of configuring the following order settings which ECHO will enforce during order creation:

- **Duplicate Order Item Support** – This indicates whether or not the provider supports separate order items with the same catalog item in the same order. If this is set to false ECHO will not allow an order to contain more than one order item to have a particular catalog item.
- **Maximum Items Per Order** – This indicates the maximum number of order items which can be added to a single order. If no value is set, then order sizes are unbounded.

7.1.1.5 Secure Socket Layer (SSL)

In order to support SSL encrypted communication between ECHO and the Data Partner's Order Fulfillment service, the Data Partner's SSL Certificate must be configured in ECHO. Once the public, PEM-encoded key from the certificate has been entered or updated in the provider policies, ECHO Operations must acknowledge and activate the changes. ECHO Operations will be notified via email from the ECHO system when changes are made. Once activated, the new SSL certificate will be used during secure communication. It is important that Data Partners ensure that their server hosting the order fulfillment service is correctly configured to utilize SSL.

While troubleshooting SSL communication between ECHO and the Data Partner's Order Fulfillment service, the following items should be verified:

- The configured endpoint in PUMP must specify the https:// protocol
- Firewall access must be granted by the Data Partner allowing ECHO to communicate to the secure port.
- SSL communication must be enabled by the Data Partner in the PUMP "Provider Policies."
- ECHO Operations must activate an updated SSL certificate in PUMP.
- The certificate placed in PUMP should be PEM encoded. An example is shown below.

```
-----BEGIN CERTIFICATE-----
MIIDWzCCAsSgAwIBAgIDD0nJMA0GCSqGSIb3DQEBBQUAME4xCzAJBgNVBAYTA1VT
MRAwDgYDVQQKEwdFcxVpZmF4MS0wKwYDVQQLEyRfcXVpZmF4IFNlY3VyZSBDZXJ0
aWZpY2F0ZSBBdXRob3JpdHkwHhcNMTAwMjA5MjAwMTI3WhcNMTIwMzEzMDEwNDU5
WjCB5TEpMCcGA1UEBRMgaEZRmWR2Vld6Sk12OXJ4LXBNC00ZUtDdS1WODNYQnQx
CzAJBgNVBAYTA1VTMRGwFgYDVQQKFA8qLmVjaG8ubmFzYS5nb3YxEzARBgNVBASt
cmNlcy9jCHMgKGMpMTAxLzAtBgNVBAStJkRvbWVpbiBDb250cm9sIFZhbG1kYXRl
ZCAtIFJhcGlkU1NMKFIPMRGwFgYDVQQDFA8qLmVjaG8ubmFzYS5nb3YwgZ8wDQYJ
KoZIHvcNAQEBBQADgY0AMIGJAoGBAMNvxaof0SEs3RS3k6Y2enOK07mnAhE5/vNt
AridvGBWbAYzZqfBQUX6RBj0H5xeudZnPyv5XbMEAm3yroRtXC9XJamoalZz3Q6
4c6eZE0Ly3W2C1xP3KnwmJh4nV4kNwtjxjhvkA1C1CD4VM1B79cfVBjKp08KNWBt
Pjdmy6ozQhJNZh2t8Gu4ATA6BgNVHR8EMzAxMC+gLaArhilodHRwOi8vY3JsLmdl
b3RydXN0LmNvbS9jcmxzL3NlY3VyZW50LmNybDAfBgNVHSMEGDAWgBRI5mj5K9Ky
lddH2CMgEE8zmJCf1DAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAWIwDQYJ
OTAdPBuQbFq9RsevLaqeScdMprNSBXVflvDo/nNMyQvZpuZZGFMKBAGyMbVqXNIu
FG7QYHL+JoDFHN4BDQw9c01Ic018uioiF5HXAJ9VUURh1zmfeznhhZ+RlFNayWM=
-----END CERTIFICATE-----
```

7.1.1.6 Custom Properties

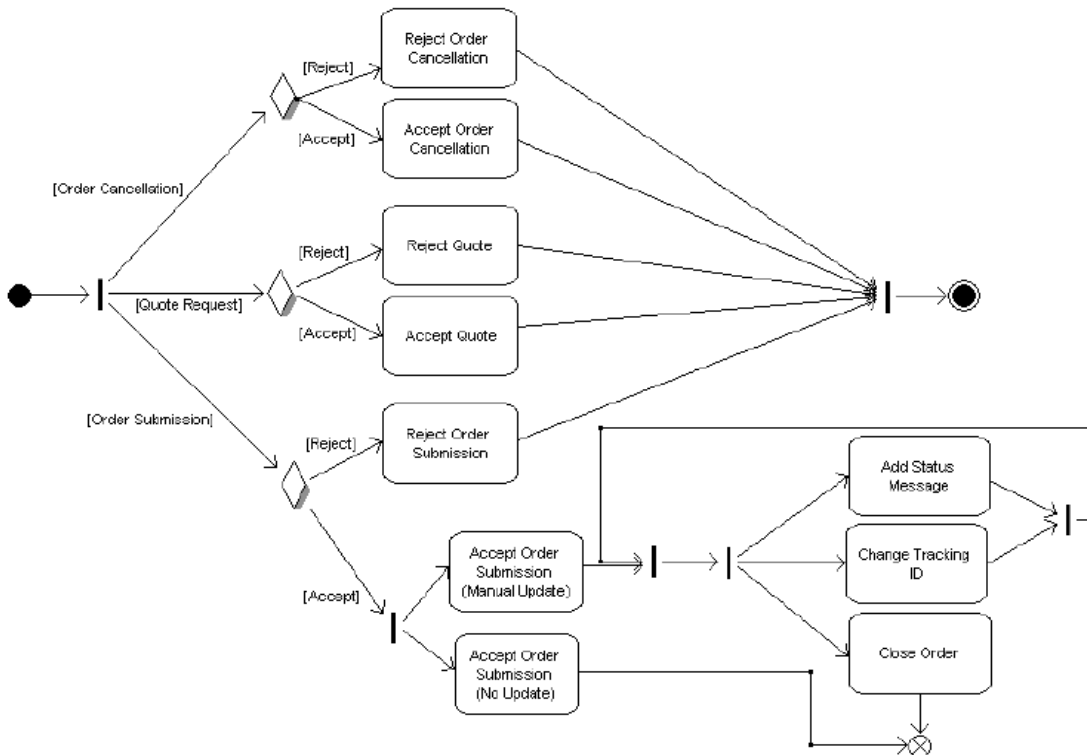
Data Partners wishing to have additional information provided to their Order Fulfillment service during order communication may configure a custom XML block of information which ECHO will include in every order transaction.

7.2 ORDER WORKFLOW

It is an ECHO Client's responsibility to facilitate the creation of an order within ECHO. ECHO will subdivide an order by data center so that each Data Partner will be able to interact with the portion of the order for which they are responsible for fulfilling. The following diagram shows the workflow of an order as it passes into a Data Partner's order Order Fulfillment Service, described in Section 7.1. The entry point to this workflow is when ECHO requests an order quote, submission, or cancellation. Order cancellation and quotes are typically atomic synchronous activities, so their workflows are shown as terminating after the action is accepted or rejected.

Order submissions may be accepted with either an 'automatic' or a 'manual' distinction. An 'automatic' distinction signifies that the order has been completed by the provider and the order will be closed by ECHO. If the 'manual' distinction is used, this indicates that the provider will notify ECHO through the OrderProcessingService as the order is fulfilled. Shown in the diagram are the most common updates, including the addition of a provider tracking id, status updates, and the final closure.

Note that quote, submissions, and cancellations and quotes may also be postponed, however these workflows are not displayed.



An ECHO Order associated with a single data partner will have on of the following states during its lifecycle:

- **Validated** – Order has been validated and all information is correct
- **Not_Validated** – Order has not been validated
- **Quoting** – Order has been sent to provider for quoting
- **Quote_Rejected** – Provider does not support quoting
- **Quote_Failed** - Order failed quoting
- **Quoted** – Provider has responded with quote
- **Submitting** – Order has not yet reached the provider
- **Submit_Rejected** – Order rejected by provider
- **Submit_Failed** – Order failed to send to provider
- **Processing** – Provider received order and process
- **Cancelling** – In the process of Cancelling
- **Cancelled** – Order has been cancelled (no change from Closed status)
- **Closed** – Order has been completed

7.3 ORDER INFORMATION

The following sections outline the information included in an ECHO order transmitted to a Data Partner through the Order Fulfillment Service.

7.3.1 ECHO Order vs Provider Order

Within ECHO, there are two distinct items which may be referred to as an “order.” These include an “ECHO Order” and a “Provider Order.” These two concepts are described below:

- **Provider Order** – A provider order contains information regarding the items which have been ordered from a Data Partner’s holdings, an associated provider tracking ID (discussed later), and other quote and state information. A provider order does not include order information.
- **ECHO Order** – An ECHO order contains information about the user and client submitting the order, notification level, and a list of provider orders containing the information listed above.

An “ECHO Order” is identifiable by an **OrderGuid**, which is a unique identifier for that order within ECHO. The “Provider Order” is identifiable by a combination of the **OrderGuid** and **ProviderGuid**, the internal identifier used by ECHO to identify a Data Partner.

7.3.2 Provider Tracking ID

Data Partners often have an internally managed order tracking system which uses a different ID than the ECHO order ID. In order to facilitate improved order tracking capabilities, a Data Partner’s Order Fulfillment Service may choose to update the provider’s ECHO order with this internal tracking ID. Data Partners should update the corresponding order in ECHO with their unique ID as soon as it is assigned.

7.3.3 Order Items

Each order contains a listing of order items, which match to records within the Data Partner’s ECHO holdings. Each order item included in the order includes the following information:

- **Granule UR** – The Data Partner’s unique granule identifier which will contain a value if the order item is a granule.
- **Producer Granule ID** – The granule’s producer granule identifier which will contain a value if the order item is a granule and the provider has supplied a value for the metadata field.
- **Dataset ID** – The Data Partner’s unique collection identifier to which the granule belongs if the order is for a granule, and the identifier of the collection being ordered if the order is for a collection.
- **Catalog Item Guid** – The ECHO unique identifier for the catalog item being ordered.
- **Quantity** – The quantity of this order item being ordered. If the order is for electronic delivery (that is, FTP), then the quantity should always be one. If the order is for the delivery of pre-packaged media, then the quantity can be considered the number of media that are being requested.
- **Price** – The predetermined price of the item.
- **Option Selection** – Additional ordering options for this item. The option selection comes from the client created output of the option definition ECHO Form which the user has completed. The selection will be in an XML format as defined by the ECHO Form specification. If the original item had multiple option definitions, the name of the option selection can be used to determine which set of options the user selected.

7.3.4 User Information

Each order contains information about the user who has submitted the order. ECHO will enforce that guest users submit the necessary information as well as registered users. This user information includes the following information:

- **User Id** – The ECHO User ID (e.g. john.doe)
- **Shipping Address** – The user's shipping address indicating where an order should be physically shipped.
- **Billing Address** – The user's billing address identifying who will be paying the bill for the order.
- **Contact Address** – The user's contact address identifying who is placing the order.
- **User Domain** – The domain within which the user is associated. (e.g. K12)
- **User Region** – The region in which the user is located. (e.g. USA)

As shown, the user's information may contain three address fields, however only the Contact Address is required. Each address field contains the same fields, but may contain different information depending on how the user has submitted their order. The following fields are included in each of the three addresses:

- **Name** – The user's first and last name and role (e.g. research assistant).
- **Address** – The user's street address. (Optional)
- **Phone Number(s)** – A list of the user's phone numbers. (Optional)
- **Email Address** – The user's email address.
- **Organization** – The user's organization name.

Some additional information for these fields is given in the following sections.

7.3.4.1 Address

The address field is a structure containing the following fields:

- **Name** – The name that the user uses to describe the address (for example: Home, Work, Project, etc.)
- **US Format Flag** - Indicates whether a provider can rely on the address looking like a standard US address
 - 1) The rule is that the system will validate those addresses that have the US Format flag set by checking that Street1, City, State, Zip Code, and Country are all set
 - 2) If the US Format flag is clear, then only Street1 and Country are required.
- **Streets** – ECHO allows for up to 5 street address lines, each 32 characters long.
- **City** – The user's city.
- **State** – The user's state, if applicable.
- **Zip** – The user's zip code, if applicable.
- **Country** – The full name of the country. This should conform to the published list of approved country codes.

The ECHO system allows user to enter a freeform string to define their contact, shipping, or billing address. In cooperation with the EOSDIS User Services Working Group, ECHO has created a list of approved country names that complies with the [ISO 3166](#) standard. This list of names will appear in the ECHO supported PUMP and WIST tools for for user registration and order creation in WIST. ECHO will support all

countries on this list and additional user-entered countries. ECHO Data Partners are encouraged to use this list in their applications to provide a consistent list of countries to all ECHO users. The full list is available at the following location:
<http://www.echo.nasa.gov/documents/common/ApprovedCountryList.xls>.

7.4 ORDER OPTIONS

The term "order option" refers to the mechanism by which a Data Partner can request specific information from a user at the time of ordering. For example, an order option may require that users provide ftp-push information, or specific subsetting information. The ECHO Forms Specification defines the structure and content of the XML documents which are used in ECHO to specify the order options assigned to a collection within the ECHO data catalog. The term "ECHO Form" is used to refer to the XML document created using the standards in the ECHO Forms Specification.

It is the responsibility of the ECHO Data Partners to design, upload, and assign order options to their catalog items using the ECHO Forms Specification. If a provider's collection does not allow ordering, then an order option may not be assigned. It is also the responsibility of the Data Partner to verify that their order options correctly define the information needed to complete the order process.

It is the responsibility of an ECHO Client Partner which is offering the service of ordering ECHO data to implement functionality that will display the ECHO Form. It is strongly encouraged that a Client Partner implementing the ordering functionality ensure that the entire ECHO Forms Specification can be represented. ECHO Data Partners may choose to utilize any and all parts of the specification. If ordering is not a feature which is included in the scope of a Client Partner's work, then they need not implement the ECHO Forms Specification.

Note: Not all catalog items have associated definitions – in which case they do not need option selections when ordering. However, if there are any options that are required by that catalog item, they will have to be filled out before an order can be validated and/or submitted (or else an error will be returned by ECHO).

7.4.1 ECHO Forms Specification

As mentioned previously, an ECHO Form is an XML document outlining the information which is to be requested from a user during the order process and how an ECHO client should facilitate the information gathering. The ECHO Forms Specification, outlines the complete set of syntax and features utilized by ECHO Data Partners when creating ECHO Forms. The specification also describes the workflow and user interface elements which are used as form controls.

For further information regarding ECHO forms, including the specification and sample forms, please visit the "Data Partners >> Fulfilling Orders" portion of the ECHO Website.

7.4.2 Option Definitions

Data Partners may manage their option definitions through PUMP or the DataManagementService API methods. An ECHO option definition contains the following information:

- **Name** – The identifying name for the form which must be unique amongst all option definitions created by a Data Partner. This value will be displayed to the users to identify an option definition.
- **Scope** – The scope of the option definition. This may be 'system' or 'provider' and is described further in section 7.4.2.1.
- **ECHO Form** – The XML ECHO Form outlining the information which is to be requested from a user during the order process and how an ECHO client should facilitate the information gathering.

- **Description** – A textual description of the form which will be displayed to the user to give additional information regarding the form. This is useful when multiple option definitions are assigned to the same collection. The description can be used to help a user know which definition they want to use.
- **Sort Key** – This optional field may be used to define a sort order for definitions. This is also useful when multiple option definitions are assigned to the same collection. The values in this field will be sorted based on a standard text sort order.

7.4.2.1 Provider vs. System Forms

The ECHO Forms that are created and uploaded by Data Partner are referred to as "provider level forms." This designates the fact that they are visible only to the Data Partner who uploaded the form. There are also ECHO Forms that are created and uploaded by the ECHO Team, and they are referred to as "system level forms." These forms are visible to all providers. The purpose of system level forms is to centralize common option definition elements which can be shared by all providers in order to reduce duplication of form components.

7.4.2.2 Definition Management

An option definition cannot be edited once it has been created. This is due to the fact that an order may have been created against the option definition. Were the definition to be changed before the order was submitted, then the option selection within that order would be invalid for all items it had been applied to. For that reason, forms can only be deprecated or deleted.

If a change is needed, a new form should be created, assigned to the appropriate collection(s) and the previous option definition should be deprecated. Deprecation will allow the option definition to persist within ECHO. Orders which have been created or submitted against this option definition will continue to process successfully. New orders will not be able to use this deprecated definition.

When an option definition is deleted, all orders that have been created and validated with that form, but not submitted, will no longer be valid. All orders that have been submitted with the deleted form will not be affected.

Note: When deleting an option definition, ECHO must perform internal bookkeeping to disassociate order items with the deleted item. Due to the potentially large number of order items, this activity may take a significant amount of time. Providers are encouraged to utilize the deprecation capability and wait at least 60 days before deleting the option definition. ECHO has a 60 day retention period, so waiting beyond this time will eliminate the potential performance issue

7.4.3 Option Assignments

Within ECHO, an option assignment must be created between an option definition and collection in order to signify that order options exist during the ordering process. Multiple definitions may be assigned to a single collection, and a definition can be assigned to multiple collections. An option assignment may also include an XPath statement which will filter on a granule's metadata to determine whether the granule should use the assigned option definition, or not. For instance, the following XPath expression may be used as part of an Option Assignment to include a specific option definition if the "SWIR_ObservationMode" and "VNIR1_ObservationMode" additional attributes have a value of "ON". As you can see, the XPath starts with the `/results/provider/result/` path. The remaining path, beginning with the `GranuleURMetadata` element, is evaluated against the Granule Results DTD, found here: <http://api.echo.nasa.gov/echo/dtd/ECHOGranuleResults.dtd>. Any combination of expressions and metadata fields from the granule results may be used as a part of an Option Assignment XPath statement.

```
boolean(/results/provider/result/GranuleURMetaData/AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName='SWIR_ObservationMode' and  
  AdditionalAttributeValue='ON']) and
```

```
boolean(/results/provider/result/GranuleURMetaData/AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName='VNIR1_ObservationMode' and  
  AdditionalAttributeValue='ON']) and
```

CHAPTER 8. GETTING STARTED

8.1 INTRODUCTION

As an ECHO Data Partner, the ECHO operations staff will be your primary contact. There are also a number of online resources at your disposal to assist you as well. The ECHO Home page, located at <http://www.echo.nasa.gov/>, is a very good start. All the documentation needed to assist further can be found as one navigates down in the ECHO Home Page to include the, “ECHO Data Partner's Guide” located at http://www.echo.nasa.gov/documents/data_partners/ECHO_Data_Partner_User_Guide.pdf .

Before ingest of a new provider's metadata takes place, ECHO will need to have an understanding of expected metadata volume, characteristics, delivery mechanism, frequency, relationship to the study of Earth science, as well as contact information and accessibility of the new provider. This information prepares ECHO with the information needed to provide any needed services and hardware support to the new provider.

8.2 ECHO ENVIRONMENTS

ECHO Operations supports three environments for utilization by providers. A short description is given below. For additional information, reference the “ECHO Systems” portion of the ECHO website (http://www.echo.nasa.gov/?goto=systems_overview).

- **Operations** – The Operational system is the high performance, volume, and availability environment utilized by the external science community. This system is monitored and supported 24/7. New ECHO versions are released monthly.
- **Partner Test** – The Partner Test system provides a stable test environment for ECHO's client and data partners. New ECHO versions are released into this system approximately 2 weeks prior to the schedule Operational release date to allow for sufficient testing. ECHO Partners are encouraged to verify ingest and order capabilities in this system when a new version has been released.
- **Testbed** – The Testbed system provides a preview of new ECHO functionality and an environment for testing capabilities which may not become operational. There is often very little metadata available in this mode, but it is fully functional. Use is subject to a Memorandum of Understanding. The next operational ECHO version is released into this system approximately 45 days before its schedule Operational release date.

8.3 ECHO PROVIDER IDENTIFICATION

8.3.1 Data Center ID

You will need to identify your data center id/data provider for use in ECHO. The name chosen for this will become the basis of your provider's existence in ECHO. Changing this value after data has been ingested is possible, but strongly discouraged. The data center id chosen will need to be 10 characters or less and unique against all other providers in the operational ECHO. ECHO Operations can assist with suggestions if needed. This data center id will appear in the following locations within the ECHO system:

- ECHO API
- ECHO Metadata Item IDs
- WIST Calendar
- WIST Metadata Results
- ECHO Holdings Report
- ECHO Website

- PUMP
- Ingest Reports
- Ingest FTP Space

8.3.2 Provider Administrators

ECHO Operations will ask for an initial listing of ECHO user IDs which will be granted administrator access to your ECHO provider. If individuals you wish to have administrative access do not have an ECHO registered account, they may create one by visiting either WIST or PUMP for the mode in which your provider is being configured. See the ECHO Website (http://www.echo.nasa.gov/?goto=systems_overview) for system-specific links to both of these applications.

8.4 ECHO INGEST SETUP

The following sections outline Ingest related topics which ECHO Operations will be discussing with you, as a new provider. Section 5 of the Data Partner User Guide provides detailed information about the ECHO Ingest process.

8.4.1 Ingest Notifications

ECHO Ingest will be configured to send email notifications to Data Partner associated individuals regarding Ingest activity. It is recommended that more than one person associated with an ECHO Data Partner receive Ingest related notifications. ECHO Operations will coordinate with you regarding the correct email addresses. For a full list of notifications Ingest will send, refer to Section 5.9.2 in the Data Partner User Guide.

8.4.2 Ingest Summary Reports

After job completion, ECHO Ingest generates an XML Ingest Summary Report (ISR) containing job statistics including the number of records processed, rejected, deleted, updated, replaced, inserted, as well as verification and inventory results for the three metadata types; collections, granule and browse. The ISR also contains details regarding individual job, file, and/or item level errors encountered during processing. ECHO Ingest places this report file in the Data Partner's ftp space for retrieval and manual or machine processing. See Section 5.9.1 of the Data Partner User Guide for additional information regarding Ingest Summary Reports.

8.4.3 Packaging Method

ECHO ingest supports two ftp metadata delivery mechanisms; package and single file. Single file delivery accepts individual collection, granule, and browse metadata files and creates an ingest job from received files according to a configured frequency. Package file delivery accepts zip files containing metadata files and a manifest specifying package contents and other identifying information. Browse image files are to be transferred as independent files, irrespective of metadata delivery mechanism. It is recommended that providers utilize the package delivery mechanism for metadata submission, as it offers greater control of the processing order to the provider and allows for additional checks for completeness of jobs before ingest commences. For additional information regarding this topic, refer to Section 5.2 of the Data Partners User Guide.

8.4.4 Delivery Directory Structure

ECHO recursively searches the configured input directory for Data Partner submitted ingest files. Data Partners, with the help of ECHO Operations, can create a directory hierarchy, if desired, for receiving Ingest files.

8.4.5 Input Detection Frequency & Threshold

ECHO Ingest will scan for received metadata files or packages according to a configured interval. ECHO Operations will work with you to determine what the optimal configuration value is, balancing metadata currency needs and job creation rates.

8.4.6 No input warning threshold

ECHO ingest will keep track of the time when the last metadata was received. If a provider defined interval of time has passed without ECHO receiving additional metadata for ingest, an email notification will be sent alerting the data provider and the ECHO operator of this situation. This threshold can be adjusted to suit the submission schedule of the data provider.

8.4.7 Out of Sequence Notification Interval

Data providers that utilize the package delivery mechanism for metadata submission, and specify a sequence number within the package file, will have the desired order of ingest for their various packaged jobs enforced. If a package is received that does not contain the next correct sequence, the ingest process will wait until that sequence is received. If a provider configurable period of time is exceeded without receiving the properly sequenced package, the ECHO operator and data provider will receive an email notification alerting of this condition.

8.5 ECHO METADATA

8.5.1 Metadata Data Model

The following three metadata constructs are utilized by the ECHO system:

- **Collection** - A grouping of science data that all come from the same source, such as a modeling group or institution. Collections have information that is common across all the granules they contain and a template for describing additional attributes not already part of the metadata model.
- **Granule** - The smallest aggregation of data that can be independently managed (described, inventoried, and retrieved). Granules have their own metadata model and support values associated with the additional attributes defined by the owning collection.
- **Browse** - An image which provides a high-level view of the associated granule or collection metadata item. Browse images are not spatially enabled, but are very useful during data discovery and cross-referencing to other granules or collections.

For additional information regarding the ECHO data model, refer to the following HTML documentation <http://www.echo.nasa.gov/ingest/schemas/operations/docs/index.html> and Sections 4.1 and 4.2 of the Data Partner User Guide.

8.5.2 Spatial Representations

The ECHO system accepts spatial data represented in the Cartesian and Geodetic coordinate systems. ECHO also accepts spatial information representing orbital data. Each coordinate system has associated limitations. You should choose a coordinate system based on the size and projection of the original data of the spatial area covered. You may not combine spatial types for granules within the same collection as the representations are mutually exclusive. For additional information, refer to Section 3 of the Data Partner User Guide.

8.5.3 Special Considerations

The following items should be considered as you are generating your metadata for submission to ECHO. Refer to the associated sections in Section 4 of this document.

- Additional Attributes
- TwoD Coordinate System Representation
- Collection Spatial Representation
- Granule Spatial Representation
- Science Keywords
- GCMD Dif Entry IDs
- Platform/Instrument/Sensor

8.6 DATA MANAGEMENT

The following sections discuss topics relevant to data management within ECHO. This is an integral function that you will perform to control access to your provider's information and catalog items. For more detailed discussions of these topics, refer to Section 6 of the Data Partner User Guide.

8.6.1 Provider Object Access

As an ECHO Data Provider, you will have the ability to control access to information about your provider (e.g. ordering policies & groups). Permissions for these items are assigned to groups containing lists of ECHO registered users. It is important that a data provider exercise appropriate control over the accessibility of their provider's information. In some cases, a provider will only have one group of individuals to whom permission will be granted. However, other providers may wish to separate responsibilities into separate "roles," such as Data Managers and User Services.

8.6.2 Catalog Item Access

As an ECHO Data Provider, you will also have the ability to control access to catalog items (e.g. collections & granules). Permissions for all of these items are assigned to groups containing lists of ECHO registered users. It is important that a data provider exercise appropriate control over the accessibility of their data holdings. In some cases, a provider will not require any restrictions on the visibility or orderability of their data, but there are a variety of reasons why this would not be or stay the case; i.e. hiding old versions of data, data integrity issues, access restricted to certain members of the science community, cost, etc.

8.6.3 Provider Administrators

Your provider will have an "Administrator" group which will have permissions to create, read, update, and delete all information about your provider. This group will also be view all catalog items. Members of this group are referred to as "provider administrators." These individuals will also be able to create additional groups (e.g. data managers & user services) and assign permissions to access provider information. ECHO Operations will coordinate with you regarding an initial set of ECHO registered users who will be added to your "Administrator" group. Subsequent changes can be made by members of that group.

8.6.4 Groups

As has been discussed, groups are a core component of assigning permissions to access provider information and catalog items. ECHO Operations will work with you to determine the right strategy for group creation according to your data management needs.

8.7 ORDERING

The following sections discuss topics relevant to the access of data represented in ECHO. For more detailed discussions of these topics, refer to Section 7 of the Data Partner User Guide.

8.7.1 Ordering or Not

As an ECHO Data Partner you can choose to accept orders brokered through ECHO. ECHO has a published API which, when made available by a Data Partner, is used to transmit ECHO order information. If you choose to support ordering, you will need to implement the ECHO OrderFulfillment service and then register it with ECHO. Whether or not you choose to support ordering, you may supply online access URLs in your metadata, which ECHO and WIST will make available to end users facilitating direct downloads.

8.7.2 Order Options

In order to request additional order information (e.g. ftp push location), you can create ECHO Order Forms based on the ECHO Forms Specification. This specification can be used to specify the input mechanisms for gathering information from users, and how that data will be represented in an XML block sent to your order fulfillment service. WIST currently supports the visualization of this specification. For more information, refer to the ECHO Website (Data Partners > Fulfilling Orders).

8.8 WIST

8.8.1 Button Mapping

All collections available for data discovery through WIST are logically separated by discipline/topic pairings (i.e. Land/ASTER) and then assigned to a radio button appearing on the main search page. When a user chooses the search interface's radio button, all assigned collections will appear in the dataset listing box, as seen in Figure 1.

Values for disciplines should be chosen from: Atmosphere, Cryosphere, Land, Oceans, and Solar/Other. Values for topics are unique labels which facilitate finer-grained categorization of collections. Some example topic values are: ASTER, Field/In Situ and LIGHTNING. Data partners will need to work with ECHO Operations to determine whether their collections fit within existing discipline and topic pairings, or whether new entries and radio buttons will be needed. Collections are internally mapped based upon the following criteria:

- Data Center ID
- Instrument
- Instrument/Platform/Data Center ID
- Instrument/Platform/Science Keyword Topic
- Shortname/Data Center ID

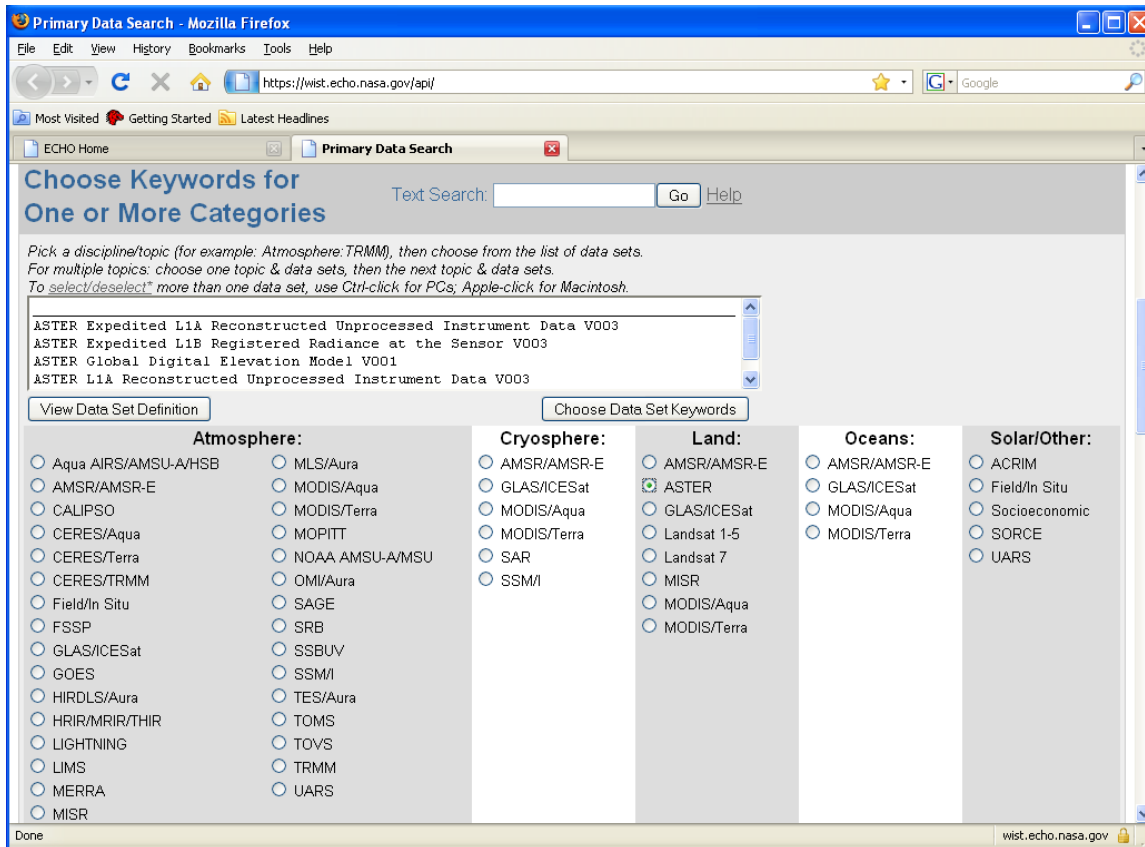


Figure 6 - WIST Main Search Page

8.8.2 Calendar

The WIST calendar allows Data Partners to inform the public of alerts and events for their data center. Each Data Partner will need to identify an individual who will be given the ability to manage access to their WIST calendar. This individual will be able to maintain events and alerts in the WIST calendar, and also grant or revoke permissions for other users to manage the calendar.

Alerts do not have an associated date or time, and are therefore displayed in WIST until they are manually removed. Events do have an associated date and time, and are automatically displayed when relevant and removed when they have expired. Events may be pre-configured prior to the event date in order to simplify calendar management.

All configured alerts and events scroll at the top of all WIST pages below the NASA logo. Figure 2 shows an example of the scrolling alerts displayed at the top of the WIST page.

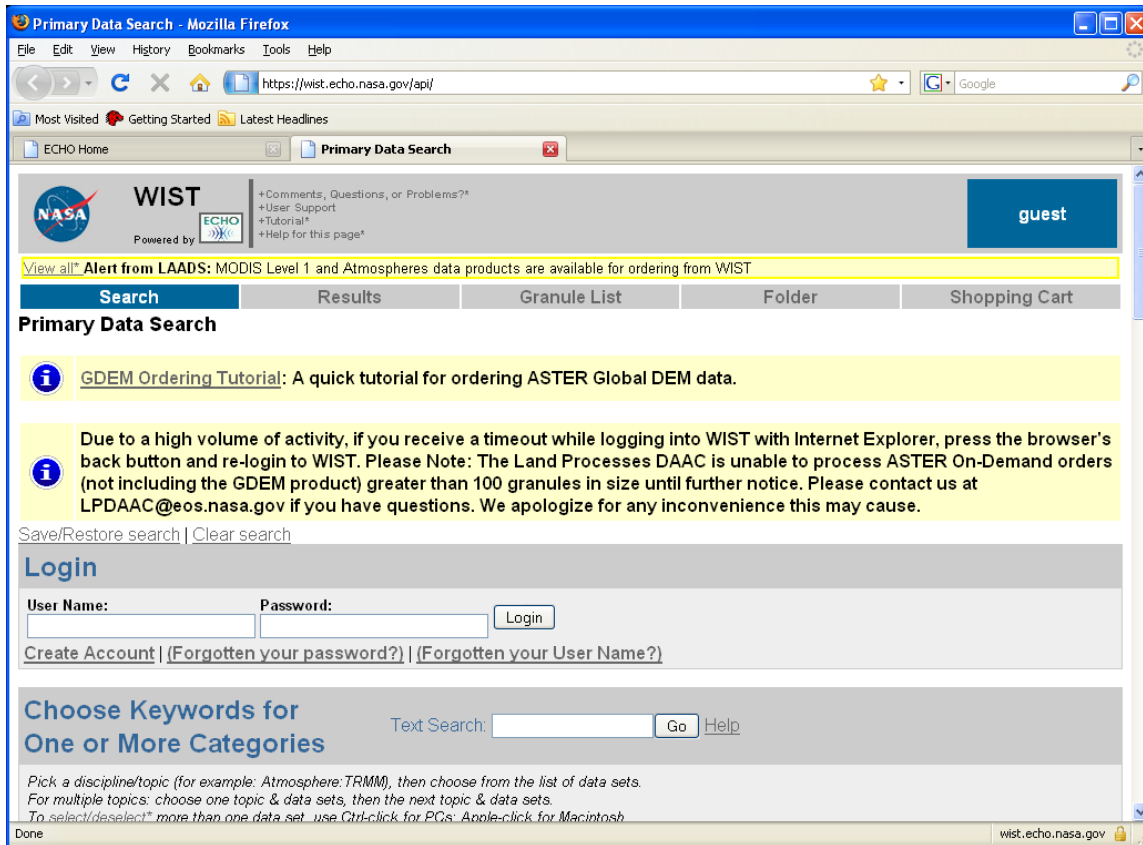


Figure 7 - Scrolling calendar alerts in WIST

Users may select the “View all” link from the scrolling alerts bar to view all alerts and events for all data centers on a separate calendar screen. On this screen, alerts are displayed in yellow and events are displayed in red. Figure 3 shows an example of the separate calendar screen.

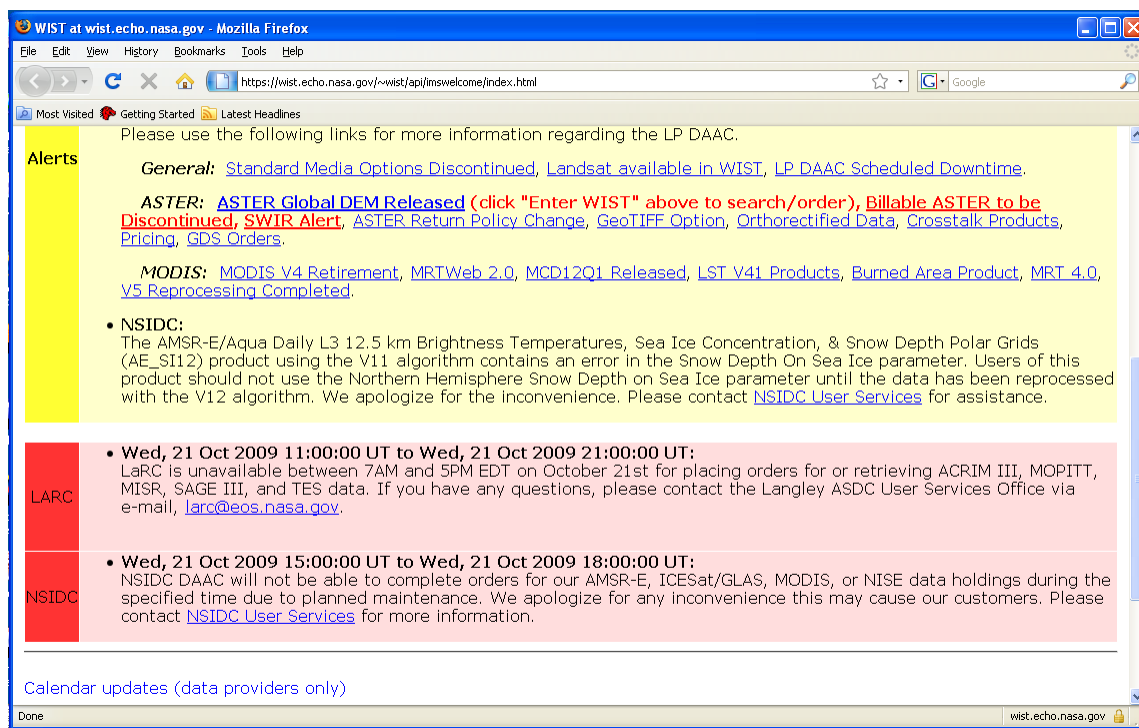


Figure 8 - WIST calendar page

Data Partners may log into the WIST Calendar to make updates by selecting the “Calendar updates (data providers only)” link found at the bottom of the calendar, as seen in Figure 3. After selecting this link, a user will be prompted to enter their configured user name and password. For authentication issues, please contact ECHO Operations. After authenticating, a user will be able to edit events and alerts for all providers to which they have been granted access. Figure 4 shows the calendar editing screen which will be presented. For additional information, please refer to the WIST help page found at <https://wist.echo.nasa.gov/~wist/uptime-cal/help/>.

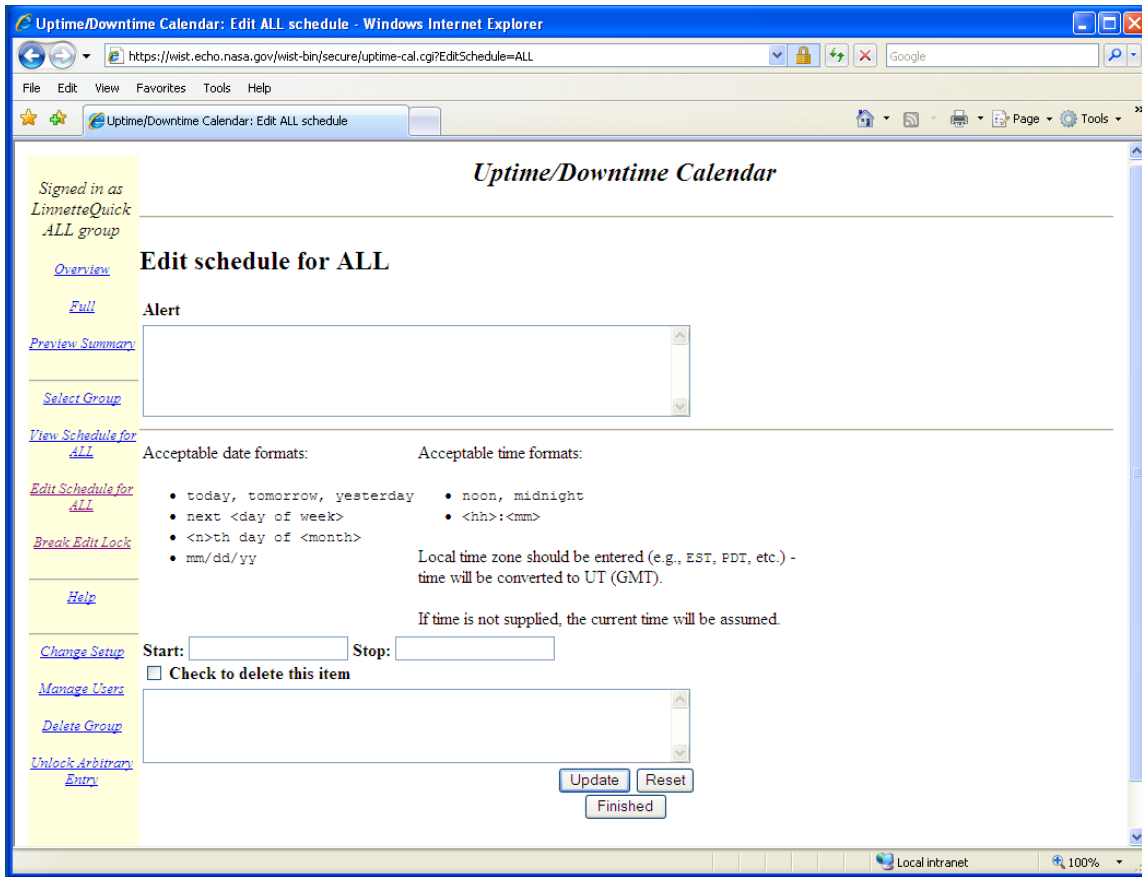


Figure 9 - Sample data center calendar.

8.8.3 Additional Attributes

WIST assigns additional attributes found in each available collection to a category which allows for enhanced end-user data discovery. User's can search for additional attributes by selecting "By Categories/Attributes" in the WIST main search page, as seen in Figure 5.

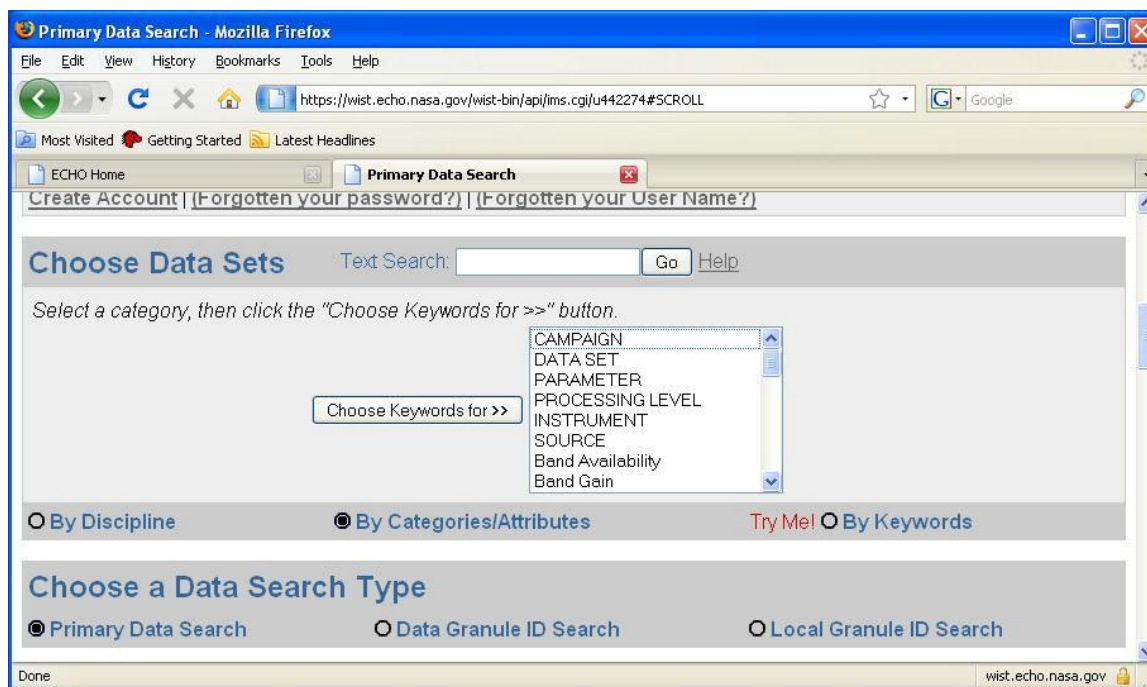


Figure 10 - WIST attribute searching

When first detected by WIST, additional attributes will be assigned to the UNKNOWN category. Data partners should coordinate with ECHO Operations to determine the category for any additional attributes which are currently not included in WIST. ECHO Operations will review new attributes and verify that the type and range validations are the same as the metadata additional attributes. The following categories are currently available in WIST:

- Band Availability
- Band Gain
- Calibration
- Cloud Amount
- Collection Details
- Component
- Content Classification
- Database Identifier
- File Identifier
- Granule Details
- Model
- Observation Mode
- Orbit
- Parameter Identifier
- Parameter Values
- Pointer
- Product generation Algorithm
- Quality
- Resampling Method
- Sensor Characteristics

- Solar Angle
- Solar/Lunar Events
- Spatial
- Storage Medium
- Temporal

APPENDIX A. ACRONYMS USED IN ECHO

Acronyms used in this document are contained in this appendix. A complete list of the acronyms frequently used in discussions of ECHO can be found on the ECHO website (<http://www.echo.nasa.gov>) in the "About ECHO" > "FAQ / Acronyms" section.

ACL	Access Control List
API	Application Programming Interface
AQL	Alternative Query Language
ASF DAAC	Alaska Satellite Facility DAAC
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
BMGT	Bulk Metadata Generation Tool
COTS	Commercial Off The Shelf
DAAC	Distributed Active Archive Center
DB	DataBase
DTD	Document Type Definition
ECHO	EOS Clearinghouse
ECS	EOSDIS Core System
EDC	EROS Data Center
EDG	EOS Data Gateway
EJB	Enterprise JAVA Beans
EMD	EOSDIS Maintenance and Development
EOS	Earth Observing System
EOSDIS	EOS Data and Information System
EROS	Earth Resources Observation Systems
ESDIS	Earth Science Data and Information System
ESIP	Earth Science Information Partner
ETC	ECHO Technical Committee
FTP	File Transfer Protocol
GCMD	Global Change Master Directory
GES DAAC	GSFC Earth Sciences DAAC
GHRC	Global Hydrology Resource Center
GIS	Geographic Information System
GML	Geography Markup Language
GMT	Greenwich Mean Time
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface

Acronyms Used in ECHO

GUID	Globally Unique Identifier
IIMS	Independent Information Management Subsystem
J2EE	Java 2 Enterprise Edition
LAADS	Level 1 and Atmosphere Archive and Distribution System
LP DAAC	Land Processes DAAC
MISR	Multiangle Imaging SpectroRadiometer
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NSIDC DAAC	National Snow and Ice Data Center DAAC
ODL	Object Description Language
OGC	OpenGIS Consortium
ORNL DAAC	Oak Ridge National Laboratory DAAC
PGE	Product Generation Executives
PO.DAAC	Physical Oceanography DAAC
PSA	Product Specific Attribute
PUMP	Provider User Management Program
QA	Quality Assurance
SEDAC	Socioeconomic Data and Applications Center
SOAP	Simple Object Access Protocol
SSC	Stennis Space Center
SSL	Secure Sockets Layer
UDDI	Universal Description, Discovery, and Integration
UI	User Interface
UR	Universal Reference
UR	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Universal Time, Coordinated (also called GMT/UTC)
WIST	Warehouse Inventory Search Tool
WGS	World Geodetic System
WRS	Worldwide Reference System
WSDL	Web Services Description Language
XML	eXtensible Markup Language
XSLT	eXtensible Style Language Transformation

APPENDIX B. FUNCTIONAL BREAKDOWN BY USER/ROLE TYPE

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
AdministratorService				
GetAuditAction			√	
GetAuditReport			√	
GetErrorMessages	√	√	√	√
GetMetricDataPointSample				
InitializeEchoSystemuser				
InitializeErrorCodes				
InitializeTaxonomies				
PurgeArchiveRecords				
SetErrorMessages				
SyncUddiRegistry				
AuthenticationService				
ExpiresSoon	√	√	√	√
GetECHOVersion	√	√	√	√
GetSecurityTokenInformation				
login		√	√	√
logout		√	√	√
RemoveTokens				
RevokeAllTokensForUsers				
ReokeTokens				
CatalogService				
ExecuteQuery	√	√	√	√
GetCatalogItemMetadata	√	√	√	√
GetCatalogItemNamesByDataSetId	√	√	√	√
GetQueryResults	√	√	√	√
ResolveMetadataPaths			√	

Functional Breakdown By User/Role Type

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
DataManagementService				
All			√	
EventNotificationService				
All		√	√	√
ExtendedServicesService				
ActivateWSAdvertisements				
ActivateWSGuis				
ActivateWSImplementations				
ActivateWSInterfaces				
CreateWSAdvertisements				√
CreateWSGuis				√
CreateWSImplementations				√
CreateWSInterfaces				√
GetOperationsByImplementation	√	√	√	√
GetServiceNamesByTaxonomyEntry	√	√	√	√
GetWSAdvertisements	√	√	√	√
GetWSDLForWSInterface	√	√	√	√
GetWSGuis	√	√	√	√
GetWSImplementationNamesByWSInterface	√	√	√	√
GetWSImplementations	√	√	√	√
GetWSInterfaces	√	√	√	√
RemoveWSAdvertisements				√
RemoveWSGuis				√
RemoveWSImplementations				√
RemoveWSInterfaces				
UpdateWSAdvertisements				√
UpdateWSGuis				√
UpdateWSImplementations				√
UpdateWSInterfaces				

Functional Breakdown By User/Role Type

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
GroupManagementService				
CreateGroups			√	
GetGroupNames		√	√	√
GetGroupNamesByManager		√	√	√
GetGroups		√	√	√
NotifyManagers		√	√	√
NotifyMembers		√	√	√
RemoveGroups (only group manager)		√	√	√
UpdateGroups (only group manager)		√	√	√
OrderManagementService				
AddOrderItems	√	√	√	√
CancelOrder	√	√	√	√
CancelProviderOrder	√	√	√	√
CreateAndSubmitOrder	√	√	√	√
CreateOrder	√	√	√	√
GetCatalogItemOrderInformation	√	√	√	√
GetCatalogItemOrderInformation	√	√	√	√
GetOrderGuidsByStateAndOwner	√	√	√	√
GetOrderGuidsByStateDateAndOwner	√	√	√	√
GetOrderItemNamesByOrder	√	√	√	√
GetOrderItemNamesByProviderOrder	√	√	√	√
GetOrderItems	√	√	√	√
GetOrders	√	√	√	√
GetProviderOrderGuidsByStateAndOwner		√	√	√
GetProviderOrderGuidsByStateAndProvider			√	√
GetProviderOrderGuidsByStateDateAndProvider			√	√
QuoteOrder	√	√	√	√
RemoveOrderItems	√	√	√	√
RemoveOrders	√	√	√	√
RemoveProviderOrders	√	√	√	√
SetAuthenticationKey		√	√	√
SetUserInformationForOrder	√	√	√	√

Functional Breakdown By User/Role Type

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
SubmitOrder	√	√	√	√
UpdateOrderItems	√	√	√	√
ValidateOrder	√	√	√	√
OrderProcessingService				
All			√	
ProviderService				
ActivateProvider				
AddAuthenticatorDefinitions			√	
CreateProvider		√	√	√
GetAuthenticatorDefinitions	√	√	√	√
GetProviderNames	√	√	√	√
GetProviderNamesByProviderId	√	√	√	√
GetProviderPolicies			√	√
GetProviders	√	√	√	√
GetProviderSupportedTransactions	√	√	√	√
GetProviderNamesByProviderId	√	√	√	√
RemoveAuthenticatorDefinitions			√	
RemoveProvider				
RemoveProviderPolicies			√	
SetProviderPolicies			√	
SetSslCertProviderPolicyActivated				
UpdateProvider			√	√
SubscriptionService				
All		√	√	√
TaxonomyService				
AddTaxonomyEntry				
CreateTaxonomy				
CreateVirtualTaxonomy				

Functional Breakdown By User/Role Type

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
GetRootPath	√	√	√	√
GetTaxonomies	√	√	√	√
GetTaxonomyEntries	√	√	√	√
GetTaxonomyEntry	√	√	√	√
UserService				
CreateAuthenticator		√	√	√
CreateUser	√			
GetAuthenticators		√	√	√
GetAuthenticatorNames		√	√	√
GetCurrentUser		√	√	√
GetUserNames		√	√	√
GetUserNamesByUserId		√	√	√
GetUserNamesByOrganizationAndFirstAndLastName			√	√
GetUserNamesByRole (provider role)			√	√
GetUserPreferences		√	√	√
GetUsers				√
GrantAccess (provider access)			√	
RecallUserId	√	√	√	√
RemoveAuthenticators		√	√	√
RemoveUserPreferences		√	√	√
RemoveUsers				
ResetUserPassword	√	√	√	√
RevokeAccess (provider access)			√	
SetUserPassword		√	√	√
SetUserPreferences		√	√	√
UpdateUsers		√	√	√

APPENDIX C. ECHO PATH URIS

ECHO Path URI is the name given to a specific URI which maps into the metadata of a granule or collection. ECHO paths are based on XPath, a language for finding information in an XML document and navigating through elements and attributes. Refer to Section 7 for information about using XPath as part of the order fulfillment process.

C.1 FORMAT

The general format for the URI is `<echoltemType>://<echoHost>/<echoltemId>[/xpath]`

- f. **echoltemType** := granule (or collection)—This is the syntax to indicate whether the URI is pointing to a granule or collections metadata.
- g. **echoHost** is the address of the ECHO server the metadata is on. This should be left blank for now, as this feature is not currently supported.
- h. **echoltemId** is the item ID of the of the granule or collection (for example, C14016455-PSATEST).
- i. **xpath** is optional. It is an XPath statement that maps into the XML payload returned from a GetMetadata request using the echoltemId.

Here is an example in the correct format. Note that the echoHost is left blank.

```
collection:///C14016455-
PSATEST/%2Fresults%2Fprovider%2Fresult%2FCollectionMetaData%2FECHOItemid%2Ftext%28%29
```

The XPath above is escaped to be put in the URI. The un-escaped format looks like:
"/results/provider/result/CollectionMetaData/ECHOItemid/text()"

C.2 BEHAVIOR

The granule mapping URI will retrieve data from the XML metadata of a granule or collection. The behavior that appears depending upon the state of the XPath selection appears below.

Table 4. ECHO Path URI Behavior

XPath Selection	Granule	Collection
No XPath included	Returns the entire metadata XML	Returns the entire metadata XML
XPath selects multiple nodes	Returns XML fragment representing multiple nodes	Returns XML fragment representing multiple nodes
XPath selects a node with child nodes and attributes	Returns XML fragment representing node and child nodes	Returns XML fragment representing node and child nodes
XPath selects a node with a single value	Returns single value	Returns single value

APPENDIX D. ECHO ERROR HANDLING

The ECHO 10.0 Web Service API has advanced error-reporting capabilities. There are 12 types of faults reported by ECHO. They are:

- a. **AuthorizationFault** – Reported by ECHO when a user is not authorized to invoke an operation
- b. **DataSizeLimitFault** – Reported by ECHO to indicate that the data size limit has been exceeded
- c. **DuplicateIdFault** – Reported by ECHO to indicate that an entity with the same ID exists in ECHO already
- d. **InternalFault** – Reported by ECHO when an internal error occurs
- e. **InvalidArgumentFault** – Reported to indicate that one or more arguments passed were invalid
- f. **InvalidStateFault** – Reported to indicate that an action by the client would put an object in ECHO in an invalid state
- g. **InvalidURLFault** – Reported to indicate invalid syntax in a URL or an element of the URL that does not exist
- h. **ItemNotFoundFault** – Reported when the client attempts to access one or more objects that do not exist
- i. **ParseFault** – Reported to indicate that some value could not be parsed
- j. **RemovalFault** - Reported to indicate an error that has occurred during the removal of an object from ECHO
- k. **UnsupportedFeatureFault** - Reported to indicated that a feature was selected that is not supported
- l. **ValidationFault** - Reported to indicate that an object in or passed to ECHO is not valid

All the above fault types extend the basic **EchoFault** type. An **ECHOFault** is reported by ECHO when an error occurred during the invocation of an ECHO operation

All faults will include an **ErrorCode**, **SystemMessage**, and **Timestamp** of when the error occurred and an **ErrorInstanceid**. An **EchoFault** may also have an **OpsMessage** (message specified by ECHO Operations).

Error codes are strings that uniquely identify an error case in ECHO. Some error codes are reused, such as when a required parameter to an operation was not provided. ECHO Operations may associate different messages with specific error codes. If ECHO Operations has a message configured for an error code, then that message will be returned with the **EchoFault** in the **OpsMessage** element.

In most instances, receiving a fault from ECHO occurs by catching an **EchoFault** and displaying the Ops Message, System Message, and Error Instance ID to the user.

InternalFaults capture errors that are internal to ECHO, such as when ECHO cannot talk to the metadata catalog or due to an invalid configuration file. Clients receiving an **InternalFault** should report any information provided with the error to ECHO Operations. Depending on the cause of the **InternalFault**, a second request may succeed.

Code Listing 67: Catching Exceptions from ECHO

```
try
{
    // Create authentication service
    AuthenticationServiceLocator authServiceLocator =
        new AuthenticationServiceLocator();
    AuthenticationServicePort authenticationService =
        authServiceLocator.getAuthenticationServicePort();

    ClientInformation clientInfo =
        new ClientInformation();
    clientInfo.setClientId("A Client");
    clientInfo.setUserIpAddress("192.168.1.1");
```

```
    // Call login with jdoe as username, mypass as password,
    // and client information
    authenticationService.login("jdoe", "mypass",
        clientInfo, null, null);
}
catch (EchoFault e)
{
    // This exception was likely caused by user input.
    String message = "Could not login to ECHO:";

    if (e.getOpsMessage() != null
        && e.getOpsMessage().length() > 0)
    {
        message += "\nOps Message : " + e.getOpsMessage();
    }
    message +=
        "\nMessage: " + e.getSystemMessage()
        + "\nError Instance Id: "
        + e.getErrorInstanceId();
    System.out.println(message);
}
catch (RemoteException e)
{
    // ECHO could not be reached.
    System.out
        .println("Could not communicate with ECHO : "
            + e.toString());
}
catch (ServiceException e)
{
    // An error occurred while creating the service.
    System.out.println("Could not create ECHO Service : "
        + e.toString());
}
```

APPENDIX E. INGEST SCHEMAS AND DTDs

The following schemas associated with each system can be downloaded from the ECHO website (www.echo.nasa.gov) in the “Resources” section of each system under the “ECHO Systems” section of the website.

- Collection Metadata
- Granule Metadata
- Browse Metadata

APPENDIX F. BEST PRACTICES

The following tips and other recommended practices are in outline form rather than paragraphs for quick reading.

F.1 TIPS FOR PREPARING YOUR METADATA

- a. File naming conventions can help you track a file in the final Ingest report
 - 1) A metadata file name should be self-describing and unique, including ID of data provider, metadata type, short name, and version_id. The date coverage of metadata and generation date is helpful, for example:
NSCGAMSR200701620070160101.20070116113342.XML
EDCBMOLT200701620070170202.20070117005857.XML
- b. Validate before submission
- c. Perform ingest of new metadata on partner test system
- d. There are separate schemas that govern the XML format for the various metadata types, available from the ECHO website at <http://www.echo.nasa.gov>
Click: Data Partners → Data Development
- e. There are also examples of minimum collection and granule metadata XML requirements located on the ECHO website at <http://www.echo.nasa.gov>
Click: Data Partners → Getting Started

F.2 FACTORS THAT AFFECT INGEST RATES

- a. Metadata type being ingested (collection, granule, browse, update)
- b. Type of action being taken (insert, delete, replacement)
- c. Amount of information in a metadata record (sparse vs. dense)
- d. Number of records in a provider schema
- e. Amount of time required for preprocessing such as adapting to the current format
- f. Amount of competition for database and system resources

F.3 COMMON ERRORS

- a. Use of incorrect data type for element:
 - 1) Use of string for a date or number field:
 - a) <DeleteTime>none</DeleteTime>
 - b) <SwathWidth>2600 km</SwathWidth>
- b. Exceeding the character limitations for an element—refer to entity definitions on ECHO website.
- c. Incomplete dates
- d. Not matching short name and version_id of granule to what is used with its associated collection.
- e. Violating unique constraints within a collection or granule

APPENDIX G. INGEST SUMMARY REPORT

The following is a sample Ingest Summary Report:

```
<?xml version='1.0' encoding='utf-8'?>
<IngestReport dataCenterId="DAAC"
jobId="4E412651-8B3D-92E3-DDC2-AAAAAAAAAAAA"
reportDate="2009-05-10T12:00:24Z">
  <Overview sequenceNumber="2339" jobName="2696"
endDate="2009-05-10T08:00:24.260-04:00"
startDate="2009-05-10T07:41:18.743-04:00">
    <JobErrors />
    <ProcessingTotals>
      <CollectionProcessingTotals processed="1" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0"
verifications="0" inventories="1"/>
      <GranuleProcessingTotals processed="1562" rejected="2"
deleted="30" updated="500" replaced="0" inserted="30"
verifications="1000" inventories="0"/>
      <BrowseProcessingTotals processed="1070" rejected="10"
deleted="0" updated="30" replaced="10" inserted="520"
verifications="500" inventories="0"/>
    </ProcessingTotals>
  </Overview>
  <Details>
    <MetadataFiles>
      <MetadataFile name="DAAC_recon_20091306_1025.001.XML">

        <ProcessingTotals>
          <CollectionProcessingTotals processed="1" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0"
verifications="0" inventories="1"/>
          <GranuleProcessingTotals processed="1500" rejected="0"
deleted="0" updated="500" replaced="0" inserted="0"
verifications="1000" inventories="0"/>
          <BrowseProcessingTotals processed="1000" rejected="0"
deleted="0" updated="0" replaced="0" inserted="500"
verifications="500" inventories="0"/>
        </ProcessingTotals>
        <FileErrors />
      </MetadataFile>

      <MetadataFile name="DAAC_granule_20091306_1025.001.xml">
        <ProcessingTotals>
          <CollectionProcessingTotals processed="0" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0"
verifications="0" inventories="0"/>
          <GranuleProcessingTotals processed="60" rejected="0"
deleted="30" updated="0" replaced="0" inserted="30"
verifications="0" inventories="0"/>
          <BrowseProcessingTotals processed="0" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0"
verifications="0" inventories="0"/>
        </ProcessingTotals>
        <FileErrors />
      </MetadataFile>
    </MetadataFiles>
  </Details>
</IngestReport>
```

```
</MetadataFile>

<MetadataFile name="DAAC_browse_20091306_1025.001.xml">
  <ProcessingTotals>
    <CollectionProcessingTotals processed="0" rejected="0"
      deleted="0" updated="0" replaced="0" inserted="0"
      verifications="0" inventories="0"/>
    <GranuleProcessingTotals processed="0" rejected="2"
      deleted="0" updated="0" replaced="0" inserted="0"
      verifications="0" inventories="0"/>
    <BrowseProcessingTotals processed="70" rejected="10"
      deleted="0" updated="30" replaced="10" inserted="20"
      verifications="0" inventories="0"/>
  </ProcessingTotals>
  <FileErrors />
  <ItemErrorGroups>
    <ItemErrorGroup errorCode="GRANULE_NOT_EXISTS">
      <ItemError itemType="GRANULE" itemId="GRANULE:253135734"
        level="CRITICAL">
        <Message>Validation error, GRANULE:253135734. Granule
          does not exist.</Message>
      </ItemError>
      <ItemError itemType="GRANULE" itemId="GRANULE:253137497"
        level="CRITICAL">
        <Message>Validation error, GRANULE:253137497. Granule
          does not exist.</Message>
      </ItemError>
    </ItemErrorGroup>
  </ItemErrorGroups>
</MetadataFile>
<MetadataFile name="DAAC_granule_20091306_1025.002.xml">
  <ProcessingTotals>
    <CollectionProcessingTotals processed="0" rejected="0"
      deleted="0" updated="0" replaced="0" inserted="0"
      verifications="0" inventories="0"/>
    <GranuleProcessingTotals processed="0" rejected="0"
      deleted="0" updated="0" replaced="0" inserted="0"
      verifications="0" inventories="0"/>
    <BrowseProcessingTotals processed="0" rejected="0"
      deleted="0" updated="0" replaced="0" inserted="0"
      verifications="0" inventories="0"/>
  </ProcessingTotals>
  <FileErrors>
    <Error errorCode="FILE_TYPE_INDETERMINABLE"
      level="CRITICAL">
      <Message>Unable to determine file type of a processing
        file for metadata file granule_2009-05-10_1.xml</Message>
    </Error>
  </FileErrors>
</MetadataFile>
</MetadataFiles>
</Details>
</IngestReport>
```

APPENDIX H. INDEX

Browse

- delete example, 114
- DTD, 61, 110
- insert example, 112
- update example, 112

Catalog

- GetCatalogOrderInformation, 18
- introduction to the term, 15
- items, 18

Collections

- delete items, 71, 105

DTD

- browse, 61, 110
- for collection metadata, 105
- location for all ECHO DTDs, 14
- metadata updated DTD*, 71, 101

ECHO

- concept, 13
- design, 14
- entities, 17
- features, 14
- graphical representation of concept, 14

Error handling

- catching exceptions, 116
- types of errors, 116

Error messages

- example of XML validation error, 75
- invalid XML file error, 75

Exception catching, 116

FTP, 14

- ingest*, 14

GCMD, 58

Granules

- delete items, 71, 105

GUID

- defined, 17

Ingest

- FTP location, 62
- metadata mapping, 66
- overview, 62

Metadata mapping

- ingest*, 66

Order options

- defined, 20