EOS ClearingHOuse (ECHO) Client Training Class

Keith Wichmann
ECHO System Engineer/Architect
wichmann@gst.com
(240) 542-1117

Robin Pfister
ESDIS
Robin.Pfister@gsfc.nasa.gov
(301) 614-5171

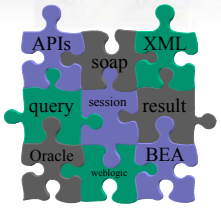
Global Science & Technology, Inc.

Agenda

Wednesday, November 5th, 2003 – Client Class				
11/5/2003	8:30 AM	0:30	9:00 AM	Coffee
11/5/2003	9:00 AM	0:30	9:30 AM	Overview of Client Interaction
11/5/2003	10:00 AM	0:30	10:30 AM	Java Client Toolkit
11/5/2003	10:30 AM	0:15	10:45 AM	Break
11/5/2003	10:45 AM	0:30	11:15 AM	Guest and Registered Users
11/5/2003	11:15 AM	0:45	12:00 PM	Catalog Service
11/5/2003	12:00 PM	1:30	1:30 PM	Lunch
11/5/2003	1:30 PM	1:00	2:30 PM	AQL-Query Language
11/5/2003	2:30 PM	0:15	2:45 PM	Provider Profile Service
11/5/2003	2:45 PM	0:15	3:00 PM	Break
11/5/2003	3:00 PM	0:45	3:45 PM	Orders
11/5/2003	3:45 PM	0:15	4:00 PM	User Preferences
11/5/2003	4:00 PM	0:30	4:30 PM	Subscription Service
11/5/2003	4:30 PM	0:30	5:00 PM	Ops Team Support

ECHO Overview

- What is ECHO?
- Goal of ECHO
- Goal of Workshop
- History of ECHO
- Guidance/Advisory Groups
- Architecture Overview
- Current State of ECHO



What is ECHO?

- Clearinghouse of spatio-temporal metadata.
- Order Broker
- User and provider account service
- Services clearinghouse and broker (future):
  - Data Services (e.g. Subsetting)
  - Search Services (e.g. Coincidence Searching)
  - Administrative Services (e.g. B&A)

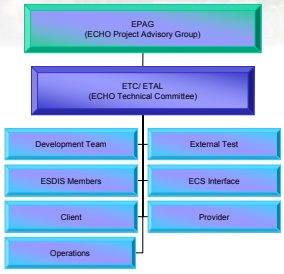
Goal of ECHO

- Respond to user needs for more flexible interfaces for access to EOS data.
- Serve as a portal to Earth Science metadata.
- Allow providers of data to share their metadata and offload some of their search responsibilities.
- Broker orders from clients to the appropriate providers, providing tracking services for both the client and the provider.
- Present a messaging interface based on XML.
- NOT a GUI.

Goal of Training Workshop

- Train Client Developers on the use of the ECHO Client APIs.
- All-way, informal interaction among participants. We need your feedback.
- Familiarize Client Developers with how to use the materials: web site and users manual.
- Collect any requirements we may have missed.
- Get feedback on the processes for client developer participation in ECHO.

ECHO Committees



Guidance/Advisory Groups

- **ECHO Technical Committee:**
  - Originated to study IMS alternatives, independent of the ECS architecture.
  - Representatives from various contractors, ESDIS, and ECHO providers.
  - Continues to provide technical input into the direction of ECHO on a monthly basis.
- **Advisory Panel: made up of ESDIS and non-ESDIS representatives.**

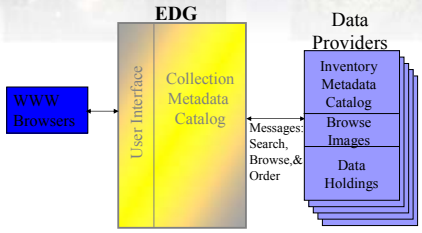
Architecture Overview:

System Drivers

- **Present an API for organizations to connect their own user interfaces and programs to**
- **Make it easy for providers of Earth Science data and services to participate in the system**
- **Provide searches that respond quickly**
- **Broker orders for both data and services**
- **Minimize operational costs**
- **Build upon advances in industry and use e-commerce systems as a model**
- **Build a system that can be scaled up to handle large numbers of requests**

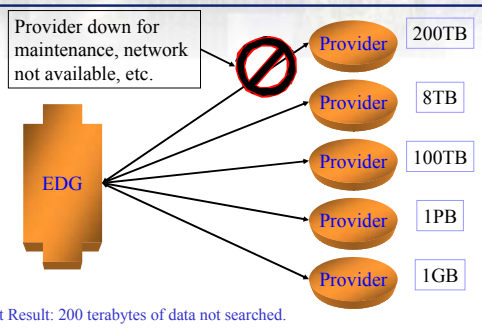
Architecture Overview:

Current Information Architecture



Architecture Overview:

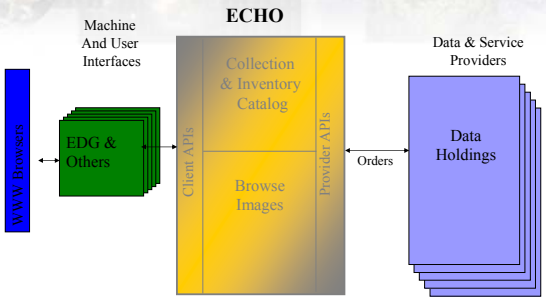
Current Metadata Access



Net Result: 200 terabytes of data not searched.
Search time dependent on lowest common denominator.

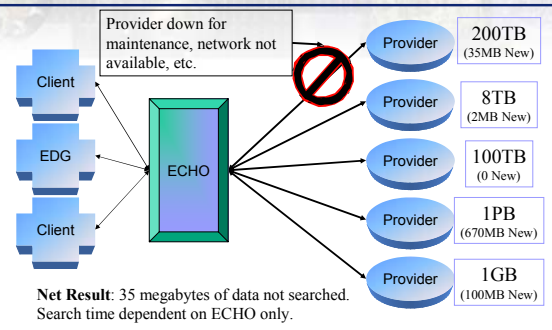
Architecture Overview:

Information Architecture w/ECHO



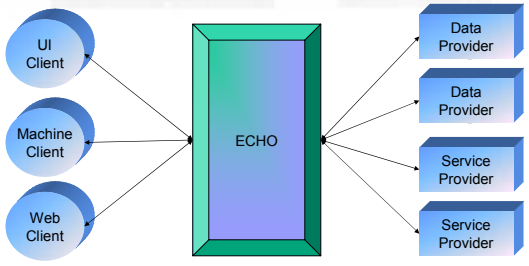
Architecture Overview:

ECHO Metadata Access

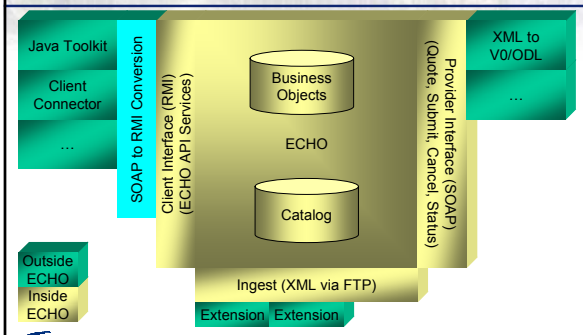


Net Result: 35 megabytes of data not searched.
Search time dependent on ECHO only.

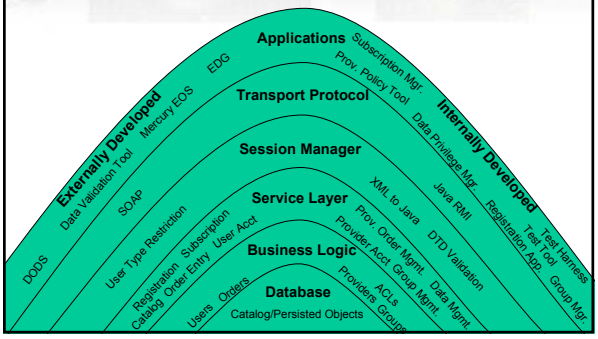
High Level ECHO Context



ECHO Interfaces



ECHO's Layered API Architecture



Current State of ECHO

- **Second Operational Version (Release 2- Version 5.0.1):**
  - Functional but not complete or necessarily completely correct.
    - Vetted through previous workshops and previous year of operations
    - Clients are under development and provided feedback
  - Changes are coming
    - Data Model Review changes will cause some basic changes in the conceptual model of the system
    - XML has been leveraged to try to minimize impact of changes
    - Some structuring of the messages and tag names will be impacted
    - New functions will be added
    - Some parts of the model that are not being used will be removed
- **Clients under development**
  - Mercury-EOS, EDG, DVUI, Gizmo, more
- **Your feedback is desired, welcome and needed!**
  - What will make your client work better?
  - Are there missing requirements?

Vocabulary

- **ECHO - EOS Clearing HOuse**
- **EOS - Earth Observing System**
- **EJB - Enterprise Java Bean**
- **J2EE - Java 2 Enterprise Edition**
- **SOAP - Simple Object Access Protocol**
- **DTD - Data Type Definition**
- **XML - eXtensible Markup Language**

Introduction to the Client API

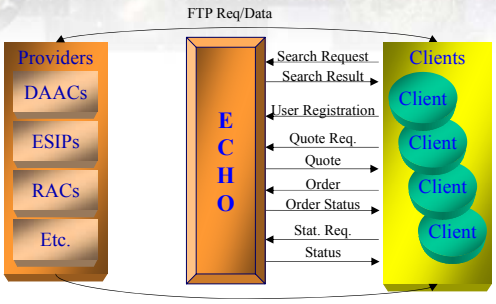
ECHO provides an interface to the outside world through XML messages carried over various protocols.

Global Science & Technology, Inc.

What is a Client Developer?

- **An ECHO Client Developer uses ECHO APIs to provide services to their client community. The client interface can make of all or only some part of the services that ECHO offers. It is intended for the Client Developer to focus on providing the best interface possible for their target audience and let ECHO focus on keeping the Earth Science metadata up to date and dealing with the differences among Earth Science Data Providers.**
- **The term Client Provider is used to refer to operational systems that use ECHO services**

Client Context



Types of Client Providers

- **GUIs**
  - These clients are geared to interface directly with a user
  - They may use ECHO account services, catalog services, order entry services as either guest or as a registered user (on behalf of their user)
    - They may use only a subset of the above services
- **Machine-to-Machine**
  - These clients do not directly involve a person
  - They will typically come from other existing system using ECHO as an archive, or from specialized GUIs that need some static information updated on some schedule
  - All ECHO requests can be performed in this way

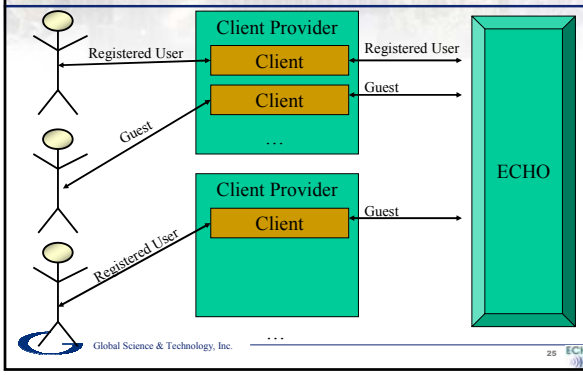
Client Provider Connections

- **A Client Provider is running a program X on some machine Y that is providing services to a given user Z.**
- **Program X can connect to ECHO in two ways currently:**
  - SOAP: XML sent over an HTTP connection
  - RMI: XML sent over an RMI connection
- **Program X can represent the user Z in two ways:**
  - Guest user: the user will have the capability to search public metadata and retrieve, but other transactions are limited
  - Registered user: the user is a known user to ECHO, has an account, and is logged in for using the system

Concept of Logging In

- **Most ECHO transactions are available to guests (users who have not logged in), but registered users (who have logged in) have some additional ones they can use**
- **It is presumed that ECHO Clients will have a user that they are representing when they interact with ECHO. That user is who the Client should log in as, not one representing the Client Provider.**
  - Existing systems may have their own set of users which is independent of ECHO
  - New systems may decide to rely on ECHO for user account management

Users and ECHO



Seven Basic ECHO Functions

- **When a client interacts with ECHO, there are really only 7 basic functions**
  - **Login**: Login to ECHO as a registered user by providing userid and password
    - If this is not done, then the user is a guest
    - Login information if maintained for the length of the session (timeout or logout, by IP)
  - **Logout**: End an ECHO session
  - **Perform**: Send an XML message, get one back
  - **Identify**: Provide a client identification string
  - **SetProviderContext:** Let ECHO know what Provider you are representing
  - **SetUserContext:** Let ECHO know what user an admin is acting as
  - **Remove**: End the session connection with ECHO (or set it to NULL)
- **Both SOAP and RMI versions are available, others are feasible**
- **Each client will be assigned a string for it to use as its identity when communicating with ECHO**
  - This is useful for debugging issues, and for tracking how often a client is used
  - The identity of the client will be logged and passed along with orders

The Session Concept

- **As soon as a client grabs a reference to the ECHO system, a session is begun**
- **A session has a user, which may be a guest, and a client identification string**
- **State is maintained until the session ends**
  - A session ends by calling the remove command or by timing out
- **The user represented in the session is controlled by login and logout**
  - Until a login command is executed, the user is considered a guest user
  - After a logout command is executed, the user is considered a guest user

ECHO Request/Response Paradigm

- **ECHO's Perform command allows a client to send a string which must be an XML message to the system**
  - This is referred to as the request message
  - The ECHO naming scheme appends "Request" to the message name
  - E.g. "QueryRequest"
- **ECHO's Perform command returns a string which will be an XML message from the system**
  - This is referred to as the response message
  - The ECHO naming scheme appends "Response" to the message name
  - E.g. "QueryResponse"

Understanding XML DTD References

- **All Perform transactions take an XML message as input**
- **There is a set of DTDs that describe the different messages**
- **The DTDs are available at** http://api.echo.eos.nasa.gov/echo/dtd/
- **When sending a message, it is important that internal DTD references are consistent**

The ECHO URLs

- www.echo.eos.nasa.gov **- Also,** http://eos.nasa.gov/echo
  - Official web pages describing the system
  - Will be available November 1, 2002
- http://api.echo.eos.nasa.gov
  - The place where the XML messages that constitute the API are sent
- **Browse.echo.eos.nasa.gov**
  - Browse data hosted at ECHO will have this as the domain name of the URL
- **Unofficial web sites with useful information**
  - Dangermouse.gst.com/dmr
  - Canyon.gst.com/public

Client Connectors

- **ECHO's preferred method of communication is through SOAP**
- **Since SOAP is a relatively new standard, ECHO has provided some code to facilitate sending SOAP messages directly from 2 different client languages more easily**
  - Perl, Python
- **They offer direct access to a way to call the seven different basic ECHO commands: login, logout and perform, set user context, set provider context, identify and remove**
- **Currently available at http://www.echo.eos.nasa.gov/clientpartner-resources.shtml**

Client API Future Items

- **ECHO Services as Web Services**
  - As ECHO starts to represent Earth Science Data Services, we intend to register ECHO services in the services directory as well
  - ECHO will provide WSDL descriptions of its services and provide compliant SOAP access
    - We are considering a finer grain object view with each ECHO service being its own service, rather than the course grained 4 method service currently available

The Java Client Toolkit (aka Façade)

An easy mechanism for Java developers to communicate with ECHO.

Global Science & Technology, Inc.

Façade Motivation

- **In developing management clients to ECHO, the development team found out a few things:**
  - A large amount of code was shared among the different clients
  - The creation and parsing of XML makes for less readable code in your development language
  - "I'm coding my client in Java. ECHO is in Java. Isn't there an easier way for me to access ECHO without going through this XML middle layer?"
- **ECHO originally chose XML as its message interface format because it enables heterogeneous clients**
- **SOAP has provided a new way to leverage XML but still have your client interface in your own language**
  - ECHO has a SOAP interface that mimics the way the system was originally designed, but will be moving to a web service approach
  - In the meantime, the Java Client Toolkit (aka Façade) can be used by Java developers to communicate with ECHO without using XML (except in queries and presents)

Pre-Façade Code Example

```
Document requestDoc = new DocumentImpl();
Element service =
    requestDoc.createElement("UserAccountService");
requestDoc.appendChild(service);
Element transaction =
    requestDoc.createElement("AddAddressRequest");
service.appendChild(transaction);
Element addressInformationElement =
    requestDoc.createElement("AddressInformation");
transaction.appendChild(addressInformationElement);
Element idElement = requestDoc.createElement("AddressId");
addressInformationElement.appendChild(idElement);
Element theIdElement =
    requestDoc.createElement("AddressId");
theIdElement.appendChild(requestDoc.createTextNode(addressI
    d));
IdElement.appendChild(theIdElement);
Element usFormatElement =
    requestDoc.createElement("USFormat");
usFormatElement.appendChild(requestDoc.createTextNode(usF
    ormat));
addressInformationElement.appendChild(usFormatElement);
// ... and so on for Street1-5, City, State, and Zip Code
```

```
// submit the message and get the response
OutputFormat format = new OutputFormat(requestDoc);
String dtdAddress =
http://api.echo.eos.nasa.gov/dtd/UserAccountService.dtd;
format.setDoctype(null, dtdAddress);
StringWriter stringWriter = new StringWriter(); XMLSerializer
    serializer = new XMLSerializer(stringWriter, format);
serializer.asDOMSerializer();
serializer.serialize(requestDoc.getDocumentElement());
String requestMessage = stringWriter.toString();
String responseMessage = null;
try {
responseMessage = echoToken.perform(requestMessage);
} catch (XMLServiceException e) {
// do something intelligent
}
// now parse the response message
```


Façade Code Example

```
AddressInformationDO address = null;
try {
address = new AddressInformationDO(new AddressID("home"), true, street1,
  street2, street3, street4, street5, city, state, zip, country);
} catch (ValidationException e) {
// some validation can be performed client side within remote invocation
}
try {
UserAccountService.AddAddress(address, echoToken);
} catch (XMLServiceException e) {
// some validation can only be performed on the server, this is where those
  errors are reported
}
```

Service Level Common Functions

static java.lang.String[ ][ ] getParameterDocs(java.lang.reflect.Method method)	Returns an array of String[] that represent the descriptions of the formal parameters, in declaration order, of the method represented by the Method object.
static java.lang.String[ ] getParameterNames(**java.lang.reflect.Method method**)	Returns an array of String that represents the names of the formal parameters, in declaration order, for the Method object provided.
static java.lang.Class[ ] getRequiredParameterTypes(**java.lang.reflect.Method method**)	Returns an array of Class objects that represent the formal required parameter types, in declaration order, of the method represented by the Method object.
static java.lang.String[ ] getServiceDocs()	Returns an array of String that represents the description of service.
static java.lang.String[ ] getTransactionDocs(**java.lang.reflect.Method method**)	Returns an array of String that represents the description of the method provided.

Façade Primary Classes

- **One primary class for each service in the system**
  - AdministrationService
  - CatalogService
  - DataManagementService
  - GroupManagementService
  - OrderEntryService
  - ProviderAccountService
  - ProviderOrderManagementService
  - ProviderProfileService
  - RegistrationService
  - SubscriptionService
  - UserAccountService
- **The state of the session is maintained through ECHOToken**

What is the ECHO Token?

- **Clients use ECHO's Session Manager to establish the session state**
  - Associate a user
  - Identify the client
  - Define provider that the user is representing
  - Define user that an admin is representing
- **Since SOAP is stateless, the ECHO Token is used to embody the state, and is passed in with each request**
- **Instead of calling the Session Manager directly in the Façade, you create an ECHO Token**

ECHO Token Methods

- boolean authenticationAttempted() - Returns true if a user authentication was attempted
- void connectTo(String server) - Connects to the server specified.
- String getProviderContext() - Returns the provider the current user is acting on behalf of, or null if the user is a guest.
- String getTargetUrl() - Returns the target URL.
- String getUserContext() - Returns the user that the current user is acting on behalf of, or the current username of the user, or null if the user is a guest.
- String getUserName() - Returns the user name of the current user, or null if no user is currently authenticated.
- void identify(String identity) - Sets the identity for the session.
- boolean isAuthenticated() - Returns true if the user is authenticated
- boolean isConnected() - Returns true if the token is connected to Echo
- void login(String userName, String password) - Authenticates the user specified using the password provided.
- void logout() - Logs the current user out and resets the connection to be that of a guest.
- String perform(String request) - Submits the message specified and returns the response.
- boolean setProviderContext(String providerId) - Sets the provider that the current user is acting on behalf of.
- void setTargetUrl(String server) - Sets the target URL.
- boolean setUserContext(String userName) - Sets the user that the current user is acting on behalf of.
- String version() - Returns the version of the EchoToken.

Client Toolkit Link

- **The Client Toolkit (Façade) can be found linked on the main ECHO web page:**
  - http://www.echo.eos.nasa.gov/clientpartner-resources.shtml
- **Javadocs:** http://www.echo.eos.nasa.gov/echo-toolkit/i_4_8_0_b6/docs/api/
- **Client Toolkit Library:** http://www.echo.eos.nasa.gov/echo-toolkit/i_4_8_0_b6/lib/client_toolkit.jar
- **Source code:** http://www.echo.eos.nasa.gov/echo-toolkit/i_4_8_0_b6/src/
- **These links will be updated to 5.0.1 with the transition**

Break

Global Science & Technology, Inc.

Registration Service Transactions for Clients

The Registration Service allows a guest to create a new registered user.

User Registration

- **CreateUserRequest**
  - Creates a new registered user within the system
  - UserName, password, first name, last name, email address are all required
  - Address and Phone are optional

Sample User Registration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE RegistrationService SYSTEM 'http://api.echo.eos.nasa.gov/echo/dtd/RegistrationService.dtd'>
<!-- Create a new user -->
<RegistrationService>
        <CreateUserRequest>
                <UserName>kwichma</UserName>
                <Password>password</Password>
                <UserInformation>
                        <FirstName>Keith</FirstName>
                        <LastName>Wichmann</LastName>
                        <EmailAddress>wichmann@gst.com</EmailAddress>
                </UserInformation>
        </CreateUserRequest>
</RegistrationService>
```

Note: The XML version and DOCTYPE tags are vital. They will not be illustrated any more in this class.

User Account Service

The User Account Service allows a registered user to manage their account information including address, phone numbers, email, user preferences. This service also provides access to submitted orders.

User Address Related Transactions

- **AddAddressInformation**
- **UpdateAddressInformation**
- **PresentAddressInformation**
- **DeleteAddressInformation**

Describing Addresses in ECHO

- **A Registered User can maintain any number of addresses**
  - Each address will have a name that references the address
  - The client can present the list of names and allow the user to pick which address they want to use
- **Address Information specifics**
  - The name of the address (AddressID) is unique for the user the address belongs to.
  - There are 5 address lines.
  - The address can be in either the standard US format or in a somewhat free format for international addresses.
  - The US format requires at least one street address to be filled as well as the city, state, zip code, and country fields.
  - The non-US format only requires one street address and the country field.
  - The country field must follow the convention of the ISO 3-letter country code.

AddAddressInformation

- **This transaction simply adds a new address to the registered user's profile**
- **Caveats**
  - Adding an existing address is not allowed. If this is attempted, then the following error message will be returned: "The <address-name> address already exists." where <address-name> is the name of the address that already exists. If the intention of the user was to update an existing address, then the UpdateAddressInformation transaction should be used instead.
  - All required address fields in an address should be filled. If one of the required fields is not completed, then the following error message will be returned: "The <field-name> address field is missing" where the <field-name> is the name of the field that is empty.

UpdateAddressInformation

- **This transaction enables a registered user to update one or many named addresses to the user's persisted address book.**
  - If the user updates multiple addresses and one of the address additions fails, then the whole transaction fails.
- **Caveats**
  - Updating a new address is not allowed. If this is attempted, then the following error message will be returned: "The <address-name> address does not exist." where <address-name> is the name of the address that does not exist. If the intention of the user was to add a new address, then then AddAddressInformation transaction should be used instead.
  - All required address fields in an address should be filled. If one of the required fields is not completed, then the following error message will be returned: "The <field-name> address field is missing" where the <field-name> is the name of the field that is empty.

PresentAddressInformation

- **This transaction enables a registered user to view one or many persisted addresses.**
  - If the user requests to present multiple addresses and one of the address presentation transactions fails, then the whole transaction fails.
  - The user has the choice of either specifying or not specifying the address names in the request.
    - If an address name is specified, the address information of the specified address is returned.
    - If no address name is specified, then all the persisted addresses are returned.
- **Caveats**
  - Presenting an non-existing address is not allowed. If this is attempted, then the following error message will be returned: "The <address-name> address does not exist." where <address-name> is the name of the address that does not exist.

DeleteAddressInformation

This transaction enables a registered user to delete one or many persisted addresses.

- If the user deletes multiple addresses and one of the address deletions fails, then the whole transaction fails.

Caveats:

- **Deleting an non-existing address is not allowed. If this is attempted, then the following error message will be returned: "*The <address-name> address does not exist*." where <address-name> is the name of the address that does not exist.**

User Phone Related Transactions

- **AddPhoneInformation**
- **UpdatePhoneInformation**
- **PresentPhoneInformation**
- **DeletePhoneInformation**

AddPhoneInformation

- **This transaction enables a registered user to add one or many phone numbers to the user's phone book.**
  - If the user adds multiple phone numbers and one of the them fails, then the whole transaction fails.
- **Caveats:**
  - Adding an existing phone number is not allowed. If this is attempted, then the following error message will be returned: "The <phone-name> phone already exists." where <phone-name> is the name of the phone number that already exists. If the intention of the user was to update an existing phone, then the UpdatePhoneInformation transaction should be used instead.
  - All phone fields in a phone should be filled out. If one of the fields is not completed, then the following error message will be returned: "The <field-name> phone field is missing" where the <field-name> is the name of the field that is empty.

UpdatePhoneInformation

- **This transaction enables a registered user to update one or many named phones to the user's phone book.**
  - If the user updates multiple phones and one of the phone additions fails, then the whole transaction fails.
- **Caveats:**
  - Updating a new phone is not allowed. If this attempted, then the following error message will be returned: *"The <phone-name> phone does not exist."* where <phone-name> is the name of the phone that does not exist. If the intention of the user was to add a new phone, then then AddPhoneInformation transaction should be used instead.
  - All phone fields in an phone should be filled. If one of the fields is not completed, then the following error message will be returned: *"The <field-name> phone field is missing"* where the <field-name> is the name of the field that is empty.

PresentPhoneInformation

- **This transaction enables a registered user to view one or many phone information records in user's phone book.**
  - If the user requests to present multiple phones and one of the phone presentation transactions fails, then the whole transaction fails.
  - The user has the choice of either specifying or not specifying the phone names in the request.
    - If an phone name is specified, the phone information of the specified phone is returned.
    - If no phone name is specified, then all the persisted phones are returned.
- **Caveats:**
  - Presenting a non-existing phone is not allowed. If this is attempted, then the following error message will be returned: "The <phone-name> phone does not exist." where <phone-name> is the name of the phone that does not exist.

DeletePhoneInformation

- **This transaction enables a registered user to delete one or many persisted phones.**
  - If the user deletes multiple phones and one of the phone deletions fails, then the whole transaction fails.
- **Caveats:**
  - Deleting a non-existing phone is not allowed. If this is attempted, then the following error message will be returned: "The <phone-name> phone does not exist." where <phone-name> is the name of the phone that does not exist.

User Information Transactions

- **PresentUserInformation**
  - This transaction enables a registered user to view a user's information including first name, last name, email address, opt-in, and/or organization name.
    - The opt-in field is not currently used, but is a way for a user to say that they are willing to have their personal information shared within the ECHO community. If this field is not set, then the provider should only use the user's information for actual order handling.
- **UpdateUserInformation**
  - This transaction enables a registered user to update his information including first name, last name, email address, opt-in, and/or organization name.
- **ChangeUserPassword**
  - This transaction enables a registered user to change his password.

Order Related Transactions

- **PresentOrderHistory**
- **PresentOrderHistorySummary**
- **PresentPendingOrder**
- **PresentPendingOrderSummary**
- **CancelOrder**

PresentOrderHistory

- **This transaction enables a registered user to view detailed information about the user's orders that have been fulfilled, or are in terminating order states.**
- **The user has the choice of either specifying or not specifying the terminating order states in the request.**
  - If order states are specified, the order information of the orders in the specified order states are returned.
  - If no order state is specified, then all the persisted orders that are in terminating order states are returned.

PresentOrderHistorySummary

- **This transaction enables a registered user to view a summary of the user's orders that have been fulfilled, or are in terminating order states.**
- **The order summary that is returned includes order ID and order state.**
- **The user has the choice of either specifying or not specifying the terminating order states in the request.**
  - If order states are specified, the order information of the orders in the specified order states are returned.
  - If no order state is specified, then all the persisted orders that are in terminating order states are returned.

PresentPendingOrders

- **This transaction enables a registered user to view detailed information about the user's orders that have not been fulfilled, or are in pending order states.**
- **The user has the choice of either specifying or not specifying the pending order states in the request.**
  - **If order states are specified, the order information of the orders in the specified order states are returned.**
  - **If no order state is specified, then all the persisted orders that are in pending order states are returned.**

PresentPendingOrderSummary

- **This transaction enables a registered user to view a summary of the user's orders that have not been fulfilled, or are in pending order states.**
- **The order summary that is returned includes order ID and order state.**
- **The user has the choice of either specifying or not specifying the pending order states in the request.**
  - If order states are specified, the order information for the orders in the specified order states are returned.
  - If no order state is specified, then all the persisted orders that are in pending order states are returned.

CancelOrder

- **A registered user may use this message to request cancellation of an order or a specific provider order that has been submitted and has not yet been fulfilled.**
  - There may be a cost associated with cancelling the order.
  - Either a whole order can be specified (by the OrderID) or a specific provider order can be specified (by the ProviderOrderID).
  - Both should not be specified, but if for some reason both are specified, then only the ProviderOrderID will be used, thus only that specific provider order will be cancelled.

Lunch

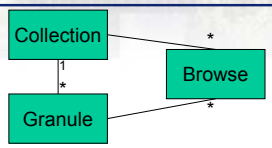
Global Science & Technology, Inc.

Catalog Service

Guests and registered users can use this service to perform searches on the Provider metadata hosted in the clearinghouse, and to retrieve those results. This is one of the services of primary interest to most clients.

Global Science & Technology, Inc.

Simplified ECHO Metadata Model



- **At the highest level, there are three main entities that are part of the ECHO conceptual data model**
  - Collection – A grouping of granules typically based on a common source of the granules
  - Granule – The lowest level item retrievable from a provider that is uniquely described in ECHO
  - Browse – Some kind of binary or ASCII file used to provide a user with a quick view of the data. This could be a scaled down version of the imagery, or a histogram of the data, or some other representation

Catalog Service Features

- **Searching and retrieving science metadata stored by ECHO is done by the Query and Present transactions of the Catalog Service**
- **In addition, ECHO supports**
  - storage of user queries
  - retrieval of previously stored user queries
  - execution of stored user queries
  - storage of the results of a query
  - present previously saved results

Query

- **Query ECHO for metadata constrained by user specified search parameters**
- **The QueryRequest message can contain**
  - Query in IIMSAQL language enclosed in CDATA section or
  - Name of a previously saved query (only for registered users)
- **Stateful vs. Stateless Query Execution**
  - Stateful: ECHO maintains a result set maintained for the user
    - The ID for the result is returned to the client
  - Stateless: ECHO returns a list of IDs to the client
    - A maximum of N items can be returned in one result
- **Users can execute multiple queries simultaneously (using multiple sessions)**
  - These results will be available at least until the session times out or the registered user logs out
  - Registered users can save these results across sessions by explicitly saving the results using the SaveResultSet transaction
  - Guest's result sets will only last for the session
- **ECHO supports storage and retrieval of previously stored queries by registered users**
  - A user may execute a previously stored query to get more recent data
  - User queries cannot be viewed, executed or removed by another user

QueryRequest message

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CatalogService SYSTEM "http://api.echo.eos.nasa.gov/echo/dtd/CatalogService.dtd">
<CatalogService>
  <QueryRequest>
    <QueryExpression>
      <query>
<![CDATA[
  User query goes here. >
]]>
      </query>
      <namespace>none</namespace>
      <QueryLanguage>
        <IIMSAQL/>
      </QueryLanguage>
    </QueryExpression>
    <ResultType>
      <RESULTS/>
    </ResultType>
      <IteratorSize>10</IteratorSize>
      <Cursor>1</Cursor>
      <PresentationDescription>
        <TupleType>
          <attributeName>GranuleUR</attributeName>
          <PrimitiveTypeName><String/></PrimitiveTypeName>
        </TupleType>
        <TupleType>
          <attributeName>DataSetId</attributeName>
          <PrimitiveTypeName><String/></PrimitiveTypeName>
        </TupleType>
        ..... more TupleType specification ....
        <DTDType>
          <ECHO/>
        </DTDType>
      </PresentationDescription>
  </QueryRequest>
</CatalogService>
```

this query must be in this Query language

Can also take values RESULT_SET_ID, ITEM_IDS and HITS

QueryResponse Message

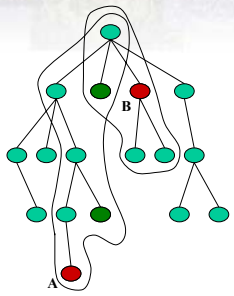
```
<CatalogService>
  <QueryResponse>
        <BooleanResult>
        <BooleanResultType>
            <REQUEST_SUCCEEDED />
        </BooleanResultType>
    </BooleanResult>
    <ResultSetID>R133684703_RS_992003928262</ResultSetID>
    <ResultType>
        <RESULT_SET_ID />
    </ResultType>
    <Status>
        <SUCCESS_RESULTS_AVAILABLE />
    </Status>
    <Hits>821</Hits>
  </QueryResponse>
</CatalogService>
```


Results

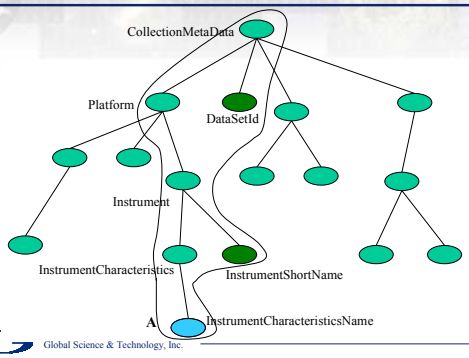
- **Retrieving results of a previously executed stateful query is done by sending a Present message to ECHO with the ResultSetID from the query response**
- **A Present message consists of**
  - a ResultSetID
  - results presentation specification
    - Number and position of results to be returned
    - The attributes of each result to be returned
- **Response message contains results in a CDATA section**
  - The results are XML embedded in a CDATA
  - Need to extract the results and validate against its own DTD
- **ECHO supports storage and retrieval of previously stored results**
  - User results cannot be viewed or removed by another user

Features of results

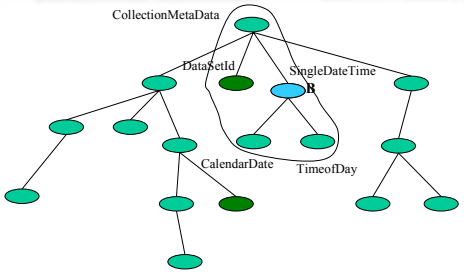
- Returns all metadata fields
- Optionally return parts of the metadata of interest to the client
- Can return results that conform to BMGT and ECHO DTDs
- "Template-based" definition of return values



Results - Horizontal Subsetting



Results - Horizontal Subsetting



Result Details

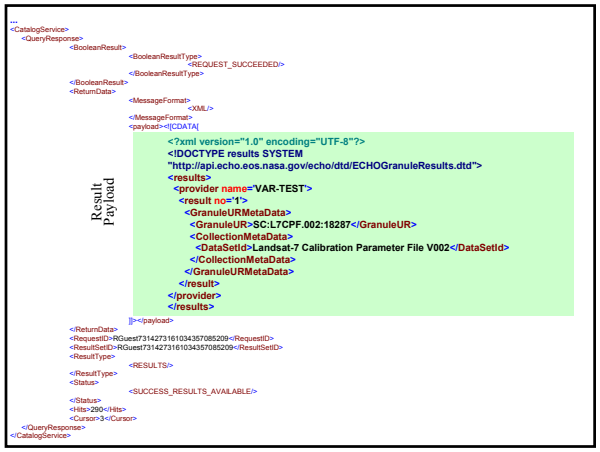
- **Presentation Specification**
  - DTDType
    - ECHO supports results in ECHO ingest DTD format with a few additions like ECHOInsertDate etc. which are not in the ingest DTD*
      - The DTDs for granules and Collections are different.
      - Defaults to the ECHO DTD format
  - Cursor
    - The result set is numbered from 1 to N, the cursor specifies the offset of the first result to be presented within this result set
      - Defaults to 1
  - IteratorSize
    - Starting from the result at the cursor position, iterator size specifies the number of results to be returned
      - A maximum of 2000 results can be presented at a time
      - Defaults to 10
  - TupleType
    - Define what part of the metadata to be returned
      - If all the metadata is to be returned, no need to specify TupleTypes
      - Any element (with a few exceptions) in the result DTD can be specified in the TupleType to be presented
      - The valid values for TupleTypes for a particular result depend on whether the result set contains granules or collections
      - While the TupleType is structured, it is actually treated only as a list

Present Results

- **All elements specified in the results DTD corresponding to the DTDType can be specified in the TupleTypes.**
  - The only exception are the elements that describe the spatial extent of the collection or granule
  - The spatial extent of the collection or granule is considered as a single entity and cannot be split up
- **The list of spatial elements that cannot be as TupleTypes are:**
  - Point, Circle, BoundingRectangle, GPolygon, Polygon, PointLatitude, PointLongitude, CenterLatitude, CenterLongitude, Radius, WestBoundingCoordinate, NorthBoundingCoordinate, EastBoundingCoordinate, SouthBoundingCoordinate, Boundary, ExclusiveZone, SinglePolygon, MultiPolygon, OutRing, InnerRing.

Present Request Message

```
<CatalogService>
<PresentRequest>
 <ResultSetID>R294722418_RS_987548721952</ResultSetID>
 <PresentationDescription>
   <TupleType>
     <attributeName>GranuleUR</attributeName>
     <PrimitiveTypeName><String/></PrimitiveTypeName>
   </TupleType>
      <TupleType>
     <attributeName>DataSetId</attributeName>
     <PrimitiveTypeName><String/></PrimitiveTypeName>
   </TupleType>
      <TupleType>
     <attributeName>CampaignShortName</attributeName>
     <PrimitiveTypeName><String/></PrimitiveTypeName>
   </TupleType>
   <DTDType><ECHO/></DTDType>
 </PresentationDescription>
 <IteratorSize>5</IteratorSize>
  <Cursor>1</Cursor>
</PresentRequest>
</CatalogService>
```



Results - Structure

- The results are grouped per provider
- Each result is numbered

```
<results>
  <provider name="ORNL">
    <result number='1' itemId= 'ORNL1'>
    <result number='2' itemId= 'ORNL2'>
    . . .
  </provider>
  <provider name="LPDAAC">
   <result number='100' itemId= 'LPDAAC1'>
    . . .
  </provider>
</results>
```

- Future versions will include an index of the first result of each provider to facilitate clients to present a few results from each provider to the user

Results - Deleted data and ACL controlled data

- Deleted granules or collections have no effect on Query
- A collection or granule may be deleted after it is in a result set
  - When the result set is presented, there is no metadata to present
  - Will appear like

```
<result number='1' itemId='GSFC70958'>GSFC70958 is
 deleted</result>
```

- A collection or granule may be in the result set but the user may not have permissions to view it
  - Will appear as

```
<result number='1' itemId='GSFC23996'>GSFC23996 is
not visible</result>
```

  - The user can talk with the ops team to find out what procedure they should follow to gain access to the data, if at all

Catalog Service transactions

- Transactions to interact with queries in ECHO
  - Query
  - SaveQuery
  - ListSavedQueries
  - PresentSavedQuery
  - RemoveSavedQuery

SaveQueryRequest

```
<CatalogService>
  <SaveQueryRequest>
    <QueryName>mq1</QueryName>
      <QueryExpression>
        <query>
          <![CDATA[
Query
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE query SYSTEM http://api.echo.eos.nasa.gov/echo/dtd/IIMSAQLQueryLanguage.dtd">
<query>
 <for value="granules"/>
  <dataCenterId>
   <all/>
  </dataCenterId>
 <where>
  <granuleCondition>
   <CampaignShortName><value>'Boreas'</value></CampaignShortName >
  </granuleCondition>
 </where>
</query>
          ]]>
        </query>
        <namespace></namespace>
        <QueryLanguage><IIMSAQL></IIMSAQL></QueryLanguage>
      </QueryExpression>
  </SaveQueryRequest>
</CatalogService>
```


ListSavedQueries

- Request

```
<CatalogService>
  <ListSavedQueriesRequest/>
</CatalogService>
```

- Response

```
<CatalogService>
  <ListSavedQueriesResponse>
    <BooleanResult><BooleanResultType><REQUEST_SUCCEEDED/>
      </BooleanResultType></BooleanResult>
    <QueryName>mq1</QueryName>
    <QueryName>boreasCollections</QueryName>
  </ListSavedQueriesResponse>
</CatalogService>
```

PresentSavedQuery

```
<CatalogService>
  <PresentSavedQueryRequest><QueryName>mq1</QueryName>
    </PresentSavedQueryRequest>
</CatalogService>
```

```
<CatalogService>
  <PresentSavedQueryResponse>
    <BooleanResult>
    <BooleanResultType><REQUEST_SUCCEEDED/></BooleanResultType></BooleanResult>
      <QueryExpression>
        <query><![CDATA[
            <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE query SYSTEM "http://api.echo.eos.nasa.gov/echo/dtd/IIMSAQLQueryLanguage.dtd">
  <query> <for value="granules"/>
   <dataCenterId> <all/> </dataCenterId>
  <where> <granuleCondition> <CampaignShortName><value>'Boreas'</value></CampaignShortName>
      </granuleCondition> </where> </query>
    ]]>
    </query><namespace /><QueryLanguage><IIMSAQL/></QueryLanguage>
    </QueryExpression>
    </PresentSavedQueryResponse>
    </CatalogService>
```


QueryRequest (with saved query)

```
<CatalogService>
    <QueryRequest>
   <QueryName>boreasCollections</QueryName>
            <ResultType>
                    <HITS/>
            </ResultType>
    </QueryRequest>
</CatalogService>
```

```
  <CatalogService>
   <QueryResponse>
    <BooleanResult><BooleanResultType>   <REQUEST_SUCCEEDED/>
     </BooleanResultType>   </BooleanResult>
    <RequestID>RU480210352286052 82</RequestID>
    <ResultSetID>RU480210352286052 82</ResultSetID>
    <ResultType>
      <HITS /> </ResultType>
    <Status><SUCCESS_RESULTS_AVAILABLE/>
    </Status>
    <Hits>292</Hits>
   </QueryResponse>
  </CatalogService>
```


Catalog Service transactions

- **Transactions to interact with result sets in ECHO**
  - Present
  - SaveResultSet
  - ListSavedResultSets
  - RemoveSavedResultSet
- **Remember that saved Result Sets will not be updated with newly inserted data that matches**
  - However, updates to the metadata of existing entries in the result set will be presented upon request
  - Deleted data will no longer show up
  - If a metadata update changes the fact that a particular entry should be in the result set, that change is not reflected

GetMetadata

- **Present is used to look at the metadata in the result set generated by a query**
- **GetMetadata bypasses the need to query to look at metadata**
  - You can specify the unique ECHO item ID to look at its metadata
  - You can specify the Presentation Description to only return certain metadata fields as in Present
- **The response looks like a Present response**
  - There is a payload with the metadata in it

Sample Get Metadata Request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CatalogService PUBLIC "-//ECHO CatalogService (v5.0)//EN"
"http://api.echo.eos.nasa.gov/echo/dtd/CatalogService.dtd">
<CatalogService>
        <GetMetadataRequest>
                <ItemID>C10800-ORNL_DAAC</ItemID>
                <ItemID>C10802-ORNL_DAAC</ItemID>
                <PresentationDescription>
                        <TupleType>
                                <attributeName>Sensor</attributeName>
                                <PrimitiveTypeName>
                                        <String/>
                                </PrimitiveTypeName>
                        </TupleType>
                        <TupleType>
                                <attributeName>Campaign</attributeName>
                                <PrimitiveTypeName>
                                        <String/>
                                </PrimitiveTypeName>
                        </TupleType>
                        <DTDType>
                                <BMGT/>
                        </DTDType>
                </PresentationDescription>
        </GetMetadataRequest>
</CatalogService>
```


ExplainSearchParameter - Dynamic Valids

- **Valids for search criteria on the fly**
  - The valids will honor the access control lists in the system for the user who is requesting the valids
- **Valids for one search criteria can be constrained based on constraints using other search criteria in AQL**
  - E.g. What are the valids for Campaign Short Name if the data is restricted to North American spatial region for ORNL provider
  - Helps the client to develop the query in a drill down fashion
  - Reduces the possibility of a zero hit query

Explain Search Parameter Example

```
<CatalogService>
        <ExplainSearchParameterRequest>
                <SearchParameterName>cloudCover</SearchParameterName>
                <QueryExpression>
                        <query><![CDATA[
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE query PUBLIC "-//ECHO CatalogService (v5.0)//EN"
"http://api.echo.eos.nasa.gov/echo/dtd/IIMSAQLQueryLanguage.dtd ">
<query> <for value="granules"/>
<dataCenterId><value>LPDAAC_ECS</value></dataCenterId>
<where><granuleCondition><onlineOnly/></granuleCondition></where>
</query>
]]></query>
                        <namespace>none</namespace>
                        <QueryLanguage>
                                <IIMSAQL/>
                        </QueryLanguage>
                </QueryExpression>
        </ExplainSearchParameterRequest>
</CatalogService>
```


Explain Search Parameter Response

```
<CatalogService>
 <ExplainSearchParameterResponse>
  <BooleanResult>
   <BooleanResultType>
    <REQUEST_SUCCEEDED/>
   </BooleanResultType>
  </BooleanResult>
  <Valid>
   <Category>
    <CategoryName>cloudCover</CategoryName>
    <CategoryType>GRANULES</CategoryType>
    <CategoryDescription>Scene Cloud Coverage in percentage</CategoryDescription>
    <NegationAllowed>N</NegationAllowed>
    <CriteriaList>
     <Criteria>
      <CriteriaName>cloudCover</CriteriaName>
      <CriteriaType>float</CriteriaType>
      <CriteriaRange>
       <RangeMin>0</RangeMin>
       <RangeMax>100</RangeMax>
      </CriteriaRange>
     </Criteria>
    </CriteriaList>
   </Category>
  </Valid>
 </ExplainSearchParameterResponse>
</CatalogService>
```


ECHO Query Language

Global Science & Technology, Inc.

Query Language

- **Currently ECHO supports the IIMSAQL query language**
  - Domain based query
  - XML-based and must be validated by its own DTD
  - Supports querying for collections (discovery) and granules (inventory search)

IIMSAQL - Skeleton

Discovery (Search for collections)

```
<query><for value="collections"/>
<dataCenterId><value>ORNL</value></dataCenterId>    <where>
    <collectionCondition>....</collectionCondition>
    <collectionCondition>....</collectionCondition>
    .....
  </where>
</query>
```

Inventory Search (Search for granules)

```
<query><for value="granules"/>
  <dataCenterId><all/></dataCenterId>
  <where>
    <granuleCondition>....</granuleCondition>
    <granuleCondition>....</granuleCondition>
    .....
  </where>
</query>
```


Sample Query to ECHO

```
<CatalogService>
 <QueryRequest>
  <QueryExpression>
   <query>
    <![CDATA[
```

AQL Section

```
<query>
  <for value="granules"/>
    <dataCenterId>
      <value>ORNL</value>
    </dataCenterId>
  <where>
    <granuleCondition>
      <CampaignShortName><value>'Boreas'</value></CampaignShortName>
    </granuleCondition>
  </where>
</query>
```

```
]]>
    </query>
    <namespace>none</namespace>
    <QueryLanguage><IIMSAQL/></QueryLanguage>
  </QueryExpression>
  <ResultType> <HITS/></ResultType>
 </QueryRequest>
</CatalogService>
```

IIMSAQL – Types of searches

- **Spatial search on the spatial coverage of the data**
- **Temporal search on the temporal coverage of the data**
- **Keyword search on campaigns, platform, spatial keywords, etc.**
- **Number based search on cloud cover, PSA etc**
- **Date based search on ECHO insert and last update dates and PSAs with dates**
- **Function based search like global granules, online granules, etc.**

* The following slides are not presented in this order

IIMSAQL – Keyword search

- Keyword search on campaigns, platform, spatial keywords etc.
  - List of keywords that are ORed or wild card character based search
    - List cannot exceed more than 1000 values
  - All keyword searches are case insensitive

IIMSAQL Spec	Search for

`<campaign>`
`  <value>'River'</value>`
`</campaign>`
	Campaign 'River'

`<campaign>`
`  <list>`
`  <value>'River'</value>`
`  <value>'Lake'</value>`
`  </list>`
`</campaign>`
	Campaign 'River' or 'Lake'

`<campaign>`
`  <textPattern operator="LIKE">`
`'Boreas%'`
`  </textPattern>`
`</campaign>`
	Search using Wild card characters. Wild Cards must conform to oracle wild cards. Search for campaign starting with 'Boreas'.

IIMSAQL – Keyword search

- **Pattern matching in Oracle using wildcards**
  - Percent '%' matches zero or more characters
    - '$M%' could match SMITH, SMITHS etc
    - % cannot match null values
  - Underscore '_' matches exactly one character
    - 'MAT_' will match MATE but not match MAT
  - ESCAPE Option
    - To match 'A_B' or 'A%B' escape the '_' or '%' with a '\'
    - 'A_B' will match 'A_B'
  - Performance
    - Oracle index on text, indexes on the first character
    - If '%' or _ is the first character in the pattern the index is not used and the search will take much longer
    - E.g. Search for '_hyme' that matches rhyme and thyme will not use the index

IIMSAQL – Number search

- **Number based search on cloud cover, PSA etc**
  - Search for a list of values or range of values
    - *List cannot exceed more than 1000 values*

IIMSAQL Spec	Search for

`<cloudCover>`
`  <range low="10" high="20">`
`</cloudCover>`
	Return only those granules that have cloud cover between 10% and 20%. (cloud Cover is a percentage, mandated by IIMSAQL)

`<psa>`
`<psaName>'QAStatPctBadPixels'</psaName>`
`  <psaValue>`
`  <range upper='20'/>`
`  </psaValue>`
`</psa>`
	Search for data with a *QAStatPctBadPixels*
	PSA in which that PSA has less than or equal to 20% bad pixels. (the value 20 is a percentage is implied by the definition of the PSA and does not need to be spelled out)

IIMSAQL – Date search

- Date based search on ECHO insert and last update dates and PSAs with dates
  - Search for data inserted or last updates on a date or within a date range

IIMSAQL Spec	Search for

`<echoInsertDate>`
`  <dateRange><startDate>`
`  <Date YYYY="2001" MM="04" DD="05"/>`
`  </startDate></dateRange>`
`</echoInsertDate>`
	Return only data inserted into ECHO on or after April 5, 2002

`<echoLastUpdate>`
`  <dateRange>`
`  <startDate>`
`  <Date YYYY="2001" MM="01" DD="01"/>`
`  </startDate>`
`  <stopDate>`
`  <Date YYYY="2001" MM="06" DD="01"/>`
`  </stopDate>`
`  </dateRange>`
`</echoLastUpdate>`
	Return only data last updated in ECHO between January 1 2001 and June 1, 2001 (both days included)

IIMSAQL – Temporal search

- **Temporal search on the temporal coverage of the data**
  - Search for data that intersects with the date range or recurring date range
- **Range Search**
  - Searches for granule/collections whose temporal range overlaps with the time between May 12, 1990 at 5:30 am to June 13, 1992

```
<temporal>
   <startDate><Date YYYY="1990" MM="5" DD="12" HH="5" MI="30" SS="0"/></startDate>
   <stopDate><Date YYYY="1992" MM="6" DD="13"/></stopDate>
</temporal>
```

- **Recurrent Search**
  - Searches for granules/collections whose temporal range overlaps with the time ranges: May 16, 1990 - July 14, 1990 or January 5, 1991 - March 5, 1991 or January 5, 1992 - February 10, 1992

```
<temporal>
   <startDate><Date YYYY="1990" MM="5" DD="12"/></startDate>
   <stopDate><Date YYYY="1992" MM="2" DD="10"/></stopDate>
   <startDay value="5"/><endDay value="60"/>
</temporal>
```

IIMSAQL – Spatial search

- **Spatial search using IIMSPolygon**

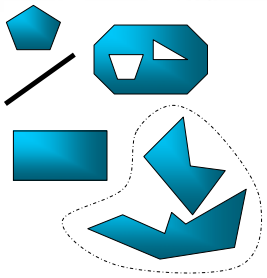
```
<spatial><IIMSPolygon>
          <IIMSLRing>
            <IIMSPoint long='50' lat='-20'/>
            <IIMSPoint long='70' lat='-20'/>
            <IIMSPoint long='70' lat='30'/>
            <IIMSPoint long='50' lat='30'/>
            <IIMSPoint long='50' lat='-20'/>
          </IIMSLRing>
  </IIMSPolygon></spatial>
```

- **Spatial search using OGC XML elements**

```
<spatial operator="RELATE">
  <Polygon>
    <LRing>
      <CList>-120, -30, -100, -60, 5, -90, -120, -60, 160, 5, 160, 60, 120, 85, 5, 85,
        -120, 30, -120, -30</CList>
    </LRing>
    <LRing><CList>80, 20, 80, 60, 20, 60,20, 20, 80, 20 </CList></LRing>
  </Polygon>
</spatial>
```

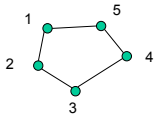

IIMSAQL - Types of Spatial search

- **Support GML 1.0 and IIMS geometries**
- **Polygons and Polygons with holes**
- **Multi polygons**
- **Lines**



IIMSAQL - Spatial Details

- **Polygon can consist of more than one ring**
  - Simple polygon will only have 1 outer ring
  - Polygons with holes can have 1 outer ring and multiple inner rings that indicate the holes
- **Each Ring must have the same first and last point.**
- **Points must be specified in counter-clockwise order**
- **When using GML elements the point must be specified with longitude first, followed by latitude.**

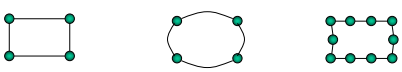


IIMSAQL - Spatial Details Cont'd

- **Providers are registered to use Cartesian or Geodetic coordinate system for spatial data**
  - Cartesian system
    - Allows the coverage to be as large as the whole earth but does not allow data that covers the pole or crosses the date line
    - Search window also needs to conform to these restrictions
  - Geodetic system
    - Allows the data to cross the date line and the poles but the total area of a single polygon cannot exceed half the earth
    - Search window also needs to conform to these restrictions

IIMSAQL - Spatial Details Cont'd

- **A search polygonal window is converted to Cartesian polygon when searching data from providers using Cartesian coordinates, and to geodetic polygons while searching data from providers using Geodetic coordinates**
- **Polygons in Cartesian coordinates are connected using straight lines while those in Geodetic system are converted using great circle arcs.**
  - For the same polygon (represented using the same points) the area covered using the Cartesian coordinate system may be different from the area covered using the geodetic system
- **To ensure better accuracy represent the polygon with more than just the corner points.**
  - Experiments have shown acceptable performance with 50 point polygons

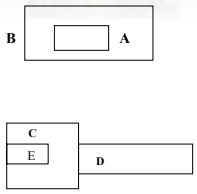


IIMSAQL - Spatial Details Cont'd

- **Oracle Spatial Search**
  - 2 step search
    - Fast filter of bounding box of the data against the query window
    - Slower more accurate polygonal geometry search for exact results on the results from the fast filter
  - A small window results in more data being eliminated by the first filter, so the search will run faster
  - Skewed polygonal windows or data will have lots of wasted space in the bounding box will result in more false positives passing through the fast filter

IIMSAQL - Types of Spatial Search Operators

- **WITHIN :** Interior and boundary of the first object is completely contained in the interior of the second.
  - Search region B, A is within
- **CONTAINS:** The interior and boundary of the second object is completely contained in the interior of the first.
  - Search region A, area B contains A
- **EQUAL:** They have the same boundary and interior.
- **TOUCH :** The boundaries intersect but the interiors do not.
  - C and D touch
  - C and E do not touch
- **RELATE:** The objects are non-disjoint, i.e. their body and/or boundaries intersect.



Global data

- **Some data have global coverage**
  - ECHO does not spatially index global data
- **Recommend that Collections that are global register "Global" as a spatial keyword for the Collection**
  - All data may not follow this convention
  - Search for collections using spatial constraint will not return global collections
- **Recommend that granules that are global have their global flag set**
  - Search for granules with spatial constraint will not return global granules
  - Search using globalGranulesOnly function search

IIMSAQL - Function search

- Function based search
  - –Not based on the value of a particular metadata field
  - –Search for online data, search for data with browse, globalGranules, etc.

IIMSAQL Spec	Search for

`<granuleCondition>`
`  <browseOnly/>`
`</granuleCondition>`
	Search for granules that have browse associated with them

`<granuleCondition>`
`  <dayNightFlag value='DAY'/>`
`</granuleCondition>`
	Search for granules gathered at least partly during daylight.

`<collectionCondition>`
`  <onlineOnly/>`
`</collectionCondition>`
	Search for collections that are available online

IIMSAQL - Discovery search criteria

Search Criteria	Search for collections based on
Campaign Short Name	'Soil Collections','River Discharge (RIVDIS)'
Dataset ID	'15 MINUTE STREAM FLOW DATA: USGS (FIFE)'
ECHO Insert Date	The date it was inserted into ECHO; e.g. from July 3, 2001 at 5:30 pm to June 2, 2002
ECHO Last Update	The date the collection level metadata for this collection was last updated in ECHO.
Online Collections Only	Only those that are available online
Parameter	Geo-physical parameter; e.g.
Processing Level	1A, 2B, etc.
Sensor Name	'BAROMETER', 'CAMERA', 'SWIR'
Short Name	'AST_L1A', 'MYDPT1KN'
Source Name	'DIGITAL ELEVATION MODEL', 'NOAA-9'
Spatial	Polygon, polygon with holes, multi polygon, lines
Spatial Keywords	'Global', 'United Kingdom', 'Solar'
Temporal	Date range and periodic ranges search over the temporal coverage of the collection
Temporal Keywords	'January', 'Spring', 'Stone age', 'Weekly'
PSA Names	'DAR_ID', 'QAFRACTIONGOODQUALITY','FIREPIXELS'

IIMSAQL - Inventory search criteria

Search Criteria	Search for granules based on
Only Granules with Browse Data	Granules that have associated browse in ECHO
Campaign Short Name	'Soil Collections','BOREAS'
Percentage of Cloud Cover	Percentage of cloud cover between 5-10%
Dataset ID	Granules from the collection identified by dataset id
ECHO Insert Date	Date it was inserted into ECHO
ECHO Last Update	granule metadata last updated in ECHO
Either Day or Night granules	Whether they are taken in day time, night time or both
Only Global Granules	Whether the granules are global
Search on granule Ids (ECHO specific)	Only search for specific granules
Search on Granule UR	'SC:AST_L1A.003:2007226177', 'VEMAP_1_SITE.w'
Online Granules Only	Only those that are available online
Path Row Range	Path row coverage; e.g. path 6, row 100
Sensor Name	'BAROMETER', 'CAMERA', 'SWIR'
Source Name	'DIGITAL ELEVATION MODEL', 'NOAA-9'
Spatial	Polygon, polygon with holes, multi polygon
Temporal	Temporal coverage of the granule
PSA	e.g. granule for DAR_ID value '345f3c'
Local Granule ID	The local identifier used by providers

Validate! Validate! Validate!

Validate IIMSAQL query against its own DTD!

Valids for query

- **Valid values and ranges for IIMSAQL search criteria**
- **Static Valids**
  - ECHO wide valid values for each search criteria
  - No valids for spatial coverage
  - Refreshed periodically to reflect state of system
  - Available at a known place for clients
- http://browse.echo.eos.nasa.gov/valids/aqlvalids.xml

Sample Static Valids

```
<ECHOStaticValids>
 <Category>
  <CategoryName>sensorName</CategoryName>
  <CategoryType>GRANULES, COLLECTIONS</CategoryType>
  <CategoryDescription>Sensor short names</CategoryDescription>
  <CriteriaList>
   <Criteria>
    <CriteriaName>sensorName</CriteriaName>
    <CriteriaType>String</CriteriaType>
    <CriteriaLength>1024</CriteriaLength>
    <CriteriaValues>
     <CriteriaValue>AAS</CriteriaValue>
     <CriteriaValue>ADVANCED SOLID-STATE ARRAY SPECTRORADIOMETER</CriteriaValue>
     <CriteriaValue>AIR THERMOMETER</CriteriaValue>
     <CriteriaValue>AIRBORNE VISIBLE AND INFRARED IMAGING SPECTROMETER</CriteriaValue>
     <CriteriaValue>ALGORITHM</CriteriaValue>
    </CriteriaValues>
   </Criteria>
  </CriteriaList>
 </Category>
</ECHOStaticValids>
```


Getting Best Performance out of Catalog Service

- **Query times so far, seem adequate**
- **Presenting metadata is the bottleneck**
- **Query using HITS as the ResultType instead of RESULTS**
  - Sometimes a query executes successfully but the results take longer and the entire transaction times out, leaving the client to re-execute the query
    - This is a bug that will be fixed in 7.0
  - Problem will go away if no results are requested with the Query
- **Use iterator size to control the number of results being returned at a time**
- **Use TupleTypes to get back the minimum metadata that is needed**
  - Optimal number of results being returned at a time may depend on the amount of metadata needed per result
- **Support drilling down into metadata on user requests**

ECHO Resources

- **Use ECHO resources judiciously**
  - Use search on ECHO insert date or last update if only updated data is required rather than searching the entirety of ECHO holdings
  - Search ECHO for data in a small spatial region
    - Large spatial regions (like the majority of the Earth) will evaluate the majority of the data and will take a while
  - Use static valids to increase the probability of searching for data that interests the user
  - Delete result sets that are no longer needed
    - In the future, ECHO will put a cap on number of results per user
  - Retrieve minimum required metadata for each result
  - Use subscription service if bulk metadata updates are required on a regular basis
  - Don't hog the system!

Future Requirements

- **Asynchronous queries**
- **Search on more attributes in the ingest DTD**
- **Orbital Search**
- **Rename IIMSAQL to ECHOQL**

Future Requirements – Asynchronous Queries

- **Queries executed by ECHO in the background, offline**
- **Completion of execution notice will be sent to users via email**
- **Guest will not be allowed to perform Asynchronous queries**
- **Result sets of asynchronous queries are automatically saved for a system (or user) specified time period**

Future Requirements – Search Criteria

- **Search on more attributes in the ingest DTD**
  - Determine list of search criteria
  - Current list for additions includes
    - Producer granule id
    - Measured parameters
    - Parameter keyword stack
    - Search function lastECHOInsert and lastECHOUpdate
      - Return granules inserted (or updated) in ECHO in the last 10 days
    - ANDs for certain parameters that make sense
    - Ability to query for the most recent N granules in a collection based on the granule's temporal coverage

Future Requirements – Orbital Search

- **Oracle Spatial search uses R-tree indexing**
  - R-tree indexes work best with small, square-like spatial data
- **Spatial coverage of some earth science data are swaths**
  - To accurately represent swaths in ECHO requires a lot of points to represent them
  - Oracle spatial search may be slow for these types of data
- **Orbital search converts search for spatial data into a search on simple pre-processed numeric data that is much faster**
- **Currently ECHO is updating its data model to ensure it contains sufficient information to perform this type of search**

Future Requirements – Rename query language

- **Rename IIMSAQL to ECHOQL**
- **Long Overdue!**

Validate! Validate! Validate!

Validate IIMSAQL query against its own DTD!

Break

Global Science & Technology, Inc.

Provider Profile Service

A mechanism for clients to discover what providers are participating in ECHO, and what capabilities they offer.

Global Science & Technology, Inc.

List Providers

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProviderProfileService PUBLIC "-//ECHO ProviderProfileService
(v5.0)//EN"
"http://api.echo.eos.nasa.gov/echo/dtd/ProviderProfileService.dtd">
<ProviderProfileService>
<ListAllProvidersRequest/>
</ProviderProfileService>
```


List Providers Response

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE ProviderProfileService SYSTEM
"http://api.v50.echo.eos.nasa.gov/echo/dtd/ProviderProfileService.dtd">
<ProviderProfileService>
        <ListAllProvidersResponse>
                <BooleanResult>
                        <BooleanResultType>
                                <REQUEST_SUCCEEDED/>
                        </BooleanResultType>
                </BooleanResult>
                <ProviderID>GSFCECS</ProviderID>
                <ProviderID>ORNL-DAAC</ProviderID>
                <ProviderID>USGSNODAAC</ProviderID>
                <ProviderID>NSIDC_TS1</ProviderID>
                <ProviderID>LPDAAC_ECS</ProviderID>
        </ListAllProvidersResponse>
</ProviderProfileService>
```


Present Provider Profile

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProviderProfileService PUBLIC "-//ECHO ProviderProfileService
(v5.0)//EN"
"http://api.echo.eos.nasa.gov/echo/dtd/ProviderProfileService.dtd">
<ProviderProfileService>
<PresentProviderProfilesRequest>
<ProviderID>ORNL-DAAC</ProviderID>
</PresentProviderProfilesRequest>
</ProviderProfileService>
```



```
<ProviderProfileService>
        <PresentProviderProfilesResponse>
                <BooleanResult>
                        <BooleanResultType>
                                <REQUEST_SUCCEEDED/>
                        </BooleanResultType>
                </BooleanResult>
                <ProviderProfile>
                        <ProviderID>ORNL-DAAC</ProviderID>
                        <OrganizationName>Oak Ridge National Laboratory Distributed Active Archive Center for
Biogeochemical Dynamics</OrganizationName>
                        <SpatialProjectionType>
                                <CARTESIAN/>
                        </SpatialProjectionType>
                        <ProviderContact>
                                <ContactRole>system engineer</ContactRole>
                                <ContactFirstName>Tim</ContactFirstName>
                                <ContactLastName>Rhyne</ContactLastName>
                                <AddressInformation>
                                        <AddressID>ornl contact address</AddressID>
                                        <USFormat>True</USFormat>
                                        <Street1>PQB 2008, Bldg. 1507, MS-6407</Street1>
                                        <City>Oak Ridge</City>
                                        <State>TN</State>
                                        <Zip>37831-6407</Zip>
                                        <Country>USA</Country>
                                </AddressInformation>
                                <PhoneInformation>
                                        <PhoneName>voice</PhoneName>
                                        <CountryCode>US</CountryCode>
                                        <AreaCode>865</AreaCode>
                                        <ExchangeCode>574</ExchangeCode>
                                        <Phone>7447</Phone>
                                </PhoneInformation>
                                <EMailAddress>rhynebl@ornl.gov</EMailAddress>
                        </ProviderContact>
                </ProviderProfile>
        </PresentProviderProfilesResponse>
</ProviderProfileService>
```

Order Entry Service

ECHO is a clearinghouse of metadata, but does not touch the actual data. ECHO brokers orders to the providers of the data, dealing with the intricacies of communicating with multiple providers asynchronously. This service allows a client to create, validate and submit an order, as well as quote one.

Global Science & Technology, Inc.

Order Entry Service

- **Allow users to create, modify, quote, and submit orders.**
  - Some providers post the actual data on the web with its online URL included in the query result. In this case, users can browse the data without going through the order process provided in Order Entry Service.
- **One order consists of one or more provider orders. Each provider order represents the catalog items that a user wants from a particular provider.**
- **Each provider order consists of one or more order line items, with each line item describing the catalog item that is being requested including any requested options for that item.**
  - Catalog Items in ECHO are either granules or collections
  - Only items that are orderable are Catalog Items

Order Information

- **An order consists of several pieces of information**
  - Order ID
  - Order State
  - Shipping, Billing and Contact Addresses
  - Order Price
    - This is not implemented, so is currently ignored
  - Options at the order level
    - Currently, there are none defined
  - For each Provider, provider order state, a list of catalog items to be ordered, quantity and options

Components of An Order

```
<Order>
    <OrderID>2958</OrderID>                                  ← order ID
    <OrderState> <NOT_VALIDATED/></OrderState>               ← order state
    <ShippingAddress>                                        ← shipping address
      <shipToBusinessName/>
       <ContactName><FirstName/><LastName/></ContactName>
      <AddressInformation>
        <AddressID>Shipping</AddressID>
        <Street1/><Street2/><City/><State/><Zip/><Country/>
      </AddressInformation>
    </ShippingAddress>
    <BillingAddress>                                         ←billing address
      <billToBusinessName/>
      <ContactName><FirstName/><LastName/> </ContactName>
      <AddressInformation>
        <AddressID>Billing</AddressID>
        <Street1/><Street2/><City/><State/><Zip/><Country/>
      </AddressInformation>
    </BillingAddress>
    <ContactAddress>                                         ←contact address
  <ContactName><FirstName/><LastName/></ContactName>
  <PhoneString/><EmailString/>
  <AddressInformation>
    <AddressID>Contact</AddressID>
    <Street1/><Street2/><City/><State/><Zip/><Country/>
  </AddressInformation>
 </ContactAddress>
```


Components of An Order (Continued)

```
<orderPrice>0.0</orderPrice>                  ← order price
<OptionSelection>… </OptionSelection> ← order level option
<ProviderOrder>                               ← first provider order  from ECHO-TEST
  <ProviderOrderID>
    <ProviderID>ECHO-TEST</ProviderID>
    <OrderID>2958</OrderID>
  </ProviderOrderID>
  <OrderLineItem>                             ← contains only one item
          <CatalogItemID>740</CatalogItemID>
          <quantityOrdered>1</quantityOrdered>
          <OptionSelection>… </OptionSelection> ← line item level option
  </OrderLineItem>
  <ProviderOrderState> <NOT_VALIDATED/></ProviderOrderState> ← provider order state
  <totalPrice>0.0</tatalPrice>
</ProviderOrder>
    <ProviderOrder>                           ← second provider order from ECHO-TEST2
      <ProviderOrderID>
        <ProviderID>ECHO-TEST2</ProviderID>
        <OrderID>2958</OrderID>
      </ProviderOrderID>
      <OrderLineItem>
        <CatalogItemID>1626</CatalogItemID>   ←the  first  item
        <quantityOrdered>2</quantityOrdered>
      </OrderLineItem>
      <OrderLineItem>
        <CatalogItemID>1627</CatalogItemID>  ← the second  item
        <quantityOrdered>4</quantityOrdered>
      </OrderLineItem>
      <ProviderOrderState><NOT_VALIDATED/></ProviderOrderState>
      <estimatedPriceOfItems>0.0</estimatedPriceOfItems>
  </ProviderOrder></Order>
```


Options in Order Entry Service

- **Options can be displayed and set for most of the Order Entry Service objects. Most notably:**
  - Orders (no order-level options currently exist)
  - Provider Orders
  - Order Line Items (where most options are located)
  - Also used in other transactions, but are called policies or preferences.

What are Options?

- **ECHO uses a common infrastructure for allowing abstract, definable items to be specified at run-time**
  - If the structure of something is not known when ECHO is developed, then the option infrastructure can support representing that structure without changes to the ECHO code
  - The options infrastructure allows a client to see what the template (Option Definitions) for a structure is, as well as any previous selections
  - It also allows a client to set (select) values in the structure according to the rules in the Option Definitions Template

Options

- **Option Definitions - describe the options available for a given item. Some options may be declared as required.**
- **Option Selections - the actual value associated with an option definition for a particular item.**

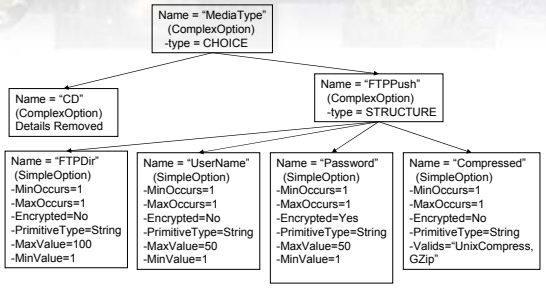
Option Definitions

- **Simple Option Attributes**
  - Option Name- The string that is used to refer to this node of the option tree
  - MinOccurs - The minimum number of times this node should appear in the option tree
  - MaxOccurs - The maximum number of times this node should appear in the option tree
  - Encrypted - When representing this field for display or input, use a text box that hides what is being typed (password box)
  - Primitive Type - The type of the option being represented
    - String, Integer, Boolean, Date, Double
  - Valids - If a string, an enumeration of valid values, if a number, an enumeration of valid values
  - Min Value - If a number, the minimum value, or if a string, the minimum length
  - Max Value - If a number, the maximum value, or if a string, the maximum length
- **Complex Option Attributes**
  - Structure - All contained options should be present in the Option Selection
  - Choice - Only 1 of the contained complex options should be present in the Option Selection

Option Selections

- **The Option Selection tree should parallel the Option Definitions tree**
- **Simple Option Attributes**
  - Option Name - Should equal Option Definition name at this node in the tree
  - Value - The value that the user wants assigned
    - This will be validated against the information in the Option Definition for this node
- **Complex Option Attributes**
  - Option Name - Should equal Option Definition name at this node in the tree
  - ComplexValue - This is a container for placing the Option Selections for the corresponding child Options
- **Option Selections are validated against the Option Definition when Options are set**

Options Definition Illustration Example

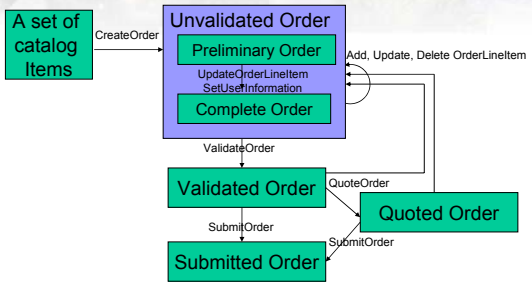



```
<OptionSelection>
    <ComplexOptionSelection>          ← complex option consists of
                                        ← an option name  and a complex value
      <OptionName>ECS-TEST PKG1</OptionName>
      <ComplexValue>                  ← the complex value consists of
                                        ← zero or more simple option
                                        ←and zero or more complex option
        <SimpleOptionSelection>                    ←simple option consists of
          <OptionName>Production Option</OptionName>  ← an option name
          <Value>Native Granule</Value>              ← and one or more value
        </SimpleOptionSelection>
        <ComplexOptionSelection>
          <OptionName>Media Type</OptionName>
          <ComplexValue>
            <ComplexOptionSelection>
              <OptionName>CDROM</OptionName>
              <ComplexValue>
                <SimpleOptionSelection>
                  <OptionName>Compression</OptionName>
                  <Value>GZip</Value>
                </SimpleOptionSelection>
                <SimpleOptionSelection>
                  <OptionName>DDISTMEDIAFMT</OptionName>
                  <Value>Rockridge</Value>
                </SimpleOptionSelection>
              </ComplexValue>
            </ComplexOptionSelection>
          </ComplexValue>
        </ComplexOptionSelection>
      </ComplexValue>
    </ComplexOptionSelection>
</OptionSelection>
```

Options in Order Entry Service

- **Options are used here to represent the Provider defined settings that are needed to completely describe an order but are unique to a provider**
- **Order Entry Service allows for options at 3 levels**
  - Order - Options that apply across the entire order
    - Currently not used
  - Provider Order - Options that apply to every item being sent to a Provider
    - Currently not used, but could be
  - Line Item - Options that apply only to the smallest unit being ordered
    - Package options are specified here (media, shipping, etc.)
    - This is specified when you create or update the Order Line Item
- **Order options are further validated when the Validate Order command is sent**
  - An option setting does not have to be provided when the Order Line Item is first created, but if needed, it must be provided before Validation

Basic Order Flow



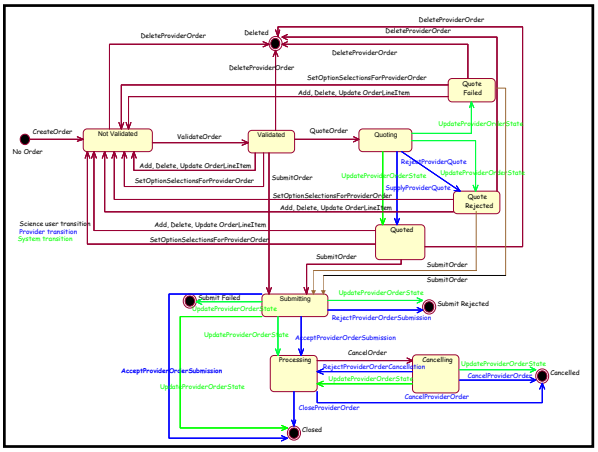
States of a Provider Order

Open Order States

- Not Validated
- Validated
- Quoting
- Quoted
- Quote Failed
- Quote Rejected
- Submitting
- Processing
- Cancelling

Closed Order States

- Deleted
- Submit Failed
- Submit Rejected
- Cancelled
- Closed



States of an Order

- The state of the order is calculated based on the Provider Order states
- The client has the ability to see both the order state and the provider order states and can therefore represent the order in any way they deem appropriate

- Not Validated
- Validated
- Quoting
- Quoted
- Quoted with exceptions
- Submitting
- Submitted with exceptions
- Processing
- Processing with exceptions
- Cancelling
- Closed with exceptions
- Closed
- Cancelled

OrderEntryService API

- **PresentCatalogItem (CatalogItemID)**
  - Displays information about a specific catalog item including product description, product list price, and a list of options (required or available) for that catalog item.
  - It is presumed that any item that the client will need to run this transaction for any item it is going to place into an order so that it can get the Option Definitions.
- **ListUnsubmittedOrderSummary (UnsubmittedOrderState)**
  - List a summary of the orders that are in the specified states.

Order Level Transactions

- **CreateOrder (CatalogItem)**
  - create an order by specifying one or more order line items and return back a unique OrderID.
  - Option Selections are optionally provided during creation
- **PresentOrder (OrderID)**
  - Displays the current description of the order that was specified.
- **DeleteOrder (OrderID)**
  - delete a specified order from the system. This only applies to orders that have not been submitted. If they have been submitted, use the CancelOrder transaction in UserAccountService.
- **SetUserInformationForOrder (OrderID, ShippingAddress, BillingAddress, ContactAddress, NetworkAddress, CreditAccountID, PurchaseOrderID)**
  - Specify specific shipping and billing address for a particular order.

Order Level Options

- **PresentOptionDefinitionsForOrder (OrderID)**
  - Displays a list of options (required or available) for the specified order. These options affect all provider orders within this order.
- **SetOptionSelectionsForOrder (OrderID, OptionSelection)**
  - Set the desired values for the available options for the specified order.
- **Note that there are no order level options currently, and none planned for the immediate future**

Provider Order Level Transactions

- **PresentProviderOrder (ProviderOrderID)**
  - Displays the portion of the order that represents catalog items from the specified provider.
  - ProviderOrderID is made up of the Provider ID and the Order ID to make a unique identifier for that part of an order that goes to a single provider
- **PresentOptionDefinitionsForProviderOrder(ProviderOrderID)**
  - Displays the list of options (required or available) for the specified provider order.
- **SetOptionSelectionsForProviderOrder (ProviderOrderID, OptionSelection)**
  - Set the desired values for the available options in a particular provider-order.
- **DeleteProviderOrder (ProviderOrderID)**
  - Remove a provider order from the larger order context.
- **SetAuthenticationKeyRequest (ProviderOrderID, AuthenticationKey?)**
  - Not currently used

Line Item Transactions

- **UpdateOrderLineItem (OrderID, OrderLineItem)**
  - Modify one or more order line items in an order, including the quantity and the Option Selections for that order line item
  - The line item to be updated is determined by the CatalogItemID contained in the OrderLineItem
  - Replaces the entire line item
- **AddOrderLineItem (OrderID, OrderLineItem)**
  - Add one or more order line items to an order
- **DeleteOrderLineItem(OrderID, CatalogItemID)**
  - Remove one or more order line items completely from an order

Order Action Transactions

- **ValidateOrder (OrderID)**
  - Validates that the specified order has all the necessary contact fields filled out and the required order options are specified. Must be done before a user is allowed to submit the order. Returns the Order being validated
- **QuoteOrder (OrderID)**
  - To request a quote for a specified order.
  - In effect, each provider order that makes up the full order will be sent to their representative provider.
  - Requires a response from providers, thus no changes may be made to the order until all providers have responded.
- **SubmitOrder (OrderID, NotificationMechanism)**
  - Request that the specified order be submitted to the necessary providers.
  - In effect, each provider order that makes up the full order will be sent to their representative provider.
  - NotificationMechanism specifies whether the user will manually check for updated status of their order or be emailed the status.
    - Notification Level- Verbose, Detail, Info, Critical, None

Caveats

- **Order line item could become un-orderable before the order is submitted**
  - Data provider sets new restriction to prevent ordering the item
  - Data provider revokes permission for ordering the item
  - Data provider deletes the item
- **An error message will be generated indicating the reason for not being able to order the item**
  - Checking is done in any OES transaction except Present Order

Future Order Features

- **Hierarchical setting of options**
- **Line Item Order Status**
- **Handle prices/quotes better**
- **CreateAndSubmitOrder as a single transaction**
  - This will be in addition to existing transactions

User Preferences

ECHO uses a generic policy/ option/ preference mechanism to provide a run-time way of changing values that the business object operates on. This section describes this mechanism and how it applies to Users.

User Preferences

- **Preferences that a user sets to change the behavior of ECHO when managing their interactions with ECHO.**
- **Examples**
  - Default billing address
  - Default shipping address
  - Default contact address
- **User Preference Transactions**
  - PresentOptionDefinitionsForUser
  - SetOptionSelectionsForUser
  - PresentOptionSelectionsForUser

Order Best Practice

- **Use User Account Service to maintain a list of addresses and phone numbers**
- **Use User Preferences to identify which of the addresses, emails and phone numbers are the user's default setting**
- **When creating the order, pre-fill in the order shipping, billing and contact information from the address book in UAS and as specified by the defaults in User Preferences**
- **Allow the user to choose other addresses and phone numbers that are in UAS easily**
- **Allow the user to add a new address, email or phone number easily, and then offer the user the opportunity to add them into the address book in UAS**

Subscription Service

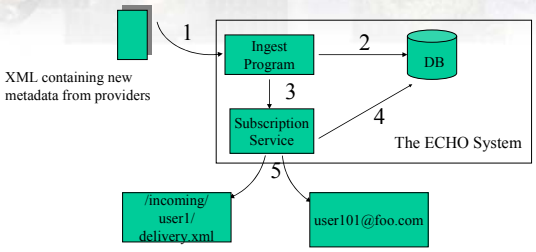
This service allows a user of the system to get all metadata updates that ECHO receives for a specific collection

Global Science & Technology, Inc.

Why do we have Subscription Service?

- **Keep user/system informed of updated metadata**
- **Makes it easier to obtain data from ECHO**
  - Don't have to issue a query every night
  - Only updated metadata if sent
  - Push rather than pull model
- **Helps power users and near-ECHO applications**
  - Extension point from ECHO
    - Allows for pre-processing of ECHO's providers' metadata
    - Allows for data massaging
      - Creation of OLAP view (data cube)
- **Must be a registered user to use this service**
  - Email address of user is used to report errors
- **The subscription service honors the Provider's Access Control Lists by only delivering data that the registered user has access to**

Subscription Process Chart



1. New metadata received
2. Store new metadata
3. Notification of ingest
4. Query for data
5. Deliver data

Subscription Key Information

What can you subscribe to?

- **All collections**
  - Receive collection metadata updates of all collections that of different providers
- **A collection**
  - Receive collection metadata updates of a collection
- **A collection's granules**
  - Receive granule metadata updates of a collection
- **A collection and its granules**
  - Receive both collection metadata and granule metadata updates of a collection

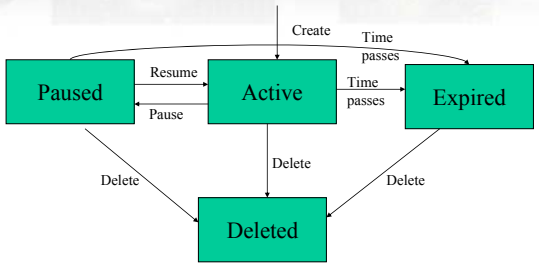
Subscription Key Information

- **What dataset (collection) are you interested in?**
  - You can use the Catalog Service to get the list of datasets (collections) from the system
- **Where should ECHO send the metadata when metadata is updated?**
  - Two delivery mechanisms: FTP and E-Mail
- **What file format to be sent?**
  - Uncompressed text or Compressed (Gzip)

Creating a Subscription

- **Required parameters**
  - Name
  - Provider and Dataset
  - Presentation format, compression option, type of metadata to be sent
  - Delivery information
    - FTP
      - Delivery address (anonymous@ftp.foo.com)
      - Password
      - Delivery folder
    - E-Mail
      - Delivery address (user@foo.com)
  - Expiration date
- **Delivery Address Format**
  - For FTP: username@machinename:portnumber
  - For email: username@domain

Subscription Life Cycle



Manage Subscription

- **Create Subscription**
- **Present Subscription**
- **Update Subscription**
- **Pause Subscription**
- **Resume Subscription**
- **Delete Subscription**
- **List Subscription Names**

Subscription Service Caveats

- **Delivery Address is used for both email and FTP addresses**
  - The format is username@host
- **Ingests that match a subscription but have all metadata filtered out for some reason will generate an empty delivery**
  - There will be a delivery with no metadata contained in it
- **When new metadata is not delivered because of an ACL issue, only a change in science metadata will trigger sending that metadata out**
  - There is a risk that new metadata will never be sent out to an unprivileged user

Future Items for Subscription Service

- **Ability to subscribe to one provider's set of collections**
- **Ability to subset collection on spatial**
- **Ability to renew an expired subscription**
- **ECHO might consider adding the capability to create orders automatically for data upon receipt of new metadata**
  - However, this sounds like feature creep

ECHO Operations Group (ECHO Ops) Support for Client Partners

Chao-Hsi Chang
Global Science and Technology
c-hchang@gst.com

Global Science & Technology, Inc.

ECHO

What is an ECHO Client Partner?

- **Client Partners participate in ECHO by developing a software application that communicates with ECHO to allow end users to access its metadata catalog and functionality;**
- **There are multiple ways for Clients to provide an end user interface to ECHO, e.g.**
  - Many clients will provide a graphical user interface (GUI) to assist users navigating and exploring ECHO for data/services;
  - Others may "harvest" ECHO metadata periodically and use them for serving other needs.

Overview of ECHO Ops Support for Client Partners

- **Client Partner Application and Setup**
- **Acclimation and Test Support**
- **Client Operations Support**
- **Partner Relations Management**

Client Partner Application and Setup

- **Client Partner Application**
  - Partner completes application form
  - Reviewed by ECHO Ops in conjunction with Project Managers to determine compatibility with ECHO scope and goals set forth by the ECHO Technical Committee (ETC)
- **Initial Setup**
  - ECHO Ops provides info and resources to facilitate client development
    - System requirements
    - ECHO User's Guide for Programmers
    - API documentation
    - Tools, e.g. XML Message Test Facility, *ECHO Façade* (Java toolkit)
    - Access to the ECHO test system
    - ECHO user accounts (if needed)
- **Operations Agreement (OA)**
  - ECHO Ops works with Client Partner to establish an OA based on the ESDIS ECHO template OA (currently being finalized)

Acclimation and Test Support

- **API and Metadata Support**
  Assist Client Partners in
  - Understanding ECHO business model and APIs
  - Interpretation of metadata DTD and valids
  - Gaining a clear understanding of ECHO capabilities (e.g. anticipated growth in metadata holdings and system functionality)
- **Test Support**
  - Maintain a catalog of representative metadata on the ECHO test system
  - Provide resources for Partner end-to-end testing
    - "Dummy" Providers
    - Canned XML scripts
- **Client Review**
  - Prior to final deployment (connection to operational ECHO system), ECHO Ops conducts a final check-out review to ensure readiness for operations

Client Operations Support

End users of ECHO Client applications are customers of Client Partner organizations <u>not</u> ECHO!

- **Client Partner organizations <u>must</u> provide their own user support services and maintain the Client application's operational environment.**
- **ECHO Ops will help to ensure successful Client operations by**
  - Establishing reasonable expectations for Client operations through up-to-date information on ECHO system status, metadata holdings, functionality, and performance;
  - Providing support for timely resolution of ECHO system problems reported by Client Partners;
  - Monitoring Client activity to ensure that requested services are provided as anticipated (e.g. orders are transmitted to Data Partners and filled as appropriate).

Partner Relations Management

ECHO Ops aims to achieve Partner satisfaction through

- **High Quality User Services**
  - Carefully tracking problems and issues reported by Partners
  - Monitoring time to resolve Partner issues
- **High Quality User Support Products**
  - Working with Partners to identify new/evolving requirements for ECHO user support
  - Conducting surveys of events, materials, and tools
- **System Availability and Performance Monitoring**
  - With feedback to ECHO Dev for HW/SW requirements planning and system evolution
- **Outreach Activities**
  - Promoting project visibility in NASA ESE and external communities
  - Promoting and successfully engaging new Data Partners
  - Advertising the availability and capabilities of public clients that are operational or under active development

Client Partner Resources – URLs

- **Project web site** http://www.echo.eos.nasa.gov
- **API documentation**
  http://api.echo.eos.nasa.gov/echo/message_detail.html
- **Operational system access points**
  - XML Message Test Facility:
    http://api.echo.eos.nasa.gov/echo/rmi/EchoTestFacility.jsp
  - Client access via SOAP (incl. PUMP):
    api.echo.eos.nasa.gov/soap/servlet/rpcrouter
- **Test system access points**
  - XML Message Test Facility:
    http://beamish.gsfc.nasa.gov:4800/echo/rmi/EchoTestFacility.jsp
  - Client access via SOAP (incl. PUMP):
    beamish.gsfc.nasa.gov:4800/soap/servlet/rpcrouter

Client Partner Resources – Documents and Tools

- **ECHO Documentation is maintained online at**
  http://www.echo.eos.nasa.gov/echo-docs.shtml
  - ECHO 5.0 Features/Functionalities *(coming soon ECHO 5.0.1)*
  - ECHO API Changes between Version 4.5 and 5.0
  - ECHO 5.0 User's Guide
  - ECHO DTD Tag Directory
  - ECHO Acronym List
- **ECHO Connectors**
  - Perl SOAP Proxy: http://www.echo.eos.nasa.gov/documents/PERLProxy/
  - Python SOAP Proxy: http://www.echo.eos.nasa.gov/documents/PythonProxy/
- **ECHO Façade (Java toolkit) is available at:**
  - Javadocs: http://www.echo.eos.nasa.gov/echo-toolkit/i_4_8_0_b6/docs/api/
  - Library: http://www.echo.eos.nasa.gov/echo-toolkit/i_4_8_0_b6/lib/client_toolkit.jar
  - Source: http://www.echo.eos.nasa.gov/echo-toolkit/i_4_8_0_b6/src/

ECHO Ops – Current Priorities (11/2003)

- **Complete deployment of *ECHO Scout***
  - Monitors overall ECHO system availability to end users in real-time on both the operational and test systems;
  - Alert function notifies ECHO Ops and logs errors with preliminary diagnostic information.
- **Publish summary of current holdings and data sets currently scheduled for ingest with anticipated date of public release**
- **Assist in ECHO Version 5.0.1 transition to operations**

ECHO Ops Contact Info

- **Need help from ECHO Ops? Contact us by email at** echo-ops@killians.gsfc.nasa.gov **-- we'll make sure you get a human to assist you ASAP!**
- **Key ECHO Ops members include**

Medora Macie	GSFC	NASA Project Manager	301-614-6812
Beth Weinstein	GSFC	NASA Project Manager	301-614-5318
Jackie Kendall	SSAI	Task Lead / Coordinator	301-867-2026
Chao-Hsi Chang	GST	Partner Coordinator	301-867-2059
Alex Lai	GST	API Support	301-867-2061
Frank Corprew	GST	Ingest Manager	301-867-2058
Claudia Castaneda	GST	Database Administrator	301-867-2124