ERDAS APOLLO
User Guide

December 2010

erdas
The Earth to Business Company

Printed in the United States of America.

Manager, Technical Documentation
ERDAS, Inc.
5051 Peachtree Corners Circle
Suite 100
Norcross, GA 30092-2500 USA.

The information contained in this document is subject to change without notice.

Table of Contents

List of Figures

List of Tables

Introduction

In this chapter:

What is Covered in This Guide

The ERDAS APOLLO system provides a web client so that external users can search, view, and download the data in the catalog that has been made accessible to them.

You can also interact with the ERDAS APOLLO catalog directly using a Web Coverage Service (WCS) interface, a Web Map Service (WMS) interface, the Web Feature Service (WFS) interface, and the Web Registry Service (WRS) interface. All of these interfaces are compliant with the standards established by the Open Geospatial Consortium (OGC).

Audience

This guide is intended for people who will be using the ERDAS APOLLO web client or the OGC-compliant interfaces to interact with the ERDAS APOLLO Catalog.

Additional Help

- The ***ERDAS APOLLO Concepts Guide*** explains the terminology and technology that you need to know in order to understand the ERDAS APOLLO system. ERDAS recommends that you review this guide before trying to install or configure the product.
- The ***ERDAS APOLLO QuickStart Guide*** provides instructions for installing a basic, working ERDAS APOLLO system, as well as instructions for upgrading or replacing an older system.
- The ***ERDAS APOLLO Data Manager Guide*** provides instructions for the ERDAS APOLLO Data Manager, which is the ERDAS APOLLO system component that you will be using to create and manage your imagery catalog and/or service providers.
- The ***ERDAS APOLLO User Guide*** gives instructions on using the ERDAS APOLLO Web Client and instructions on using HTTP requests to interact with your ERDAS APOLLO Server.

- The ERDAS APOLLO Administrator’s Guides contain instructions for advanced installation, configuration, and optimization of the server component for ERDAS APOLLO. There are two books that cover this material:

  - ERDAS APOLLO Administrator’s Guide (Advanced/Professional edition) provides information about the features that are unique to that level of the product, such as:

    - The ERDAS APOLLO Imagery Catalog
    - Clip, Zip, and Ship image downloading
    - Geoprocessing (WPS)

  - ERDAS APOLLO Administrator’s Guide (Essentials-SDI edition) provides information about how to create nd configure OGC-compliant service providers, how to use the different tools and utilities available in ERDAS APOLLO to help you work with your data, and how to use the ERDAS APOLLO Style Editor.

ERDAS APOLLO Web Client

In this chapter:

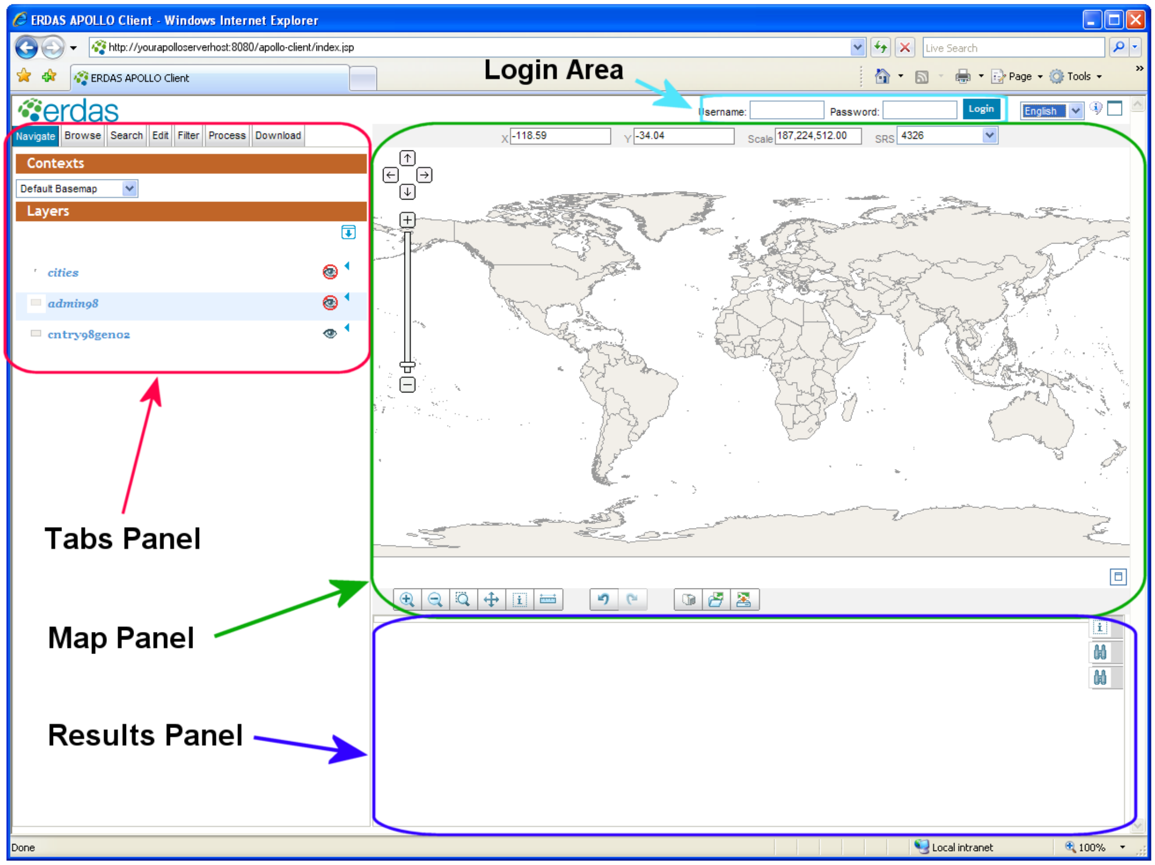
- Introduction to the Web Client
- The Map Panel
- The Tabs Panel
- The Results Panel

Introduction to the Web Client

The ERDAS APOLLO web client is the part of the ERDAS APOLLO system that allows you to make your imagery and data available to all of the different people who need it. The web client was installed and published when you installed the ERDAS APOLLO Server software on your server computer and can be accessed over the internet.

The default URL of the ERDAS APOLLO Web Client is http://*localhost*:8080/apollo-client/.

Figure 1: ERDAS APOLLO Web Client



ERDAS APOLLO Client - Windows Internet Explorer

http://yourapolloserverhost:8080/apollo-client/index.jsp

X -118.59 Y -34.04 Scale 187,224,512.00 SRS 4326

Contexts: Default Basemap

Layers: cities, admin98, cntry98gen02

Overview of the Layout

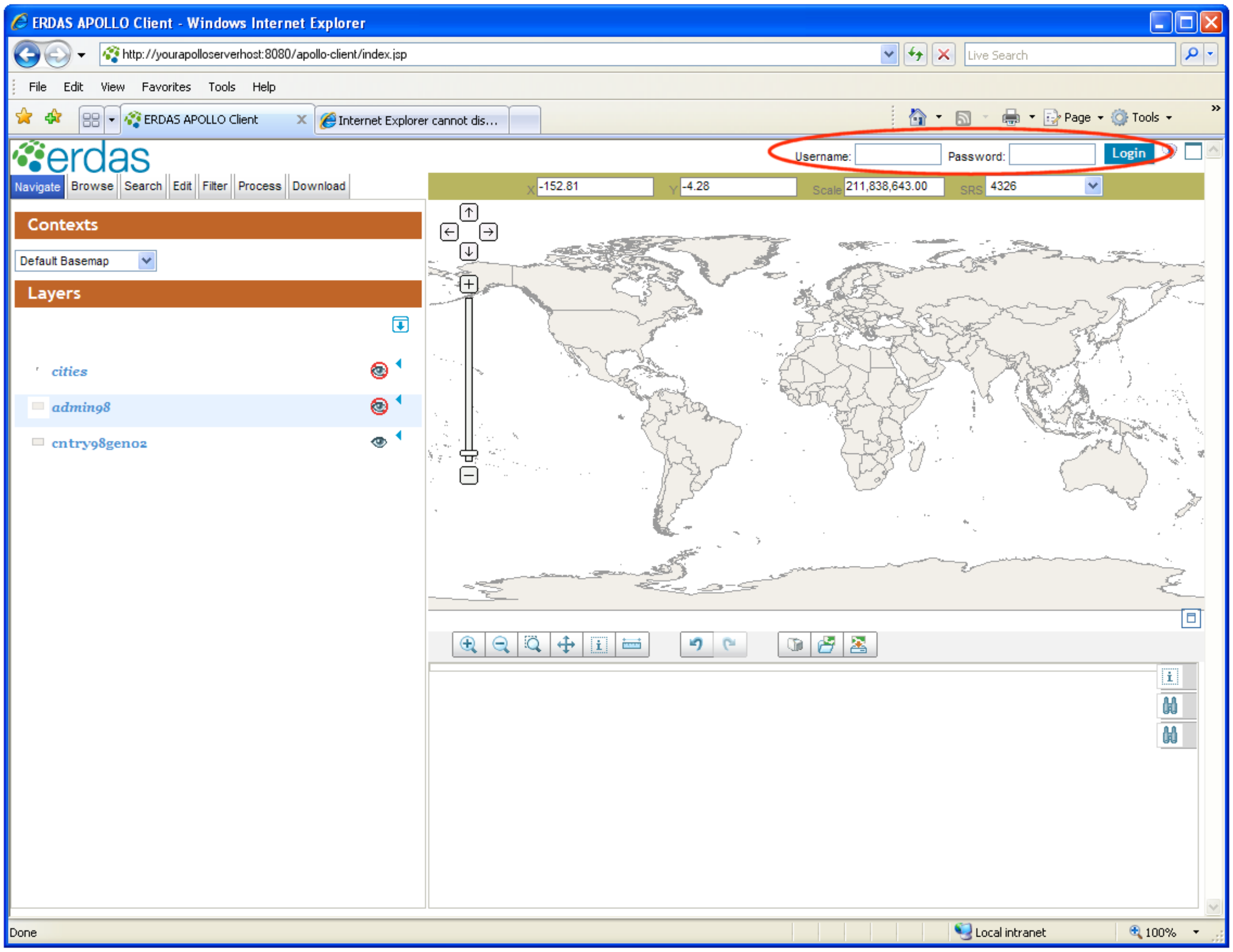
When you first open the ERDAS APOLLO Web Client, you will notice the following general parts:

- **The Login Area**
- **The Map Panel**
- **The Tabs Panel**
- **The Results Panel**

The Login Area

On the top right corner of the page you will see the **Username** and **Password** fields and the Login button, which allow users with an account to log in to the ERDAS APOLLO Web Client.

You can use the ERDAS APOLLO web client without logging in to an account, but you will only have access to public data.



http://yourapolloserverhost:8080/apollo-client/index.jsp

X -152.81 Y -4.28 Scale 211,838,643.00 SRS 4326

Default Basemap

cities
admin98
cntry98geno2

The Map Panel

Just under the username and password fields, you will see the map panel. The map panel allows you to see a graphical representation of georeferenced data.

The Base Map

When you first open a standard installation of the ERDAS APOLLO Web Client, the map panel will show a base map of the Earth.

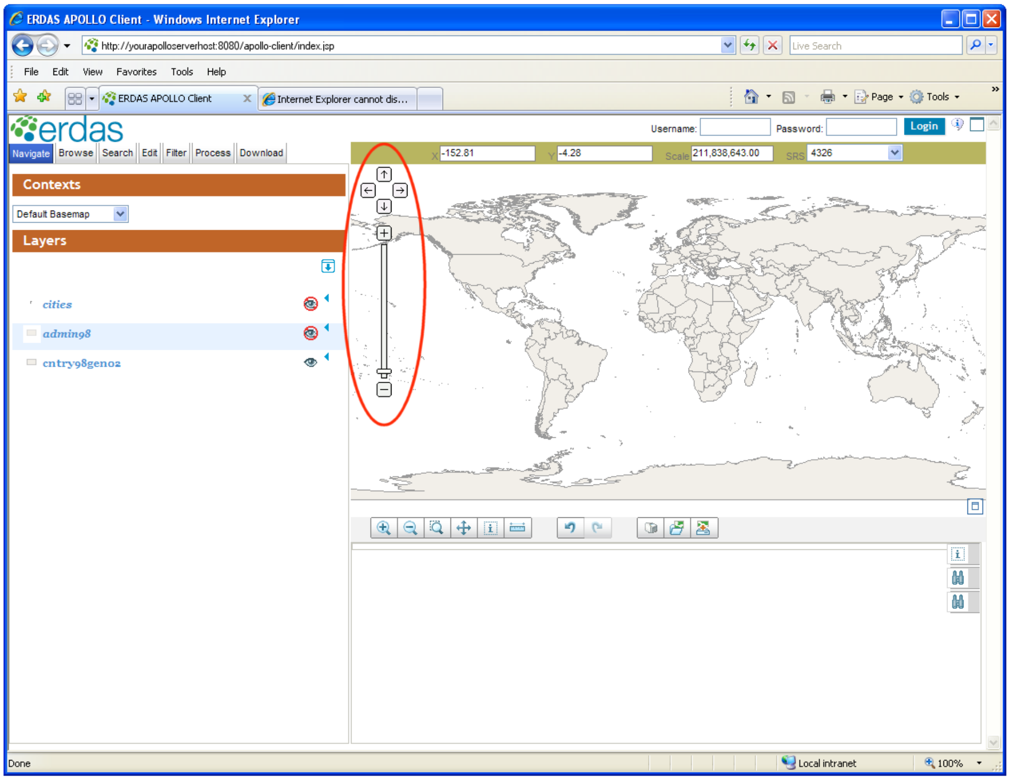
You can view your data as layers on top of your base map in order to get an idea of the geographic location of the data.

It is possible to change the base map that is used if you want. See the ERDAS APOLLO 2010 Admin Guide for instructions.

The QuickAccess Zoom and Pan Tools

The quick access zoom and pan tools are located on the left side of the map panel. You can use the slider bar to quickly zoom in and out on the map. You can use the arrow cross to quickly pan to a different area on the map.

Figure 2: Quick Access Zoom and Pan Tools



http://yourapolloserverhost:8080/apollo-client/index.jsp

X -152.81 Y -4.28 Scale 211,838,643.00 SRS 4326

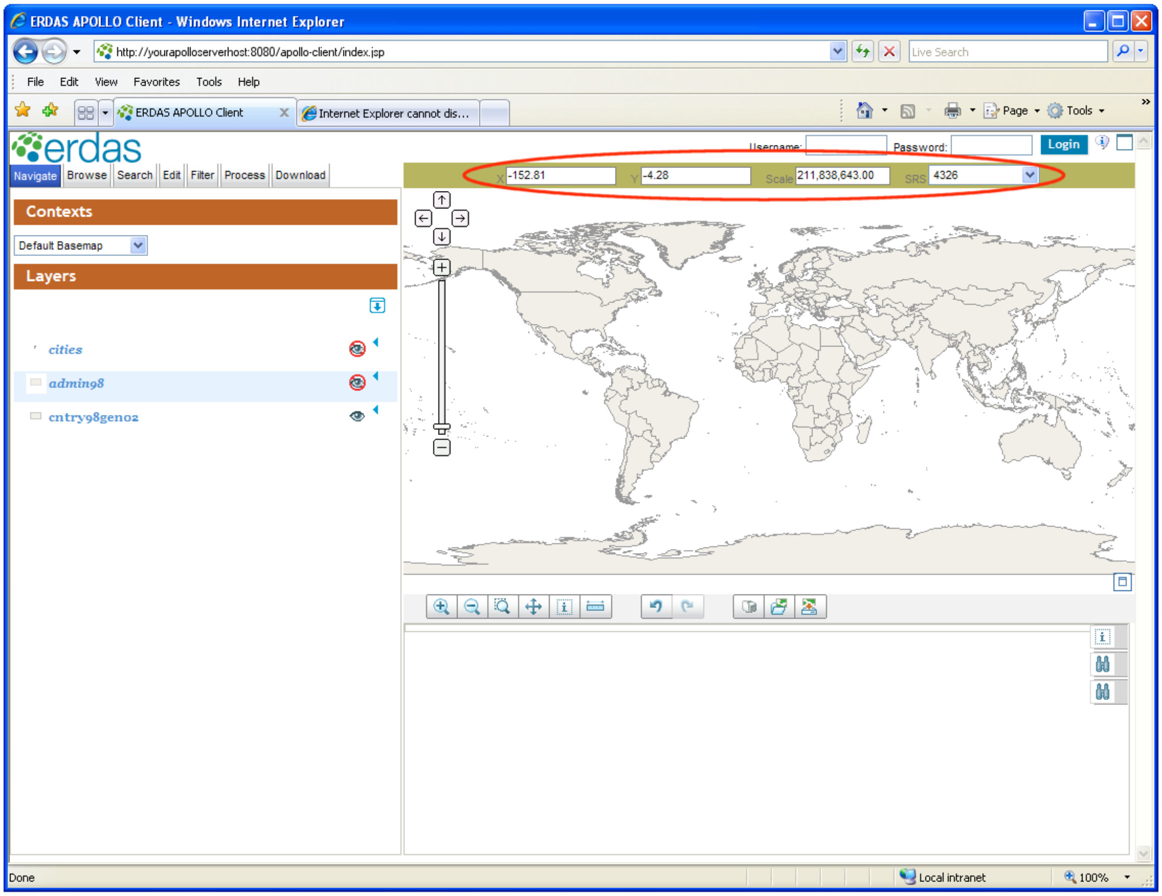
Contexts: Default Basemap

The Info Bar

The info bar is located along the top of the map panel and indicates some basic information about the map such as:

- the **X** and **Y** coordinates of the location of your cursor on the map.
- the **Scale** at which you are viewing the map.
- the **SRS** in which you are currently viewing the map.

Figure 3: The Info Bar on the Map Panel



X -152.81 Y -4.28 Scale 211,838,643.00 SRS 4326

X and Y Coordinates

By pointing your cursor at the map,. you can see the x and y coordinates of that location on the map.

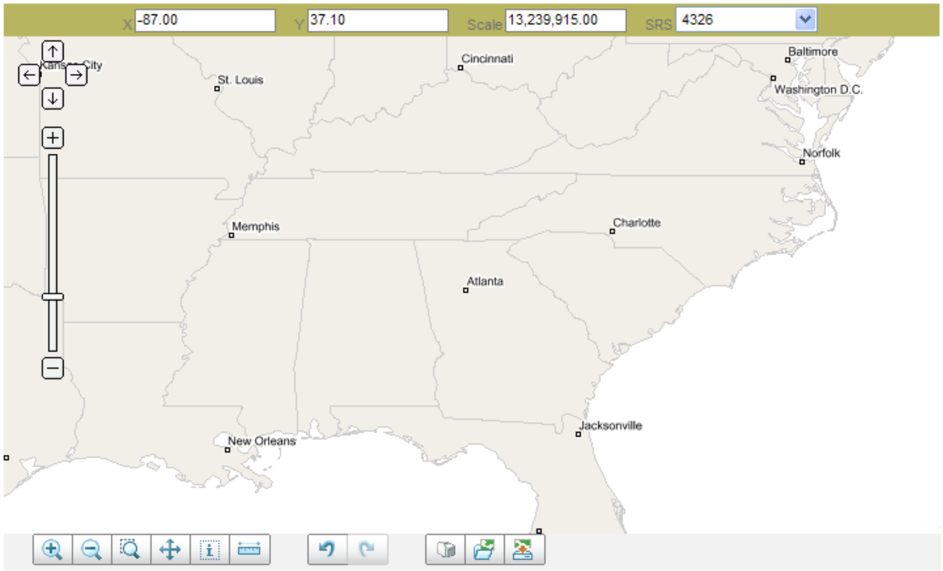
Scale

The number in the scale field indicates how many units on the actual Earth are equivalent to one unit on this map. The actual units used do not matter, so long as they are the same for the map and the Earth.

For example, in the map shown below, 1 unit on the map is equivalent to 13, 239, 915 units on the actual Earth. So, any of the following could be true:

> 1 ***inch*** on the map = 13, 239, 915 ***inches*** on the Earth
>
> 1 ***centimeter*** on the map = 13, 239, 915 ***centimeters*** on the Earth
>
> 1 ***foot*** on the map = 13, 239, 915 ***feet*** on the Earth

You can perform conversions to make a meaningful scale if you want. For example, by converting inches to the equivalent number of miles, you could change the first example given above to
1 ***inch*** on the map = 208.96 ***miles*** on the Earth



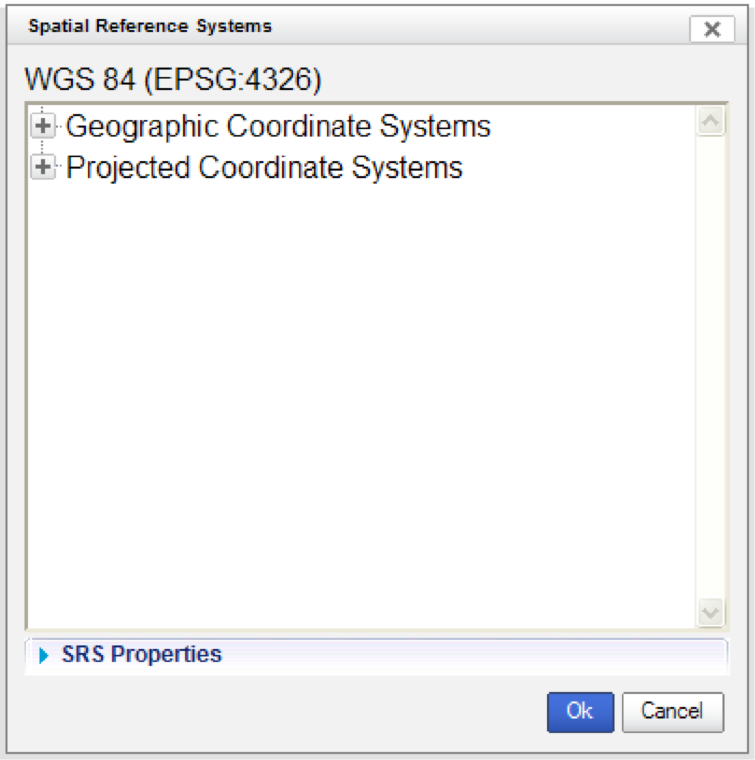
X -87.00 Y 37.10 Scale 13,239,915.00 SRS 4326

SRS

The SRS field indicates the Spatial Reference System in which the map is currently projected. For example, in the image above, it is shown in EPSG:4326.

There are two ways to change the SRS. If you know the EPSG code of the SRS you want to use, click the down arrow in the field and select **Enter EPSG Code**. A box will appear and you can type the number in the box and click OK. That map will then be reprojected in that SRS.

You can also click the down arrow in the field and select **Select SRS**. The Spatial Reference Systems dialog box opens and allows you to select the SRS that you want to use. After you select it and click **OK** in the dialog, the map will reproject using the SRS that you selected.

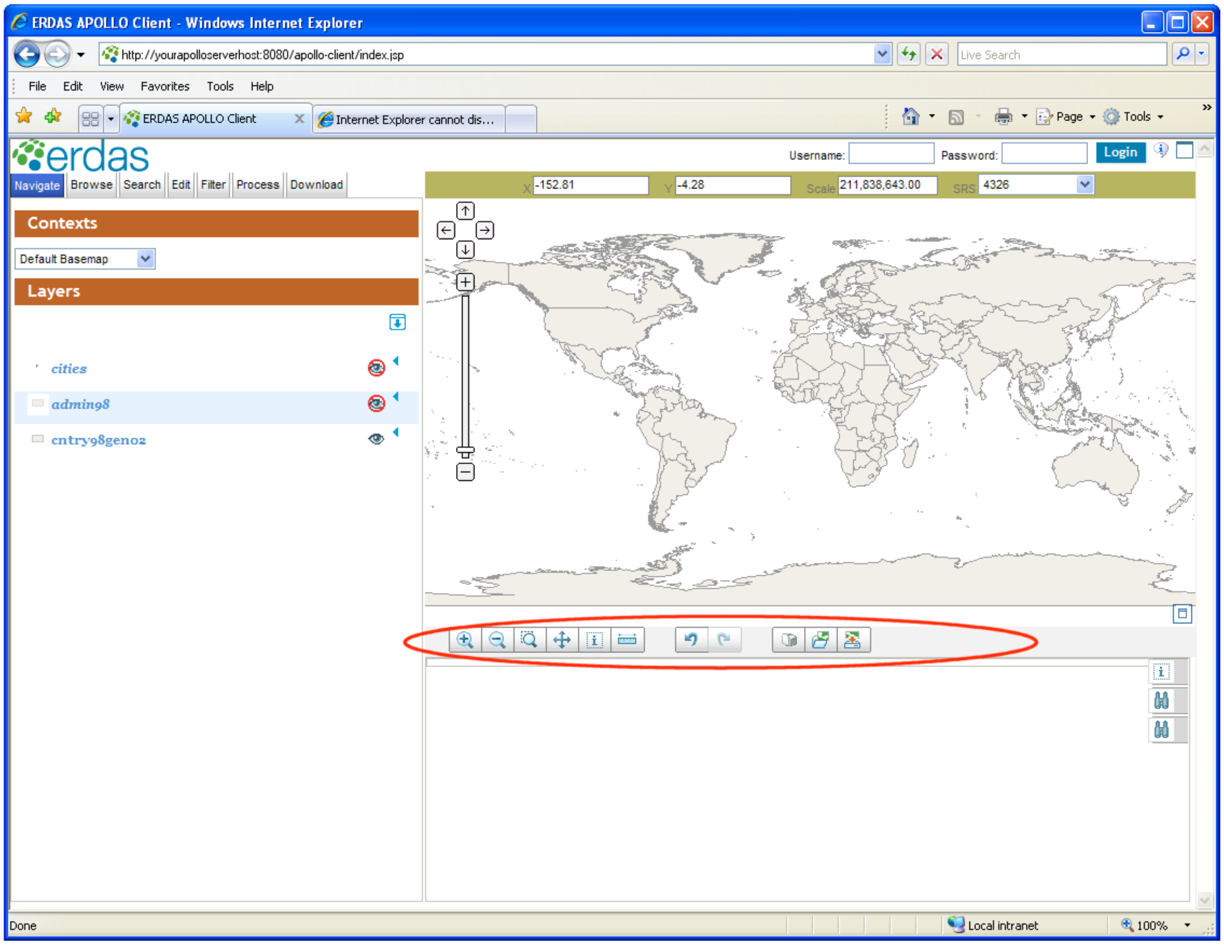

Spatial Reference Systems
WGS 84 (EPSG:4326)
Geographic Coordinate Systems
Projected Coordinate Systems
SRS Properties
Ok
Cancel

Basemap

Use the Basemap dropdown to display a basemap. The default is Bing Roads.

The Map Toolbar

There is a toolbar along the bottom of the map panel that allows you to zoom in and out, view details about the layer, measure the distance between two points on the map, print the map, and work with map contexts.


ERDAS APOLLO Client - Windows Internet Explorer
http://yourapolloserverhost:8080/apollo-client/index.jsp
Live Search
File Edit View Favorites Tools Help
ERDAS APOLLO Client
Internet Explorer cannot dis...
Page
Tools
erdas
Username:
Password:
Login
Navigate Browse Search Edit Filter Process Download
Contexts
Default Basemap
Layers
cities
admin98
cntry98geno2
Done
Local intranet
100%

X -152.81 Y -4.28 Scale 211,838,643.00 SRS 4326

The toolbar provides access to the following tools:

- **Zoom In**
  Allows you to zoom in to see the map in more detail. If you click the button and then click the map, the scale will decrease by about half. If you click the button and then draw a box on the map, the map will zoom in until the box fills the viewing area of the map panel.

- **Zoom Out**
  Allows you to zoom out to see a larger extent of the map. If you click the button and then click the map, the scale will increase by about half.

- **Interactive Zoom**
  Allows you to drag the mouse to zoom in and out.

- **Pan**
  Allows you to drag the mouse to move the entire map so that a different section of it appears in the viewing area of the map panel.

- **Layer Info**
  Allows you to see details about the layers shown in the map panel.

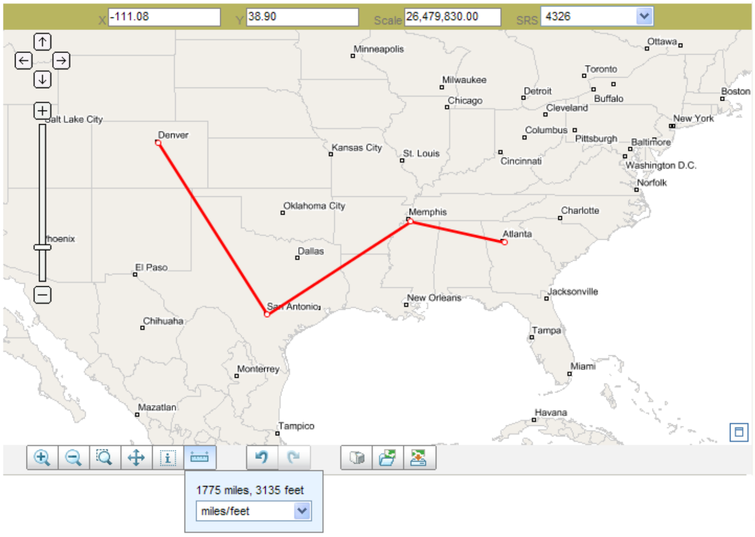
  To use this tool, click the Layer Info button, then click the map or draw a box on the map. The Layer Info tab of the Results panel will then show the details for any of the layers in the map panel whose extents contain that point or intersect with that box.

  You can restrict the layers that appear in the Results panel by opening the Navigate tab in the Tabs Panel and clicking the **Select Layer** icon for the layers you want to view in the Results panel. If no layers are selected, the map will show all of them.

- **Measure**
  Allows you to measure the distance between points on the map.

  To use this tool, click the button. Select the unit of measurement in the box that appears. Then, place your points on the map by clicking on it.

Figure 4: The Measure Tool on the Map Panel.



X -111.08 Y 38.90 Scale 26,479,830.00 SRS 4326

1775 miles, 3135 feet

miles/feet

- **Back**
  Allows you to undo the previous 10 changes to the map.

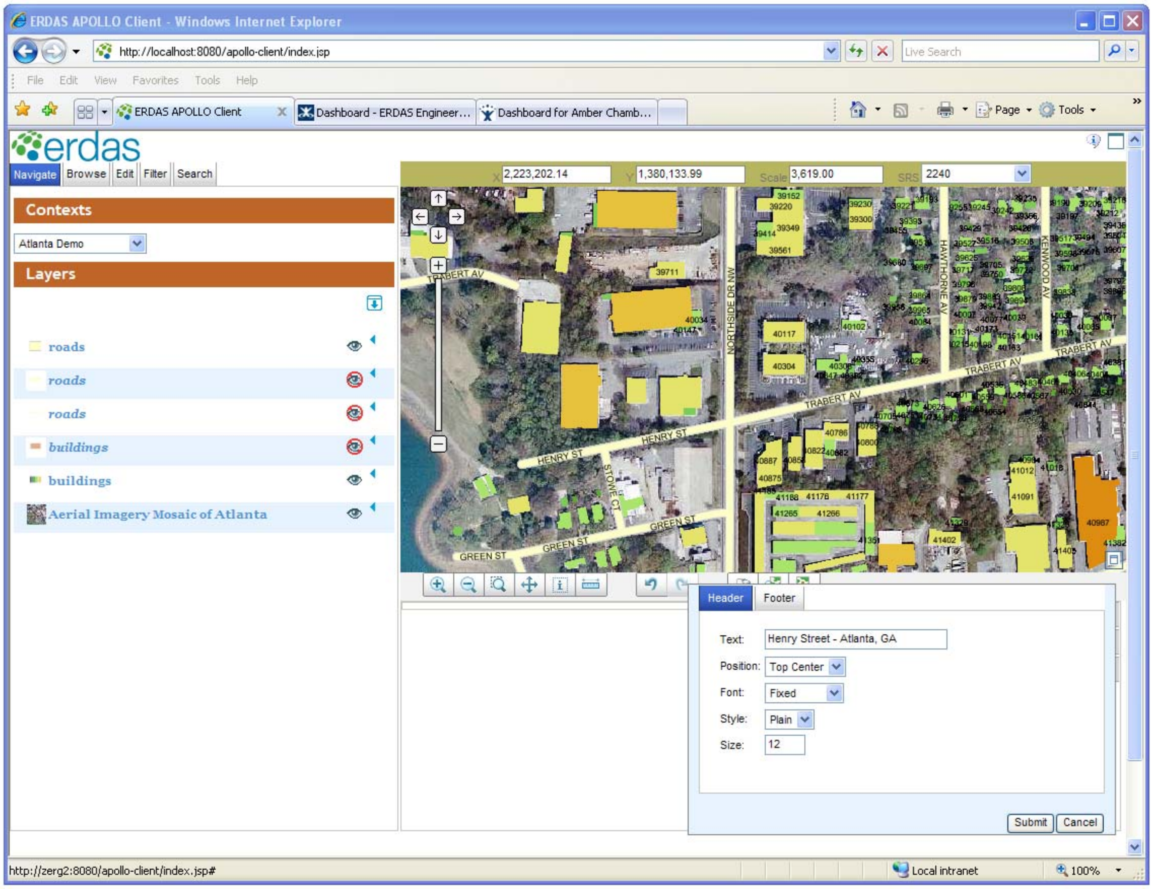
- **Forward**
  Allows you to redo the previous 10 changes to the map that you had undone.

- **Print**
  Places the contents of the viewable area of the map panel in a printer-ready format.

  To use the print tool, click the button. A box with header and footer settings for the printed document will appear. When you finish selecting the header and footer options, click the Submit button. A separate window will open that contains a printable view of the map. You can click the Print button in this window to send the map to your printer.

The Advantage/Professional product tier does not support printing.

Figure 5: The Header and Footer Setup Boxes for the Print Map Tool.


ERDAS APOLLO Client - Windows Internet Explorer
http://localhost:8080/apollo-client/index.jsp
File Edit View Favorites Tools Help
ERDAS APOLLO Client
Dashboard - ERDAS Engineer...
Dashboard for Amber Chamb...
Page
Tools
erdas
Navigate
Browse
Edit
Filter
Search
Contexts
Layers
Header
Footer
Submit
Cancel
http://zerg2:8080/apollo-client/index.jsp#
Local intranet
100%

Contexts: Atlanta Demo

Layers:
- roads
- roads
- roads
- buildings
- buildings
- Aerial Imagery Mosaic of Atlanta

X 2,223,202.14 Y 1,380,133.99 Scale 3,619.00 SRS 2240

Text: Henry Street - Atlanta, GA

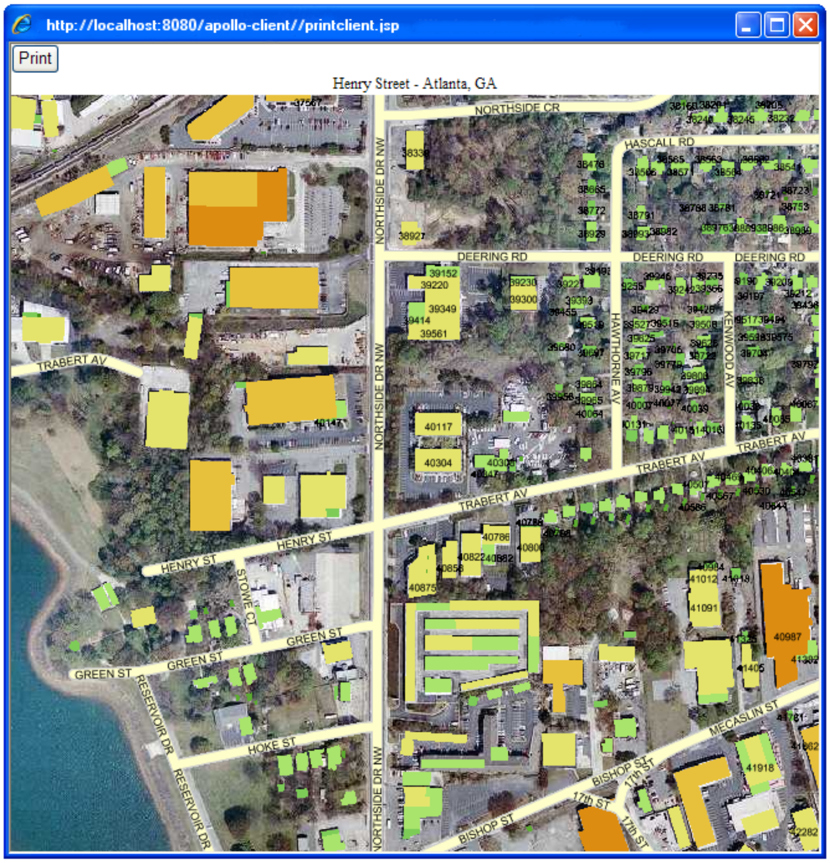
Position: Top Center

Font: Fixed

Style: Plain

Size: 12

Figure 6: The Printer-Ready Version of Your Map



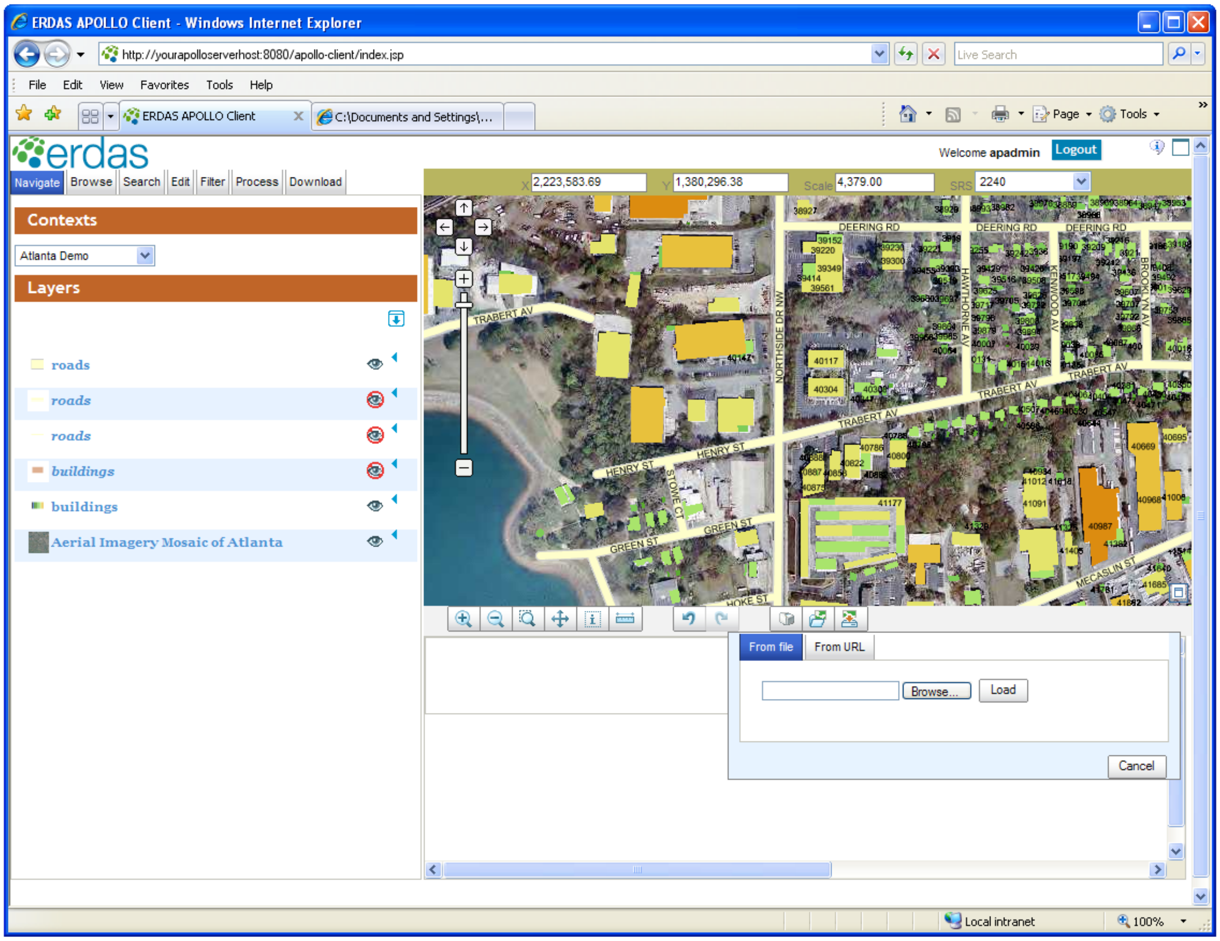
- **Load Context**
  Allows you to view a map that has been described in a Web Map Context (WMC) file.

  To load the context, click the button. A box will appear so that you can provide the location of the context file.

  If you are loading the context directly from the context file, open the **From File** tab on the box. Use the **Browse** button to find the file, and click the **Load** button to load it in the map panel.

  If you are loading the context from a service provider that has access to the context file, open the **From URL** tab on the box. Type or paste the URL into the box and click the **Load** button to load it in the map panel.

Figure 7: Loading a Context into the Map Panel by Accessing the File Directly.



ERDAS APOLLO Client - Windows Internet Explorer

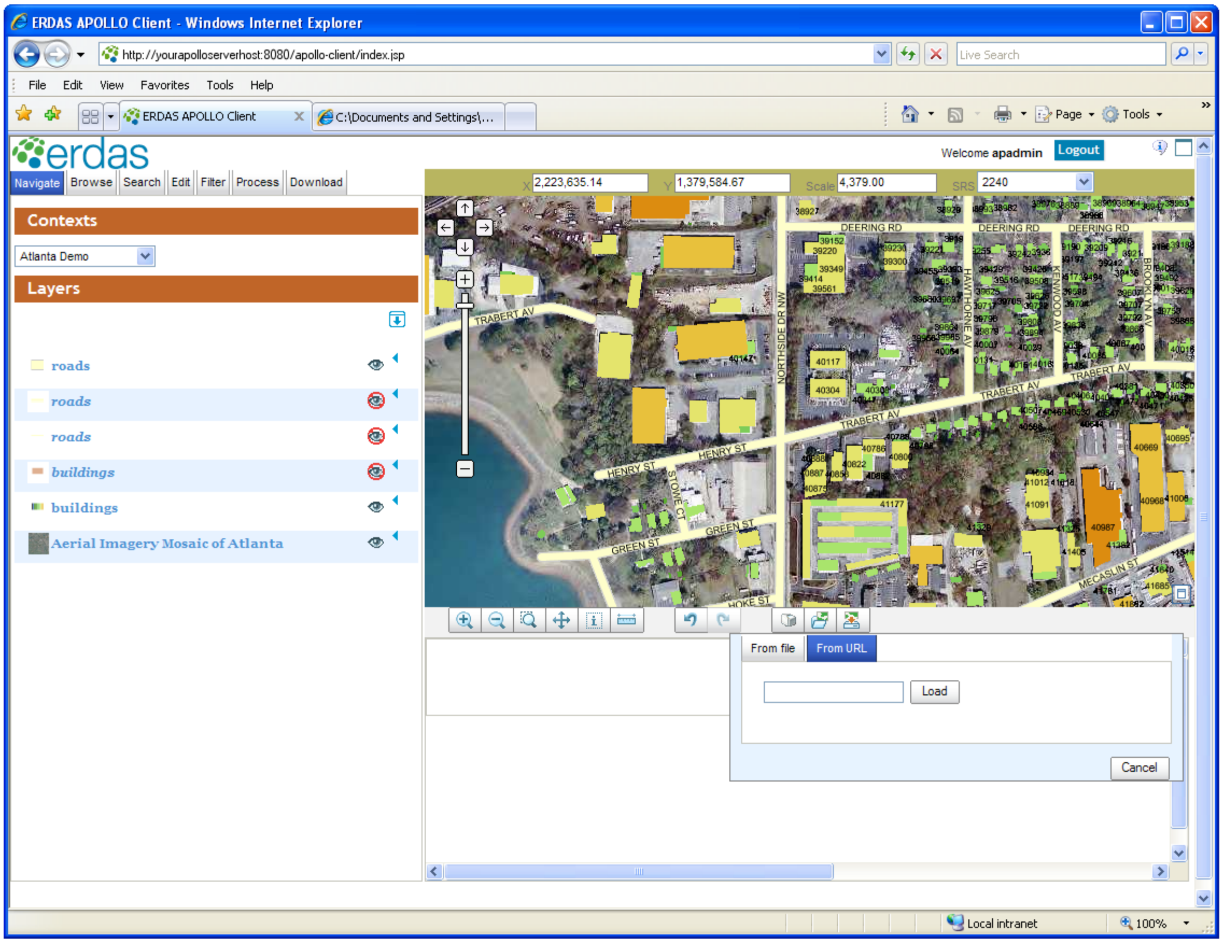
http://yourapolloserverhost:8080/apollo-client/index.jsp

X 2,223,583.69 Y 1,380,296.38 Scale 4,379.00 SRS 2240

Contexts: Atlanta Demo

Layers:
- roads
- roads
- roads
- buildings
- buildings
- Aerial Imagery Mosaic of Atlanta

Figure 8: Loading a Context into the Map Panel by Pointing to It with a URL



X: 2,223,635.14
Y: 1,379,584.67
Scale: 4,379.00
SRS: 2240

- **Save Context**
  Allows you to save the description of the map in a Web Map Context file.

The Tabs Panel

The panel to the far left of the web client page has tabs that contain all of the controls for the web client. The tabs that you see in this panel will differ slightly depending on which edition of ERDAS APOLLO you are using.

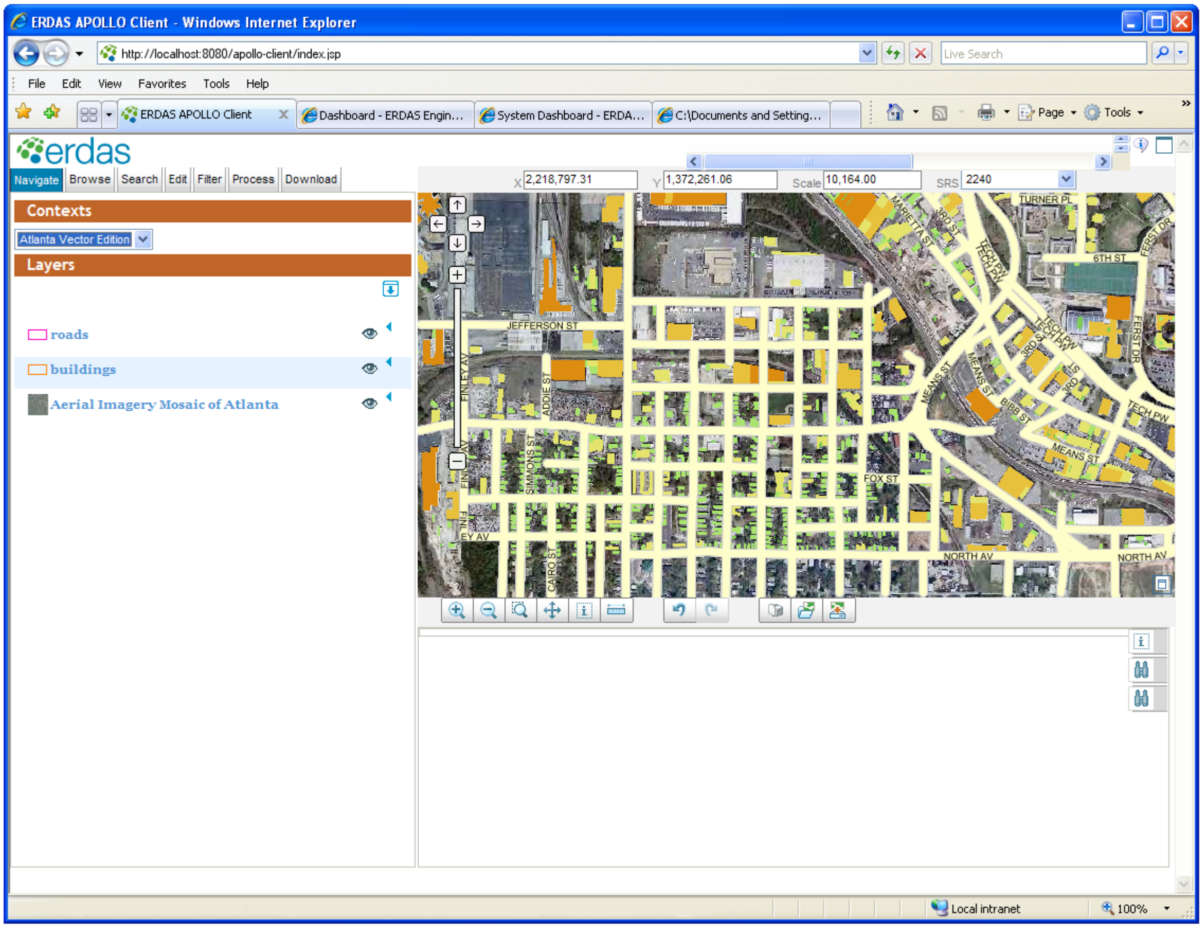
These tabs will be shown in any product level of the ERDAS APOLLO Web Client:

- **Navigate**
- **Browse**
- **Search**
- **Edit**
- **Filter**
- **Download** (available in the Advantage and Professional product levels)
- **Process** (available in the Professional product level)

The Navigate Tab

The Navigate tab allows you to see all of the contexts that you have access to and all of the layers that you can view on the map to the right of the tab area of the page.

Figure 9: The Navigate Tab



ERDAS APOLLO Client - Windows Internet Explorer

http://localhost:8080/apollo-client/index.jsp

X 2,218,797.31 Y 1,372,261.06 Scale 10,164.00 SRS 2240

Contexts: Atlanta Vector Edition

Layers:

- roads
- buildings
- Aerial Imagery Mosaic of Atlanta

Contexts

A Web Map Context is an XML document that stores details about a map (such as the layers that make up the map) and the configuration information (such as styles and scale thresholds) for each of the layers. The ERDAS APOLLO Web Client can read a Web Map Context file and create the map it describes in the map panel.

Some common reasons for using a Web Map Context file in the Web Client are:

- you want to view a map that is composed of layers that are located on different servers.

- you have a certain map that you like to use and you want the ability to open it quickly and easily.
- you want to send your map to someone else.

To use a Web Map Context file, select it in the Contexts drop down box. When you set up your ERDAS APOLLO system, you will need to specify the contexts that are available in this list. For instructions, see the section “Web Client Configuration and Customization” in the ERDAS APOLLO Administrator’s Guide (Advantage/Professional Edition).

You can have a Context file and a basemap. Not all Context files have a basemap. If you Save a Context while a Basemap is selected, the Basemap is saved to the Context

Layers

The Layers area shows you all of the layers that you can view in your map. The base map layers will appear in this section by default.

In the row for each layer, you will see the **availability/visibility** icon (a picture of an eye).

The **availability** is determined by the scale at which you are viewing the map. If you are viewing the map at a scale that is within the viewable scale range for that layer, the layer will be available.

The viewable scale range for a layer is defined at the server level, but may be further restricted by the user who is currently logged in. Layers that are not currently available to be shown on the will have a red circle with a line over the eye.

If a layer is available, you can toggle its **visibility** off and on by clicking on the eye icon. Visible layers are indicated with an open eye, invisible layers with a closed eye.

To see the options for a layer, expand the row by clicking:

- the ◀ icon on the right side of the row for the layer

- the icon on the top right corner of the layers area to expand all of the row for the layers.

After the row for a layer is expanded, you will see the following icons:

- **Zoom to Extent**
  allows closer viewing of the layer. The map will zoom in so that the entire extent of the layers fills the map viewing area.

- **Select Layer** □ (unselected) ☑ (selected)
  allows you to restrict the layer information that is shown in the Results Panel when you use the **Layer Info** map tool.

- **View Metadata**
  allows you to view the metadata for the layer. When you click this icon, the metadata information will appear in a separate window.

- **Layer Opacity**
  sets the opacity of the layer. 1 specifies that the layer is completely opaque, 0 specifies that it is completely transparent, and you can set the slider control to any value in between 0 and 1.

- **Configure Layer**
  allows you to set the options for the layer.

- **Delete Layer**
  removes the layer from the list of layers.

Configuring a Layer

To configure a layer, click the **Configure Layer** icon for that layer to open the Configure box. The Configure box contains the following tabs:

General

The General tab allows you to read/edit the basic information for the layer. This tab contains the fields:

- Title - the title of the layer (or feature type, in the case of WFS). This should be a user-friendly label for the user and will be shown in the GetCapabilities document provided by the service provider.

- **Service URL** - the URL of the service provider that serves this layer.

- **Service Type**- the service provider that serves this layer.

- **Layer Name -** the name of the layer (or feature type, in the case of WFS).

- **Namespace -** the name of the feature type (only available for WFS-based layers).

- **Image Format -** the format that the Web Client will use to show the image in the map panel.

  The web client automatically selects the image format that should lead to the best image quality for your internet browser and for the style that is being used. For example, if you are using a style that causes an object in the layer to be partially transparent, the Web Client would select the PNG image format rather than the GIF format. The PNG format is capable of showing different levels of transparency while the GIF format can only show objects as completely opaque or completely transparent.

Style

The Style tab shows the styles that are available for the layer. It also allows you to add a new style, either by creating it with the Web Client styling tools or by uploading an OGC-compliant Styled Layer Descriptor (SLD) file.

If there is a style named “default” in the list of available styles, the Web Client will automatically use that style any time you view this layer in the map panel. If there is not a style named “default” in the list, the Web Client will automatically use the style at the top of the list any time you view this layer in the map panel.

To add a new style for the layer by uploading an existing SLD file:

1. Type a name for the style in the **Name** field.

2. Click the **Browse** button to open the Choose File dialog so you can find and select the SLD file.

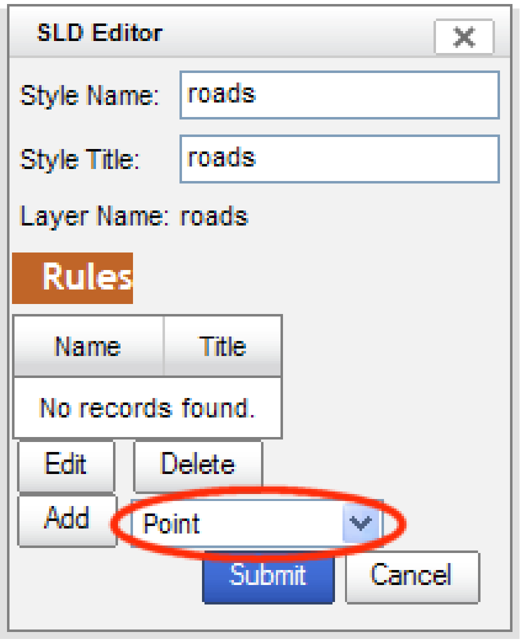
3. Click the **Upload** button.

The new style will now appear in the Available Styles list.

To create a new style using the Web Client tools:

When you create a new style in the Web Client, you will first set the name and title for the style, and then you will create one or more rules for it

1. Click the **New** button below the Available Styles list.
   The SLD Editor dialog box opens.



SLD Editor

Style Name: roads

Style Title: roads

Layer Name: roads

2. The **Style Name** and **Style Title** fields will be set automatically to the same name as the layer for which you are creating the style. You can type a different style name and style title if you want, however.

3. Select the shape of the layer objects whose appearance you want to define (use the field shown inside the red circle in the figure above).
   For example, if you want to define the appearance of roads in the layer, select *Line*.

4. Click **Add**.
   The Rule dialog box opens. This dialog box will contain different appearance settings depending on the shape you selected in the previous step.

   The dialog box always contains a **Rule Information** section where you can specify the name, title, and abstract for the rule.

It almost always contains a **Filter** section, which allows you specify that the style rule should only be applied to certain shapes within the layer. For example, if the layer contains roads and there is a property that indicates how many lanes a road has, you could configure the style to be only applied to the roads with four lanes.

5. When you are satisfied with the options you have selected for the rule, click the **OK** button in the Rule dialog box. The new rule appears in the Rules table.

6. You can create multiple rules for a style. For example, you might want to create one rule that defines how the four-lane roads look, and another that defines how the two-lane roads should look.
   To add another rule, follow steps 3 and 4 again.

   Use the **Edit** button to change a rule you have already created.
   Use the **Delete** button to remove a rule.

7. Click the **Submit** button on the SLD Editor dialog box to save the entire style.

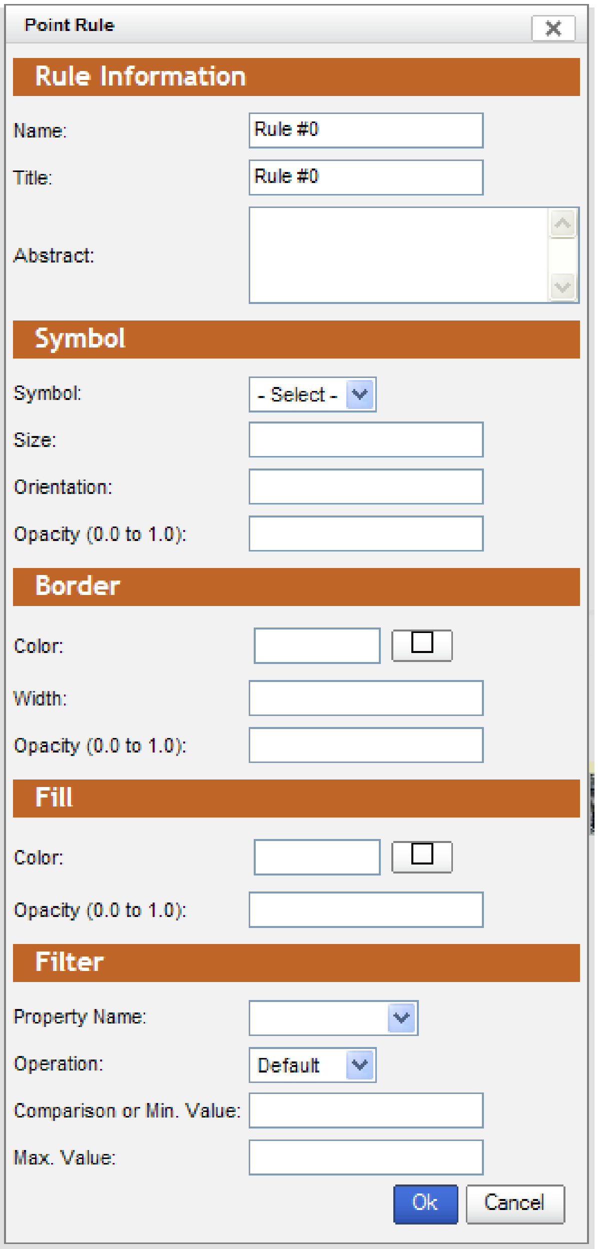
The new style now appears in the Available Styles list and the Web Client will automatically select it. To the right of any style that you created using the Web Client Tools, there is an **Edit** link and a **Download** link.

To edit the style, click the **Edit** link. The SLD Editor dialog box will open so you can change the name or title of the style or the details of the individual rules that make up the style.

To download a style as a Styled Layer Descriptor (SLD) file, click the download link.

The following figures illustrate the different options that you may see in the Rule dialog box.

Figure 10: Point Rule Dialog Box



Point Rule

Rule Information

Name: Rule #0

Title: Rule #0

Abstract:

Symbol

Symbol: - Select -

Size:

Orientation:

Opacity (0.0 to 1.0):

Border

Color:

Width:

Opacity (0.0 to 1.0):

Fill

Color:

Opacity (0.0 to 1.0):

Filter

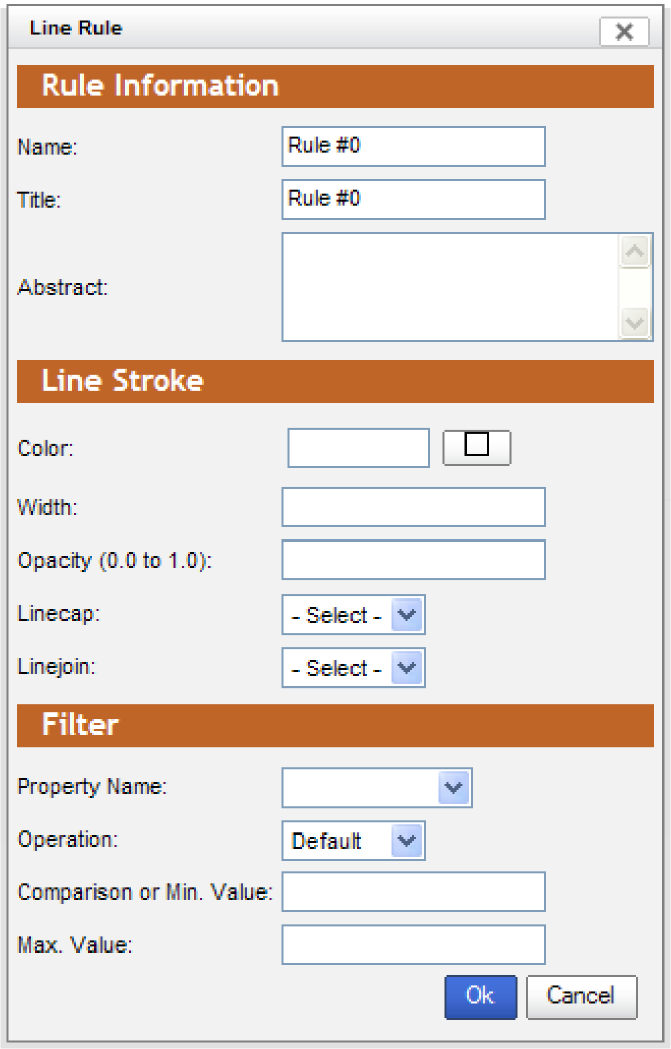
Property Name:

Operation: Default

Comparison or Min. Value:

Max. Value:

Figure 11: Line Rule Dialog Box



Line Rule

Rule Information

Name: Rule #0

Title: Rule #0

Abstract:

Line Stroke

Color:

Width:

Opacity (0.0 to 1.0):

Linecap: - Select -

Linejoin: - Select -

Filter

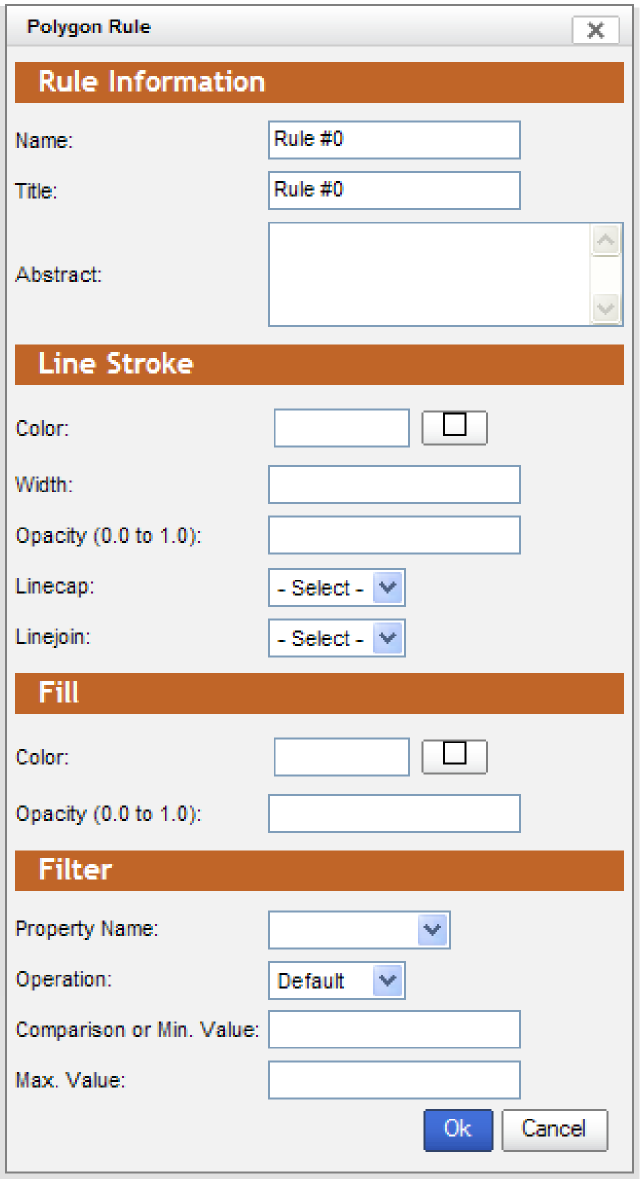
Property Name:

Operation: Default

Comparison or Min. Value:

Max. Value:

Figure 12: Polygon Rule Dialog Box



Polygon Rule

Rule Information

Name: Rule #0

Title: Rule #0

Abstract:

Line Stroke

Color:

Width:

Opacity (0.0 to 1.0):

Linecap: - Select -

Linejoin: - Select -

Fill

Color:

Opacity (0.0 to 1.0):

Filter

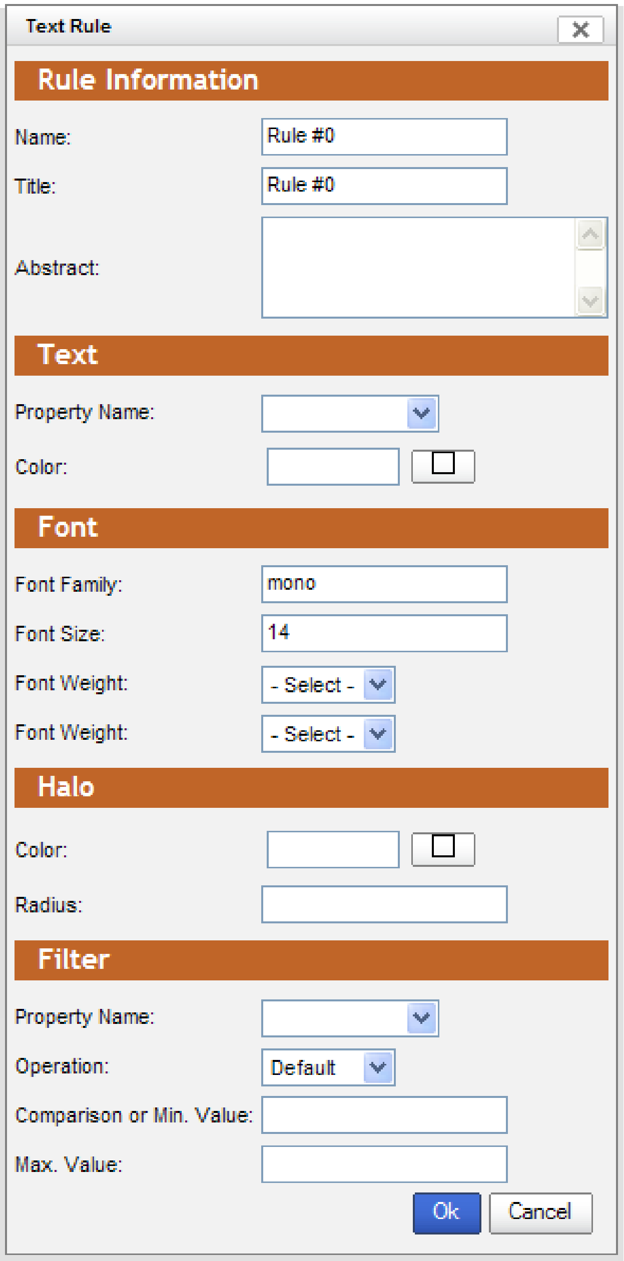
Property Name:

Operation: Default

Comparison or Min. Value:

Max. Value:

Figure 13: Text Rule Dialog Box



Text Rule

Rule Information

Name: Rule #0

Title: Rule #0

Abstract:

Text

Property Name:

Color:

Font

Font Family: mono

Font Size: 14

Font Weight: - Select -

Font Weight: - Select -

Halo

Color:

Radius:

Filter

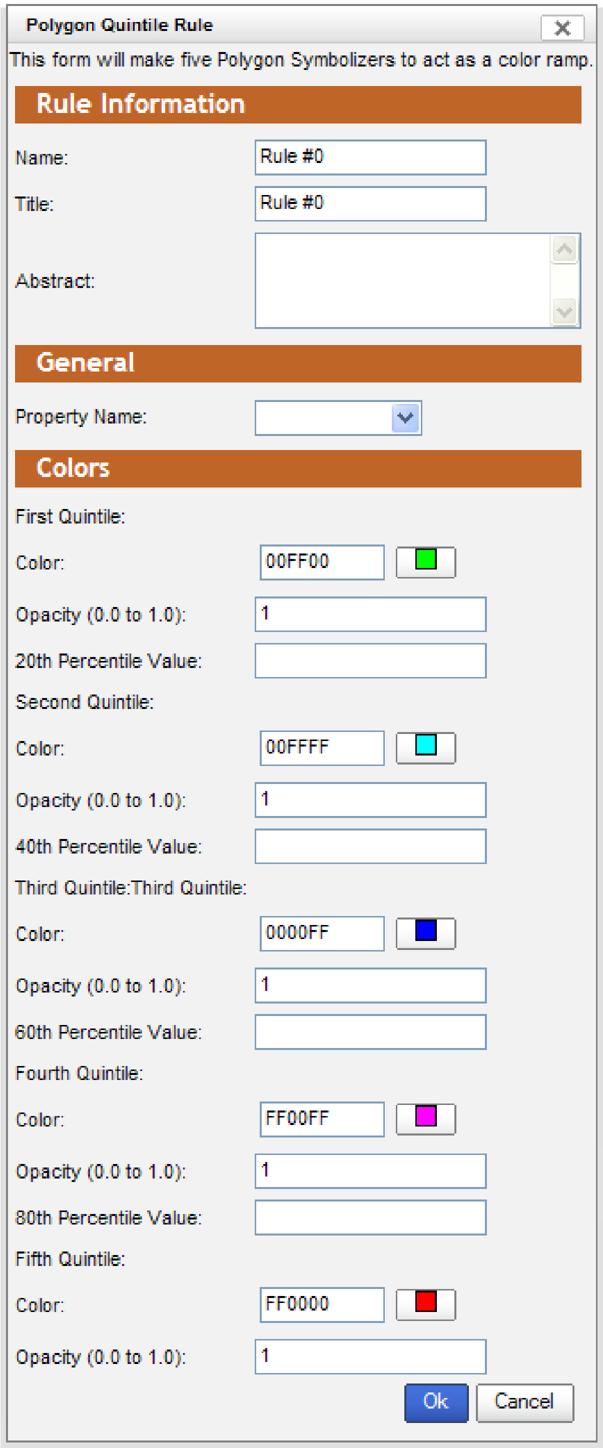
Property Name:

Operation: Default

Comparison or Min. Value:

Max. Value:

Figure 14: Polygon Quintile Rule Dialog Box



Polygon Quintile Rule

This form will make five Polygon Symbolizers to act as a color ramp.

Rule Information

Name: Rule #0

Title: Rule #0

Abstract:

General

Property Name:

Colors

First Quintile:

Color: 00FF00 🟩

Opacity (0.0 to 1.0): 1

20th Percentile Value:

Second Quintile:

Color: 00FFFF

Opacity (0.0 to 1.0): 1

40th Percentile Value:

Third Quintile:Third Quintile:

Color: 0000FF 🟦

Opacity (0.0 to 1.0): 1

60th Percentile Value:

Fourth Quintile:

Color: FF00FF 🟪

Opacity (0.0 to 1.0): 1

80th Percentile Value:

Fifth Quintile:

Color: FF0000 🟥

Opacity (0.0 to 1.0): 1

Layer Type

Use this to set the default delivery protocol for this layer.

- **Simple** - WMS GetMap
- **Tiled** - ImageX if available or WMS GetMap (tiled with 256x256 blocks for the requested extent)
- Streamed (**ECWP**) - ECWP Streaming (for both ECWP and JP2)
- Streamed (**JPIP**) - JPIP Streaming (for JP2 only)

Scale

The viewable scale range for a layer is primarily defined at the server level. The Scale tab allows you to further limit the scales at which the layer can be seen. Then, if you zoom in or out too much (out of the range boundaries), nothing will be shown.

- **Min** - The lowest scale at which the layers should be viewable.
- **Max** - The highest scale at which the layer should be viewable.

Refresh

The Refresh tab allows you to force the layer to refresh and set the amount of time that should elapse before the layer is refreshed again. This is useful when the back-end server has real-time data that is updated continuously.

- **Apply** - if this option is selected, the Web Client will refresh this layer in the map panel at the specified time intervals.
- **Delay** - specified the amount of time, in milliseconds, before the layer is refreshed again.

The Browse Tab

This tab allows you to see all of the data that you are able to access through the Web Client.

Remote Services

The Remote Service node shows all of the service providers that you can access through the web client. These service providers are categorized by the type of protocol that they use.

- **Maps (WMS)** - web map service providers. Under this node, you will also see the base map that is being used in the maps panel. The label shown for the top-level node for a WMS comes from the title parameter for the WMS (which is shown in the GetCapabilities document for the WMS). Under that node, the layers are shown in their hierarchical format, in accordance with OGC specs for WMS.

- **Vectors (WFS)** - web feature service providers. The top-level node for WFS comes from the title parameter of the WFS. Under that node, the layers are shown in a flat format, since the WFS specification provided by the OGC does not allow the nesting of feature types.

- **Coverages (WCS)** - web coverage service providers

- **Coverage Catalog** - contents of the image catalog for the ERDAS APOLLO Server that is associated with this Web Client.

  All of the aggregates in the catalog are shown as nodes under the ROOT aggregate for the catalog. You can open an aggregate node to see the child aggregates, if there are any. If you click on an aggregate, the Browse Results tab in the Results Panel will display more information about the aggregate and a list of the datasets that it contains.

- **ECWP (IWS)** - enhanced compression wavelet protocol (powered by the Image Web Server

- **Tiled Maps (TMS)** - maps that use tiles in accordance with the Tiled Map Specification (TMS), which is often supported by open- source software.

- **ArcIMS (ArcXML)** - maps originating from an Arc Internet Map Server.

To add a new Remote Service to your web client:

1. Click the **Add Service** button.
The Add Dataset box opens.

2. In the **Service Type** list, select the kind of service provider that you want to communicate with via the web client.

3. In the **Service URL** box, type (or paste) the URL of the service provider.

4. If required, specify a user name in the **Login** box and its corresponding password in the **Password** box.

5. Click **Finish**.

The new remote service will appear under the appropriate node under the Remote Services node. It blinks aqua a couple of times to make it easier for you to see where it is in the tree list.

Local Data

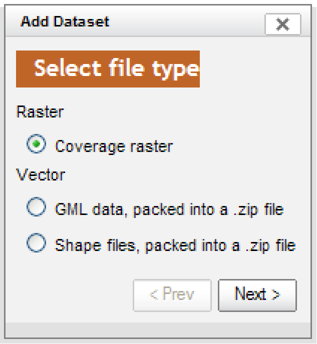
The Local Data tab allows you to use the ERDAS APOLLO Web Client to view data that you have saved on your local machine through a WCS, WMS, or WFS interface.

When you use the Web Client to view your local data, it will only be available for the rest of the current session. Once you are logged out of the Web Client, the information will not be retained.

To view local data using a WCS, WFS, or WMS interface:

1. Click the **Upload Data** button.
   The Add Dataset wizard opens.

2. Select the file type and click **Next**.



Add Dataset

Select file type

Raster

● Coverage raster

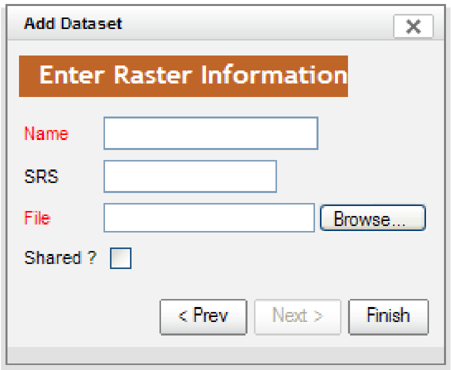
Vector

○ GML data, packed into a .zip file

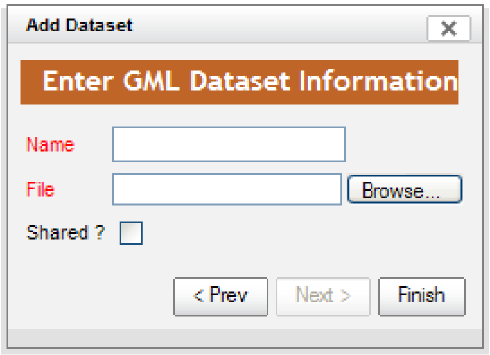
○ Shape files, packed into a .zip file

3. The next step of the wizard will look slightly different depending on what option you selected in the first step. The fields shown in red are required in order for you to upload the data.

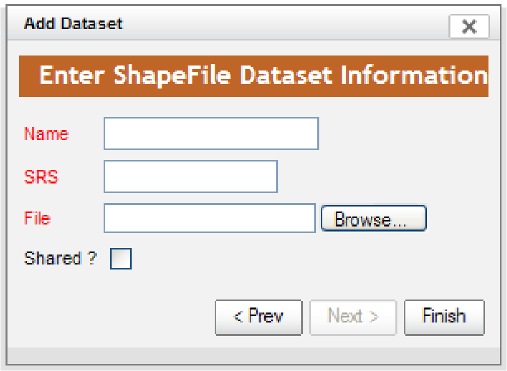
Coverage Raster



GML



ShapeFiles



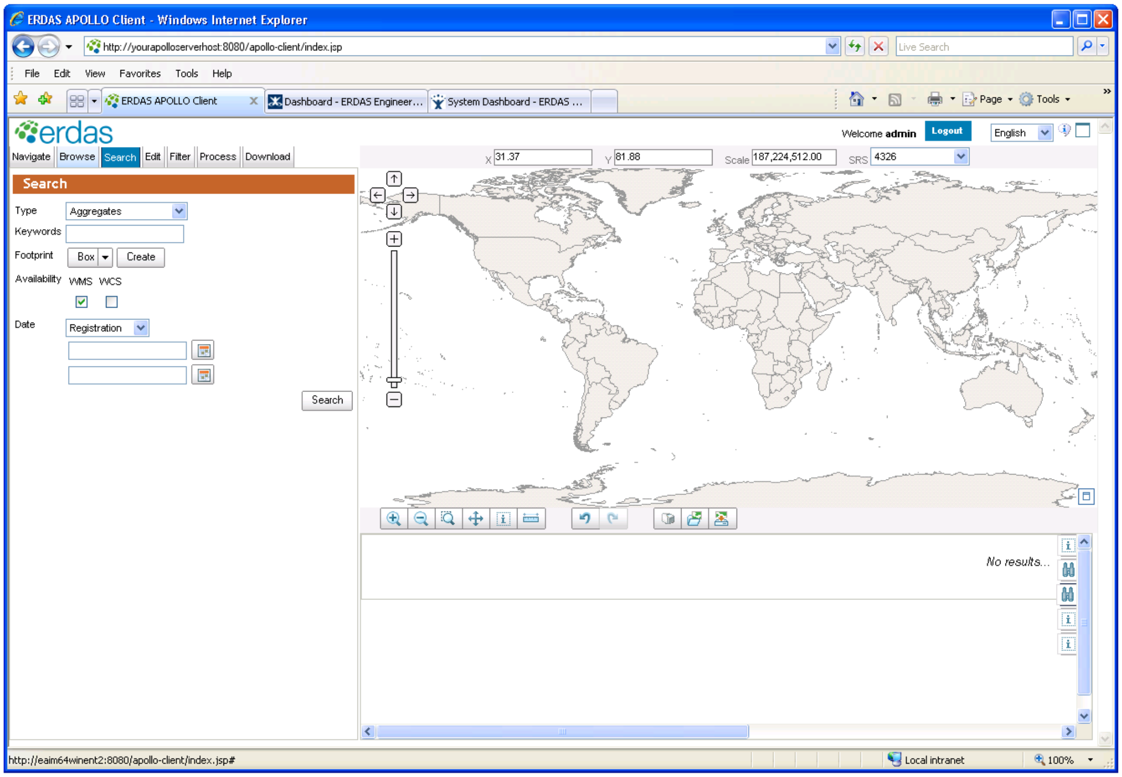
4. The **Shared?** option allows **ALL** users of the ERDAS APOLLO Web Client to view this data just as you are viewing it.
   Only click this option if you have no security concerns for this data.

5. When you are satisfied with the information you have entered, click **Finish**.

The Search Tab

This tab allows you to enter criteria for the ERDAS APOLLO Web Client to use when searching for datasets (which represent single images), aggregates (which represent collections of images), vector layers, map layers, coverage layers, or custom objects.

Some of the search criteria fields only appear when you select certain options in other fields. This tab can also be customized to include other search criteria fields in addition to these default fields.



http://yourapolloserverhost:8080/apollo-client/index.jsp

English

X 31.37 Y 81.88 Scale 187,224,512.00 SRS 4326

Search

Type: Aggregates

Keywords:

Footprint: Box

Availability: WMS ☑ WCS ☐

Date: Registration

No results...

http://eaim64winent2:8080/apollo-client/index.jsp#

Standard Search Fields

Type

Specifies the kind of information that you are looking for. You can search for:

- Any

- Datasets & Aggregate
- Datasets
- Aggregates
- Vector Layers
- Map Layers
- Coverage Layers
- Custom Objects (this option allows you to search for items such as text files, PDF files, sound files, and movie files)

Keywords

Returns items whose ID or keywords match one of the requested keywords.

The Web Client automatically appends a "wildcard" to the end of each keyword you enter in this field. For example, if you enter "Ma" as a keyword, the search would return any item with an ID or keyword that begins with Ma, and your search results could include "Massachusetts", "Malaysia", and "Maine".

.

Footprint

Allows you to specify a shape to use while looking for items. The search will return any item whose footprint intersects the specified shape.

You can use the following shapes:

- Box
- Point
- Multi Point
- Line
- Multi Line
- Polygon
- Multi Polygon

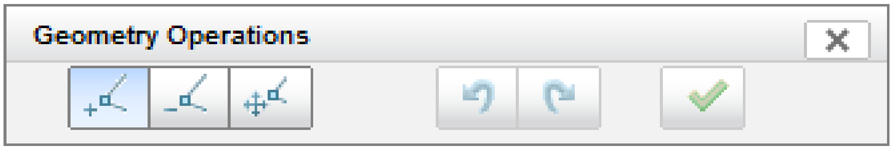
To specify the footprint search shape:

1. Select the shape in the **Footprint** dropdown box.
2. Click the **Create** button.
   (Three buttons now appear to the right of the Footprints label: Edit, Delete, and Fit.)
3. Use your mouse to draw the footprint search shape on the map to the right of the tools panel. The Geometry Operations toolbar features tools to help you draw your footprint shape. All of the tools on the toolbar are described below.
4. Click the **Ok** (green check mark) button on the Geometry Operations toolbar.

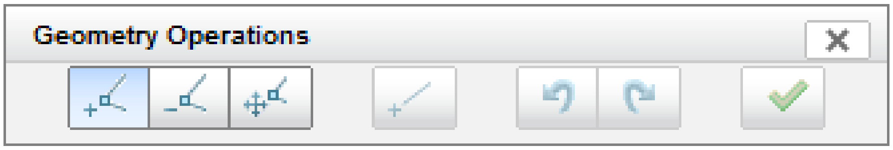
Geometry Operations Toolbar

If you are searching using a point, multi point, line, multi line, polygon, or multi polygon (in other words, anything other than a box) then the Geometry Operations toolbar will appear on the map after you click the Create button.

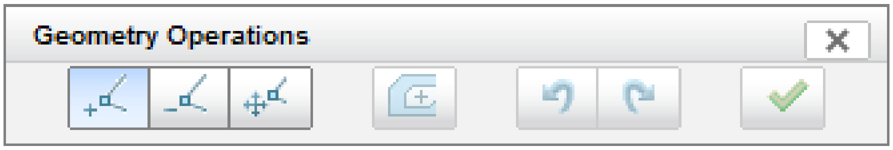
If you selected **Point**, **Multi Point**, or **Line** in the Footprints dropdown box, the Geometry Operations toolbar looks like this:



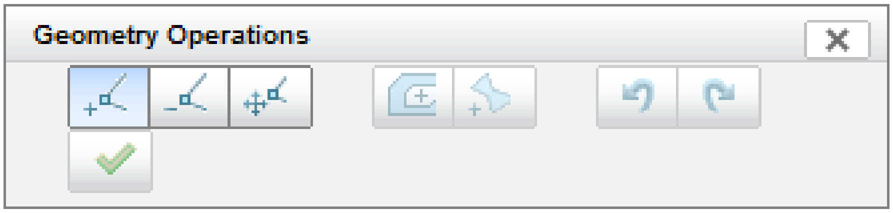
If you selected **Multi Line**, it looks like this:



If you selected Polygon, it looks like this:



If you selected **Multi Polygon**, it looks like this:



The following table describes what the tools on this toolbar will do and how to use them.

Button	Name	Description
	Add Vertex	This tool is selected by default after you click the Create button. It allows you to place points, endpoints, or vertices on the map. To use it, just click the map where you would like to place the point, endpoint, or vertex.
	Delete Vertex	Allows you to delete a point, endpoint, or vertex when you are creating a footprint that is a point, multi point, line, multiline, polygon, or multi polygon. To use this tool, click the button, then click the point that you want to delete.
	Move Vertex	Allows you to move a point, endpoint, or vertex when you are creating a footprint that is a point, multi point, line, multiline, polygon, or multi polygon. To use this tool, click the button, then click the point and drag it to its new location.
↶	**Undo Edit**	Allows you to undo up to the last five changes you made while drawing your search footprint.
↷	**Redo Edit**	Allows you to redo any changes that you have previously used the Undo tool to erase.

Button	Name	Description
✓	**Ok**	Indicates that you are finished drawing the footprint on the map. If you need to change the footprint after you have clicked this button, you can click the **Edit** button to the right of the Footprints label.
	Add Line String	Allows you to begin drawing a new, separate line. *Only available for the **Multi Line** search shape.*
	Add Ring	Allows you to begin drawing a new, separate polygon inside of an existing polygon. Only available for the **Multi Polygon** search shape.
	Add Polygon	Allows you to begin drawing a new, separate polygon in addition to an existing polygon. Only available for the **Multi Polygon** search shape.

Availability

Specifies whether the results should be accessible through ECWP streaming, maps through the WMS interface, or coverages through the WCS interface.
This field only appears when searching for datasets or aggregates!

Date

The Date field represents different things depending on what type of items you are searching for.

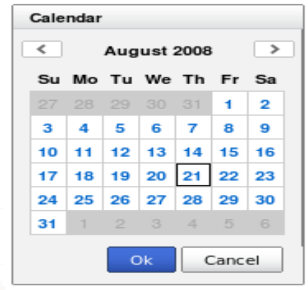
If you are searching for **vector layers**, **map layers**, **coverage layers**, or **generic items**, the Date field always represents the ***registration date***, which is the date that the layer information was registered in the ERDAS APOLLO catalog.

If you are searching for **datasets** and/or **aggregates**, you can choose which type of date to search for.
The types of dates you can search for are:

- **Registration -** the date that the item was registered in the ERDAS APOLLO catalog.
- **Creation** - the date that this item was created.
- **Modification -** the date that this item was last changed.

- **Temporal Extent** - beginning and end dates of the temporal extent you want to search for. Returns results whose temporal extent intersects the temporal extent bound by the two dates you provide here. The calendar icon to the right of each timestamp field allows you to select a date. Both timestamp fields must be filled out if you want to search by temporal extent.

Figure 15: Calendar Popup



Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

The Edit Tab

The Edit tab allows you to add new features to a vector layer or edit details about existing features on a vector layer. In order to do this, the layer must come from a transactional WFS (WFS-T) and must be visible on the map panel.

To create a new feature:

1. The dropdown box in the **Select Layer to Edit** section shows all of the transactional vector layers that are visible on the map panel. Select the one to which you want to add the new feature.

2. Click the **Create New Object** button.
   The Current Object section now displays all of the attributes for a feature in that layer.

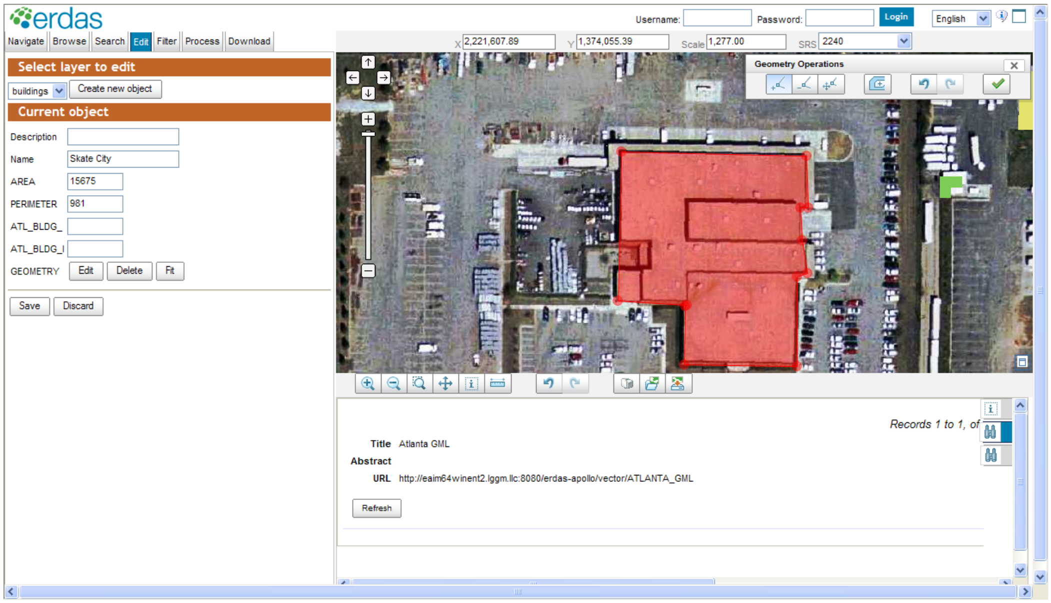
3. Fill out the attribute fields. The required attributes have asterisks (*) next to their labels.
   If you enter information incorrectly (such as placing letters in a field that only accepts numbers) the field label will turn red and stay red until the issue is corrected.

4. Specify the geometry of the new feature.

   a. Select the general shape of the feature in the **Geometry** dropdown box.

   b. Click the **Create** button to the right of the Geometry field. (Three buttons now appear to the right of the Geometry label: Edit, Delete, and Fit.)

c. Use your mouse to draw the feature shape on the map panel. The **Standard Search Fields** features tools to help you draw your shape.



X 2,221,607.89
Y 1,374,055.39
Scale 1,277.00
SRS 2240

Select layer to edit

buildings | Create new object

Current object

Description	
Name	Skate City
AREA	15675
PERIMETER	981
ATL_BLDG_	
ATL_BLDG_I	
GEOMETRY	Edit Delete Fit

Save Discard

Records 1 to 1, of

Title Atlanta GML

Abstract

URL http://eaim64winent2.lggm.llc:8080/erdas-apollo/vector/ATLANTA_GML

Refresh

5. Click the **Ok** (green check mark) button on the Geometry Operations toolbar when you are finished drawing the shape of the feature.

   If you need to change the shape after you have clicked this button, you can click the **Edit** button to the right of the Geometry label, or you can click the **Delete** button to remove this shape and create a new one.

   The **Fit** button redraws the map so the extent of your new shape fills the viewable map area.

6. When you are satisfied with the information you have entered for this new feature, click the **Save** button at the bottom of the Current Object section to save it.

To edit an existing feature:

1. Click the **Layer Info** icon on the map toolbar and then click on the feature you want to edit in the map. The layer information appears in the Results Panel below the map. You may see different tabs for different layers. Click on the tab for the layer that contains the feature you want to edit.

2. If this layer is based on a transactional WFS, you will see an Edit link in the **Feature** column on the far left.

3. Click the **Edit** link. The attributes for the feature will appear in the Current Object section.

4. Make your changes to the attribute fields. The required attributes have asterisks (*) next to their labels.

5. If you want to change the geometry of the new feature, you can click the **Delete** button to remove the existing one so you can create a new one, or you can click the **Edit** button and use the **Standard Search Fields** tools to change the existing one.

   To make the feature shape easier to see, you can click the **Fit** button so that the extent of the shape fills the viewable map area.

6. Click the **Ok** (green check mark) button on the Geometry Operations toolbar when you are finished editing the shape of the feature.

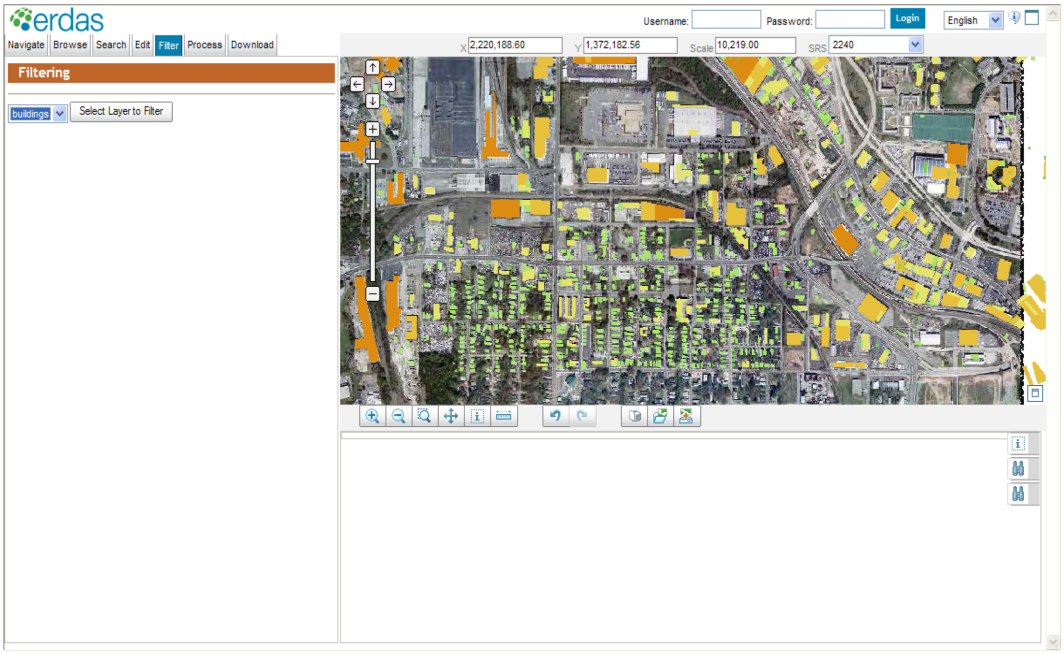
7. When you are satisfied with the information you have entered for this new feature, click the **Save** button at the bottom of the Current Object section to save it.

The Filter Tab

The filter tab allows you to temporarily restrict the features that are visible on a vector layer in the map panel.

You can select to see only the features that lie within a certain bounding box, or you can use the attributes for features in that vector layer to filter what can be seen. For example, if you have a vector layer that shows buildings, and one of the attributes for each building feature is the area, you could filter the layer so that only buildings larger than 15,000 square feet are visible.

Figure 16: A Vector Layer, Before Filtering



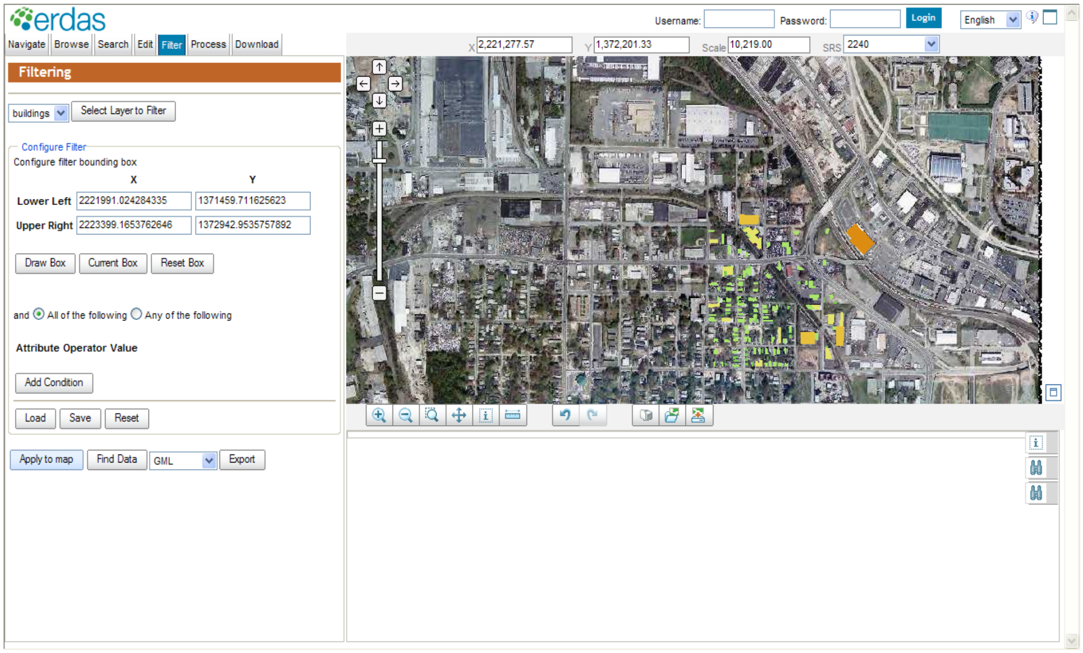
X 2,220,188.60

Y 1,372,182.56

Scale 10,219.00

SRS 2240

Figure 17: The Same Vector Layer, After Filtering With a Bounding Box



X 2,221,277.57 Y 1,372,201.33 Scale 10,219.00 SRS 2240

buildings

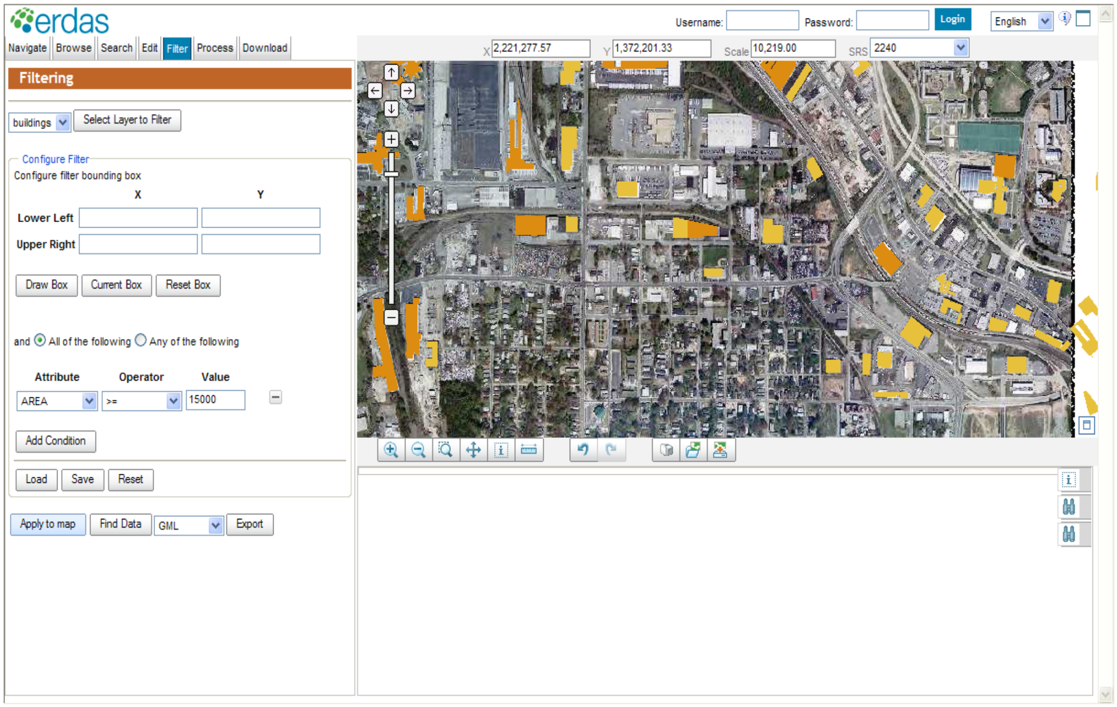
Configure filter bounding box

	X	Y
Lower Left	2221991.024284335	1371459.711625623
Upper Right	2223399.1653762646	1372942.9535757892

and ● All of the following ○ Any of the following

GML

Figure 18: The Same Vector Layer, After Filtering With the Area Attribute (Area >= 15,000)



X 2,221,277.57 Y 1,372,201.33 Scale 10,219.00 SRS 2240

buildings

Configure filter bounding box

	X	Y
Lower Left		
Upper Right		

and ● All of the following ○ Any of the following

Attribute	Operator	Value
AREA	>=	15000

GML

To create a filter:

1. Make sure the map is displaying the vector layer whose features you want to filter.

2. The vector layers that are on the map will appear in the dropdown box at the top of the tab. Select the vector layer to which you want to apply the filter in the dropdown box.

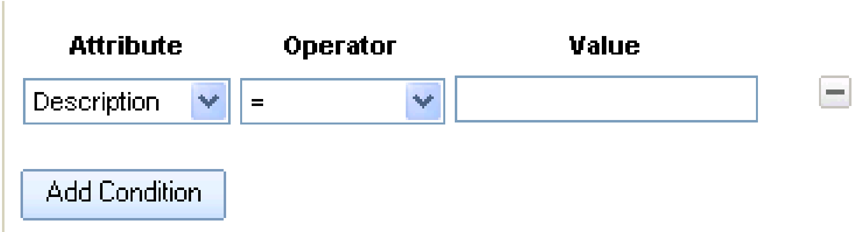
3. Click the **Select Layer to Filter** button.
   The controls that allows you to define your filter appear.

4. If you would like to use a bounding box as part of your filter, you can specify it by:

   - Clicking the **Draw Box** button and drawing the box on the map with your mouse. When you release the mouse, the bounding box values will appear in the appropriate filter fields automatically.

   - Clicking the **Current Box** button to use the extents of the current map view.

5. If you would also like to filter the features using feature attributes, click the **Add Condition** button.
   A row of fields will appear above the Add Conditions button.



Attribute	Operator	Value
Description	=	

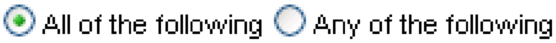
6. In the **Attribute** dropdown box, select the feature attribute you want to use in the filter.

7. In the **Operator** dropdown box, select the operator that indicates the relationship the feature attribute should have to the value.

8. In the **Value** dropdown box, type the test value to compare to the attribute value. Comparisons against a string field are not case-sensitive.

9. If you would like to add another condition, click the **Add Condition** button again.

   If you are creating multiple conditions, you should select one of the following options:



● All of the following ○ Any of the following

10. If you need to delete one of the conditions, click the minus sign (**-**) button to the right of the condition.

11. When you are satisfied with the filter options you have specified, click the Add to Map button to see the filtered layer on the map.

To save a filter you have created:

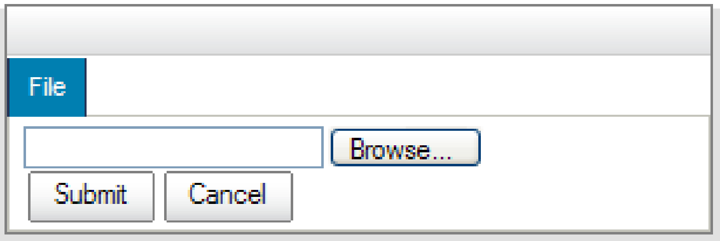
Click the **Save** button on the Filter tab. The filter specifications will be stored in an XML file that can be loaded into the Web Client.

To load a previously created filter:

1. Select the layer that the filter was created for in the dropdown box at the top of the tab.

2. Click the **Select Layer to Filter** button.

3. Click the **Load** button below the filter options controls. A box will appear on the Web Client.



4. Click the **Browse** button on the box to find the filter file.

5. Click **Submit** to load the filter.

The options for the filter will appear in the filter options controls. Click **Apply to Map** to show the filtered layer in the map.

To view the layer info for the new, filtered layer:

1. Create a filter and click the **Apply to Map** button to execute the filter. You will then be able to see the filtered vector layer in the map.

2. Click the **Find Data** button.

The Results Panel will display the detailed information about the features in the filtered layer. This feature works similarly to the **Layer Info** tool in the map toolbar.

To export the new, filtered layer:

1. Create a filter and click the **Apply to Map** button to execute the filter. You will then be able to see the filtered vector layer in the map.

2. Select the format in which to save the layer. The Web Client provides the following options:

- Keyhole Markup Language (**KML**) for use with Google Earth.

- OGC’s Geography Markup Language (**GML**)

- **Shapefile**

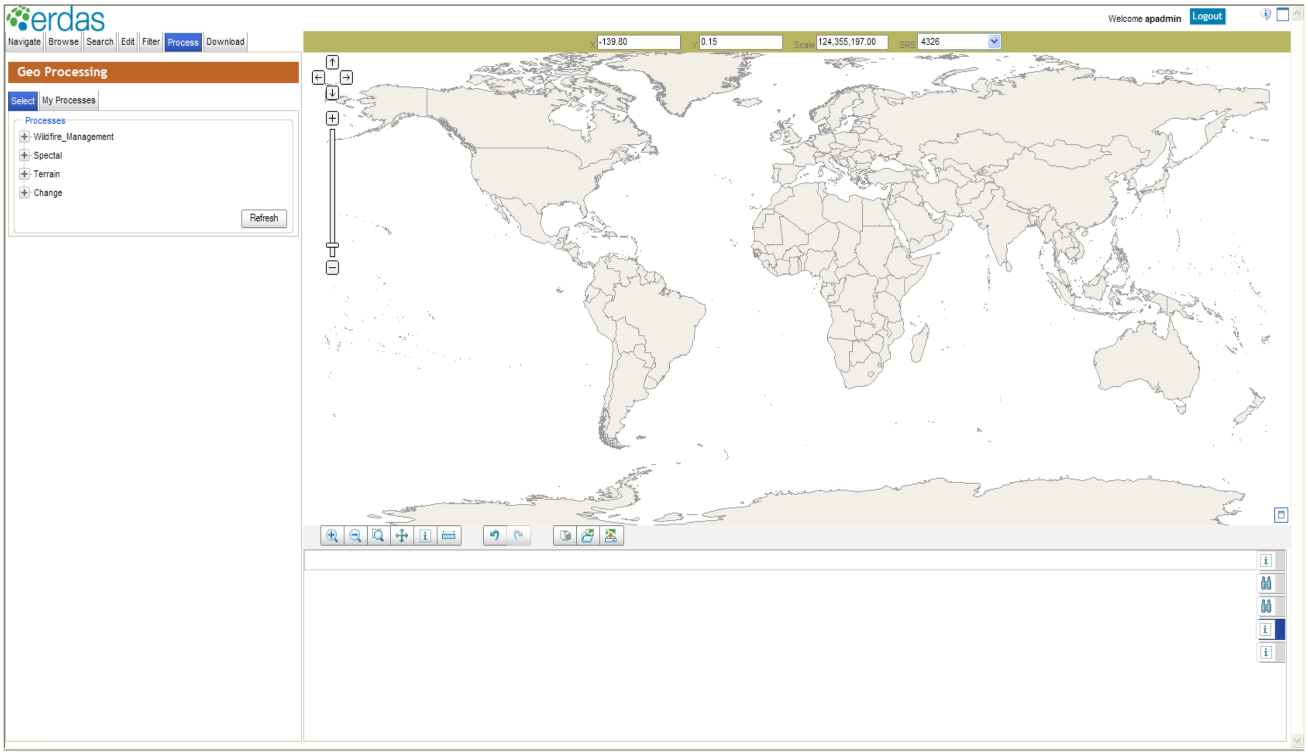
The Process Tab (ERDAS APOLLO Professional Only)

The Process tab allows you to execute certain processes that have been stored in your ERDAS APOLLO system and view the results of those processes. You must be logged in to the ERDAS APOLLO Web Client to see and use the controls on the Process tab or to see the results of executed processes.

The **Select** sub tab shows all of the processes that you have access to in the ERDAS APOLLO Web Client. Each of the nodes that you see in the tree is a process category. If you open the process category node, you will see the processes themselves.

The **My Processes** sub tab shows all of the processes that you have executed in the ERDAS APOLLO Web Client. The processes are shown in order from the most recently executed to less recently executed.

Figure 19: The Process Tab When First Opened



X -139.80 Y 0.15 Scale 124,355,197.00 SRS 4326

To execute a process in the ERDAS APOLLO Web Client:

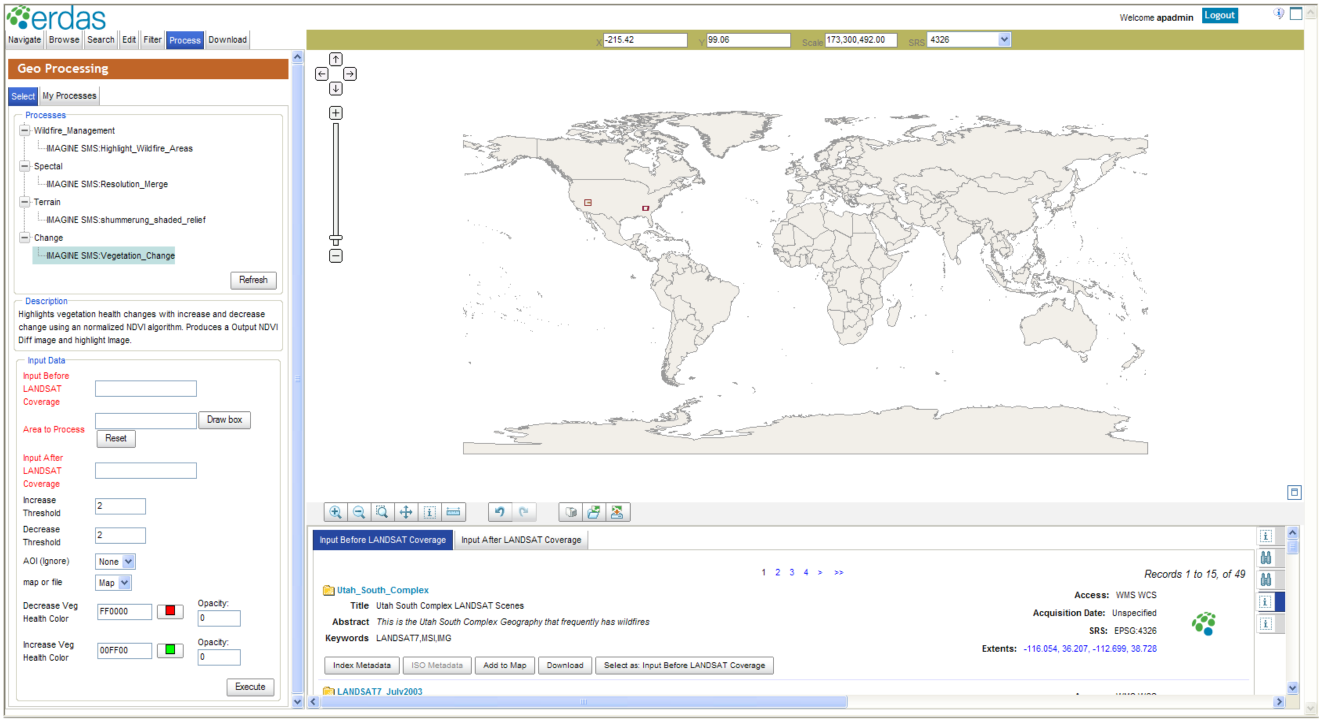
1. Locate the process in the **Processes** list on the **Select** tab. Click on it to select it.

   The **Description** and **Input Data** sections will now appear on the Search sub tab.

   The **GeoProcess Complex Inputs** sub tab will open on the Results panel. This tab shows the datasets that meet the criteria to work as reasonable inputs for this process.

   NOTE: If the process was added to the system before you began your web client session, you need to click the Refresh button to update the Processes list with the most current information. The Refresh button is located in the lower right corner of the Select sub tab.

Figure 20: The Process Tab, After Selecting a Process in the Select Sub Tab



X: -215.42 Y: 99.06 Scale: 173,300,492.00 SRS: 4326

Description: Highlights vegetation health changes with increase and decrease change using an normalized NDVI algorithm. Produces a Output NDVI Diff image and highlight Image.

Input Before LANDSAT Coverage:
Area to Process:
Input After LANDSAT Coverage:
Increase Threshold: 2
Decrease Threshold: 2
AOI (Ignore): None
map or file: Map
Decrease Veg Health Color: FF0000 🟥 Opacity: 0
Increase Veg Health Color: 00FF00 🟩 Opacity: 0

Records 1 to 15, of 49

Utah_South_Complex
Title: Utah South Complex LANDSAT Scenes
Abstract: *This is the Utah South Complex Geography that frequently has wildfires*
Keywords: LANDSAT7,MSI,IMG
Access: WMS WCS
Acquisition Date: Unspecified
SRS: EPSG:4326
Extents: -116.054, 36.207, -112.699, 38.728

2. Fill out the fields in the **Input Data** section. The fields with a red label must be filled out before you can execute the process.

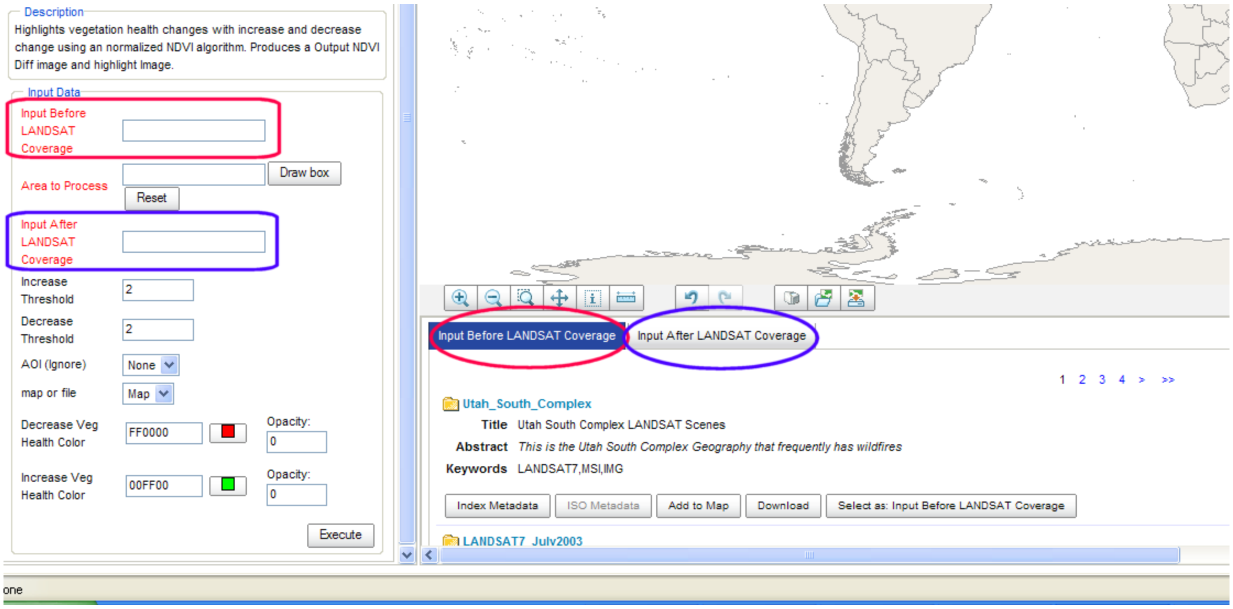
   The exact fields in the Input Data section vary depending on the process that you are trying to execute. The input data requirements are determined by the author of the process model when he or she designs it in ERDAS IMAGINE.

   Specifying the Input Data - The fields at the top of the section are generally the ones where you will indicate the coverages to be processed.

   When you select a process on the Select sub tab, the Web Client automatically filters the coverages that you have access to and identifies the ones that match the constraints established by the process author and therefore, are suitable as input for this process. These coverages are shown on the **GeoProcess Complex Inputs** tab in the Results panel.

   If there are multiple input fields, the appropriate coverages for each field are shown in separate sub tabs *inside* the GeoProcess Complex Input tab. For example, in the figure below, there are two data input fields, Input Before LANDSAT Coverage and Input After LANDSAT Coverage. Each field has its own tab within the GeoProcess Complex Inputs tab.

   To select a coverage to be the input, click the **Select as:** button in that coverage's entry in the Results panel.



Description

Highlights vegetation health changes with increase and decrease change using an normalized NDVI algorithm. Produces a Output NDVI Diff image and highlight Image.

Input Data

Input Before LANDSAT Coverage

Area to Process

Input After LANDSAT Coverage

Increase Threshold: 2

Decrease Threshold: 2

AOI (Ignore): None

map or file: Map

Decrease Veg Health Color: FF0000 🟥 Opacity: 0

Increase Veg Health Color: 00FF00 🟩 Opacity: 0

1 2 3 4 > >>

Utah_South_Complex

Title: Utah South Complex LANDSAT Scenes

Abstract: *This is the Utah South Complex Geography that frequently has wildfires*

Keywords: LANDSAT7,MSI,IMG

LANDSAT7_July2003

one

Draw box
Reset
Execute
Input Before LANDSAT Coverage
Input After LANDSAT Coverage
Index Metadata
ISO Metadata
Add to Map
Download
Select as: Input Before LANDSAT Coverage

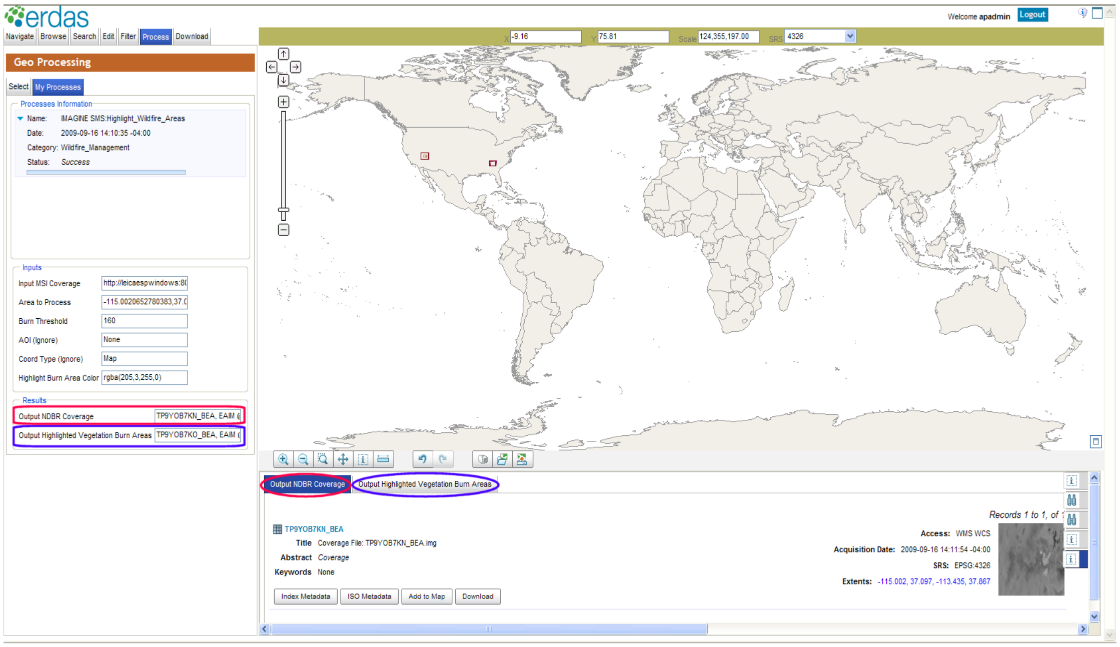
Specifying Other Process Options - Many of the other fields are simple text boxes in which you can type your information. If you have a field that requires you to specify a color, the field will be accompanied by a button that opens a color picker dialog. If you have a field that requires you to select a bounding box, it will be accompanied by the **Draw Box** and **Reset** buttons. If you click the Draw Box button and draw your bounding box on the map panel, the web client will place the correct values for the box in the field for you. If you are not satisfied with the bounding box settings, you can click the Reset button to clear the field and start over.

3. When you are satisfied with the settings you entered in the Input Data section, click the **Execute** button. The My Processes tab opens and the new process is shown at the top of the list. The Status field indicates the progress of the process execution.

When "Success" appears in the Status field, the process execution is complete.

The **Results** section of the My Processes tab now shows the names of the resulting images and a URL to the catalog that now contains the images, and the images are shown in the results panel on the **GeoProcess Complex Outputs** tab. Many times, a process will yield more than one result. Each result is shown on a separate tab inside the GeoProcess Complex Outputs tab.

You can use the Add to Map button to view the data as a layer in the Map Panel. If you click on the values in the Extents field, you can zoom to that layer's extent. If you click the Download button, the information will be added to the Download tab so that you can Clip, Zip, and Ship it.



X -9.16 Y 75.81 Scale 124,355,197.00 SRS 4326

Name: IMAGINE SMS:Highlight_Wildfire_Areas
Date: 2009-09-16 14:10:35 -04:00
Category: Wildfire_Management
Status: *Success*

Input MSI Coverage	http://leicaespwindows:8(
Area to Process	-115.0020652780383,37.(
Burn Threshold	160
AOI (Ignore)	None
Coord Type (Ignore)	Map
Highlight Burn Area Color	rgba(205,3,255,0)
Output NDBR Coverage	TP9YOB7KN_BEA, EAIM (
Output Highlighted Vegetation Burn Areas	TP9YOB7KO_BEA, EAIM (

Records 1 to 1, of 1

TP9YOB7KN_BEA
Title Coverage File: TP9YOB7KN_BEA.img
Abstract *Coverage*
Keywords None

Access: WMS WCS
Acquisition Date: 2009-09-16 14:11:54 -04:00
SRS: EPSG:4326
Extents: -115.002, 37.097, -113.435, 37.867

The Clip-Zip Ship Tab (ERDAS APOLLO Server Advantage and Professional only)

This tab provides the tools for downloading the data that you see in the ERDAS APOLLO Web Client.

The **Clip, Zip, and Ship** feature takes data that you have “clipped”, or selected, and places it in a zip file. A URL that you can use to access that zip file and download it is then “shipped” to your e-mail address.

You can download data of any type, as long as it is WCS-enabled. The data can be shipped to you in the following formats:

- *.**tif** (GeoTIFF),
- *.**img** (the ERDAS IMAGINE native format),
- *.**ntf** (National Imagery Transmission Format)
- *.**ecw** (Enhanced Compression Wavelet)
- *.**jp2** (JPEG2000)

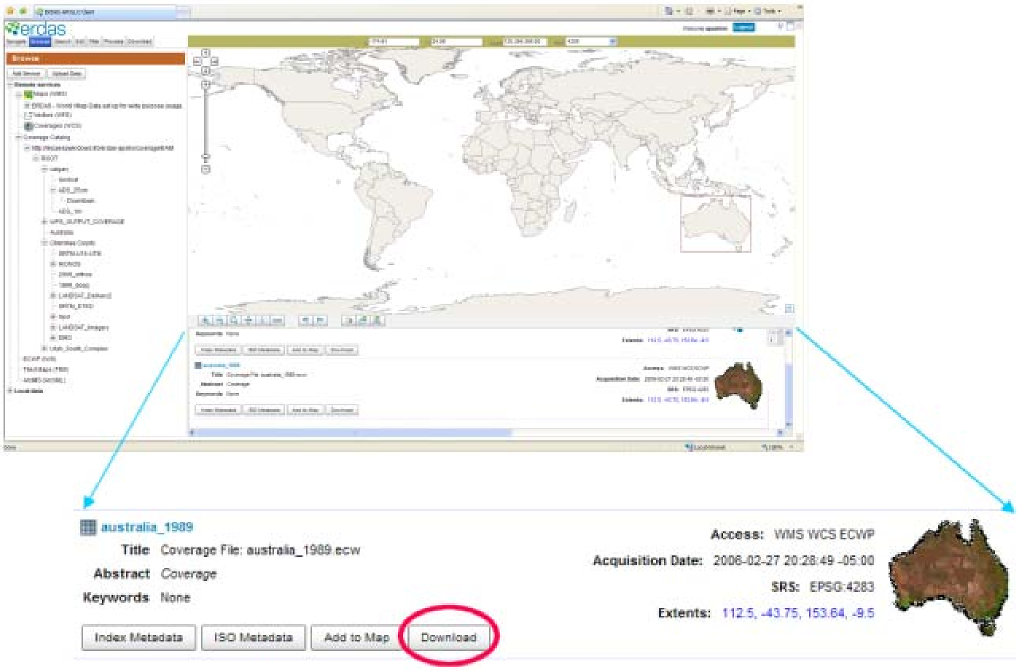
Clipping the Data

To initiate the Clip, Zip, and Ship process, you first need to indicate which coverage you want to download by clicking the **Download** button in its entry in the Results panel. The name of that coverage now appears in the coverages list on the top left of the Download tab.

You can select multiple coverages for download.

The Download button only appears for coverages that are WCS-enabled and can be downloaded.

Figure 21: Location of the Download Button in the Result Panel



australia_1989

Title Coverage File: australia_1989.ecw

Abstract *Coverage*

Keywords None

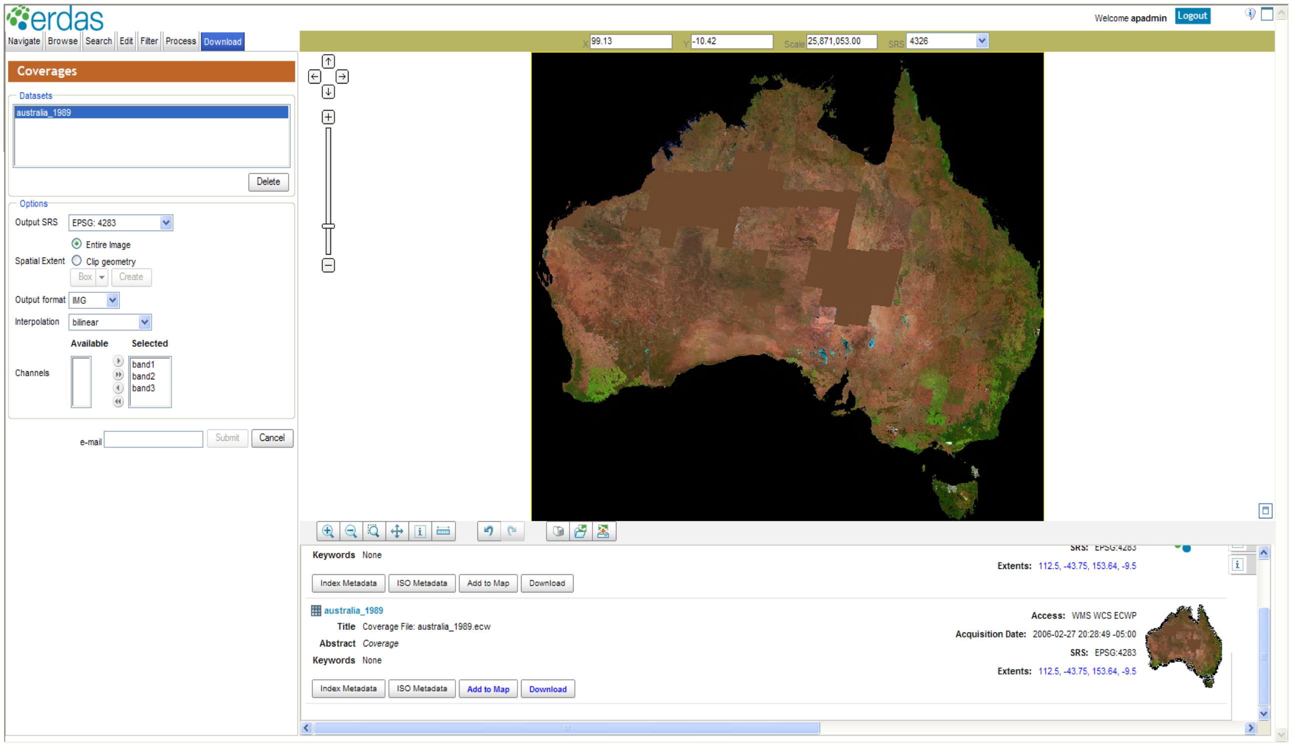
Access: WMS WCS ECWP

Acquisition Date: 2006-02-27 20:28:49 -05:00

SRS: EPSG:4283

Extents: 112.5, -43.75, 153.64, -9.5

Figure 22: The Download Tab, With an Image to be Configured for Download



X 99.13 Y -10.42 Scale 25,871,053.00 SRS 4326

Datasets: australia_1989

Output SRS: EPSG: 4283

Spatial Extent: ● Entire Image ○ Clip geometry

Output format: IMG

Interpolation: bilinear

Channels: Available | Selected: band1, band2, band3

e-mail

Keywords None

Extents: 112.5, -43.75, 153.64, -9.5

australia_1989

Title Coverage File: australia_1989.ecw

Abstract *Coverage*

Keywords None

Access: WMS WCS ECWP

Acquisition Date: 2006-02-27 20:28:49 -05:00

SRS: EPSG:4283

Extents: 112.5, -43.75, 153.64, -9.5

If you select a polygon on a coverage that is going to be shipped in Imagine or GeoTIFF format, ERDAS APOLLO will ship data from the bounding box of that polygon. The region between the outer edge of the polygon and the edge of the bounding box will consist of null values.

If you select a polygon on a coverage that is going to be shipped in NITF format, the web client will ship data from the bounding box surrounding that polygon. All of the data within the bounding box will be valid data from the source coverage.

Zipping the Data

Resampling Options

After you select the coverages that you want to download, you need to specify if and how you want the web client to resample the coverages that it will ship to you.

Just below the list of coverages on the Download tab, you will see the options that allow you to specify how you want them to be resampled.

The options have to be set individually for each coverage. To do this, you need to highlight each coverage in the list before you set the options.

If the coverage you select is in the NITF format, the SRS, Resolution, and Band options will be disabled. NITF images contain additional metadata that other image formats do not, and resampling can cause you to lose some of that metadata.

The options are:

- **Output SRS** -The SRS in which the downloaded data will be projected. You can choose between the common SRS EPSG:4326 and the SRS of the source coverage.

- **Spatial Extent** - The extent of the selected coverage that is marked for download.

    - **Entire Image** - select this option to download the entire image.

    - **Clip Geometry** - select this option to draw a box or polygon around the specific part of the image that you want to download.

      Output Format - The data can be shipped to you in your choice of the following formats:

- *.**tif** (GeoTIFF),
- *.**img** (the ERDAS IMAGINE native format),
- *.**ntf** (National Imagery Transmission Format)
- *.**ecw** (Enhanced Compression Wavelet)
- *.**jp2** (JPEG2000)

If the data you want to download is not already in one of these formats, the ERDAS APOLLO Web Client converts it for you.

Things to Consider for Each Format

- **IMAGINE** and **GeoTIFF**

  When you download any type of data as Imagine or GeoTIFF format, each dataset that you download will be an Imagine or GeoTIFF file within the zip file. Each aggregate that you download will be an Imagine or GeoTIFF file with the child datasets of the aggregate mosaicked to form a single image.

  When you download NITF files as Imagine or GeoTIFF files there will be some loss of metadata because these image formats do not store as much information as the NITF format does.

- NITF

  NITF files are different than other image file formats because they are basically containers that store sub images and information about the sub images. The ERDAS APOLLO system treats an NITF file like an aggregate when it registers it in the catalog, and it treats each of the sub images as a child dataset of the aggregate.

  Because of these differences, there are some important things to remember when you download NITF files or download other file types as NITF files:

  - When downloading NITF files, the Clip, Zip, and Ship feature only supports nested coverages whose depth is no greater than one.
  - Each NITF aggregate that you download will be one NITF file.
  - Each NITF dataset that you download will be one NITF file that contains this dataset as a sub image.

- Other image types do not store as much information as the NITF format, so when you download other data types as NITF files, the NITF file that you receive will be missing some of the metadata that most NITF files typically contain. ERDAS APOLLO will generate a file called `warning.txt` that will appear in your zip file along with the NITF file to remind you that some of the metadata is missing.

- **ECW** and **JP2**

  The Enhanced Compression Wavelet and JPEG2000 formats are compressed formats. ERDAS APOLLO will attempt to achieve a ***target*** compression ratio, but the ***actual*** compression ratio will vary depending on the particular image.

Table 1: Target Compression Ratios and Minimum and Maximum Actual Compression Ratios Used by ERDAS APOLLO Clip, Zip, and Ship Feature

Image Type	TARGET Compression Ratio	MINIMUM Compression Ratio	MAXIMUM Compression Ratio
Grayscale	1:10	1:1	1:2000
Truecolor	1:15	1:1	1:2000
Multispectral	1:15	1:1	1:2000

- **Interpolation** - ERDAS APOLLO assigns calculated values to certain pixels in shipped images in order to make those images appear smoother. This option allows you to select how ERDAS APOLLO calculates those values. ERDAS APOLLO provides the following interpolation methods, although the ones that are currently available to you will depend on the format of the source image you are downloading.

  - **Nearest Neighbor** - assigns the value of the closest pixel.

  - **Bicubic** - assigns the weighted average of the nearest 4 pixels.

  - **Bilinear** - assigns the weighted average of the nearest 16 pixels. The pixels that are closest to the unknown pixel are weighted more heavily in that average.

- **Channels** - The bands of the source coverage that will be included in the shipped coverage. By default, all available bands are selected.

Shipping the Data

After you have set up all of the coverage download options, type your e-mail address in the **E-mail** field and click the **Submit** button.

You will receive a confirmation e-mail as soon as the server receives your request. ERDAS APOLLO will then process your coverages according to the options you specified and place them in a zip file.

You will receive a notification e-mail when the server completes your request. The notification e-mail contains hyperlink to the location of the zip file.

The Results Panel

The Results Panel shows detailed information about the data that you can access through the ERDAS APOLLO Web Client. It also often provides you with different buttons that allow you to perform specific tasks with the data.

The data is shown on different tabs within the results panel depending on how the results were obtained. The tabs are:

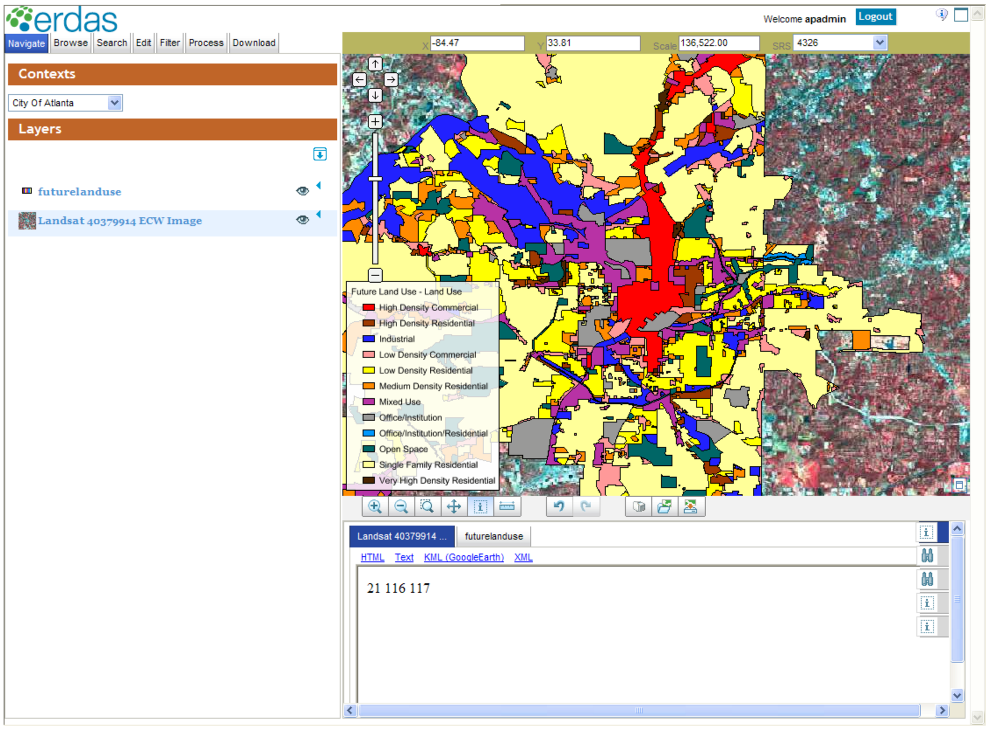
- **Layer Info**
- **Browse Results**
- **Search Results**
- **GeoProcess Complex Inputs**
- **GeoProcess Complex Outputs**

Layer Info

When you click the Layer Info ℹ button in the Map Panel toolbar and then click the map or draw a box on the map, the Layers Info tab of the Results panel will then show the details for any of the layers in the map panel whose extents contain that point or intersect with that box.

You can restrict the layers that appear in the Layer Info tab of the Results Panel by opening the Navigate tab in the Tabs Panel and clicking the **Select Layer** icon for the layers you want to view.
If no layers are selected, the map will show all of them.

The details for each layer are shown in a separate sub tab within the Layer Info tab.

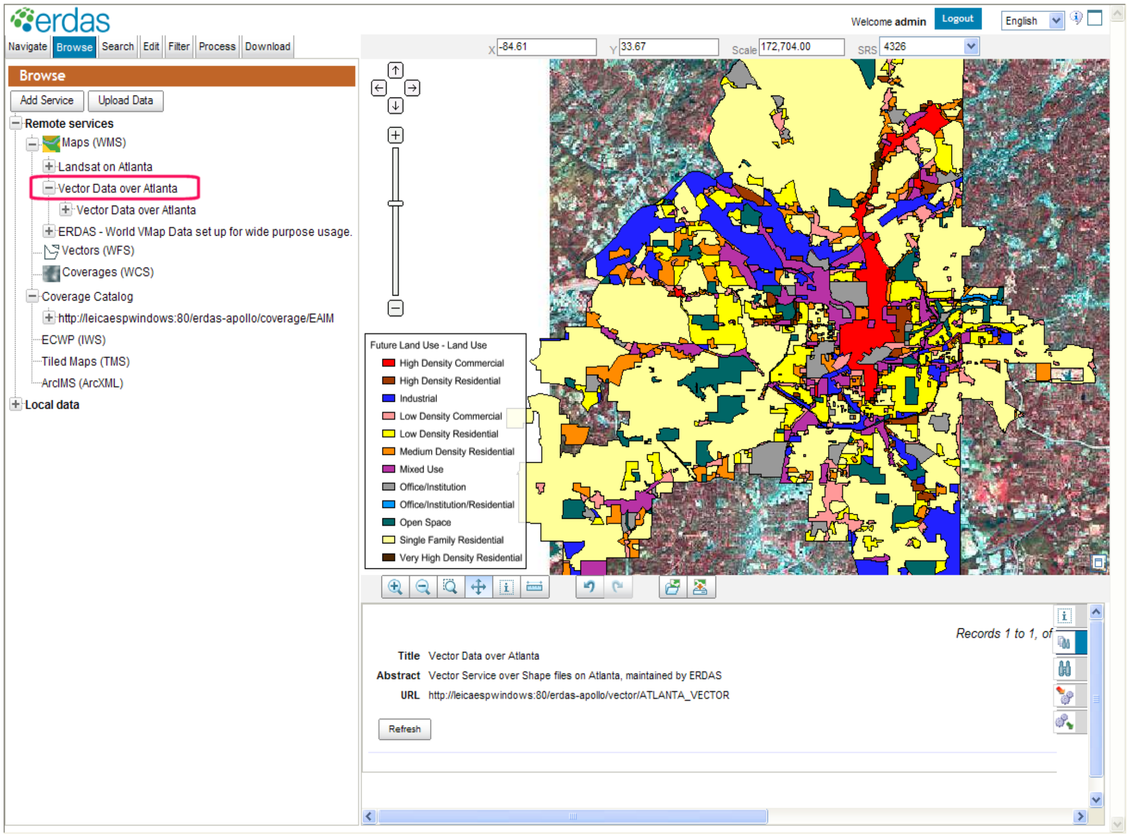


```
21 116 117
```

Browse Results

When you click on one of the services shown in the tree of the Browse tab, the Browse Results tab opens in the Results panel. The information shown in the entries will be different depending on what type of node you clicked on in the tree of the Browse tab. A

If you click on one of the top-level nodes for a service provider, the entry shown in the Results panel shows the basic information about the service provider and includes the Refresh button so you can refresh the service.



Title Vector Data over Atlanta

Abstract Vector Service over Shape files on Atlanta, maintained by ERDAS

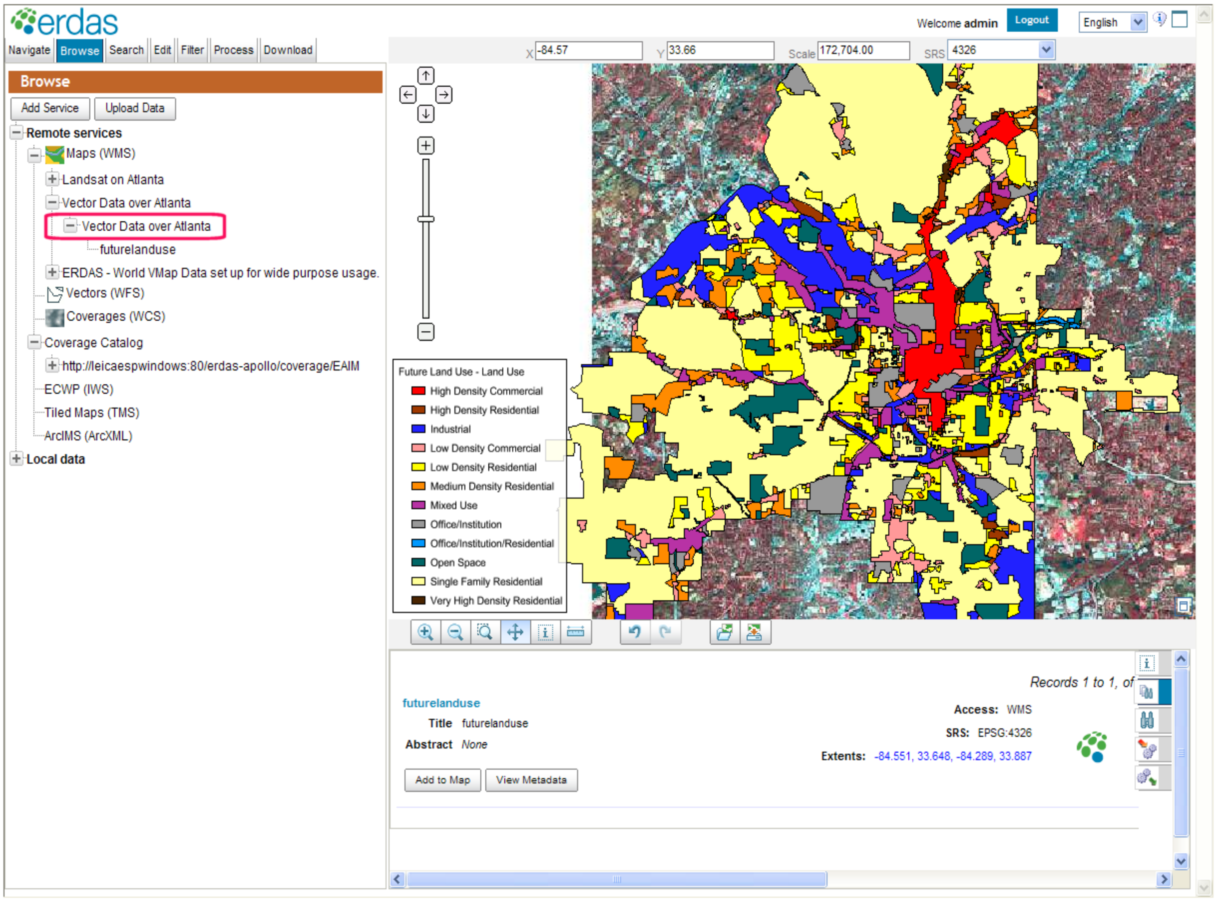
URL http://leicaespwindows:80/erdas-apollo/vector/ATLANTA_VECTOR

Refresh

If you click on one of the actual layer nodes, the entry will show the basic metadata for the layer and will provide you with the following buttons:

- **Add to Map** - click to add the layer to the map panel. For a closer look,. you can click the extents to zoom in so that the extent of the layer completely fills the viewable area of the map panel.

- **View Metadata**- click to view the metadata for the later. A box will open so you can read the information.



X -84.57 Y 33.66 Scale 172,704.00 SRS 4326

Future Land Use - Land Use

- 🟥 High Density Commercial
- ■ High Density Residential
- 🟦 Industrial
- ■ Low Density Commercial
- 🟨 Low Density Residential
- 🟧 Medium Density Residential
- 🟪 Mixed Use
- ■ Office/Institution
- 🟦 Office/Institution/Residential
- ■ Open Space
- □ Single Family Residential
- 🟫 Very High Density Residential

Records 1 to 1, of

futurelanduse

Title futurelanduse

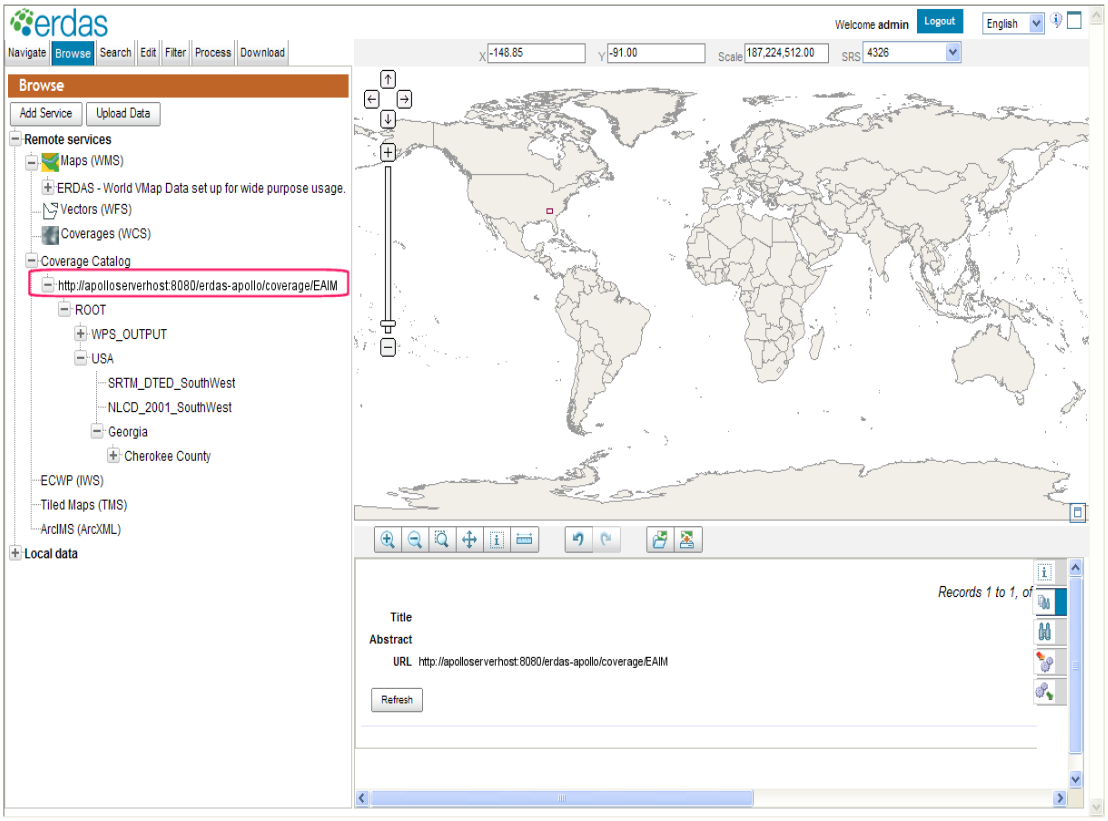
Abstract *None*

Access: WMS

SRS: EPSG:4326

Extents: -84.551, 33.648, -84.289, 33.887

If you click on a catalog node, the entry will show you the URL to that catalog.



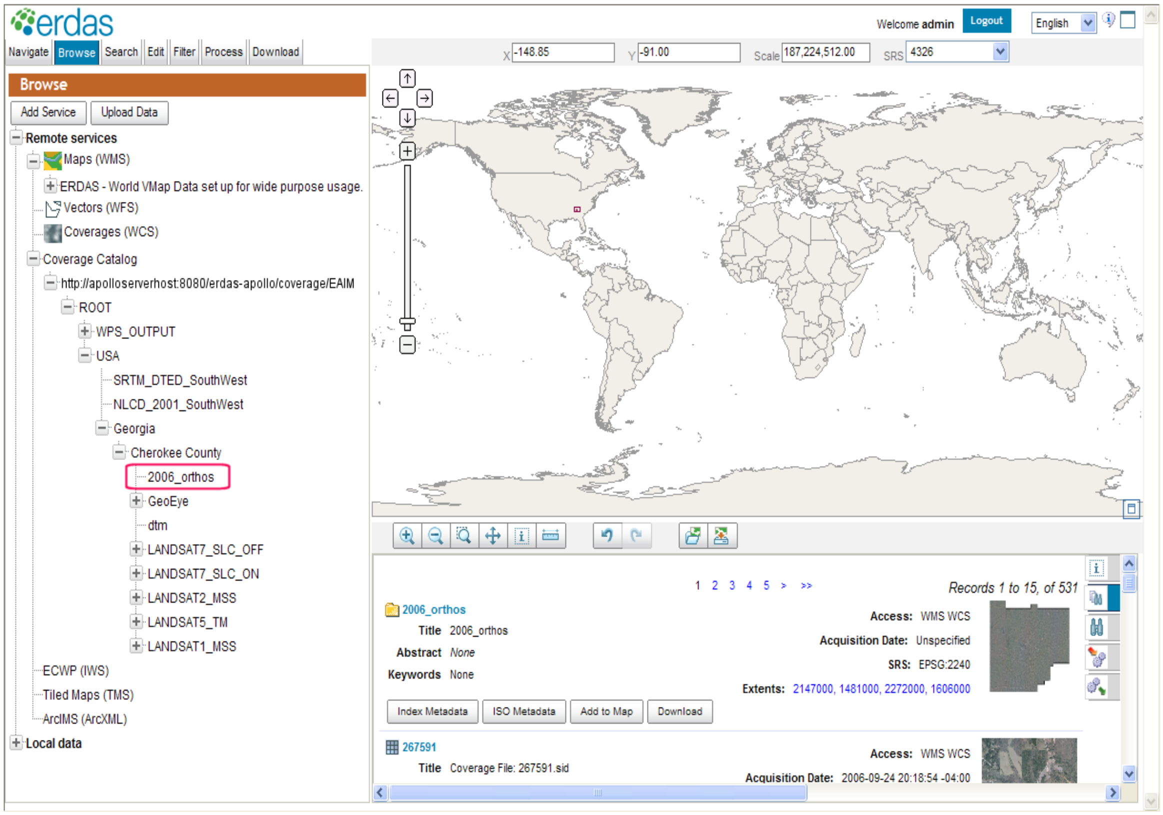
Title

Abstract

URL http://apolloserverhost:8080/erdas-apollo/coverage/EAIM

If you click one of the aggregate nodes under the catalog node, the Browse Results tab will show you all of the aggregates and datasets contained in that aggregate.

If there are more than 15 results, the results will be shown on different pages. The page numbers will be shown along the top of the Results panel and you can click on the page numbers to see the results listed on that page.



X -148.85 Y -91.00 Scale 187,224,512.00 SRS 4326

- Remote services
  - Maps (WMS)
    - ERDAS - World VMap Data set up for wide purpose usage.
  - Vectors (WFS)
  - Coverages (WCS)
  - Coverage Catalog
    - http://apolloserverhost:8080/erdas-apollo/coverage/EAIM
      - ROOT
        - WPS_OUTPUT
        - USA
          - SRTM_DTED_SouthWest
          - NLCD_2001_SouthWest
          - Georgia
            - Cherokee County
              - 2006_orthos
              - GeoEye
              - dtm
              - LANDSAT7_SLC_OFF
              - LANDSAT7_SLC_ON
              - LANDSAT2_MSS
              - LANDSAT5_TM
              - LANDSAT1_MSS
  - ECWP (IWS)
  - Tiled Maps (TMS)
  - ArcIMS (ArcXML)
- Local data

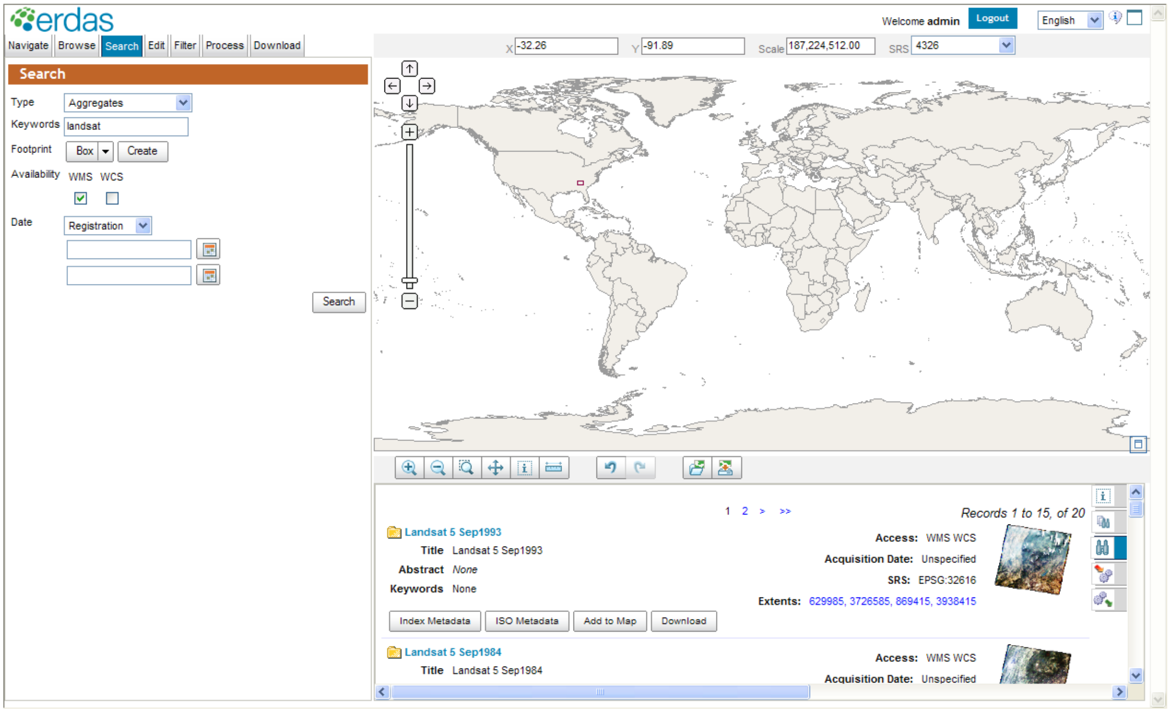
1 2 3 4 5 > >> Records 1 to 15, of 531

2006_orthos
Title 2006_orthos
Abstract None
Keywords None
Access: WMS WCS
Acquisition Date: Unspecified
SRS: EPSG:2240
Extents: 2147000, 1481000, 2272000, 1606000

267591
Title Coverage File: 267591.sid
Access: WMS WCS
Acquisition Date: 2006-09-24 20:18:54 -04:00

Search Results

When you conduct a search on the Search tab, the Search Results tab opens in the Results panel.



Type: Aggregates
Keywords: landsat
Footprint: Box
Availability: WMS ☑ WCS □
Date: Registration
X -32.26 Y -91.89 Scale 187,224,512.00 SRS 4326

Records 1 to 15, of 20

Landsat 5 Sep1993
Title: Landsat 5 Sep1993
Abstract: *None*
Keywords: None
Access: WMS WCS
Acquisition Date: Unspecified
SRS: EPSG:32616
Extents: 629985, 3726585, 869415, 3938415

Landsat 5 Sep1984
Title: Landsat 5 Sep1984
Access: WMS WCS
Acquisition Date: Unspecified

The listing for each result displays the name of the result, preceded by an icon that indicates whether that result is an aggregate (yellow folder) or a dataset (blue grid).

Each listing also displays certain metadata for each result, including the title, abstract, compatible web services, acquisition date, SRS, and spatial extent.

To highlight the spatial extent of the result on the map, hover your mouse over the listing. If you click the spatial extent information in the listing, the map will zoom in on the footprint of the result.

NOTE: If there are more than 500 results, the >> button does not display and you cannot automatically go to the last result.

Each listing also contains a group of buttons that allow you to perform certain actions on the selected result:

- **Index Metadata**: The whole set of coverage metadata extracted from the ebRIM structure.

- **View ISO Metadata**: Displays a window containing the original ISO 19115 metadata using ISO 19139 XML encoding. This document can contain additional metadata.

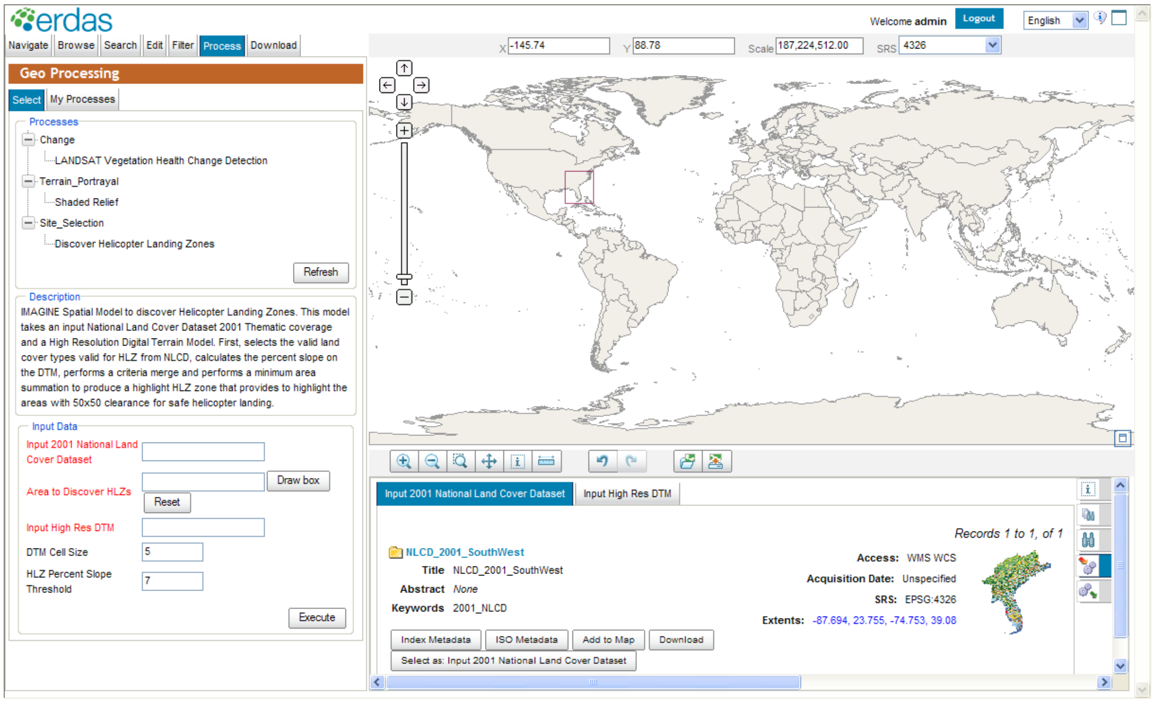
- **Add to Map**: Adds the selected element as one layer on the Navigate tab. You can add more than one result to the Navigate tab.

- **Download**: Adds the result to the Download tab.

GeoProcess Complex Inputs

The GeoProcess Complex Inputs tab is only used when you are working with processes in the Process tab.

When you click on an available process in that tab, the Web Client compares all of the datasets and aggregates to which you have access with the list of constraints for the process. It determines the datasets that will work as an input for that process and displays them on the GeoProcess Complex Inputs tab.



X -145.74 Y 88.78 Scale 187,224,512.00 SRS 4326

Description

IMAGINE Spatial Model to discover Helicopter Landing Zones. This model takes an input National Land Cover Dataset 2001 Thematic coverage and a High Resolution Digital Terrain Model. First, selects the valid land cover types valid for HLZ from NLCD, calculates the percent slope on the DTM, performs a criteria merge and performs a minimum area summation to produce a highlight HLZ zone that provides to highlight the areas with 50x50 clearance for safe helicopter landing.

Input Data

Input 2001 National Land Cover Dataset

Area to Discover HLZs

Input High Res DTM

DTM Cell Size 5

HLZ Percent Slope Threshold 7

Records 1 to 1, of 1

NLCD_2001_SouthWest

Title NLCD_2001_SouthWest

Abstract *None*

Keywords 2001_NLCD

Access: WMS WCS

Acquisition Date: Unspecified

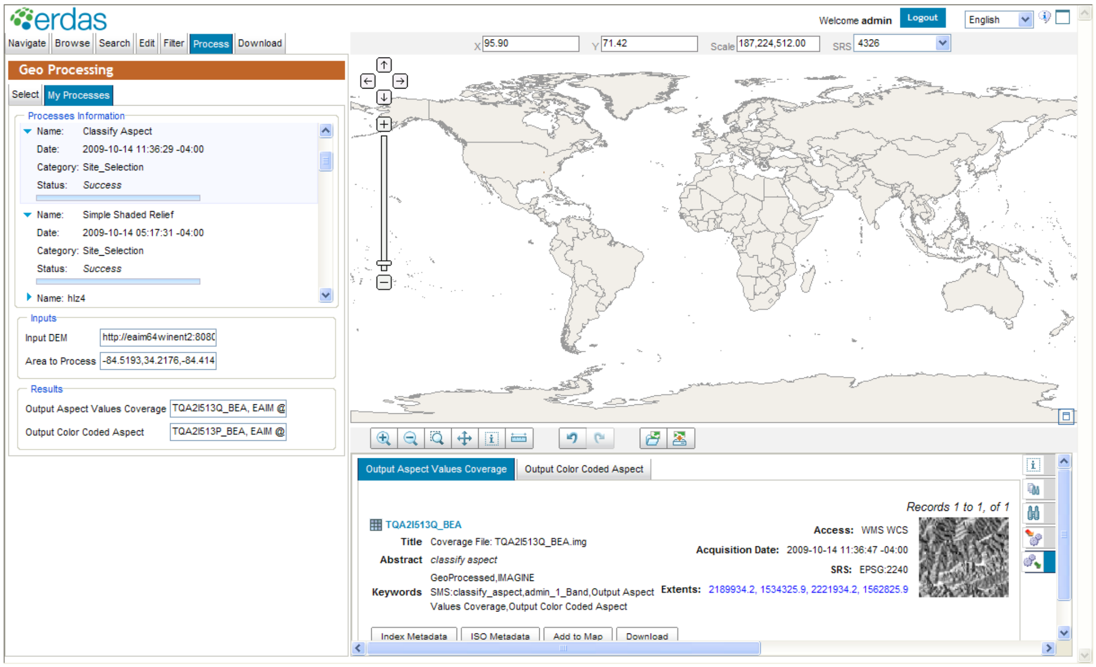
SRS: EPSG:4326

Extents: -87.694, 23.755, -74.753, 39.08

GeoProcess Complex Outputs

The GeoProcess Complex Outputs tab is only used when you are working with processes in the Process tab.

Immediately after a process has completed execution, the resulting datasets are shown on this tab. If you click on a previously executed process on the Process tab, its results are shown on the GeoProcess Complex Outputs tab.



X 95.90 Y 71.42 Scale 187,224,512.00 SRS 4326

Geo Processing

Processes Information

Name: Classify Aspect
Date: 2009-10-14 11:36:29 -04:00
Category: Site_Selection
Status: *Success*

Name: Simple Shaded Relief
Date: 2009-10-14 05:17:31 -04:00
Category: Site_Selection
Status: *Success*

Name: hlz4

Inputs

Input DEM: http://eaim64winent2:808(
Area to Process: -84.5193,34.2176,-84.414

Results

Output Aspect Values Coverage: TQA2I513Q_BEA, EAIM @
Output Color Coded Aspect: TQA2I513P_BEA, EAIM @

Output Aspect Values Coverage | Output Color Coded Aspect

Records 1 to 1, of 1

TQA2I513Q_BEA

Title Coverage File: TQA2I513Q_BEA.img
Abstract *classify aspect*
Keywords GeoProcessed,IMAGINE SMS:classify_aspect,admin_1_Band,Output Aspect Values Coverage,Output Color Coded Aspect

Access: WMS WCS
Acquisition Date: 2009-10-14 11:36:47 -04:00
SRS: EPSG:2240
Extents: 2189934.2, 1534325.9, 2221934.2, 1562825.9

Overview of the ERDAS APOLLO Web Tools

In this chapter:

Introduction

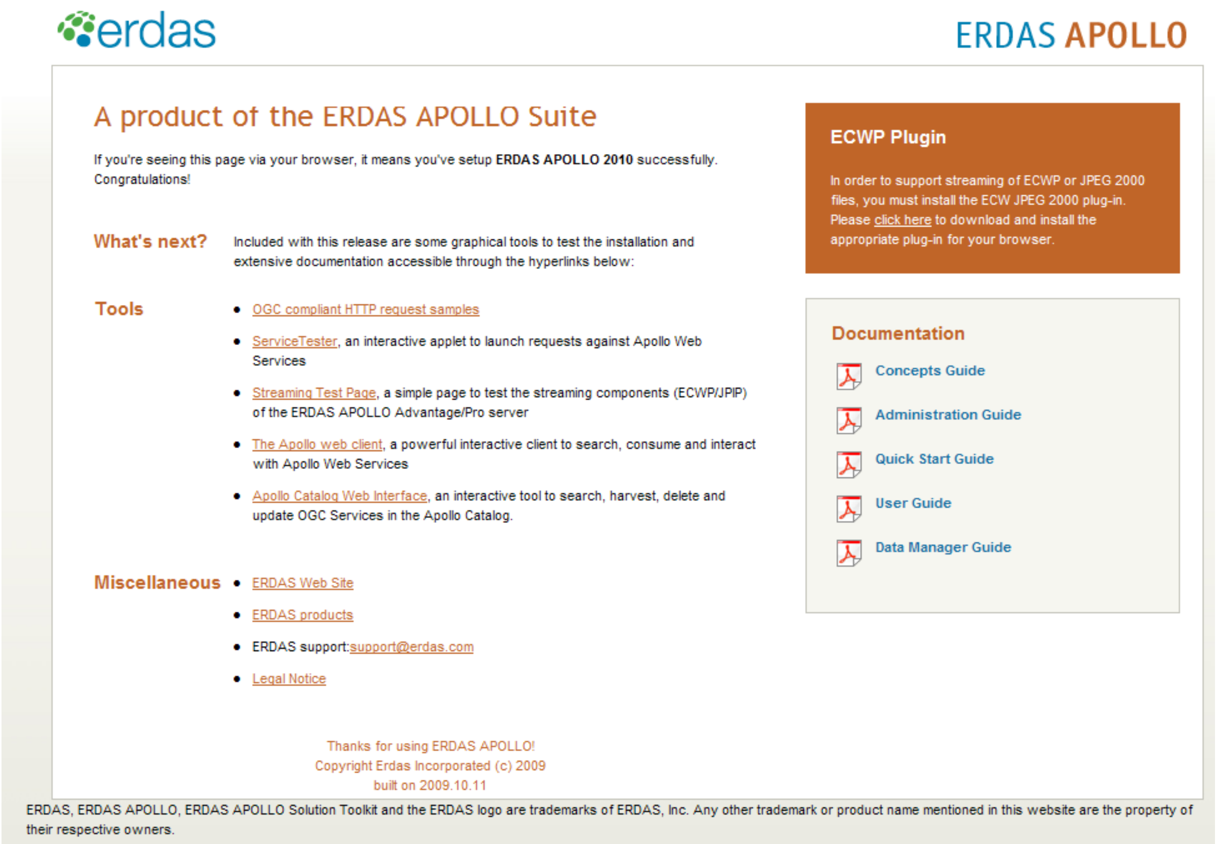
The ERDAS APOLLO products come with a web page that was installed and published when you installed the ERDAS APOLLO Server software on your server computer.

The default URL of the ERDAS APOLLO web page is http://<*server_name*>:<*portnumber*>/erdas-apollo.

From this web page, you can access the following ERDAS APOLLO tools:

- OGC compliant HTTP request samples
- ServiceTester
- Streaming Test Page
- ERDAS APOLLO web client
- ERDAS APOLLO Catalog Web Interface

Figure 23: The Main ERDAS APOLLO Web Page



A product of the ERDAS APOLLO Suite

If you're seeing this page via your browser, it means you've setup **ERDAS APOLLO 2010** successfully. Congratulations!

What's next? Included with this release are some graphical tools to test the installation and extensive documentation accessible through the hyperlinks below:

Tools

- OGC compliant HTTP request samples
- ServiceTester, an interactive applet to launch requests against Apollo Web Services
- Streaming Test Page, a simple page to test the streaming components (ECWP/JPIP) of the ERDAS APOLLO Advantage/Pro server
- The Apollo web client, a powerful interactive client to search, consume and interact with Apollo Web Services
- Apollo Catalog Web Interface, an interactive tool to search, harvest, delete and update OGC Services in the Apollo Catalog.

Miscellaneous

- ERDAS Web Site
- ERDAS products
- ERDAS support:support@erdas.com
- Legal Notice

Thanks for using ERDAS APOLLO!
Copyright Erdas Incorporated (c) 2009
built on 2009.10.11

ECWP Plugin

In order to support streaming of ECWP or JPEG 2000 files, you must install the ECW JPEG 2000 plug-in. Please click here to download and install the appropriate plug-in for your browser.

Documentation

- Concepts Guide
- Administration Guide
- Quick Start Guide
- User Guide
- Data Manager Guide

ERDAS, ERDAS APOLLO, ERDAS APOLLO Solution Toolkit and the ERDAS logo are trademarks of ERDAS, Inc. Any other trademark or product name mentioned in this website are the property of their respective owners.

Prerequisites for Use

The web client was specifically designed so that people could easily access the information in the ERDAS APOLLO system without having to got through a time-consuming installation process or learn how to use a new application.

To use the web client, you will only need:

- A compatible internet browser, either
  Microsoft Internet Explorer 7.x or later
  OR Mozilla Firefox

- the ERDAS Web Plugin

ERDAS Web Plugin - ERDAS APOLLO 2010 provides new ERDAS Web Plugins. You must uninstall any previous versions of the ERMapper Web Plugins and install the new version of the ERDAS Web Plugins before you can stream ECW and JPEG2000 files through the ECWP and JPIP protocols and use the built-in functionality of the ERDAS APOLLO Web Client.

To ensure that you have the correct plugin:

1. Uninstall any previous version of the ERMapper Web Plugin.
2. Go the server home page, http://<*hostname*>:8080/ .
3. Click the link inside the **ECWP Plugin** box on the top right of the page.
4. The File Download dialog box opens. Click **Run** to install the plugins.

The OGC Compliant HTTP Request Samples

The server installation installs a set of sample data and makes it available through OGC services. The links provided in the table invoke the predefined WMS, WFS, WCS requests for specific demo data. Results of invoking these requests are immediate when the ERDAS APOLLO Server is running and configured properly.

The ServiceTester

The ERDAS APOLLO Service Tester provides a GUI for the user to formulate the correct syntax for various WMS, WFS, and WCS requests and provides a text display window to view the results certain requests.

The Streaming Test Page

The Streaming Test Page allows you to see if your server has the browser plugin that is required in order to use Enhanced Compression Wavelet Protocol (ECWP) and JPEG 2000 Interactive Protocol (JPIP) streaming to view imagery.

If you do not have the plugin, the page provides a link so that you can download it.

If you do have it, the page provides an example of how it works.

The ERDAS APOLLO Web Client

The ERDAS APOLLO web client is the part of the ERDAS APOLLO system that allows you to make your imagery and data available to all of the different people who need it. The web client was installed and published when you installed the ERDAS APOLLO Server software on your server computer and can be accessed over the internet.

The default URL of the ERDAS APOLLO Web Client is http://*localhost*:8080/apollo-client/.

The ERDAS APOLLO Catalog Web Interface

The Catalog web interface is the out-of-the-box access point to browse and manage resources that have been registered in the catalog. It also allows the publishing of OGC services, such as those created using the Administration Console, or any other OGC-compliant service.

The Catalog web interface also offers specific administrator tools, such as a CSW interface tester to test the CSW access point of the catalog.

ERDAS APOLLO Catalog Overview

Introduction

The ERDAS APOLLO Catalog provides a way to publish, discover and administrate OGC services descriptions and metadata.

The ERDAS APOLLO Catalog uses an object model derived from the OGC OWS metadata models to store OGC services descriptions in its spatial database.

The ERDAS APOLLO Catalog offers different means to access its data, including an optimized proprietary protocol, a CSW ebRIM compliant interface, or a RESTful endpoint with various encoding formats (see the Encodings section of the Catalog Service chapter in the main guide).

This overview describes the Catalog web interface, built on top of the HTML REST representation. This web interface is the easiest way to discover and manage the content of the catalog. To understand how this web interface relates to the REST service, or for a description of the other access points of the APOLLO Catalog please see the Catalog Service section of the main guide.

Startup

Once ERDAS APOLLO Server is installed and deployed, the Catalog web interface is available at http://host:port/apollo-catalog.

When first accessing the web interface, the welcome page is displayed, with simply a query form to search for registered objects, and a display of recently published services. From there, you can either start a search, or login using the upper right authentication form.

Detailed functionalities

Querying and browsing the catalog

The query form is the entry point to query the catalog using keywords in the web interface. The search engine tries to match given keywords with the name, title, or tags of the catalog records.

Catalog results can also be narrowed to some object type, using the object type dropdown. Clicking a different object type will keep the current keyword search, while updating the object type criteria.

Publishing resources

To be able to publish resources, you must first login using the panel on the upper right. Once authenticated, several new options will appear on the screen; in particular, a 'Publish' tab will appear on the upper left (see the Catalog Configuration of the main guide to learn about configuring security in the catalog, in particular granting publishing rights to a user).

Clicking on this tab will open the publish page, where you can register new services in the catalog. The publish form takes the URL of an OGC service, and when submitted, tries to fetch its metadata and store them in the catalog.

The service type dropdown allows you to specify the type of the service to be harvested (WMS, WFS, WCS). If the type is left on its default value (Auto), the web client will try to guess the service type from the URL that is entered.

Once the service is harvested, the application is redirected to the search panel displaying the details of the newly harvested service.

Managing catalog records

Once authenticated, extra operations appear for each displayed object in the search result panel. These operations allow authorized users to delete each object, and also re-harvest service objects. Authorized users for those actions are the original publisher of the object, and any user with the admin role.

Testing the CSW interface

When logged in with an administrator role, an extra 'CSW' tab appears in the interface. This panel offers a way to send CSW requests directly to the CSW ebRIM endpoint of the APOLLO Catalog.The left panel can be filled with a valid CSW request. When clicking on 'Post request', the request will be sent to the server, and its result displayed in the right panel. A set of typical CSW requests are available in the dropdown below the left form.

Managing the Catalog

When authenticated, the admin tab (on the upper left) offers the ability to re-index the keywords. In most cases this is not necessary unless an installation of a catalog on an existing database is made.

Catalog Web Interface

The Catalog Web Interface is a web application that allows you to manager your Catalog by publishing data in it, and searching for and browsing the data in it. It also offers administration tools for the Catalog service.

Log In to the Web Application

You can search for and browse the public data in the Catalog without logging in, but before you can publish data in the Catalog or use the administration tools, you will need to log in with a user name and password.

Searching and Browsing Content

the *Browse* page (see upper left of the web interface) allows you to browse the ERDAS APOLLO Catalog. This page shows a form containing a dropdown list and a text field.

The drop-down list allows you to search for a specific object type.

The object types are :

- All Objects
  (use this option to search for custom objects such as documents, movies, sound clips, or photographs)
- All Services
- Coverage Services
- Coverage Layers
- Vector Services
- Vector Layers
- Map Services
- Map Layers

In the text field, the user can enters keywords matching data that need to be discovered. Those keywords can be specified using advanced formatting and facilities. Here is an excerpt:

Table 2: Keywords Operators for Advanced Searches

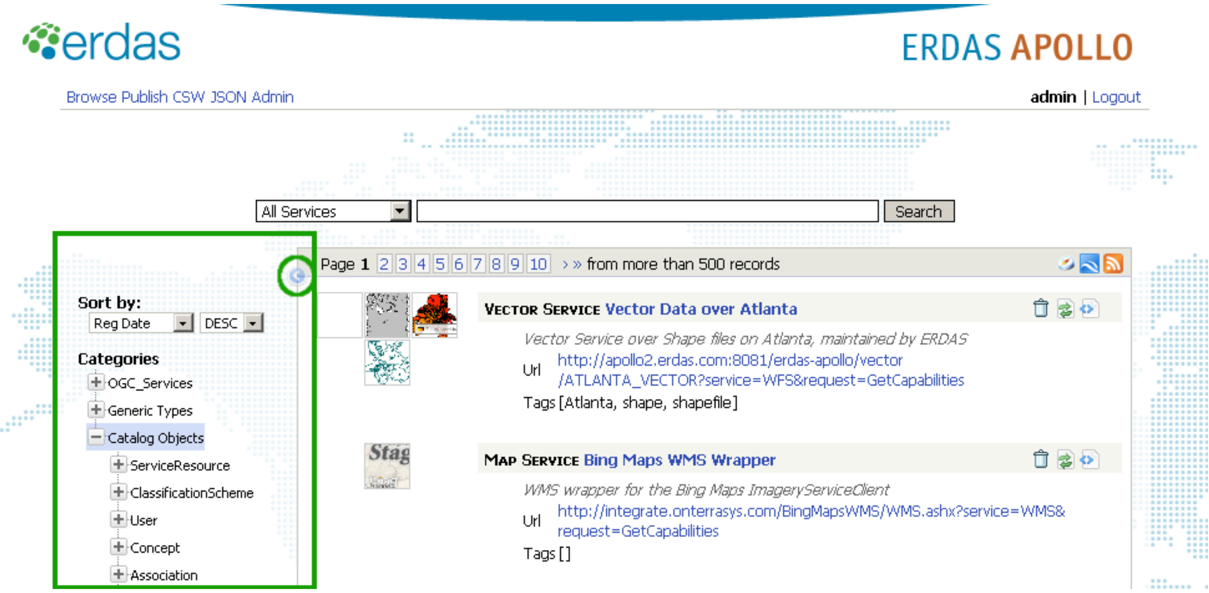
Symbol	Usage	Effect
AND	Between two keywords.	Logical AND. Example: "road AND Atlanta".
OR	Between two keywords.	Logical OR. Example: "road OR Atlanta".
()	Surrounding keywords.	Logical Group. Example: "(road OR Atlanta) AND city".

Advanced Search

The Advanced Search panel can be brought up by clicking on the arrow on the left edge of the Browse panel.

This panel offers sorting options, and also a tree view of the object types available in the ERDAS APOLLO Catalog. Single-clicking on an object type will search for records of that type. Double-clicking will bring the definition of the object type itself.

Use the contextual help (by pressing CTRL+ALT) to see more information regarding that panel.



VECTOR SERVICE Vector Data over Atlanta

Vector Service over Shape files on Atlanta, maintained by ERDAS

Url http://apollo2.erdas.com:8081/erdas-apollo/vector/ATLANTA_VECTOR?service=WFS&request=GetCapabilities

Tags [Atlanta, shape, shapefile]

MAP SERVICE Bing Maps WMS Wrapper

WMS wrapper for the Bing Maps ImageryServiceClient

Url http://integrate.onterrasys.com/BingMapsWMS/WMS.ashx?service=WMS&request=GetCapabilities

Tags []

Publishing Content

This feature allows you to register service providers, geospatial processes, and custom objects in the ERDAS APOLLO Catalog.

To publish an item to the ERDAS APOLLO Catalog:

1. Click the **Login** link located in the top right corner of the Catalog Web Interface page.

2. Enter your user name and password in the boxes that appear. Click **OK**.

3. Click the **Publish** link on the top left corner of the page.

4. Select a resource type in the **Select Resource Type** dropdown box. The options are:

   - W*S
   - WFS
   - WMS
   - WCS
   - WPS
   - process
   - generic

5. The W*S, WFS, WMS, WCS, and WPS options will only ask you to provide the URL of the object in the URL field.

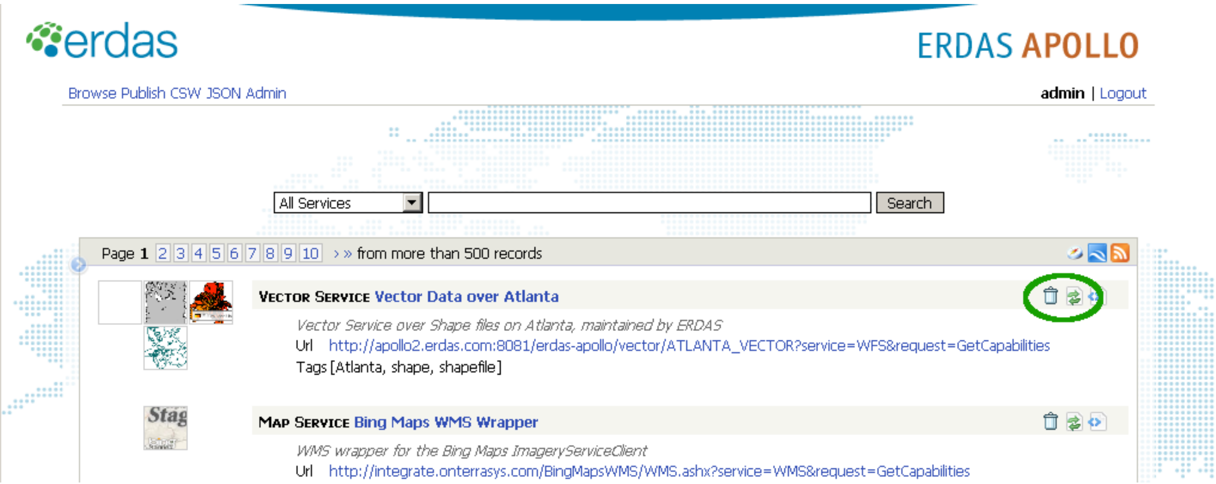
   The process options allows you to provide a URL, a file, or text .

   The generic option allows you to provide the URL to a custom object (such as a movie, pdf, or photograph), the file name of a custom object, or the text of a custom object management file that will allow you to perform bulk changes to the custom objects in the ERDAS APOLLO Catalog.

6. Click the **Publish** button (to the right of the URL field).

The current user is considered the owner of data published this way. This means that only this user and the administrator will be allowed to delete or refresh (re-publish) it using the Delete and Refresh links.

Figure 24: Advanced Operations



from more than 500 records

VECTOR SERVICE Vector Data over Atlanta

Vector Service over Shape files on Atlanta, maintained by ERDAS

Url http://apollo2.erdas.com:8081/erdas-apollo/vector/ATLANTA_VECTOR?service=WFS&request=GetCapabilities

Tags [Atlanta, shape, shapefile]

MAP SERVICE Bing Maps WMS Wrapper

WMS wrapper for the Bing Maps ImageryServiceClient

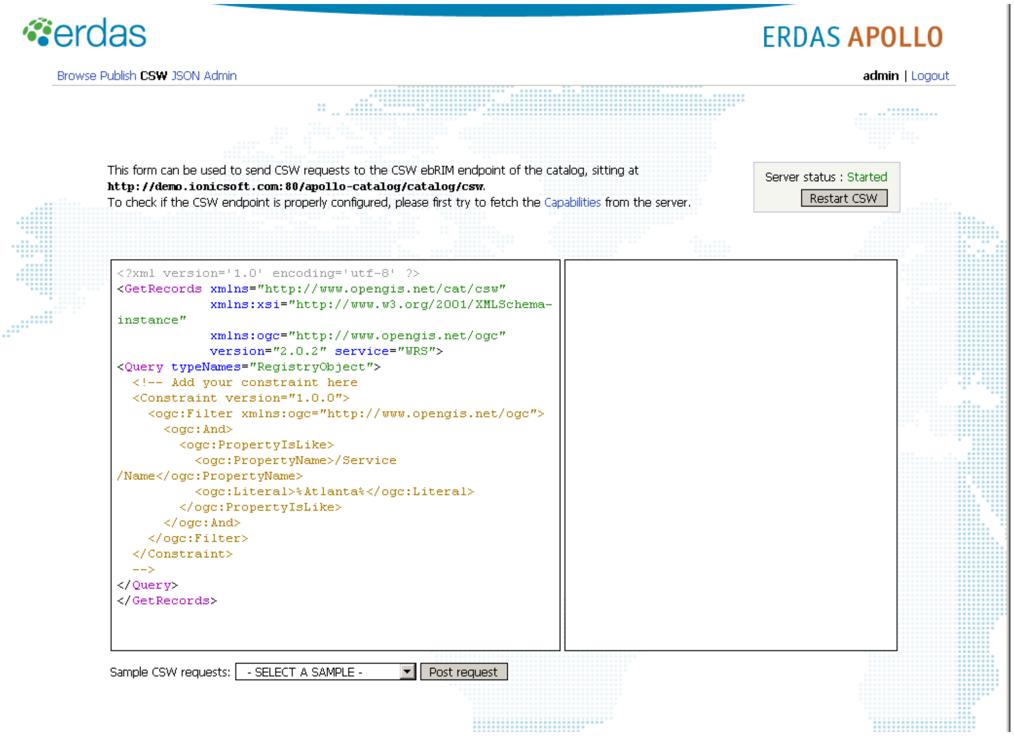
Url http://integrate.onterrasys.com/BingMapsWMS/WMS.ashx?service=WMS&request=GetCapabilities

Testing the CSW Endpoint

If you are logged in to the Catalog Web Interface as an administrative user, the **CSW** link appears at the top left of the page.

This page offers a way to send CSW requests directly to the CSW ebRIM endpoint of the Catalog, sitting at `http://<serverURL>/erdas-apollo/catalog/csw`.

Figure 25: CSW Panel



This form can be used to send CSW requests to the CSW ebRIM endpoint of the catalog, sitting at **http://demo.ionicsoft.com:80/apollo-catalog/catalog/csw**.
To check if the CSW endpoint is properly configured, please first try to fetch the Capabilities from the server.

Server status : Started

```
<?xml version='1.0' encoding='utf-8' ?>
<GetRecords xmlns="http://www.opengis.net/cat/csw"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
            xmlns:ogc="http://www.opengis.net/ogc"
            version="2.0.2" service="WRS">
<Query typeNames="RegistryObject">
  <!-- Add your constraint here
  <Constraint version="1.0.0">
    <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:And>
        <ogc:PropertyIsLike>
          <ogc:PropertyName>/Service
/Name</ogc:PropertyName>
          <ogc:Literal>%Atlanta%</ogc:Literal>
        </ogc:PropertyIsLike>
      </ogc:And>
    </ogc:Filter>
  </Constraint>
  -->
</Query>
</GetRecords>
```

Sample CSW requests: - SELECT A SAMPLE -

The left panel can be used to edit CSW requests. When clicking on 'Post request', the request will be sent to the server, and the response from the server will be displayed in the right panel. A set of typical CSW requests are available in the drop down below the left form.

On the upper right of the page, a panel indicates the status of the CSW endpoint, i.e. whether it is started or not, together with a button to force a restart of the CSW endpoint. It must be noted that the CSW endpoint will start automatically on demand; this status and button are for debug purposes only, to force a restart and a cache flush of the CSW stack.

Administration Options

The catalog web interface also offers some facilities to manage the catalog. Those functionalities are in the *Admin* page (see upper left of the web interface).

Only users with an 'admin' role can access this page: if the user doesn't have this role, the *Admin* link is hidden to him.

Here are the functionalities:

- See the list of the roles.

- Re-index the keywords: this operation is helpful if, for whatever reason, the lucene index is not synchronized anymore with the catalog content. The main case is to be able to move the server without having to move the lucene index as well.

- Manage DB schemas: this will bring up a page that lists the currently installed DB schema(s) and their version(s). If the current DB schema is not in sync with the ERDAS APOLLO software version, this page allows you to run the upgrade process.

The administrator's activity scope is extended to the data rights. Indeed, the administrator has all rights on data: he can delete or refresh everything, even he is not the owner of data.

WMS HTTP Requests

In this chapter:

WMS Versions

There are three main version sets of the OGC-WMS specification: WMS 1.0.x and earlier, WMS 1.1.1, and WMS 1.3.0 (also approved as ISO 19128). Each has a different syntax. Below is a brief summary of the differences between WMS 1.0.0, WMS 1.1.1, and WMS 1.3.0 for GetCapabilities, GetMap, and GetFeatureInfo requests. The parameter names are case-insensitive but the parameter values are case-sensitive.

Table 3: WMS GetCapabilities Parameters

WMS 1.0.0	WMS 1.1.1	WMS 1.3.0
WMTVER=1.0.0	VERSION=1.1.1	VERSION=1.3.0
REQUEST=capabilities	REQUEST=GetCapabilities	REQUEST=GetCapabilities
	SERVICE=WMS	SERVICE=WMS

Table 4: WMS GetMap Parameters

WMS 1.0.0	WMS 1.1.1	WMS 1.3.0
WMTVER=1.0.0	VERSION=1.1.1	VERSION=1.3.0
REQUEST=map	REQUEST=GetMap	REQUEST=GetMap
BBOX=xmin,ymin,xmax,ymax	BBOX=xmin,ymin,xmax,ymax	BBOX=xmin,ymin,xmax,ymax
SRS=EPSG:4326 (or EPSG:26986 or AUTO:42003,1,-100,45...)	SRS=EPSG:4326 (or EPSG:26986 or AUTO:42003,1,-100,45)	CRS=CRS:84 (or EPSG:26986 or AUTO2:42003,1,-100,45)
FORMAT=GIF (or JPEG, PNG, SVG, ...)	FORMAT=image/gif (or image/jpeg, image/png, ...)	FORMAT=image/gif (or image/jpeg, image/png, ...)

EXCEPTIONS=INIMAGE (or BLANK, XML)	EXCEPTIONS=application/ vnd.ogc.se_inimage (or application/vnd.ogc.se_blank, application/vnd.ogc.se_xml)	EXCEPTIONS=XML (or BLANK, INIMAGE)

Table 5: WMS GetFeatureInfo Parameters

WMS 1.0.0	**WMS 1.1.1**	**WMS 1.3.0**
WMTVER=1.0.0	VERSION=1.1.1	VERSION=1.3.0
REQUEST=feature_info	REQUEST=GetFeatureInfo	REQUEST=GetFeatureInfo
INFO_FORMAT=MIME (or GML.1, GML.2, GML.3)	INFO_FORMAT=text/html (or application/vnd.ogc.gml)	INFO_FORMAT=text/html (or application/vnd.ogc.gml)
QUERY_LAYERS= LAYER1,LAYER2	QUERY_LAYERS= LAYER1,LAYER2	QUERY_LAYERS= LAYER1,LAYER2
EXCEPTIONS=XML	EXCEPTIONS=application/ vnd.ogc.se_xml	EXCEPTIONS=XML

Another big difference between each OGC WMS specification is the document returned by a GetCapabilities request. Describing the whole content of the document for each version of the specification is out of the scope of this guide. Instead, the following table emphasizes the main differences between them. It does not describe all of the variations. The purpose is to give an overview of what has changed. For a complete description of the content of each specification, please refer to the appropriate OGC specification document.

In the table below, each line represents a set of attributes or subelements of the capabilities element. For WMS 1.1.1, the list is sometimes expressed by its difference compared to the WMS 1.0.0 corresponding element, and the WMS 1.3.0 lists compared to the WMS 1.1.1 corresponding element.

Each attribute or element is either not suffixed, or suffixed with "?", "+" or "*". This is inherited from the DTD syntax, and it respectively means 1..1, 0..1, 1..n and 0..n .

Table 6: WMS GetCapabilities Response

Element \ Specification Version	**WMS 1.0.0**	**WMS 1.1.1**	**WMS 1.3.0**
document mime-type	text/xml	application/vnd.ogc.wms_xml	text/xml
Validating document	DTD	DTD	XML Schema

Root element	WMT_MS_Capabilities	WMT_MS_Capabilities	WMS_Capabilities
Attributes of root element	version, updateSequence	version, updateSequence	version, updateSequence
Children of root element	Service, Capability	Service, Capability	Service, Capability
Service children	Name, Title, Abstract?, Keywords?, OnlineResource, Fees?, AccessConstraints?	Removed: Keywords? Added: KeywordList?, ContactInformation?	Added: LayerLimit?, MaxWidth?, MaxHeight?
Capability children	Request, Exception?, VendorSpecificCapabilities?, Layer?	Removed: Exception; Added: Exception?, UserDefinedSymbolization?	Removed: Exception?, UserDefinedSymbolization? (becomes a child of "VendorSpecificCapabilities"); Added:Exception
Request children	(Map | Capabilities | FeatureInfo)+	GetCapabilities, GetMap, GetFeatureInfo?, DescribeLayer?, GetLegendGraphic?, GetStyles?, PutStyles?	Same as WMS 1.1.1
Layer attributes	queryable	queryable, cascaded, opaque, noSubsets, fixedWidth, fixedHeight	Same as WMS 1.1.1
Layer children	Name?, Title, Abstract?, Keywords?, SRS?, LatLonBoundingBox?, BoundingBox*, DataURL?, Style*, ScaleHint?, Layer*	Removed: Keywords?, SRS?; Added: KeywordList?, SRS*, Dimension*, Extent*, Attribution?, AuthorityURL*, Identifier*, MetadataURL*, FeatureListURL*	Removed: SRS?, LatLonBoundingBox?, ScaleHint? Added: CRS*, EX_GeographicBoundingBox?, MinScaleDenominator?, MaxScaleDenominator?
Style children	Name, Title, Abstract?, StyleURL?	Name, Title, Abstract?, LegendURL*, StyleSheetURL?, StyleURL?	Same as WMS 1.1.1
Geographic Bounding Box name and attributes	LatLonBoundingBox(minx, miny, maxx, maxy)	Same as WMS 1.0.0	EX_GeographicBoundingBox(westBoundLongitude, eastBoundLongitude, southBoundLatitude, northBoundLatitude)

Format content	Labels (WMS-XML, PNG, ...)	Mime-Types (application.vnd.ogc.se_xml, text/html, image/png)	Mime-types for output of requests, "XML" for Exceptions
Coordinate System element name and content namespaces	SRS (EPSG, AUTO)	SRS (EPSG, AUTO)	CRS ('CRS', 'EPSG', 'AUTO2' or a URL)

WMS Key-Value Pair Parameters

The table below, extracted from the OGC-WMS 1.1.1 Implementation Specification, lists all the key-value pair (KVP) parameters that can be used in a GetMap request. For more details, please refer to the specification itself.

Table 7: The Parameters of an OGC-WMS 1.1.1 GetMap Request

Request Parameter	Required/ Optional	Description
VERSION=version	R	Request version.
REQUEST=GetMap	R	Request name.
LAYERS=layer_list	R	Comma-separated list of one or more map layers. Optional if SLD parameter is present.
STYLES=style_list	R	Comma-separated list of one rendering style per requested layer. Optional if SLD parameter is present.
SRS=namespace:identifier (In WMS 1.3.0: CRS)	R	Spatial Reference System.
BBOX=minx,miny,maxx,maxy	R	Bounding box corners (lower left, upper right) in SRS units.
WIDTH=output_width	R	Width in pixels of map picture.
HEIGHT=output_height	R	Height in pixels of map picture.
FORMAT=output_format	R	Output format of map.
TRANSPARENT=TRUE|FALSE	O	Background transparency of map (default=FALSE).
BGCOLOR=color_value	O	Hexadecimal red_green_blue color value for the background color (default=0xFFFFFF).
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=SE_XML).

TIME=time	O	Time value of desired layer.
ELEVATION=elevation	O	Elevation of desired layer.
Other sample dimension(s)	O	Value of other dimensions as necessary.
Vendor-specific parameters	O	Optional experimental parameters.
The following parameters are used only with Web Map Services that support the Styled Layer Descriptor (SLD) specification.		
SLD=styled_layer_descriptor_URL	O	URL of Styled Layer Descriptor (as defined in SLD Specification).
WFS=web_feature_service_URL	O	URL of Web Feature Service providing features to be symbolized using SLD.
The following parameters are used only with ERDAS APOLLO Web Map Services		
FILTER=ogc_filter	O	Filter using the XML syntax for the OGC Filter Encoding 1.0.0 specification, if the WMS is build on top of a WFS.
QUALITY=value	O	Image output quality between 0 and 100. For JPG format, it corresponds to the compression ratio. For PNG, values below 50 means 8 bits PNG, and 24 bits PNG upon 50.

Here is a table extracted from the OGC-WMS 1.1.1 Implementation Specification, listing all the KVP parameters that can be used in a GetFeatureInfo request.

Table 8: The Parameters of an OGC-WMS 1.1.1 GetFeatureInfo Request

Request Parameter	**Required/ Optional**	**Description**
VERSION=version	R	Request version.
REQUEST=GetFeatureInfo	R	Request name.
<Map request copy>	R	Partial copy of the map request parameters that generated the map for which information is desired, except VERSION and REQUEST already present in the GetFeatureInfo

QUERY_LAYERS=layer_list	R	Comma-separated list of one or more layers to be queried.
INFO_FORMAT=output_format	O	Return format of feature information (MIME type).
FEATURE_COUNT=number	O	Number of features about which to return information (default=1).
X=pixel_column	R	X coordinate in pixels of feature (measured from upper left corner=0)
Y=pixel_row	R	Y coordinate in pixels of feature (measured from upper left corner=0)
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=application/vnd.ogc.se_xml).

The table below describes the KVP parameters of the GetLegendGraphic request, as given in the OGC-SLD 1.1 specification (05-078r3 - SLD Profile of WMS).

Table 9: The Parameters of an OGC-SLD 1.1.0 GetLegendGraphic Request

Request Parameter	**Required/ Optional**	**Description**
SERVICE=WMS	R	Service type.
VERSION=version	R	Request version (typically "1.3.0").
SLD_VERSION=version	R	Specification version of the SLD specification (typically "1.1.0").
REQUEST=GetLegendGraphic	R	Request name.
LAYER=layer_name	R	One map layer.
STYLE=style_name	O	One optional rendering style name for the layer.
REMOTE_OWS_TYPE=service_type	O	Service type ("WFS" or "WCS").
REMOTE_OWS_URL=url_of_ows	O	URL of service providing features to be symbolized.
FEATURETYPE=re_type	O	Feature type for which to produce the legend graphic (useless if the layer has a single feature type).

Request Parameter	Required/ Optional	Description
COVERAGE=coverage	O	Coverage for which to produce the legend graphic (useless if the layer has a single coverage).
RULE=rule	O	Rule of style to produce legend graphic for, if applicable.
SCALE=scale	O	Standardized scale denominator defined in the Symbology Encoding spec.
SLD=styled_layer_descriptor_URL	O	URL of an external SLD document.
SLD_BODY=SLD_document	O	XML Content of an SLD document.
FORMAT=output_format	R	Mime type of the Output.
WIDTH=output_width	R	Width in pixels of graphic in pixels, as a hint.
HEIGHT=output_height	R	Height in pixels of graphic in pixels, as a hint.
EXCEPTIONS=exception_format	O	The mime-type of the format in which exceptions are to be reported by the WMS.

WMS HTTP POST Requests

The ERDAS APOLLO WMS services support HTTP requests using HTTP-POST. An OpenGIS project document (Ref: 02-017r1 - Web Map Service Implementation Specification - Part 2: XML for Requests Using HTTP POST) describes the structure and encoding for WMS POST requests. The greatest benefit is realized in the GetMap operation, where the comma-separated list of layer names in HTTP GET can be replaced by a sequence of XML elements, each of which is either a named or a user-defined layer, and directly associates style and filter information within each layer. The GetFeatureInfo operation, which includes most of a GetMap request, benefits the user in a similar way. The OGC document presents the XML schema for the GetCapabilities, GetMap and GetFeatureInfo requests.

The OGC proposal talks about an optional <Section> element to include in the GetCapabilities request to retrieve only one of the sections of a complete Capabilities XML document. The ERDAS APOLLO implementation does not support this feature. It always returns the entire Capabilities document.

Here is a sample GetCapabilities request on a WMS server using a POST request:

```
<GetCapabilities version="1.2.0" service="WMS" />
```

The GetMap request includes exactly one <StyledLayerDescriptor> element that defines the map layers which are to appear in the returned map. Layers are drawn in the order that they appear in the XML, with the first one drawn at the bottom. The StyledLayerDescriptor element contains named Layer(s), possibly named Style(s), or more complex expressions depending on the provider type.

Here is a sample GetMap request using a POST request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ogc:GetMap xmlns:ogc="http://www.opengis.net/ows"
            xmlns:gml="http://www.opengis.net/gml"
            xmlns:sld="http://www.opengis.net/sld"
            env:encodingStyle="http://www.w3.org/2001/09/soap-encoding"
            version="1.2.0" service="WMS">
   <sld:StyledLayerDescriptor version="1.0.0">
      <sld:NamedLayer>
         <sld:Name>HIGHWAYS</sld:Name>
         <sld:NamedStyle>
            <sld:Name>default</sld:Name>
         </sld:NamedStyle>
      </sld:NamedLayer>
   </sld:StyledLayerDescriptor>
   <BoundingBox srsName="http://www.opengis.net/gml/srs/epsg.xml#25832">
      <gml:coord>
         <gml:X>550000</gml:X>
         <gml:Y>6220000</gml:Y>
      </gml:coord>
      <gml:coord>
         <gml:X>551000</gml:X>
         <gml:Y>6221000</gml:Y>
      </gml:coords>
   </BoundingBox>
   <Output>
      <Format>image/png</Format>
      <Transparent>false</Transparent>
      <Size>
         <Width>500</Width>
         <Height>500</Height>
      </Size>
   </Output>
 </ogc:GetMap>
```

No GetFeatureInfo request has been defined using the HTTP-POST protocol.

WSDL and WMS HTTP SOAP Requests

In order to be valid in terms of a Service Oriented Architecture (SOA), a web service has to publish a Web Service Description Language (WSDL) document that describes its content and allows recording in a UDDI registry, such as a Systinet Registry. As soon as it is registered, the service can be invoked using the SOAP protocol. This is a particular type of XML encoding embedded in a classical HTTP request or response.

OGC has not yet voted on a standard for a WMS to provide its description in WSDL. We have chosen to follow one of the OGC candidate solutions, a request=getWSDL request type. The syntax is similar to a classical HTTP-GET GetCapabilities request, except that the request name is "getWSDL" and an optional "SoapOnly" parameter is allowed. Note that the "Service" parameter will determine the nature of the output WSDL, as each service type (WMS, WFS, WCS, ...) has a different WSDL description. The following is an example of a GetWSDL request:

```
http://localhost:8080/erdas-apollo/coverage/ATLANTA_MOSAIC?version=1.1.1&service=WMS&reque
st=getWSDL&SoapOnly=true
```

In the context of OGC Web Map Services, SOAP requests are very similar to POST requests: they contain a XML body which must be encapsulated into SOAP-specific elements in the case of SOAP requests. The root element is "Envelope" and its only child is a "Body" element.

Below is a sample GetCapabilities request on a WMS server using a SOAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:gml="http://www.opengis.net/gml">
 <SOAP-ENV:Body>
  <GetCapabilities version="1.2.0" service="WMS" />
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Coverages Technology Overview

IN this chapter:

- **The Coverage Concept**
- **The Web Coverage Service (WCS)**
- **The Coverage Portrayal Service (CPS)**

The *ERDAS APOLLO Concepts Guide* briefly describes coverages, Web Coverage Services, and the portrayal of these services. Please refer to that guide for an overview of this service.

This chapter provides a more detailed description of these concepts and mechanisms.

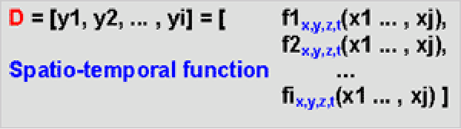
The Coverage Concept

Coverages are digital geospatial information comprised of regularly spaced locations along the 1st, 2nd, or 3rd axes of a spatial coordinate reference system representing space-varying phenomena. A coverage with a homogeneous range set defines at each location in the domain either a single (scalar) value, e.g., elevation, or a series (array / tensor) of values all defined in the same way, such as brightness values in different parts of the electromagnetic spectrum.

Fundamentally, coverages (and images) provide a n-dimensional (where n is usually 2, and occasionally 3 or higher) “view” of some (usually more complex) space of geographic features. ERDAS APOLLO sets the “view” to be geospatially registered to the Earth.

A coverage is a function from a spatiotemporal domain set to a range of values (observations). In the figure below, if x1, ..., xj are the j spatiotemporal coordinates (for example x,y,z,t or x,y,t), the coverage value attached to each of the spatiotemporal positions is an i-dimensional vector y1, y2, ..., yi where y1, y2, ..., yi are functions of the spatiotemporal coordinate.

Figure 26: Coverage Domain



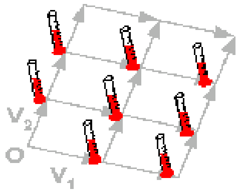
$$D = [y1, y2, \dots, yi] = [\; f1_{x,y,z,t}(x1 \dots, xj),\; f2_{x,y,z,t}(x1 \dots, xj),\; \dots\; fi_{x,y,z,t}(x1 \dots, xj)\,]$$

Spatio-temporal function

The coverage value may be a scalar (numeric or text) value, such as population density, a compound (vector or tensor) value, such as incomes by race or radiances by wavelength. The range axis descriptions are used for compound observations; they describe additional parameters, i.e., an independent variable besides space and time and the valid values of this parameter. These values are used to select subsets of coverage data similar to spatiotemporal subsetting.

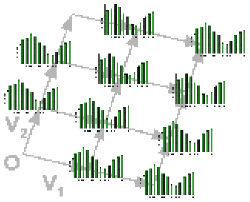
In the case of a coverage describing the ground temperature in Europe during the year 2003, the coverage value is a 1-dimensional vector (i = 1, the temperature) function of the 3-dimensional spatiotemporal coordinates (j = 3, latitude, longitude, time). No range axis is needed to describe the coverage value because it is a scalar value.

Figure 27: Scalar Observable - No RangeAxis Needed



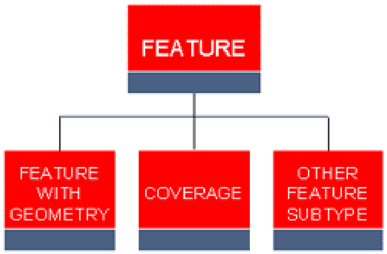
Examples of compound observations include a multispectral radiance, such as brightness by wavelength (typical of satellite imagery), age distribution (counts of people by age brackets in a census table), or climate pattern (mean rainfall by month of the year in a climate database). In these cases, the RangeAxis is needed to describe the ordinal values.

Figure 28: Spectral Response Observable - RangeAxis Used



A coverage is a special case or a subtype of a feature. The following figure demonstrates that features with geometry and coverages are two subtypes of the supertype feature. Other feature subtypes may not be directly associated with any geometry at all.

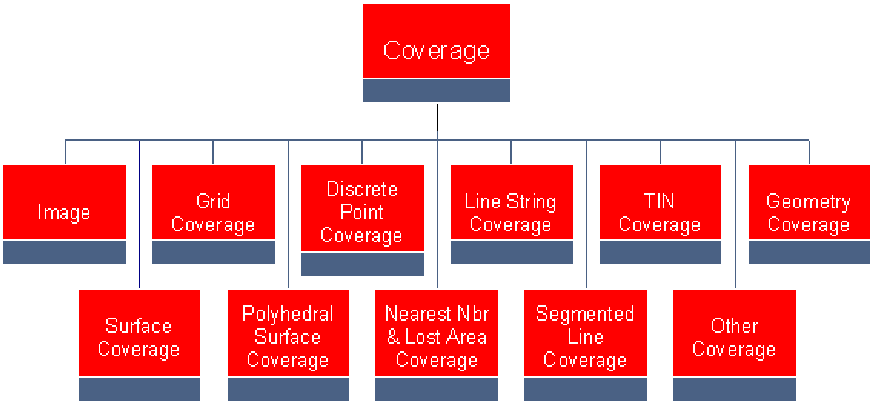
Figure 29: Coverage as a Feature Subtype



Coverage SubTypes

The coverage type itself has many important subtypes, such as Image, Grid Coverage, Surface Coverage, Discrete Point Coverage, Line String Coverage, TIN Coverage, Polyhedral Surface Coverage, Nearest Neighbor and Lost Area Coverage, Segmented Line Coverage, and Geometry Coverages. This structure is illustrated in the following figure.

Figure 30: Coverage Subtypes



The OGC-WCS specification Version 1.0.0. only supports the grid coverage type.

The grid coverage defines its domain as a regular grid of points or cells in 2, 3, or 4 dimensions. This description is suitable for digital airphotos, satellite imagery, gridded terrain models, or any other raster data.

Grid Coverage Characteristics

Grid coverages have the following characteristics:

- Variable number of bits (1,2,4,8,16,32, or 64 bits) per grid value: unsigned integer, signed integer, and real
- 1 to N bands
- 1 to N dimensions

- For grids with multiple bands, band values can be ordered by dimension. For example, a 2D grid coverage can be ordered by row-column-band (pixel interleaved), by row-band-column (line interleaved), or by band-row-column (band sequential).

- Support for a variable number of “no data values”

- Various color models are supported: gray scale, pseudocolor (any bit depth), RGB, CMYK and HSL

A grid coverage has a grid coordinate system that allows for addressing individual grid cells that are centered on the grid points. A grid has an ordering of cell values with the first cell in this ordering having grid coordinates of 0, 0. For example, a two-dimensional grid coverage with 512 rows and 512 columns would have grid coordinates with a range from rows 0 to 511 and columns 0 to 511.

The Web Coverage Service (WCS)

The Web Coverage Service (WCS) supports electronic interchange of geospatial data as "coverages". This is the definition of digital geospatial information that represents space-varying phenomena.

The Web Coverage Service provides access to intact (unrendered) geospatial information, such as temperature and cloud cover, as needed for client-side rendering and input into scientific models.

A WCS is providing one or more service offerings on coverage data:

- Spatial query (version 1.0 focuses on grid spatial request)

- Temporal query (separated from spatial)

- Logical space query (spatial query in the 'pixel' space)

- Re-projection offerings

- Range sub-settings

- Multiple interpolation offerings

- Multiple output format offerings

The Coverage Portrayal Service (CPS)

This type of service is intended to portray coverage data subsets extracted from a WCS, including application of classifications and processing. In the current OGC-CPS specification, the types of portrayal are:

- Classification

- Normalization
- Color mapping
- Hill shading
- Cloud cover filtering
- Time filtering

A CPS is providing one or more service offerings on coverage data:

- Spatial query (version 1.0 focuses on grid spatial request)
- Temporal query (separated from spatial)
- Reprojection offering(s) (optional)
- Range subsettings
- Multiple interpolation offerings
- Multiple output format offerings

The Web Coverage Service Interface

In this chapter:

Overview of the Web Coverage Service Interface

The Web Coverage Service (WCS) provides three operations: GetCapabilities, DescribeCoverage, and GetCoverage.

The GetCapabilities operation returns an XML document that describes the service and provides brief descriptions of the data collections from which clients may request coverages. Clients generally run the GetCapabilities operation and cache the result for use throughout a session, or reuse it for multiple sessions. If GetCapabilities cannot return descriptions of its available data, this information must be available from a separate source, such as an image catalog.

The DescribeCoverage operation allows clients to request a full description of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.

The GetCoverage operation of a WCS is normally run after GetCapabilities and DescribeCoverage replies have shown what requests are allowed and what data is available. The GetCoverage operation returns a coverage (values or properties of a set of geographic locations) bundled in a well-known coverage format. Its syntax and semantics bear some resemblance to the WMS GetMap and WFS GetFeature requests, but several extensions support the retrieval of coverages rather than static maps or discrete features.

Each of these request types can be formulated using HTTP-GET and HTTP-POST messaging. SOAP messaging is also supported.

The GetCapabilities Operation

This operation describes the published Web Coverage Service by providing service metadata that is based on *ISO 19119* and *WSDL.*

A critical requirement for interoperability is that each service describes itself in a standardized fashion. All OGC web services must offer service metadata packaged in an XML document. The GetCapabilities operation is used to request this metadata. According to the WCS 1.0 specification by the OGC, all WCS will need to respond to a GetCapabilities request. A server must be able to describe its capabilities. Specifically, it must indicate which coverage offerings can be queried, each coverage offering describing its bounding box, domain set extent and range sets.

NOTE: The WCS 1.0 specification does not support hierarchy.

Parameters

The following request parameters are required for the GetCapabilities request:

Table 10: The Parameters of a OGC-WCS 1.0.0 GetCapabilities Request

Request Parameter	Required/ Optional	Description
REQUEST=GetCapabilities	R	The request name
VERSION=1.0.0	O	The version of the message syntax (default: 1.0.0)
SERVICE=WCS	R	Specifies the WCS service type
SECTION=/ or /WCS_Capabilities/Service or /WCS_Capabilities/Capability or /WCS_Capabilities/ContentMetadata	O	Section of Capabilities document to be returned
UPDATESEQUENCE	O	Used for cache management

In order to formulate a GetCapabilities request using HTTP-GET, specify the URL of the WCS and the set of parameters required to return the proper result from the server.

HTTP-GET Example of a GetCapabilities

```
http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?request=GetCapabilities&version=1
.0.0&service=WCS
```

In the above example, "http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST" equals the URL of the queried WCS and "request=GetCapabilities&version=1.0.0&service=WCS" are the service parameters.

In order to formulate a GetCapabilities request using HTTP-POST, provide the XML version and encoding schema, the name of the request, the URL of the OGC specification location and the URL of the XML Schema that is referenced, and the desired output format.

HTTP-POST Example of GetCapabilities

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities service="WCS" version="1.0"
       xmlns="http://www.opengis.net/wcs"
       xmlns:ogc="http://www.opengis.net/ogc"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       outputFormat="XML"/>
```

SOAP encapsulation of HTTP-POST GetCapabilities

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   xmlns:gml="http://www.opengis.net/gml">
   <SOAP-ENV:Body>
       <GetCapabilities service="WCS" version="1.0" xmlns="http://www.opengis.net/wcs">
       </GetCapabilities>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response to a WCS GetCapabilities request is an XML document composed of 3 sections: the Service, the Capabilities, and the ContentMetadata section.

The Service section provides information about the service itself including information useful for a catalog search.

The Capability section lists the types of requests that the service supports, including the protocol and the output formats. Typically, a WCS service supports GetCapabilities, DescribeCoverage, and GetCoverage.

The Content Metadata section provides information about the data served including the names of the coverage offerings, the extents of the coverage offerings, and possibly a link to more detailed metadata in the ISO 19115 format.

When SOAP messaging is used, the resulting XML document is encapsulated in a SOAP envelope.

The following example is a subset of a successfully returned GetCapabilities response from an ERDAS APOLLO WCS.

Sample WCS Capabilities Document

```
<?xml version="1.0" encoding="utf-8" ?>
<WCS_Capabilities version="1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/wcs
http://schemas.opengis.net/wcs/1.0.0/wcsCapabilities.xsd"
 xmlns="http://www.opengis.net/wcs" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:gml="http://www.opengis.net/gml">
  <Service>
    <description>Boston Multi Simple Geotiff imagery</description>
    <name>ATLANTA_LIST</name>
    <label>Atlanta Multi Simple Coverage Server</label>
    <keywords>
      <keyword>Atlanta</keyword>
      <keyword>Coverage</keyword>
      <keyword>Geotiff</keyword>
      <keyword>Multi Simple</keyword>
    </keywords>
    <responsibleParty>
      <individualName>Luc Donea</individualName>
      <organisationName>ERDAS, Inc.</organisationName>
      <positionName>Product Manager</positionName>
      <contactInfo>
        <phone>
          <voice>+32 4 364 0 364</voice>
          <facsimile>NoFax</facsimile>
        </phone>
        <address>
          <deliveryPoint>Rue de Wallonie, 18</deliveryPoint>
          <city>Grace-Hollogne</city>
          <administrativeArea>Liege</administrativeArea>
          <postalCode>4460</postalCode>
          <country>Belgium</country>
          <electronicMailAddress>Luc.Donea@erdas.com</electronicMailAddress>
        </address>
        <onlineResource xlink:href="http://www.erdas.com" xlink:type="simple"/>
      </contactInfo>
    </responsibleParty>
    <fees>none</fees>
    <accessConstraints>none</accessConstraints>
  </Service>
  <Capability>
    <Request>
      <GetCapabilities>
        <DCPType>
          <HTTP>
            <Get>
              <OnlineResource
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?"
xlink:type="simple"/>
            </Get>
            <Post>
              <OnlineResource
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?"
xlink:type="simple"/>
            </Post>
          </HTTP>
        </DCPType>
      </GetCapabilities>
      <DescribeCoverage>
        <DCPType>
          <HTTP>
            <Get>
              <OnlineResource
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?"
xlink:type="simple"/>
            </Get>
            <Post>
              <OnlineResource
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?"
xlink:type="simple"/>
            </Post>
          </HTTP>
        </DCPType>
      </DescribeCoverage>
      <GetCoverage>
        <DCPType>
          <HTTP>
            <Get>
              <OnlineResource
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?"
xlink:type="simple"/>
            </Get>
            <Post>
              <OnlineResource
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?"
xlink:type="simple"/>
            </Post>
          </HTTP>
        </DCPType>
      </GetCoverage>
    </Request>
    <Exception>
      <Format>application/vnd.ogc.se_xml</Format>
    </Exception>
  </Capability>
  <ContentMetadata>
    <CoverageOfferingBrief>
      <metadataLink metadataType="TC211"
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST/REQUEST/getdir/DIR/me
tadata/DATA/LPR/ATLANTA__LIST/atl__tiles__1__1.xml" xlink:type="simple"/>
      <description>TIF Coverage Offering, MOD09GHK.EastCoast_x_Grid_L2g_2d.tif files,
Georectified data.</description>
      <name>MOD09GHK.EastCoast_1_Grid_L2g_2d</name>
      <label>TIF Coverage Offering</label>
      <lonLatEnvelope srsName="WGS84(DD)">
        <gml:pos>-93.343429595 39.9999840587107</gml:pos>
        <gml:pos>-65.2725515243675 49.999999996</gml:pos>
      </lonLatEnvelope>
      <keywords>
        <keyword>TIF</keyword>
        <keyword>MODIS</keyword>
        <keyword>MOD09GHK</keyword>
        <keyword>Coverage</keyword>
        <keyword>Geotiff</keyword>
      </keywords>
    </CoverageOfferingBrief>
    <CoverageOfferingBrief>
```

```
      <metadataLink metadataType="TC211"
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST/REQUEST/getdir/DIR/me
tadata/DATA/LPR/ATLANTA__LIST/atl__tiles__1__1.xml" xlink:type="simple"/>
      <description>TIF Coverage Offering, MOD09GHK.EastCoast_x_Grid_L2g_2d.tif files,
Georectified data.</description>
      <name>MOD09GHK.EastCoast_2_Grid_L2g_2d</name>
      <label>TIF Coverage Offering</label>
      <lonLatEnvelope srsName="WGS84(DD)">
        <gml:pos>-93.343429595 39.9999840587107</gml:pos>
        <gml:pos>-65.2725515243675 49.999999996</gml:pos>
      </lonLatEnvelope>
      <keywords>
        <keyword>TIF</keyword>
        <keyword>MODIS</keyword>
        <keyword>MOD09GHK</keyword>
        <keyword>Coverage</keyword>
        <keyword>Geotiff</keyword>
      </keywords>
    </CoverageOfferingBrief>
  </ContentMetadata>
</WCS_Capabilities>
```

WSDL Description

An example of a GetWSDL request to produce a WCS WSDL:

```
http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?version=1.0.0&service=WCS&request
=getWSDL&SoapOnly=true
```

Note that ERDAS APOLLO Coverage framework also supports the OGC WMS interface, and a WMS WSDL can also be obtained by running the appropriate getWSDL request:

```
http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?version=1.1.1&service=WMS&request
=getWSDL&SoapOnly=true
```

The DescribeCoverage Operation

When a client obtains summary descriptions of the coverage through a GetCapabilities request served by a particular WCS, it can build simple GetCoverage requests immediately. However, in most cases the client will need more information and will issue a DescribeCoverage request to obtain a full description of one or more coverages available. The server responds to such a request with an XML document describing one or more coverages served by the WCS, with each coverage describing its bounding box, domain set extent, and range sets.

Parameters

The following request parameters are required for a valid GetCoverage request:

Table 11: The Parameters of a OGC-WCS 1.0.0 DescribeCoverage Request

Request Parameter	Required/ Optional	Description

REQUEST=DescribeCoverage	R	The request name. Must be “DescribeCoverage”.
VERSION=1.0.0	R	The request protocol version
SERVICE=WCS	R	The service name. Must be “WCS”
COVERAGE=name1, name2, …	O	A comma-separated list of coverages to describe (identified by their name values in the Capabilities response). Default is all coverages, if supported by the server.

In order to formulate a DescribeCoverage Request using HTTP-GET, specify the URL of the WCS and the set of parameters needed to tell the server the expected result.

HTTP-GET Example of a DescribeCoverage

```
http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?request=DescribeCoverage&SERVICE=
WCS&VERSION=1.0.0
```

In the above example, "http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST" equals the URL of the queried WCS and "request=DescribeCoverage&SERVICE=WCS&VERSION=1.0.0" are the service parameters.

In order to formulate a DescribeCoverage request using HTTP-POST, provide the XML version and encoding schema, the name of the request, the URL of the OGC specification location, the URL of the XML Schema that is referenced, and the desired output format.

HTTP-POST Example of a DescribeCoverage

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeCoverage service="WCS" version="1.0.0"
        xmlns="http://www.opengis.net/wcs"
        xmlns:ogc="http://www.opengis.net/ogc"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        outputFormat="XML"/>
 <Coverage>MOD09GHK.EastCoast_1_Grid_L2g_2d</Coverage>
 <Coverage>MOD09GHK.EastCoast_2_Grid_L2g_2d</Coverage>
</DescribeCoverage>
```

The response to a DescribeCoverage request is an XML document describing the various coverage offerings served. For each one, the response document will display the computer name, a human label, an envelope, and a domain set and range set with one or more axis descriptions for each. Additional information such as supported coordinate systems, formats, and interpolation types is often provided as well. A link to metadata in the ISO 19115 format is also often provided.

When SOAP messaging is used, the resulting XML document is encapsulated in a SOAP envelope.

The following is a subset of an example of a successfully returned DescribeCoverage response from the ERDAS APOLLO WCS.

Sample DescribeCoverage Response Document

```
<?xml version="1.0" encoding="utf-8" ?>
<CoverageDescription version="1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/wcs
http://schemas.opengis.net/wcs/1.0.0/describeCoverage.xsd"
 xmlns="http://www.opengis.net/wcs" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:gml="http://www.opengis.net/gml">
  <CoverageOffering>
    <metadataLink metadataType="TC211"

xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST/REQUEST/getdir/DIR/me
tadata/DATA/LPR/ATLANTA__LIST/atl__tiles__1__1.xml" xlink:type="simple"/>
    <description>TIF Coverage Offering, MOD09GHK.EastCoast_x_Grid_L2g_2d.tif files,
Georectified data.</description>
    <name>MOD09GHK.EastCoast_1_Grid_L2g_2d</name>
    <label>TIF Coverage Offering</label>
    <lonLatEnvelope srsName="WGS84(DD)">
      <gml:pos>-93.343429595 39.9999840587107</gml:pos>
      <gml:pos>-65.2725515243675 49.999999996</gml:pos>
    </lonLatEnvelope>
    <domainSet>
      <spatialDomain>
        <gml:Envelope srsName="EPSG:4326">
          <gml:pos>-93.343429595 39.9999840587107</gml:pos>
          <gml:pos>-65.2725515243675 49.999999996</gml:pos>
        </gml:Envelope>
        <gml:RectifiedGrid dimension="2" srsName="EPSG:4326">
          <gml:limits>
            <gml:GridEnvelope>
              <gml:low>0 0</gml:low>
              <gml:high>6737 2400</gml:high>
            </gml:GridEnvelope>
          </gml:limits>
          <gml:axisName>X</gml:axisName>
          <gml:axisName>Y</gml:axisName>
          <gml:origin srsName="EPSG:4326">
            <gml:pos>-93.343429595 49.999999996</gml:pos>
          </gml:origin>
          <gml:offsetVector>0.00417 0</gml:offsetVector>
          <gml:offsetVector>0 -0.00417</gml:offsetVector>
        </gml:RectifiedGrid>
      </spatialDomain>
      <temporalDomain>
        <timePeriod>
          <beginPosition>2003-12-03T14:40:00Z</beginPosition>
          <endPosition>2003-12-03T18:00:00Z</endPosition>
        </timePeriod>
      </temporalDomain>
    </domainSet>
    <rangeSet>
      <RangeSet>
        <name>Range Set</name>
        <label>Range Set Description</label>
        <axisDescription>
          <AxisDescription>
            <name>Band</name>
            <label>Band</label>
            <values>
              <singleValue>band1</singleValue>
              <default>band1</default>
            </values>
          </AxisDescription>
        </axisDescription>
        <axisDescription>
          <AxisDescription>
            <name>Cloud</name>
            <label>Cloud</label>
            <values>
              <singleValue>0.0</singleValue>
              <default>0.0</default>
            </values>
          </AxisDescription>
        </axisDescription>
        <nullValues>
          <singleValue>36864</singleValue>
        </nullValues>
      </RangeSet>
    </rangeSet>
    <supportedCRSs>
      <requestResponseCRSs>EPSG:4326
      </requestResponseCRSs>
      <requestResponseCRSs>Image</requestResponseCRSs>
      <nativeCRSs>EPSG:4326</nativeCRSs>
    </supportedCRSs>
    <supportedFormats>
      <formats>GeoTIFF</formats>
      <formats>ECW</formats>
      <formats>DTED</formats>
      <formats>JPEG2000</formats>
      <formats>NITF</formats>
    </supportedFormats>
    <supportedInterpolations>
      <interpolationMethod>nearest neighbor</interpolationMethod>
</supportedInterpolations>
</CoverageOffering>
  <CoverageOffering>
    <metadataLink metadataType="TC211"
```

```
xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST/REQUEST/getdir/DIR/me
tadata/DATA/LPR/ATLANTA__LIST/atl__tiles__1__1.xml" xlink:type="simple"/>
    <description>TIF Coverage Offering, MOD09GHK.EastCoast_x_Grid_L2g_2d.tif files,
Georectified data.</description>
    <name>MOD09GHK.EastCoast_2_Grid_L2g_2d</name>
    <label>TIF Coverage Offering</label>
    <lonLatEnvelope srsName="WGS84(DD)">
      <gml:pos>-93.343429595 39.9999840587107</gml:pos>
      <gml:pos>-65.2725515243675 49.999999996</gml:pos>
    </lonLatEnvelope>
    <domainSet>
      <spatialDomain>
        <gml:Envelope srsName="EPSG:4326">
          <gml:pos>-93.343429595 39.9999840587107</gml:pos>
          <gml:pos>-65.2725515243675 49.999999996</gml:pos>
        </gml:Envelope>
        <gml:RectifiedGrid dimension="2" srsName="EPSG:4326">
          <gml:limits>
            <gml:GridEnvelope>
              <gml:low>0 0</gml:low>
              <gml:high>6737 2400</gml:high>
            </gml:GridEnvelope>
          </gml:limits>
          <gml:axisName>X</gml:axisName>
          <gml:axisName>Y</gml:axisName>
          <gml:origin srsName="EPSG:4326">
            <gml:pos>-93.343429595 49.999999996</gml:pos>
          </gml:origin>
          <gml:offsetVector>0.00417 0</gml:offsetVector>
          <gml:offsetVector>0 -0.00417</gml:offsetVector>
        </gml:RectifiedGrid>
      </spatialDomain>
      <temporalDomain>
        <timePeriod>
          <beginPosition>2003-12-01T14:55:00Z</beginPosition>
          <endPosition>2003-12-01T18:15:00Z</endPosition>
        </timePeriod>
      </temporalDomain>
    </domainSet>
    <rangeSet>
      <RangeSet>
        <name>Range Set</name>
        <label>Range Set Description</label>
        <axisDescription>
          <AxisDescription>
            <name>Band</name>
            <label>Band</label>
            <values>
              <singleValue>band1</singleValue>
              <default>band1</default>
            </values>
          </AxisDescription>
        </axisDescription>
        <axisDescription>
          <AxisDescription>
            <name>Cloud</name>
            <label>Cloud</label>
            <values>
```

```
              <singleValue>0.0</singleValue>
              <default>0.0</default>
            </values>
          </AxisDescription>
        </axisDescription>
        <nullValues>
          <singleValue>36864</singleValue>
        </nullValues>
      </RangeSet>
    </rangeSet>
    <supportedCRSs>
      <requestResponseCRSs>EPSG:4326</requestResponseCRSs>
      <requestResponseCRSs>Image</requestResponseCRSs>
      <nativeCRSs>EPSG:4326</nativeCRSs>
    </supportedCRSs>
    <supportedFormats>
      <formats>GeoTIFF</formats>
      <formats>ECW</formats>
      <formats>DTED</formats>
      <formats>JPEG2000</formats>
      <formats>NITF</formats>
    </supportedFormats>
    <supportedInterpolations>
      <interpolationMethod>nearest neighbor</interpolationMethod>
    </supportedInterpolations>
  </CoverageOffering>
</CoverageDescription>
```

The GetCoverage Operation

The GetCoverage operation allows retrieval of coverage data subsets from a WCS. Depending on the parameters given in the request, the output can be either a subset of an existing coverage or a new one built by assembling pieces of several coverages (by building a mosaic of images or stitching of datasets).

Parameters

The following parameters are required for the GetCoverage request:

Table 12: The Parameters of a OGC-WCS 1.0.0 GetCoverage Request

Request Parameter	**Required/ Optional**	**Description**
REQUEST=GetCoverage	R	The name of the request. Must be "GetCoverage"
VERSION=1.0.0	R	The request protocol version
SERVICE=WCS	R	The service name. Must be "WCS"
COVERAGE=name	R	The name of an available coverage
CRS=crs_identifier	O	The coordinate reference system in which the request is expressed, or "Image".

RESPONSE_CRS= crs_identifier	O	Coordinate Reference System in which to express coverage responses. The default is the request CRS.
BBOX=minx, miny, maxx, maxy, minz, maxz	One of BBOX or TIME is required	Request a subset defined by the specified bounding box, with min/max coordinate pairs ordered according to the Coordinate Reference System identified by the CRS parameter
TIME= time1,time2,… or TIME= min/max/res, …	One of BBOX or TIME is required	Request a subset corresponding to the specified time instants or intervals, expressed in an extended ISO 8601 syntax. Optional if a default time (or fixed time, or no time) is defined for the selected layer. One of BBOX or TIME is required.
PARAMETER= val1,val2, …or PARAMETER= min/max/res	R	(Included only for range sets with compound values). Request a range subset defined by constraining parameter PARAMETER. The PARAMETER key is a variable string; it must match the name of a parameter listed in the range set description of the selected coverage. For instance: band=1,5,3 (e.g., radiance values in bands 1, 5, 3) age=0/18 (e.g., counts of people with ages under 18 yrs.) Optional if the chosen range component has default values for the parameter.
WIDTH = w (integer) HEIGHT = h (integer) [DEPTH =d (integer)]	R	Request a grid of the specified width (w), height (h), and [for 3D grids] depth (d) (integer number of gridpoints). Either these or RESX, RESY, [for 3D grids] RESZ are required.
FORMAT= format	R	Requested output format of coverage. Must be one of those listed under the description of the selected coverage.
INTERPOLATION=interpolation	O	Interpolation method used to resample the data. Must be one of those listed under the description of the selected coverage.
EXCEPTIONS= application/vnd.ogc.se_xml	O	The format in which exceptions are to be reported by the server.

STORE= true	O	This option is used when the download of the output is to be deferred: the response will be an XML document containing an URL to the actual coverage. Calling that URL will do the actual download of the data. NB: Not included in the OGC-WCS specification.
(Vendor-specific parameters)	O	

In order to formulate a GetCoverage request using HTTP-GET, specify the URL of the WCS and the set of parameters needed to tell the server the expected result.

HTTP-GET Example of a GetCoverage

```
http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST?
REQUEST=GetCoverage
&SERVICE=WCS
&VERSION=1.0.0
&COVERAGE=atl__tiles__1__1
&CRS=EPSG:4326
&RESPONSE_CRS=EPSG:4326
&BAND=band1
&BBOX=-78.066296,-77.121606,36.637055,37.308619
&WIDTH=500
&HEIGHT=500
&FORMAT=GeoTIFF
&INTERPOLATION=nearest neighbor
&EXCEPTIONS=application/vnd.ogc.se_xml
```

In the above example, "http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST" equals the URL of the queried WCS and "SERVICE=WCS&VERSION=1.0.0&REQUEST=GetCoverage&COVERAGE=atl__tiles__1__1" are the service parameters.

In order to formulate a GetCoverage request using HTTP-POST, provide the XML version and encoding schema, the name of the request, the URL of the OGC specification location, and the URL of the XML Schema that is referenced.

HTTP-POST Example of a GetCoverage

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage service="WCS" version="1.0.0"
             xmlns="http://www.opengis.net/wcs"
             xmlns:gml="http://www.opengis.net/gml"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <sourceCoverage>MOD09GHK.EastCoast_1_Grid_L2g_2d</sourceCoverage>
   <domainSubset>
       <spatialSubset>
           <gml:Envelope srsName="EPSG:4326">
               <gml:pos>-78.066296 36.637055</gml:pos>
               <gml:pos>-77.121606 37.308619</gml:pos>
           </gml:Envelope>
           <gml:Grid dimension="2">
              <gml:limits>
                <gml:GridEnvelope>
                  <gml:low>0 0</gml:low>
                  <gml:high>500 500</gml:high>
                </gml:GridEnvelope>
              </gml:limits>
              <gml:axisName>X</gml:axisName>
              <gml:axisName>Y</gml:axisName>
           </gml:Grid>
       </spatialSubset>
       <temporalSubset>
         <gml:timePosition>2003-12-03T15:00:00Z</gml:timePosition>
       </temporalSubset>
   </domainSubset>
   <rangeSubset>
       <axisSubset name="Band">
           <singleValue>band1</singleValue>
       </axisSubset>
   </rangeSubset>
   <interpolationMethod>nearest neighbor</interpolationMethod>
   <output>
       <crs>EPSG:4326</crs>
       <format>GeoTIFF</format>
   </output>
</GetCoverage>
```

SOAP Encapsulation of HTTP-POST GetCoverage Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   xmlns:gml="http://www.opengis.net/gml">
   <SOAP-ENV:Body>
       <GetCoverage service="WCS" version="1.0.0" xmlns="http://www.opengis.net/wcs">
... (same content as the <GetCoverage> element in the previous example)
       </GetCoverage>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The GetCoverage operation returns a coverage (values or properties of a set of geographic locations) bundled in a well-known coverage format. In other words, the GetCoverage operation returns data, not images. Refer to Chapter 6 “The Coverage Portrayal Service (CPS)” on how to obtain an image from a coverage using a CPS.

When SOAP messaging is used, the binary result is attached to the returned SOAP message.

When the STORE=TRUE option is used, the output is an XML document referencing the actual coverage in a <CoverageData> tag. That document will look like the following:

GetCoverage Output When STORE=TRUE

```
<Coverage version="1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/wcs
http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST/REQUEST/get/DATA/LPR/CoverageType
.xsd"
 xmlns="http://www.opengis.net/wcs" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:gml="http://www.opengis.net/gml">
  <description>stored coverage result</description>
  <name>MOD09GHK.EastCoast_1_Grid_L2g_2d_TIF</name>
  <CoverageRegion>
    <description>coverage region of stored GetCoverage result</description>
    <name>MOD09GHK.EastCoast_1_Grid_L2g_2d_TIF</name>
    <domainSubset>
      <spatialSubset>
        <gml:Envelope srsName="EPSG:4326">
          <gml:pos>-78.066296 -77.121606</gml:pos>
          <gml:pos>36.637055 37.308619</gml:pos>
        </gml:Envelope>
        <gml:RectifiedGrid dimension="2" srsName="EPSG:4326">
          <gml:limits>
            <gml:GridEnvelope>
              <gml:low>0 0</gml:low>
              <gml:high>500 500</gml:high>
            </gml:GridEnvelope>
          </gml:limits>
          <gml:axisName>X</gml:axisName>
          <gml:axisName>Y</gml:axisName>
          <gml:origin srsName="EPSG:4326">
            <gml:pos>-78.066296 37.308619</gml:pos>
          </gml:origin>
          <gml:offsetVector>0.22941 0</gml:offsetVector>
          <gml:offsetVector>0 -0.22886</gml:offsetVector>
        </gml:RectifiedGrid>
      </spatialSubset>
    </domainSubset>
    <rangeSubset>
      <axisSubset name="Band">
        <singleValue>band1</singleValue>
      </axisSubset>
</rangeSubset>
    <CoverageData

xlink:href="http://localhost:8080/erdas-apollo/coverage/ATLANTA_LIST/REQUEST/gettemp/DATA/
LPR/TO4G1X303__4.tif"
     xlink:type="simple">
      <format>GeoTIFF</format>
    </CoverageData>
</CoverageRegion>
</Coverage>
```

Referencing

Georeferenced Coverages

Coverage files are usually georeferenced and provide the bounds and a spatial reference system identifier for the coverage files. For example, a GeoTIFF file has a binary header which describes the data bounding box and reference system. This information is available in the DescribeCoverage documents.

DescribeCoverage Document Subsets

```
<spatialDomain>
  <gml:Envelope srsName="EPSG:26910">
    <gml:pos>516430.13465702 4151612.8386696503</gml:pos>
    <gml:pos>589060.13465702 4224332.83866965</gml:pos>
  </gml:Envelope>
</spatialDomain>
...
<supportedCRSs>
    <requestResponseCRSs>EPSG:26910</requestResponseCRSs>
    <requestResponseCRSs>EPSG:4326</requestResponseCRSs>
    <nativeCRSs>EPSG:26910</nativeCRSs>
</supportedCRSs>
```

The spatialDomain element describes the georeferencing of the coverage data, and the nativeCRSs element describes its native reference system. The requestResponseCRSs defines the reference system identifiers that can be used in a GetCoverage request (CRS parameter) and the ones in which the result can be reprojected (RESPONSE_CRS).

Only a few request/response reference systems are described in the DescribeCoverage documents, but ERDAS APOLLO WCS can provide coordinate transforms from and to each reference system of its database. The following vendor-specific element in the GetCapabilies document expresses this Coordinate Transform/Reprojection ability:

GetCapabilities Document Subset

```
<VendorSpecificCapabilities>
    <supportDataSrs/>
</VendorSpecificCapabilities>
```

If only the bounds are provided in the source coverage file, it is possible to define the native reference system of the data using the 'srs' parameter in the providers.fac file entry.

Provider Entry

```
<CREATE ID="GLOBE" JCLASS="com.ionicsoft.wmtmap.provider.coverage.SimpleProvider" >
    <PARAM NAME="name" VALUE="world"/>
    <PARAM NAME="title" VALUE="world topography"/>
    <PARAM NAME="path" VALUE="\\Nas1\DataArchive\un\geotiff\globe30_5mln.tif" />
    <PARAM NAME="srs" VALUE="EPSG:4269"  />
```

```
</CREATE>
```

Logical Space

Grid Coverage data can be queried in its logical space. A description of that logical space is provided in the following DescribeCoverage document:

DescribeCoverage Document Subsets, Georeferenced Grid

```
<spatialDomain>
  <gml:Envelope srsName="EPSG:26910">
    <gml:pos>516430.13465702 4151612.8386696503</gml:pos>
    <gml:pos>589060.13465702 4224332.83866965</gml:pos>
  </gml:Envelope>
  <gml:RectifiedGrid dimension="2" srsName="EPSG:26910">
    <gml:limits>
      <gml:GridEnvelope>
        <gml:low>0 0</gml:low>
        <gml:high>7263 7272</gml:high>
      </gml:GridEnvelope>
    </gml:limits>
    <gml:axisName>X</gml:axisName>
    <gml:axisName>Y</gml:axisName>
    <gml:origin srsName="EPSG:26910">
      <gml:pos>516430.13465702 4224332.83866965</gml:pos>
    </gml:origin>
    <gml:offsetVector>10.0 0</gml:offsetVector>
    <gml:offsetVector>0 -10.0</gml:offsetVector>
  </gml:RectifiedGrid>
</spatialDomain>
...
<supportedCRSs>
    <requestResponseCRSs>EPSG:26910</requestResponseCRSs>
    <requestResponseCRSs>EPSG:4326</requestResponseCRSs>
    <requestResponseCRSs>Image</requestResponseCRSs>
    <nativeCRSs>EPSG:26910</nativeCRSs>
</supportedCRSs>
```

The RectifiedGrid element describes the logical space of the coverage data and the 'Image' value in a requestResponseCRSs element means that the data can be queried in its logical space. The following is a valid GetCoverage request expressed in the logical space described above:

HTTP-GET Example of a GetCoverage in Grid's Logical Space

```
http://apollo.erdas.com/erdas-apollo/coverage/SPOT5J?
request=GetCoverage
&SERVICE=WCS
&VERSION=1.0.0
&COVERAGE=TIF
&CRS=IMAGE
&Fields=ImageData4
&BBOX=0,0,7263,7272
&WIDTH=726
&HEIGHT=727
&FORMAT=GeoTIFF
&INTERPOLATION=nearest neighbor
```

If some coverage file does not provide a bounding box or a reference system, it still can be served. The data will then be exposed in its logical space and can only be requested in that logical space.

DescribeCoverage Document Subsets, Non-Georeferenced Grid

```
<spatialDomain>
  <gml:Envelope srsName="Image">
    <gml:pos>0 0</gml:pos>
    <gml:pos>400 400</gml:pos>
  </gml:Envelope>
  <gml:Grid dimension="2" srsName="Image">
    <gml:limits>
      <gml:GridEnvelope>
        <gml:low>0 0</gml:low>
        <gml:high>400 400</gml:high>
      </gml:GridEnvelope>
    </gml:limits>
    <gml:axisName>X</gml:axisName>
    <gml:axisName>Y</gml:axisName>
  </gml:Grid>
</spatialDomain>
...
<supportedCRSs>
    <requestResponseCRSs>Image</requestResponseCRSs>
    <nativeCRSs>Image</nativeCRSs>
</supportedCRSs>
```

The Coverage Portrayal Service

In this chapter:

Introduction to the Coverage Portrayal Service

The Coverage Portrayal Service (CPS) is an Open Geospatial Consortium (OGC) specification for a Web Processing Service (WPS) that adds value to the output of a Web Coverage Service (WCS). The CPS links together WMS clients and WCS services, using Styled Layer Descriptors (SLD) as a service language. The CPS interfaces are slight extensions or restrictions to the corresponding WMS interfaces.

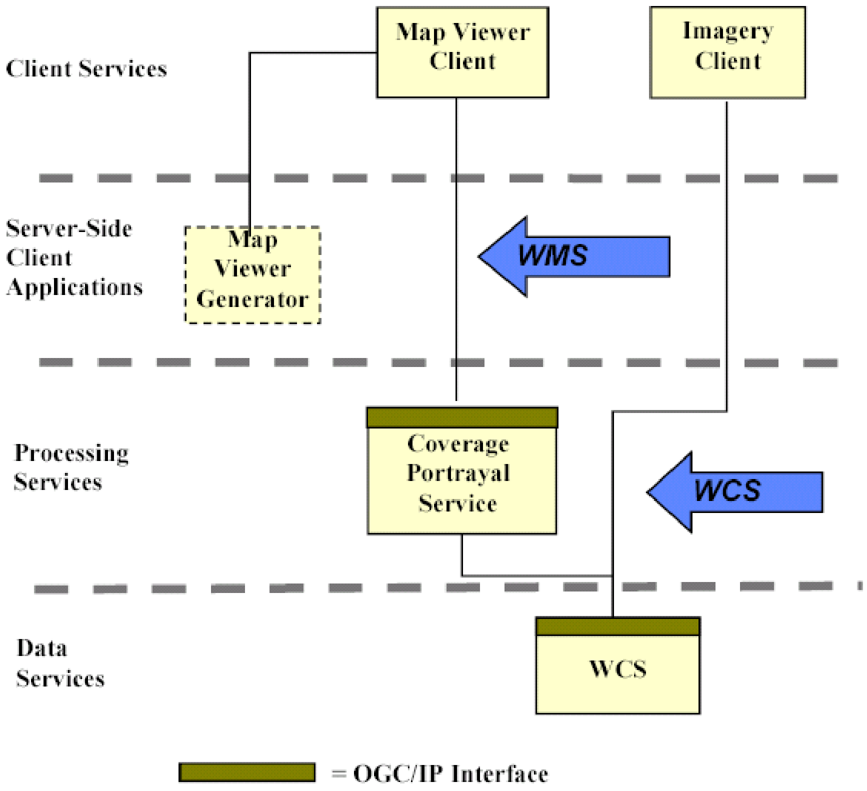
The ERDAS APOLLO Concepts Guide describes the concept of WMS and the SLD formalism. The ERDAS APOLLO Administrator's Guide describes the parameters of the various WMS requests in the "FAQ and Troubleshooting" chapter.

The purpose of the CPS is to provide a standard interface for producing visual pictures from coverage data.

Operational Concept of the Coverage Portrayal Service

The CPS implements two standard OGC interfaces, the WMS interface and the WCS interface. The figure below shows how a thin WMS map viewer client, which has been generated on the fly by a server-side client generator, can use the services of a WCS by connecting through a CPS.

Figure 31: CPS Integrates with the OGC Architecture



How the Styled Layer Descriptor (SLD) is Used

The CPS uses SLD as a language to apply a specific portrayal. In addition to request parameters, which serve to qualify the request action, additional information has to be provided to the CPS for it to work. This additional information can tell the CPS:

- Where to obtain the coverage of interest to the client.
- What part or parts of the coverage data with which to work.
- How the client wants the CPS to portray the data.

SLD is used to express this additional information and to provide a user-defined styling of the coverage data. There is a parallel between the way that a client can direct a CPS to use data from a WCS and the way that a client can direct a WMS to use data from a WFS. Using SLD for both CPS requests and WMS requests exploits this parallel.

The parts of the SLD that affect (3) above are contained in the SLD RasterSymbolizer element. The following subsections discuss some relevant aspects of this element.

The complete list of SLD tags supported by ERDAS APOLLO CPS is described in the ERDAS APOLLO Administrator's Guide, in the "Portrayal" chapter.

Data selection with SLD

The SLD RangeAxis tag enables the range axis subsetting. The following example shows the selection of the different channels of a MODIS coverage offering and their mapping on the RGB channels of the resulting image:

Figure 32: <sld:RangeAxis name="Fields"> sur_refl_b01_1,sur_refl_b02_1,sur_refl_b03_1 </sld:RangeAxis>

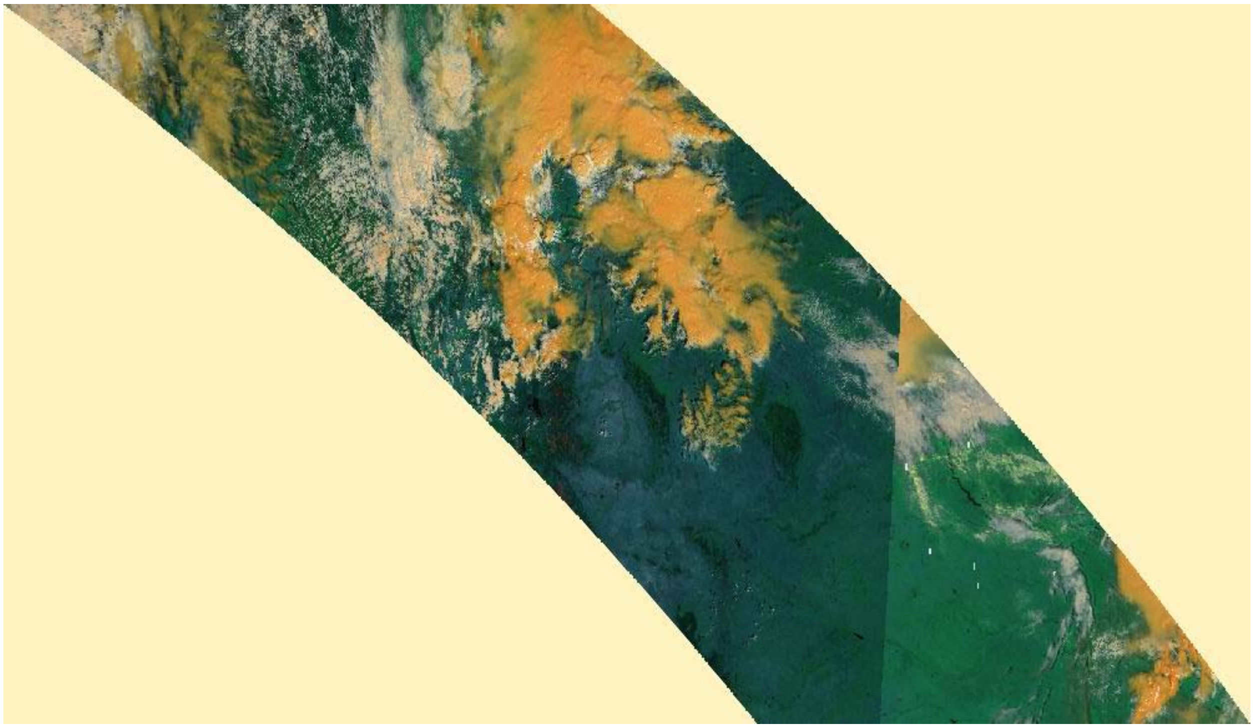
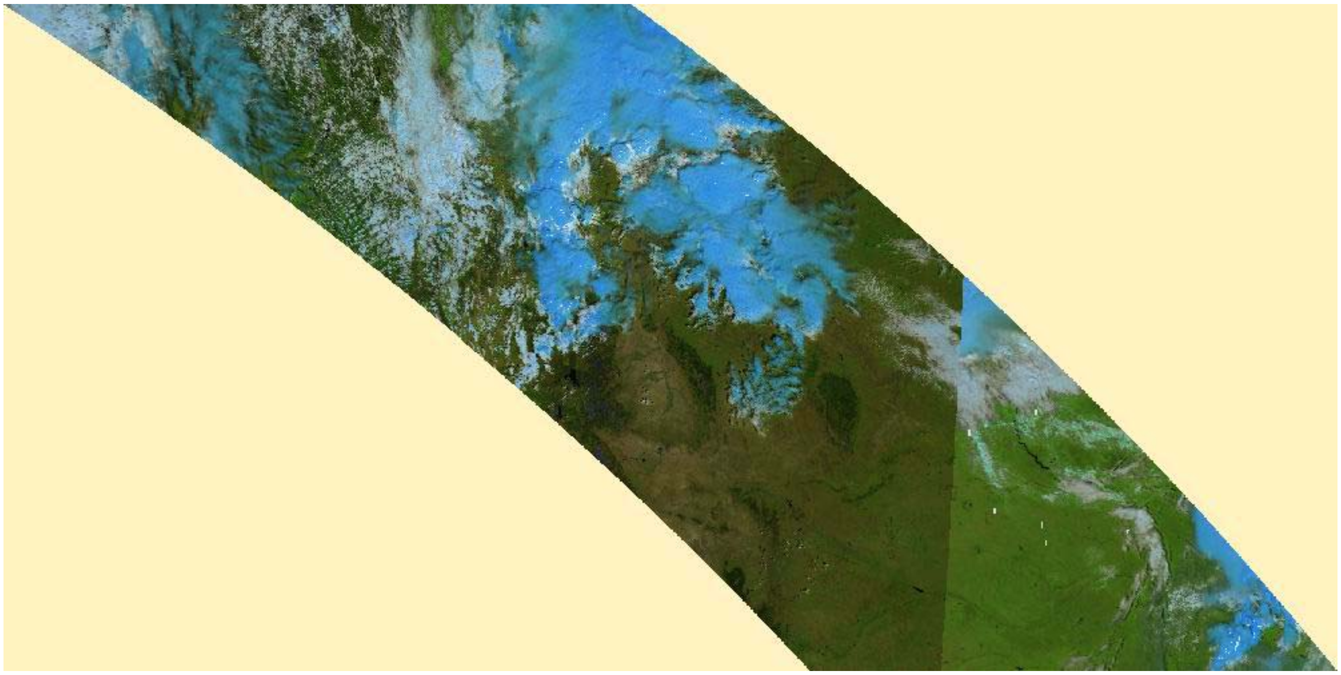


Figure 33: <sld:RangeAxis name="Fields"> sur_refl_b04_1,sur_refl_b05_1,sur_refl_b06_1 </sld:RangeAxis>



The SLD TimePeriod tag enables the temporal subsetting. The following example shows the selection of different datasets of a MODIS coverage layer, using different temporal intervals:

Figure 34: <sld:TimePeriod> 2003-12-01T14:55:00Z </sld:TimePeriod>

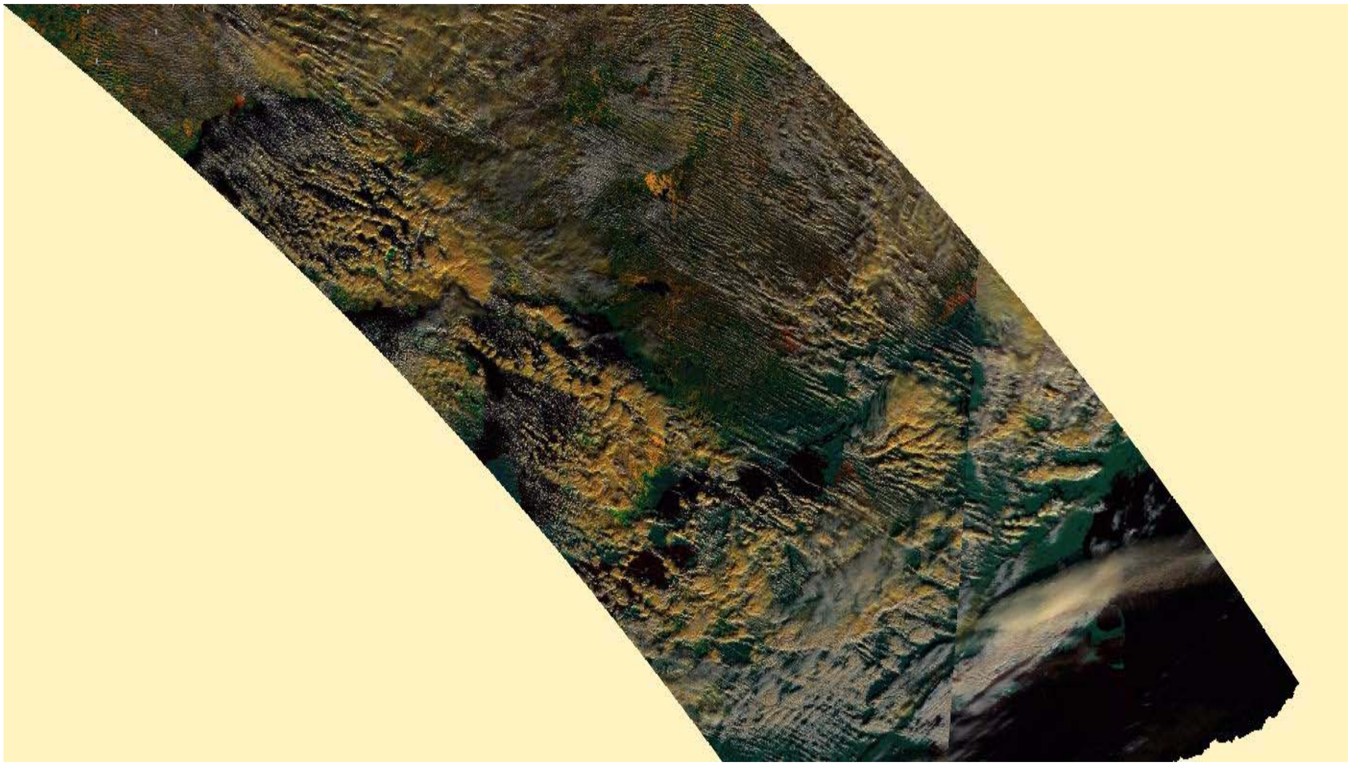
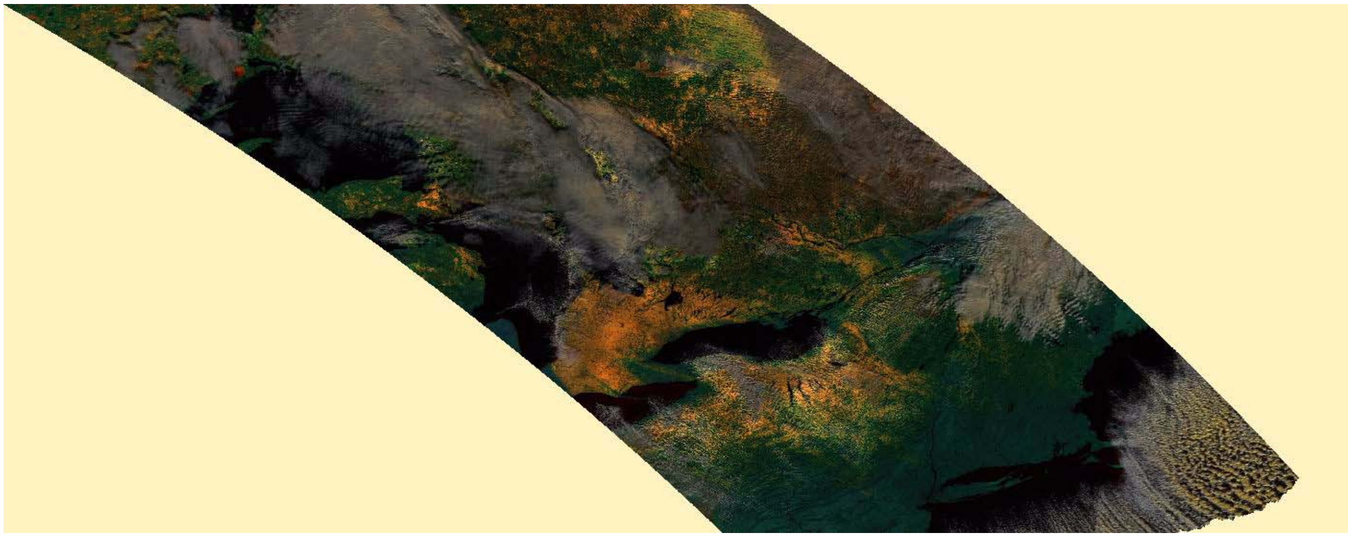


Figure 35: <sld:TimePeriod> 2003-12-03T20:03:00Z </sld:TimePeriod>



Data Styling with SLD

The SLD ContrastEnhancement tag can be used to reveal values that occupy a very small part of the data range.

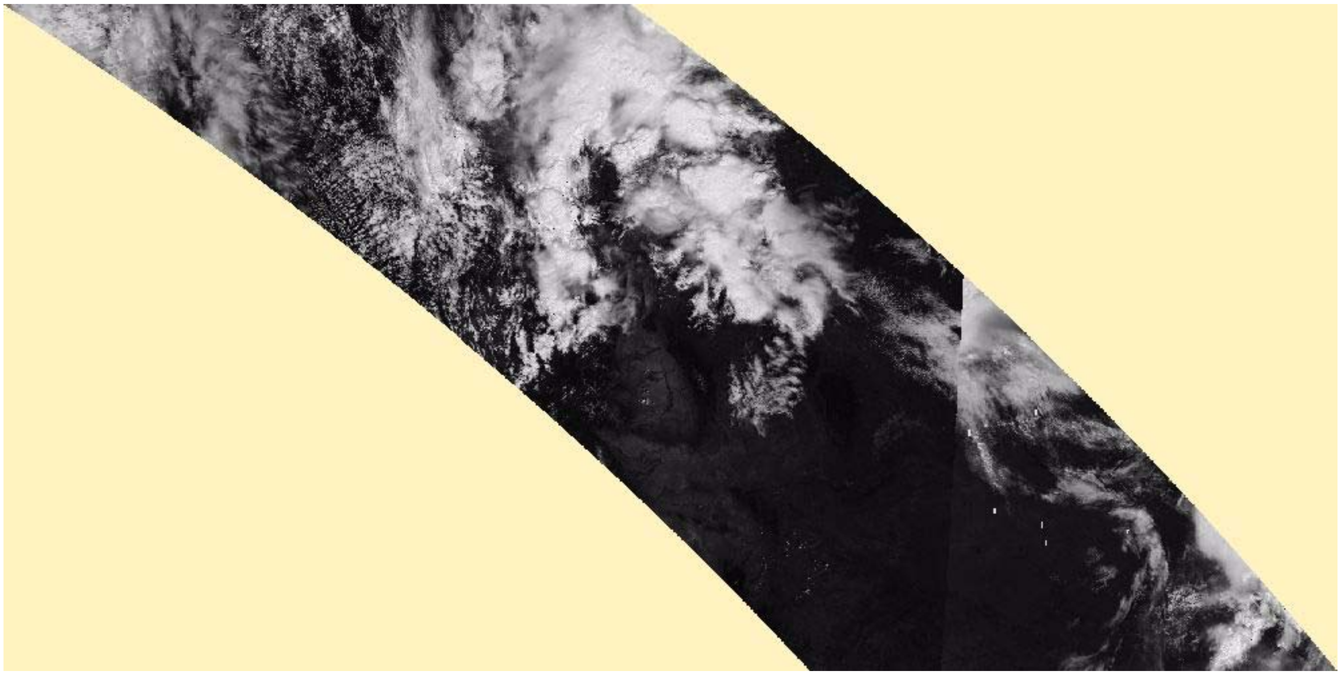
```
<sld:ChannelSelection>
     <sld:GrayChannel>
         <sld:SourceChannelName>
```

```
                Fields.sur_refl_b04_1
            </sld:SourceChannelName>
        </sld:GrayChannel>
</sld:ChannelSelection>
```

Figure 36: GrayChannel Without Contrast Enhancement

```
<sld:ChannelSelection>
        <sld:GrayChannel>
            <sld:SourceChannelName>
                Fields.sur_refl_b04_1
            </sld:SourceChannelName>
        </sld:GrayChannel>
</sld:ChannelSelection>
<sld:ContrastEnhancement>
       <sld:Normalize/>
</sld:ContrastEnhancement>
```

Figure 37: GrayChannel with Contrast Enhancement



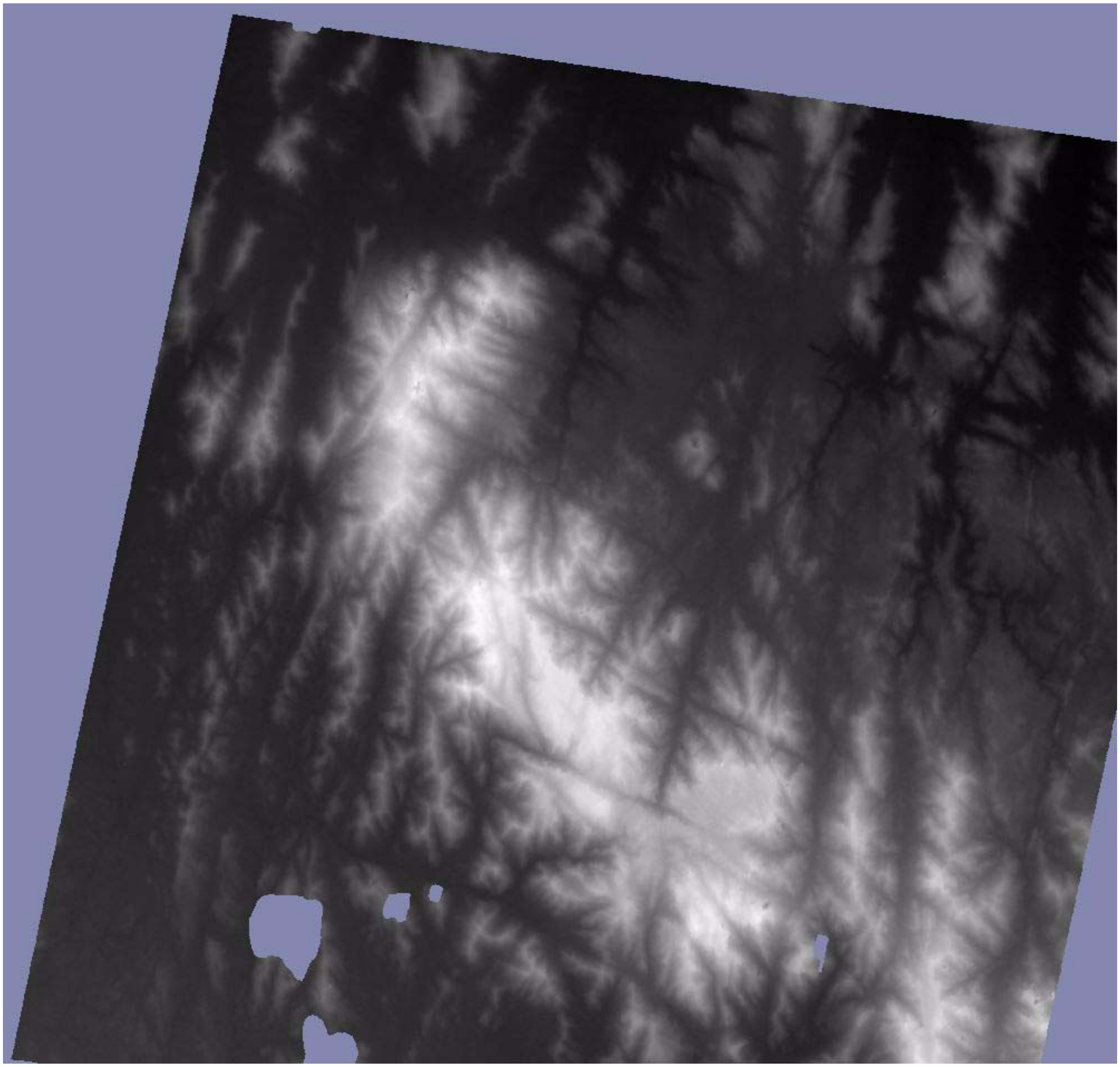
ContrastEnhancement: Defines contrast enhancement for a channel of a false-color image or for a color image. In the case of a color image, the relative grayscale brightness of a pixel color is used.

- "Normalize" means to stretch the contrast so that the dimmest color is stretched to black and the brightest color is stretched to white, with all colors in between stretched out uniformly.

- "Histogram" means to stretch the contrast based on a histogram of how many colors are at each brightness level on input, with the goal of producing equal number of pixels in the image at each brightness level on output. This has the effect of revealing many subtle ground features.

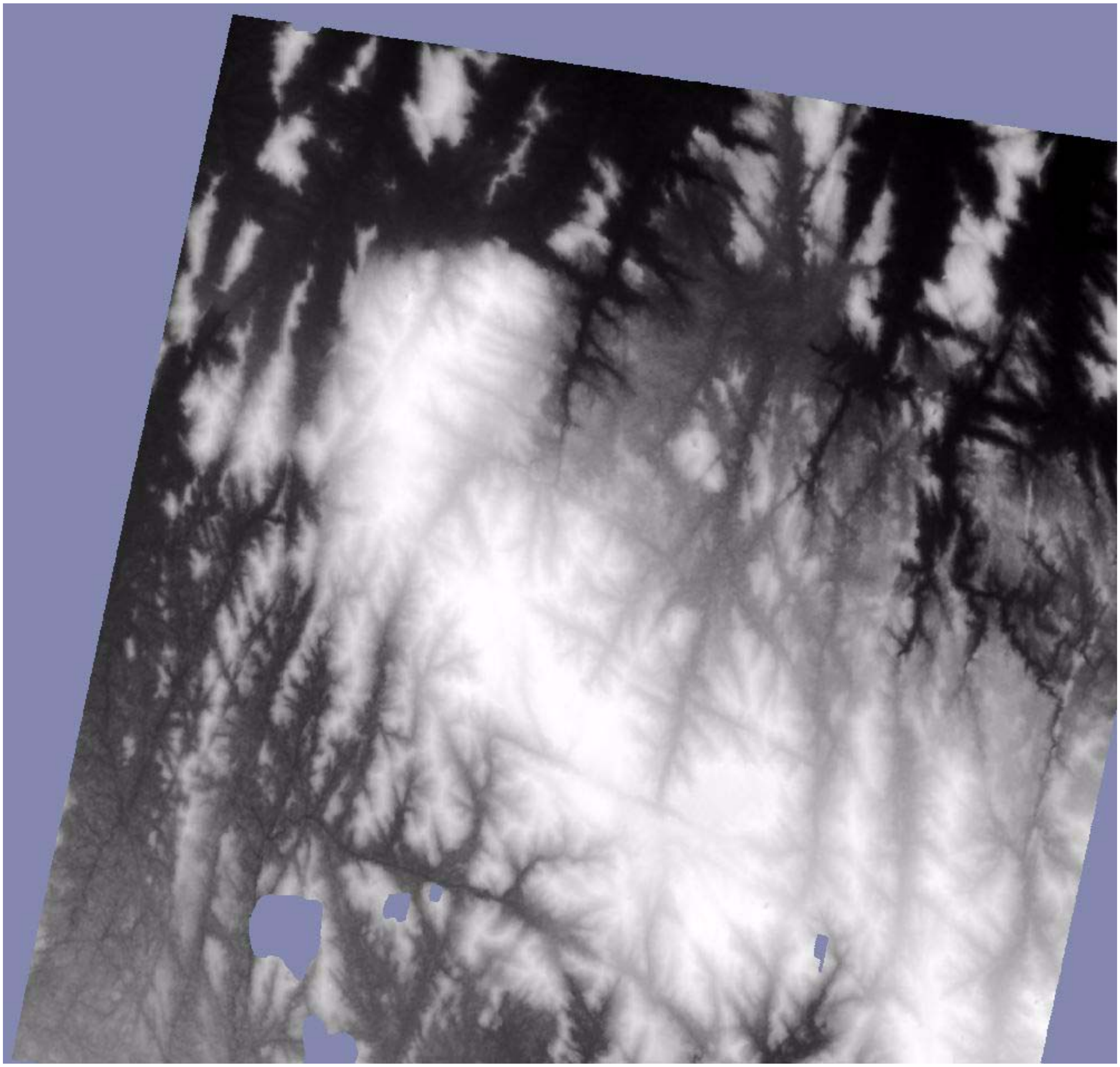
```
<sld:ContrastEnhancement>
    <sld:Normalize/>
</sld:ContrastEnhancement>
```

Figure 38: GrayChannel with "Normalize" Contrast Enhancement



```
<sld:ContrastEnhancement>
    <sld:Histogram/>
</sld:ContrastEnhancement>
```

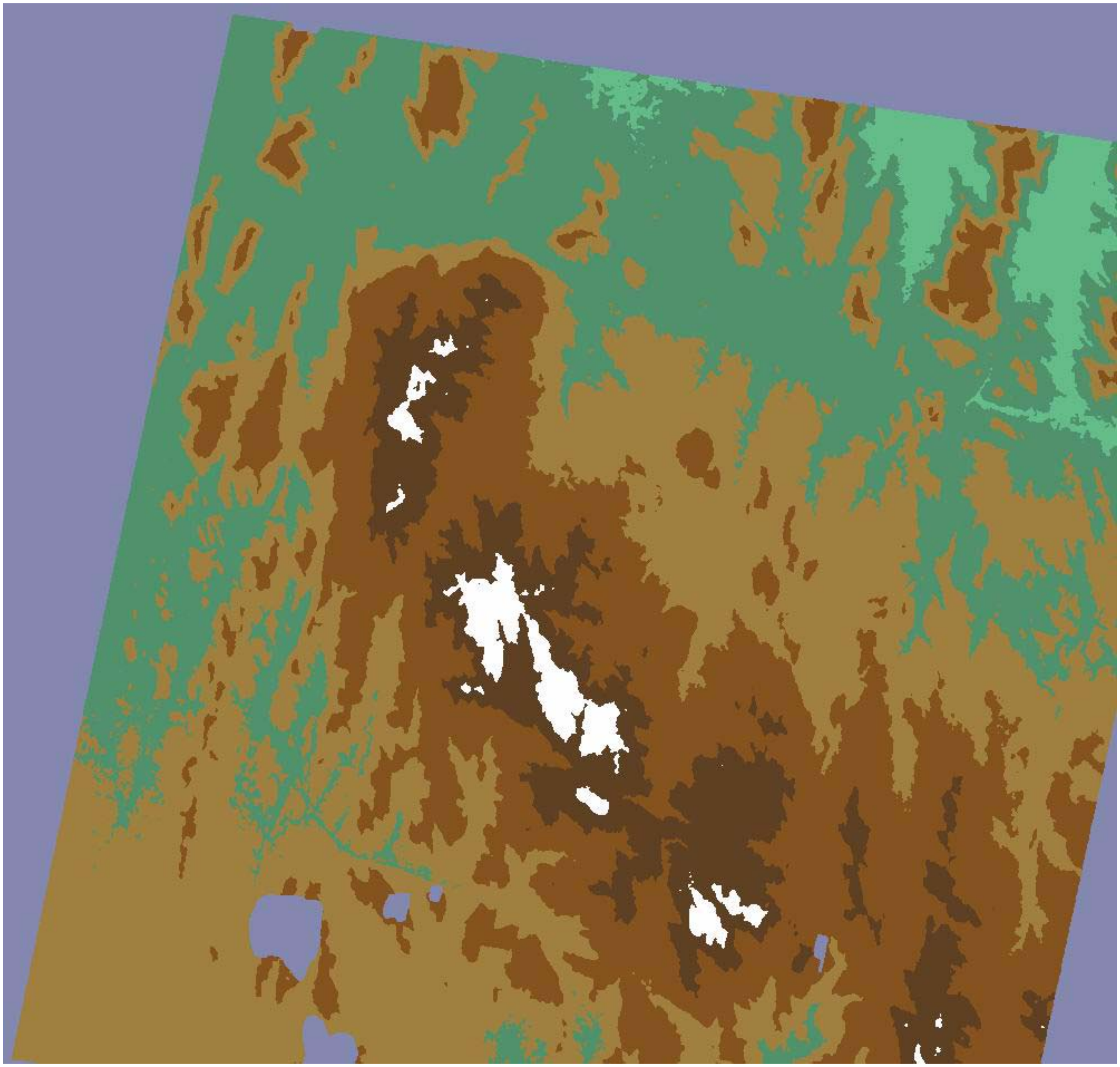
Figure 39: GrayChannel with "Histogram" Contrast Enhancement



The ColorMap SLD tag can be used to classify the values of the coverage data. One color is chosen for each data range. In the example below, the color #4327ED [hexadecimal encoding of the color RGB(4,50,126)] will be applied on all "pixels" that have a value smaller than -1.0. Note that in the example below the data is encoded in short values: this means that the data values are integers between -32767 and 32768.

```
<ColorMap>
        <ColorMapEntry color="#4327ED" quantity="-1.0"/>
        <ColorMapEntry color="#64BC89" quantity="0.1"/>
        <ColorMapEntry color="#50936C" quantity="250.0"/>
        <ColorMapEntry color="#9F803E" quantity="400.0"/>
        <ColorMapEntry color="#85531D" quantity="550.0"/>
        <ColorMapEntry color="#5E4023" quantity="1000.0"/>
        <ColorMapEntry color="#FFFFFF" quantity="1300.0"/>
</ColorMap>
```

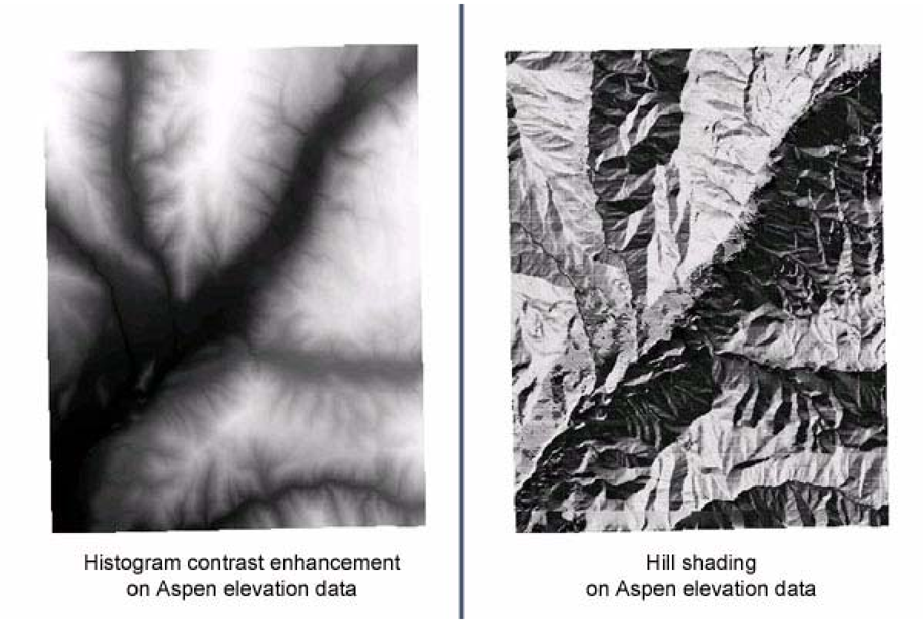
Figure 40: GrayChannel with Color Map



The ShadedRelief SLD tag can be used to apply the hill shading algorithm on elevation data.

```
<sld:ShadedRelief>
    <sld:ReliefFactor>
        1.0
    </sld:ReliefFactor>
</sld:ShadedRelief>
```

Figure 41: Hill Shading on Short Elevation Data



Histogram contrast enhancement
on Aspen elevation data

Hill shading
on Aspen elevation data

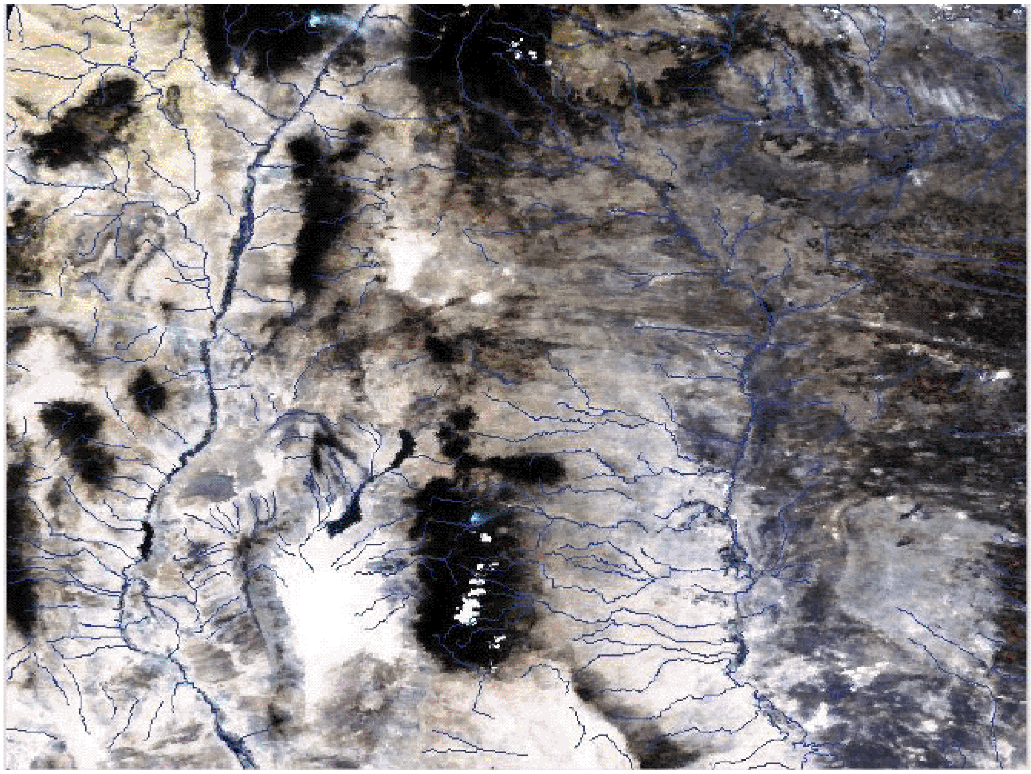
A Use Case for Portraying Coverages

This use case presents the steps that might take place during a typical CPS session.

In this use case there are three components: the WCS that provides a Spot View coverage, the CPS that styles the Spot View coverage, and a client application that knows how to define an SLD for a coverage, performs the WMS-SLD GetMap requests, and then displays the GetMap response.

The result is a visual representation of the Spot View coverage that can be overlaid with additional WMS map layers. The figure below is an example of a georectified Spot View coverage that has been overlaid with an additional map layer showing watercourses. This map layer has been requested from another WMS.

Figure 42: Layering of Watercourses Over a Portrayed Coverage



Below are the different steps:

1. A Web Coverage Service (WCS) publishes a layer containing a multiband grid of data located in a region.

```
http://apollo.erdas.com/erdas-apollo/coverage/SPOT5J - the WCS URL
```

2. The user finds that the published layer overlaps with his current region of interest and wishes to view this data in order to determine if it requires further analysis or further processing.

   This can be done by executing a GetCapabilities request followed by a DescribeCoverage for a given coverage or directly by executing a global DescribeCoverage. This action will provide information about all the coverages available on the service.

```
http://apollo.erdas.com/erdas-apollo/coverage/SPOT5J?
REQUEST=GetCapabilities&VERSION=1.0.0&SERVICE=WCS
http://apollo.erdas.com/erdas-apollo/coverage/SPOT5J?
REQUEST=DescribeCoverage&VERSION=1.0.0&SERVICE=WCS
```

3. The client application knows of a CPS and requests it to provide a view of the data that can be overlaid with familiar map features.

```
http://apollo.erdas.com/erdas-apollo/map/PORTRAY
```

4. The client builds a Styled Layer Descriptor (SLD) document, which acts as a raster descriptor, to assign red to band 1 of the grid data, green to band 2, and blue to band 3 of the WCS.

```
<sld:StyledLayerDescriptor version="0.7.2">
  <sld:UserLayer>
    <sld:RemoteOWS>
      <sld:Service>WCS</sld:Service>
      <sld:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
           xlink:type="simple"
xlink:href="http://apollo.erdas.com/erdas-apollo/map/PORTRAY"/>
    </sld:RemoteOWS>
    <sld:LayerCoverageConstraints>
      <sld:CoverageConstraint>
        <sld:CoverageName>SPVIEW_530_274_0_020809_5_1_J_3</sld:CoverageName>
        <sld:CoverageExtent>
          <sld:TimePeriod> 2002-08-09T16:45:00Z/2002-08-09T20:05:00Z </sld:TimePeriod>
          <sld:RangeAxis name="Band"> band1,band2,band3 </sld:RangeAxis>
        </sld:CoverageExtent>
      </sld:CoverageConstraint>
    </sld:LayerCoverageConstraints>
    <sld:UserStyle>
      <sld:CoverageStyle>
        <sld:Rule>
          <Name>ChannelSelection</Name>
          <Title>3 Channels Selection</Title>
          <sld:RasterSymbolizer>
            <sld:ChannelSelection>
              <sld:RedChannel>
                <sld:SourceChannelName>Band.band1</sld:SourceChannelName>
              </sld:RedChannel>
              <sld:GreenChannel>
                <sld:SourceChannelName>Band.band2</sld:SourceChannelName>
              </sld:GreenChannel>
              <sld:BlueChannel>
                <sld:SourceChannelName>Band.band3</sld:SourceChannelName>
              </sld:BlueChannel>
            </sld:ChannelSelection>
            <sld:ContrastEnhancement>
              <sld:Normalize/>
            </sld:ContrastEnhancement>
          </sld:RasterSymbolizer>
        </sld:Rule>
      </sld:CoverageStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>
```

5. The client formulates a Web Map Service (WMS) request that includes the raster descriptor along with the other requested parameters: WIDTH, HEIGHT, BBOX, and SRS.

6. The client issues this WMS request to the CPS.

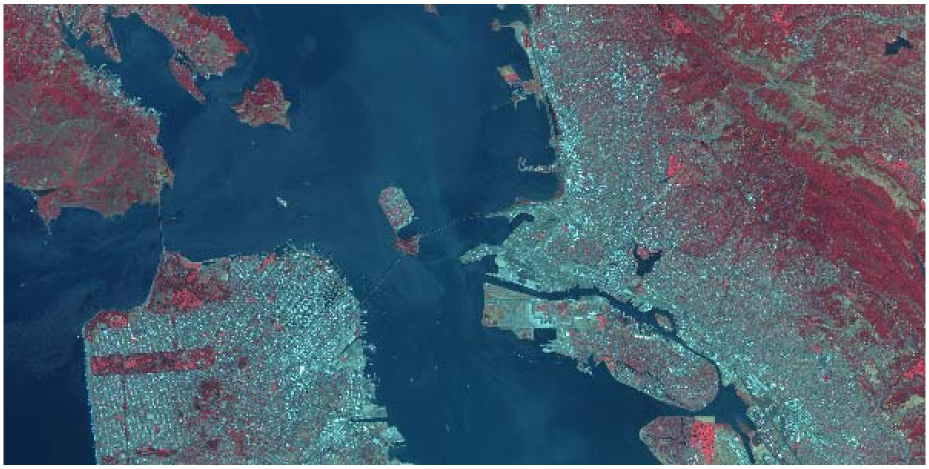
7. The CPS receives this WMS request from the client, fetches the SLD, and then formulates a WCS request for the data from the bands which were mentioned in the SLD that covers the BBOX mentioned in the WMS request and submits the WCS request.

```
http://apollo.erdas.com/erdas-apollo/map/PORTRAY?SERVICE=WCS&VERSION=1.0.0&request=GetCove
rage
&COVERAGE=SPVIEW_530_274_0_020809_5_1_J_3&CRS=EPSG:26910&Band=band1,band2,band3
```

```
&BBOX=539629.504688809,4176155.456291143,576617.010472623,4194649.20918305
&TIME=2002-08-09T16:45:00Z/2002-08-09T20:05:00Z&WIDTH=500&HEIGHT=250&FORMAT=GeoTIFF
```

8. The WCS returns the requested coverage data to the CPS. If necessary, the CPS redraws the data in the requested BBOX and SRS, applies the SLD to produce an image of the requested format with the requested WIDTH and HEIGHT, and returns the image to the client.

Figure 43: The Image Returned to the Client by the CPS



The Catalog Service

Overview

The ERDAS APOLLO Catalog service provides a mean to store, manage and discover metadata records describing various GIS-related objects, such as OGC services or imagery resources.

The resources stored in the ERDAS APOLLO Catalog service can be accessed and managed through a set of of HTTP endpoints, mainly a CSW ebRIM compliant endpoint and a RESTful endpoint. The following sections will describe both interfaces, respectively focusing on the ebRIM mapping of the catalog object model, and on the REST principles and their application to a catalog of OGC services.

CSW ebRIM Endpoint

The ERDAS APOLLO Catalog offers a HTTP endpoint compliant with the OGC CSW 2.0.2 specification and the CSW ebRIM 1.0 Application Profile. This HTTP endpoint sits at `http://localserver:8080/erdas-apollo/catalog/csw`. This section will focus on describing how the internal catalog object model is mapped into ebRIM structures and exposed in ebXML.

It must be noted that the CSW endpoint offers a view on the Catalog content; it does not rely on a separate storage specific to ebRIM. That way, any content inserted into the Catalog, be it using the Catalog Web Interface, the remote API, or the CSW interface itself, will appear through the CSW endpoint, exposed using the ebRIM models described below. Conversely, content inserted using the CSW endpoint will be mapped reciprocally using the same mappings, to fill the Catalog object model.

The Catalog Web Interface contains a tool to easily test the CSW endpoint.

ebRIM Imagery model

In the Catalog, the imagery object model consists of aggregates, coverage references (a.k.a. granules), all belonging to a datapool. The following tables will describe how those object types and properties are translated into ebRIM when accessed or queried through the CSW endpoint.

Figure 44: Imagery ebRIM Model

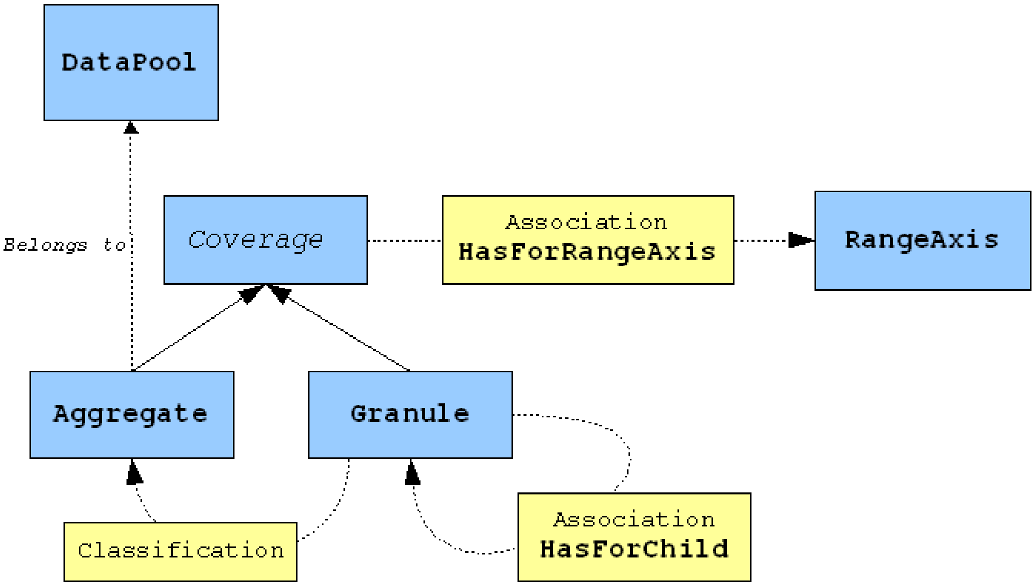


Table 13: DataPool ebRIM mapping

ebRIM mapping	Object Type or property
ClassificationScheme	**Datapool (top-level object)**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ClassificationScheme"
Id	"urn:x-ionic:internal-identifier:classificationScheme:Datapool"
Name	"Datapool"
Slot["PublicationDate"]	Catalog db object insertion date

Table 14: ebRIM mapping - Aggregate and Granule common properties

ebRIM mapping	Object Type or property
Id	System generated
Name	Coverage's name
Description	Coverage's abstract
Slot["PublicationDate"]	Catalog db object insertion date
Slot["IsBasicCoverageOffering"]	"yes"
Slot["Title"]	Coverage's title
Slot["DecoderName"]	Coverage's decoder name
Slot["Namespace"]	Coverage's namespace
Slot["NativeSRS"]	Coverage's native SRS
Slot["Wms-Wcs"]	Coverage's availability (possible values are "00","01","10","11")
Slot["PyramidInfo"]	Coverage's pyramid information
Slot["Keywords"]	Coverage's keywords (one or several values)
Slot["FootPrint"]	Coverage's spatial extent native geometry (if 4326), or corresponding 4326 geometry
Slot["DSE_SE_BoundingGeometry"]	Coverage's spatial extent native geometry
Slot["DSE_SE_XExtent"]	Coverage's spatial extent X Extent
Slot["DSE_SE_YExtent"]	Coverage's spatial extent Y Extent
Slot["DSE_SE_ZExtent"]	Coverage's spatial extent Z Extent
Slot["DSE_TE_FullExtent_BEGIN"]	Coverage's temporal extent start
Slot["DSE_TE_FullExtent_END"]	Coverage's temporal extent end

Table 15: ebRIM mapping - Aggregate specific properties

ebRIM mapping	Object Type or property
Concept	**Coverage Aggregate**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ClassificationNode"
Value	Aggregate's name
Slot["Path"]	Aggregate's path

Table 16: ebRIM mapping - Granule specific properties

ebRIM mapping	Object Type or property
Concept	**Coverage Aggregate**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:CoverageReference"
RepositoryItem	XML Serialized Granule's properties
MimeType	"text/xml"
Slot["FileURI"]	Granule's file URI
Slot["MetadataURI"]	Granule's metadata URI
Slot["CAI_AcquisitionDate"]	Granule's acquisition date
Slot["CAI_AvailabilityDate"]	Granule's availability date
Slot["CAI_LastModificationDate"]	Granule's last modification date
Slot["CAI_AcquisitionLevelMask"]	Granule's acquisition level mask
Slot["CAI_ErrorValue"]	Granule's error value
Slot["CAI_QualityValue"]	Granule's quality value
Slot["CAI_Resolution"]	Granule's resolution
Slot["CAI_ProcessingLevel"]	Granule's processing level

Table 17: ebRIM mapping - RangeAxis object type

ebRIM mapping	Object Type or property
ExtrinsicObject	**Range Axis description**
ObjectType	"urn:x-ionic:internal-identifier:ObjectType:RangeAxis"
Id	System generated
Name	Range Axis's name
Description	Range Axis's title
Slot["PublicationDate"]	Catalog db object insertion date
Slot["RA_RangeExtent_SV_as_String"]	RangeAxis single values set (one or several string values)
Slot["RA_RangeExtent_SV_as_Double"]	RangeAxis single values set (one or several double values)
Slot["RA_RangeFullExtent_MIN_as_String"]	RangeAxis variant min value (one string value)
Slot["RA_RangeFullExtent_MAX_as_String"]	RangeAxis variant max value (one string value)
Slot["RA_RangeFullExtent_MIN_as_Double"]	RangeAxis variant min value (one double value)
Slot["RA_RangeFullExtent_MAX_as_Double"]	RangeAxis variant max value (one double value)

Table 18: ebRIM mapping - hasForRangeAxis Association type

ebRIM mapping	Object Type or property
Association	**Link between a Coverage and its RangeAxis descriptions**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Association"
AssociationType	"urn:x-ionic:internal-identifier:AssociationType:hasForRangeAxis"
Id	System generated
Name	"hasForRangeAxis"
Source	A parent Granule (ExtrinsicObject) or Aggregate (Concept)
Target	A RangeAxis (ExtrinsicObject)
Slot["PublicationDate"]	Catalog db object insertion date

Table 19: ebRIM mapping - hasForChild Association type

ebRIM mapping	Object Type or property
Association	**Link between a granule and its children**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Association"
AssociationType	"urn:x-ionic:internal-identifier:AssociationType:hasForChild"
Id	System generated
Name	"hasForChild"
Source	A parent Granule (ExtrinsicObject)
Target	A child Granule (ExtrinsicObject)
Slot["PublicationDate"]	Catalog db object insertion date

ebRIM OGC Services model

The OGC Services object model consists of Service and their ServiceOperations, MapLayer, FeatureType, Coverage, Contact (User and Organization). This ebRIM mapping is based on the mapping proposed in the OGC OWS-3 Interoperability Experiment Report. The following tables will describe the structure and properties of those ebRIM object types.

Figure 45: OGC Services ebRIM Model

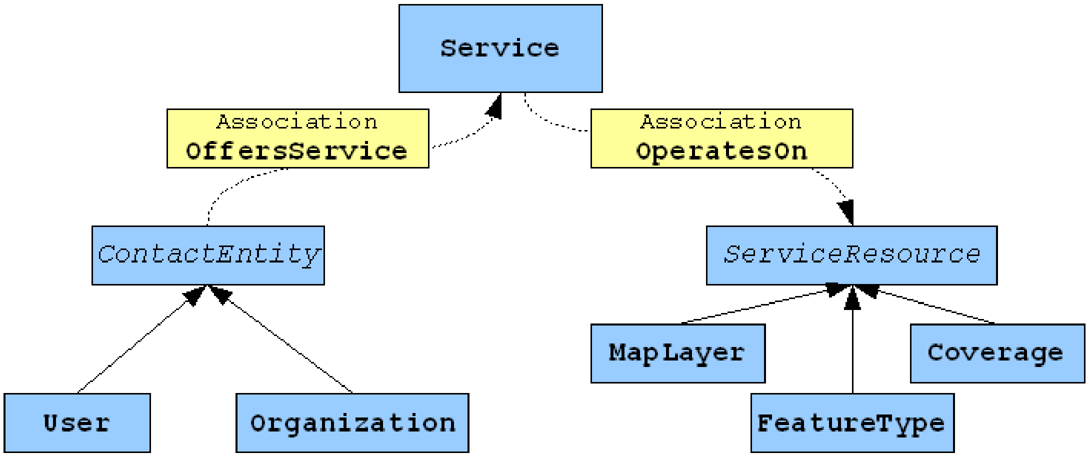


Table 20: ebRIM mapping - Service

ebRIM mapping	Object Type or property
Service	**OGC Service Object**
`ObjectType`	`"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Service"`
`Id`	System generated
`Name`	Service's title
`Description`	Service's abstract
`Slot["Identifier"]`	Service's name
`Slot["PublicationDate"]`	Catalog db object insertion date
`Slot["ServiceVersion"]`	Service's specification version number
`Slot["ServiceFees"]`	Service's usage fees, if any
`Slot["OnlineResource"]`	Service's online resource URL
`Slot["Keywords"]`	Service's descriptive keywords

Table 21: ebRIM mapping - Common properties for service resources (Map Layer, Feature Type, Coverage)

ebRIM mapping	Object Type or property
ExtrinsicObject	**Service resource**
`Id`	System generated
`Name`	Resource's title
`Description`	Resource's abstract
`Slot["Service_Layer_Name"]`	Resource's name
`Slot["PublicationDate"]`	Catalog db object insertion date
`Slot["FootPrint"]`	Resource's spatial extent
`Slot["MetadataURI"]`	Resource's metadata URI, if any

Table 22: ebRIM mapping - Object types mappings for Service resources

ebRIM mapping	Object Type or property
`WMS Layer ObjectType`	`"urn:x-ionic:internal-identifier:ObjectType:WMS_Layer"`
`Feature type ObjectType`	`"urn:x-ionic:internal-identifier:ObjectType:FeatureType"`
`Coverage ObjectType`	`"urn:x-ionic:internal-identifier:ObjectType:CoverageReference"`

Table 23: ebRIM mapping - User

ebRIM mapping	Object Type or property
User	**User (subclass of ContactEntity)**
`ObjectType`	`"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:User"`
`Id`	System generated
`Name`	User's name
`Organization`	User's organization (if any)
`Slot["PublicationDate"]`	Catalog db object insertion date

Table 24: ebRIM mapping - Organization

ebRIM mapping	Object Type or property
Organization	**Organization (subclass of ContactEntity)**
`ObjectType`	`"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Organization"`
`Id`	System generated
`Name`	Organization's name
`Users`	Organization's users
`Slot["PublicationDate"]`	Catalog db object insertion date

Table 25: ebRIM mapping - OffersService Association type

ebRIM mapping	Object Type or property
Association	**Link between a Service and its Contact point (User or Organization)**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Association"
AssociationType	"urn:x-ionic:internal-identifier:AssociationType:OffersService"
Id	System generated
Name	"OffersService"
Source	A Contact object (User or Organization)
Target	A Service object
Slot["PublicationDate"]	Catalog db object insertion date

Table 26: ebRIM mapping - OperatesOn Association type

ebRIM mapping	Object Type or property
Association	**Link between a Service and its Resources**
ObjectType	"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Association"
AssociationType	"urn:x-ionic:internal-identifier:AssociationType:OperatesOn"
Id	System generated
Name	"OperatesOn"
Source	A Service object
Target	A service resource (ExtrinsicObject)
Slot["PublicationDate"]	Catalog db object insertion date

REST Interface

REST Principles

REST, meaning "Representational state transfer", is a style of software architecture for distributed hypermedia systems such as the World Wide Web. REST strictly refers to a collection of network architecture principles which outline how resources are defined and addressed. Systems which follow REST principles are often referred to as "RESTful".

Basically, REST consists in:

- publishing every resource as a static URI, with an ID,
- using the standard HTTP commands (GET,POST,PUT,DELETE) for CRUD operations (Create,Read,Update,Delete) on resources,
- supporting multiple representations of resources, usually by using an URI extension pattern,
- linking resources by using their URIs,
- using the HTTP headers to perform content type negotiation.

In the scope of ERDAS APOLLO Catalog, a REST interface has been developed, using the **restlet framework**. This REST approach enables the catalog to be easily accessible from web-based, light clients such as browsers.

The various object types of the catalog object model are exposed through dedicated HTTP endpoints that are described in the following sections.

REST by example

Practically, this means that:

- Issuing a GET request on `http://someserver/erdas-apollo/catalog/content/items` will return you a list of all the items in the catalog (or part of it if the paging is enabled by default)
- `http://someserver/erdas-apollo/catalog/content/services` will return you a list of all the services in the catalog
- a single item with id 'abc' can be accessed directly at its URL `http://someserver/erdas-apollo/catalog/content/items/abc`

- alternate encodings can be obtained by appending the desired extension, e.g. `http://someserver/erdas-apollo/catalog/content/services.json`

- REST urls can be augmented with filtering parameters, like `http://someserver/erdas-apollo/catalog/content/services/resources?keywords=hydro&intersect=BBOX(-180 -90,180 90)&maxresults=50`

- a new service can be published by posting its url to `http://someserver/erdas-apollo/catalog/content/services`

All these URLs will be explained in the following sections.

Catalog HTTP endpoints

The REST interface base URL of ERDAS APOLLO Catalog is `http://<serverURL>/erdas-apollo/catalog/content` , so every request done through this interface will start with this pattern.

Most of the object types available in the catalog have their dedicated REST endpoints:

Table 27: Catalog REST endpoints

Object Type	**REST base url**
All object types	`http://<serverURL>/erdas-apollo/catalog/content/`*items*
OGC Services	`http://<serverURL>/erdas-apollo/catalog/content/`*services*
Web Feature Services	`http://<serverURL>/erdas-apollo/catalog/content/`*services/wfs*
Web Map Services	`http://<serverURL>/erdas-apollo/catalog/content/`*services/wms*
Web Coverage Services	`http://<serverURL>/erdas-apollo/catalog/content/`*services/wcs*
Feature Types resources	`http://<serverURL>/erdas-apollo/catalog/content/`*services/wfs/resources*
Map Layer resources	`http://<serverURL>/erdas-apollo/catalog/content/`*services/wms/resources*
Coverage resources	`http://<serverURL>/erdas-apollo/catalog/content/`*services/wcs/resources*
Coverage objects	`http://<serverURL>/erdas-apollo/catalog/content/`*coverages*

Object Type	REST base url
Contact entities	`http://<serverURL>/erdas-apollo/catalog/content/`*contacts*

Issuing a HTTP GET request to any of those URLs will return a list of corresponding objects, with no filtering applied, with default paging settings, and where the encoding is chosen as explained in the dedicated section below.

It must be noted that the first endpoint (`/items`) is the main endpoint to retrieve *any* catalog object. Other endpoints are convenient ways to subset the result set by object types. Also, some specific operations described further are available only on specific endpoints (e.g. publishing a service can only be done by POSTing to the `/service` endpoint).

Catalog REST query parameters

As shown in the previous section, all items in the catalogs can be retrieved using this URL: `http://<serverURL>/rest/catalogItems`

Any of the REST endpoints described above can be appended with the id of an object, thereby providing a way to obtain a single object. E.g. `http://someserver/erdas-apollo/catalog/content/services/{id}` or `http://someserver/erdas-apollo/catalog/content/coverages/{id}` (where {id} is replaced by the given id). However, since the `item` endpoint gives access to all the object types, it is common to always use that endpoint to link to a single object, whatever its actual object type.

Besides this id mechanism, Catalog REST endpoints support various querying parameters, depending on the object type that is being queried. These parameters mainly allow for filtering, ordering and paging of results; some of them are only available for specific object types.

Table 28: REST query parameters

parameter name	value type	supported by type	description
keywords	space separated set of string keywords; default to `none`	all	performs a search in the catalog db and tries to match the given keywords against the name, title or tags of the catalog items

parameter name	value type	supported by type	description
orderby	name of an object attribute; defaults to `none`	all	return the results ordered according to the given attribute name. The attribute must be a valid attribute for the object type being queried. E.g. `./rest/items?orderBy=registrationDate` will return all the objects, with the most recent objects first.
start	integer, defaults to 0	all	paging parameter; specifies the start index for the objects to return.
maxresults	integer, defaults to 10	all	paging parameter; specifies the maximum number of results to return.
intersect	a WKT-encoded geometry; defaults to `none`	FeatureType, MapLayer, Coverage	performs filtering using spatial intersection. E.g. `./rest/services/wms/resources?intersect=BBOX(-180 -90,180 90)`
inurl	part of a domain name	Ogc Services, FeatureType, MapLayer, Coverage	performs filtering on the content of the online resource URL, if any. E.g. `./rest/services?inurl=erdas.com`

Result encodings

Any REST query can be retrieved in various encodings. The encoding is specified by adding the corresponding extension to the URL, before the query parameters. E.g.:
`http://someserver/erdas-apollo/catalog/content/services`**.json**`?keywords=hydro`

If no encoding is specified, the server checks the HTTP headers for preferred encodings, and decides what is the best encoding to choose. Typically, browsers send their HTTP GET requests with HTML/XHTML as preferred encodings; therefore omitting the encoding when typing a URL such as `./rest/services` in browser will directly retrieve the HTML encoding.

The following encodings are currently supported:

Table 29: REST encodings

Encoding	URL extension	description
HTML	`.html`	Using this encoding, results are encoded using an HTML template defined on the server, ready to be displayed in a standard HTML browser. It must be noted that the ERDAS APOLLO Catalog Web client is actually the HTML encoding of the REST interface.

Encoding	URL extension	description
Plain Text	`.txt`	This encoding is more of a debugging feature, to get a quick and simple display of the results. In particular, this encoding can be useful to output the URLs of the service currently stored in the catalog, for a quick backup or transfer, using a URL like: `./rest/services.txt.`
JSON	`.json`	This encoding returns the set of results encoded in JSON, and ready to be interpreted in a Javasacript environment. This is especially useful to build lightweight javascript applications that can run in a browser and can access the catalog data without the need for a server-side middle-tier.
GeoRSS	`.georss`	Using this encoding, the results will be returned as a RSS feed, following the GeoRSS specification that defines ways to add location tags to RSS entries. F.i., this can be useful to be notified of the latest services harvested, by subscribing to the following feed: `./rest/services.georss?orderBy=registrationDate&maxResults=50`
KML	`.kml`	The KML encoding returns the set results encoded using the Google KML encoding, following the ML 2.2 specification. Results are encoded using their spatial extent when available; furthermore, if results are MapLayers, the KML document will contain special overlays that allow you to display the original data from the service directly in the KML browser (typically Google Earth).

The Web Feature Service (WFS)

WFS GetCapabilities

The WFS Capabilities request in HTTP-GET has to mention at a minimum the following parameters (values are case-sensitive, not parameter names):

```
VERSION=1.0.0
SERVICE=WFS
REQUEST=GetCapabilities
```

For a WFS 1.1.0 server, the "AcceptVersions" parameter can also be used and the "Service" parameter is optional and defaults to "WFS". The "updateSequence" and "Sections" parameters are ignored. The "AcceptFormats" parameter is understood but as our WFS only outputs its capabilities in XML, a single format is possible.

In ERDAS APOLLO, a WFS can either support the WFS 1.0.0 specification or the WFS 1.1.0 one, but it does not fully support both. This is due to the fact that a WFS relies on a GML application schema, GML 2 for a WFS 1.0.0 and GML 3 for a WFS 1.1.0. Even though a WFS 1.0.0 can sometimes produce GML 3 output, the opposite is only true for simple schemas (no complex properties, no GML3 types, ...).

A WFS 1.0.0 server will respond to a GetCapabilities request with an XML document with a root element of WFS_Capabilities. It contains a set of sections, namely: Service, Capabilities, FeatureTypeList and Filter_Capabilities.

Since the WFS 1.1.0 specification is based on the "OWS Common Implementation Specification", it will respond with a capabilities document containing a structure with the following set of OWS-Common sections: ServiceIdentification, ServiceProvider and OperationsMetadata. Additionally, the following sections are specific to the WFS: FeatureTypeList, ServesGMLObjectTypeList (not produced by ERDAS APOLLO WFS), SupportsGMLObjectTypeList and Filter_Capabilities.

The WFS Capabilities request in HTTP-POST has to mention at least the following XML elements and attributes:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ogcwfs:GetCapabilities version="1.0.0" service="WFS"
```

```
              xmlns:ogcwfs="http://www.opengis.net/wfs" />
```

For a WFS 1.1.0, the GetCapabilities request can also mention additional parameters in a way similar to the HTTP-GET syntax. The response will have a different structure due to the inheritence from the OWS Common Implementation Specification on which it is based.

WFS DescribeFeatureType

The WFS DescribeFeatureType request in HTTP-GET must declare the following parameters (values are case-sensitive, not parameter names):

```
VERSION=1.0.0
SERVICE=WFS
REQUEST=DescribeFeatureType
TYPENAME=type1,type2,...   (this parameter is optional. By default, all types are described)
```

The WFS DescribeFeatureType request in HTTP-POST must declare the following XML elements and attributes (type names have to be replaced with actual names):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ogcwfs:DescribeFeatureType outputFormat="XMLSCHEMA"
              xmlns:ogc="http://www.opengis.net/ogc"
              xmlns:ogcwfs="http://www.opengis.net/wfs"
              version="1.0.0" service="WFS">
  <ogcwfs:TypeName>TYPE1</ogcwfs:TypeName>
  <ogcwfs:TypeName>TYPE2</ogcwfs:TypeName>
</ogcwfs:DescribeFeatureType>
```

WFS GetFeature

The WFS GetFeature request in HTTP-GET must declare the following parameters (values are case-sensitive, not parameter names):

```
VERSION=1.0.0
SERVICE=WFS
REQUEST=GetFeature
TYPENAME=type1,type2,...
```

Note that the type names have to be replaced with actual names.

GetFeature can be rather complex including explicit output property names, filters on several properties, etc. It is recommended that the HTTP-POST syntax of a GetFeature be used as basic parameters (version, service, request, typename, maxFeatures) are not enough.

The WFS GetFeature request in HTTP-POST must declare the following XML elements and attributes (type and property names have to be replaced with actual names as well as the operator name "PropertyIsEqualTo" and the literal value "Val1"):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ogcwfs:GetFeature maxFeatures="20"
              xmlns:ogc="http://www.opengis.net/ogc"
              xmlns:ogcwfs="http://www.opengis.net/wfs"
              version="1.0.0" service="WFS" >
 <ogcwfs:Query typeName="TYPE1">
   <ogc:PropertyName>TYPE1.PROP1</ogc:PropertyName>
   <ogc:Filter>
     <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>TYPE1.PROP2</ogc:PropertyName>
        <ogc:Literal>Val1</ogc:Literal>
     </ogc:PropertyIsEqualTo>
   </ogc:Filter>
 </ogcwfs:Query>
</ogcwfs:GetFeature>
```

Note that a wide variety of GetFeature requests can be built, with many types of filtering operators. Please refer to the "WFS Parameters of a GetFeature" section below, or to the OGC WFS 1.0.0 and 1.1.0 Implementation Specifications for the exact syntax and schema.

WFS Transaction

If ERDAS APOLLO Server WFS is configured to support editing and modification, it is considered to be transactional (WFS-T). It will support "Transaction" operations such as Insert, Update, Delete and possibly LockFeature.

Even though the WFS-T request supports three operations (Insert, Update, Delete), it only allows the Delete operation for HTTP-GET. Transactional Insert and Update operations, in practice, would be very lenghthy expressions not suitable syntax for the HTTP-GET request.

The WFS-T Transaction request for a Delete operation in HTTP-GET must declare the following parameters (values are case-sensitive, not parameter names):

```
VERSION=1.0.0
SERVICE=WFS
REQUEST=Transaction
OPERATION=Delete
TYPENAME=type1,type2,...  (optional if FEATUREID is mentionned)
FEATUREID                 (optional)
RELEASEACTION             (optional)
FILTER                    (optional)
BBOX                      (optional)
```

The WFS-T Transaction request for a Delete operation in HTTP-POST must declare the following XML elements and attributes (type names have to be replaced with actual names):

```
<?xmlversion="1.0"?>
<wfs:Transaction version="1.0.0" service="WFS"
xmlns:myns="http://www.someserver.com/myns"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
 <wfs:Delete typeName="myns:TYPE1">
  <ogc:Filter>
   <ogc:FeatureId fid="TYPE1.1013"/>
  </ogc:Filter>
 </wfs:Delete>
</wfs:Transaction>
```

The WFS-T Transaction request for an Insert operation in HTTP-POST must declare the following XML elements and attributes (type names have to be replaced with actual names):

```
<?xmlversion="1.0"?>
<wfs:Transaction version="1.0.0" service="WFS"
xmlns:au1="http://www.erdas.com/wfs"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
 <wfs:Insert>
  <au1:hydro fid="hydro.0">
   <au1:FNODE_>2</au1:FNODE_>
   <au1:TNODE_>1</au1:TNODE_>
   <au1:LENGTH>53.126</au1:LENGTH>
   <au1:HD_ID>378</au1:HD_ID>
   <au1:TILE_NAME>126</au1:TILE_NAME>
   <au1:CLASS/>
   <au1:MINOR_NUM>204</au1:MINOR_NUM>
   <au1:GEOMETRY>
    <gml:LineString srsName="EPSG:26986">
     <gml:coordinates>
      240397.25,902849.25 240447.109375,902849.5 240450.375,902849.5625
     </gml:coordinates>
    </gml:LineString>
   </au1:GEOMETRY>
  </au1:hydro>
 </wfs:Insert>
</wfs:Transaction>
```

The WFS-T Transaction request for an Update operation in HTTP-POST must declare the following XML elements and attributes (type names have to be replaced with actual names):

```
<?xmlversion="1.0"?>
<wfs:Transaction version="1.0.0" service="WFS"
xmlns:au1="http://www.erdas.com/wfs"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
 <wfs:Update typeName="au1:hydro">
  <wfs:Property>
   <wfs:Name>TNODE</wfs:Name>
   <wfs:Value>8</wfs:Value>
  </wfs:Property>
  <ogc:Filter>
   <ogc:FeatureId fid="hydro.0"/>
  </ogc:Filter>
 </wfs:Update>
</wfs:Transaction>
```

WFS LockFeature

The WFS LockFeature request in HTTP-GET must declare the following parameters (values are case-sensitive, not parameter names):

```
VERSION=1.0.0
SERVICE=WFS
REQUEST=LockFeature
TYPENAME=type1,type2,... (optional if FEATUREID is mentionned)
FEATUREID                (optional)
EXPIRY                   (optional)
LOCKACTION               (optional)
FILTER                   (optional)
BBOX                     (optional)
```

The WFS LockFeature request in HTTP-GET must declare the following parameters (values are case-sensitive, not parameter names):

```
<?xmlversion="1.0"?>
<LockFeature version="1.0.0"
             service="WFS" lockAction="SOME"
             xmlns="http://www.opengis.net/wfs"
             xmlns:au1="http://www.erdas.com/wfs"
             xmlns:ogc="http://www.opengis.net/ogc"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs../wfs/1.0.0/WFS-transaction.xsd">
<Lock typeName="au1:hydro">
<ogc:Filter>
<ogc:FeatureId fid="hydro.0"/>
<ogc:FeatureId fid="hydro.1"/>
</ogc:Filter>
</Lock>
```

```
</LockFeature>
```

Other WFS Request Types

The WFS GetGmlObject and GetFeatureWithLock are defined in the WFS specification but unsupported by ERDAS APOLLO Server WFS.

WSDL and WFS SOAP requests

In order to be valid in terms of a Service Oriented Architecture (SOA), a web service has to publish a WSDL document that describes its content and allows recording in a UDDI registry, such as a Systinet Registry. As soon as it is registered, the service can be invoked using the SOAP protocol. This is a particular type of XML encoding embedded in a classical HTTP request or response.

OGC has not voted a standard for a WFS to provide its description in WSDL. At ERDAS, we have chosen to follow one of the OGC candidate solutions, a request=getWSDL request type. The syntax is similar to a classical HTTP-GET GetCapabilities request, except that the request name is "getWSDL" and an optional "SoapOnly" parameter is allowed. Note that the "Service" parameter will determine the nature of the output WSDL as each service type (WMS, WFS, WCS, ...) has a different WSDL description. Example of a GetWSDL request:

```
http://localhost:8080/erdas-apollo-
demo/vector/ATLANTA_VECTOR?version=1.0.0&service=WFS&request=getWSDL&SoapOnly=true
```

Note that the ERDAS APOLLO Server Feature Framework also supports the OGC WMS interface, and a WMS WSDL can also be obtained by running the appropriate getWSDL request:

```
http://localhost:8080/erdas-apollo-
demo/vector/ATLANTA_VECTOR?version=1.1.1&service=WMS&request=getWSDL&SoapOnly=true
```

In the context of OGC Web Feature Services, SOAP requests are very similar to POST requests; they contain a XML body which must be encapsulated into SOAP-specific elements in the case of SOAP requests. The root element is "Envelope" and its only child is a "Body" element.

Below is a sample GetCapabilities request on a WFS server using a SOAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
```

```
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                    xmlns:gml="http://www.opengis.net/gml">
 <SOAP-ENV:Body>
  <ogcwfs:GetCapabilities version="1.0.0" service="WFS"
                xmlns:ogcwfs="http://www.opengis.net/wfs" />
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

WFS Parameters of a GetFeature

The table below lists all the parameters that can be used in an HTTP-GET GetFeature request. Most parameters are from the OGC-WFS 1.0.0 Implementation Specification. However, some of the parameters are part of the OGC WFS 1.1.0 specification but absent from the OGC WFS 1.0.0 specification. These are identified by "*", in the Required/Optional column. For more details, please refer to the OGC specifications.

Table 30: The Parameters of an OGC-WFS 1.0.0 GetFeature HTTP-GET Request

Request Parameter	Required/Optional	Description
VERSION=version	R	Request version (default is 1.0.0 or 1.1.0).
REQUEST=GetFeature	R	Request name.
SERVICE=WFS	R	Service name.
TYPENAME=feature_type_name_list	R	Comma-separated list of one or more feature type names to query. Optional if the FEATUREID parameter is given.
PROPERTYNAME=property_name_list	O	Comma-separated list of one or more property names. "*" means all properties. Default: all properties.
FEATUREID=fid	O	An enumerated list of feature instances to fetch identified by their feature identifiers.
FEATUREVERSION=version_number	O	Directs the WFS on which feature version to fetch. Possible values are "ALL" or a number. Default: ALL.
MAXFEATURES=nb_features	O	Maximum number of features that the WFS should return in response to the query.
OUTPUTFORMAT=output_format	O*	Output format of the feature collection. Default is GML2. Other values are GML3, SHAPE.
FILTER	O	URL-encoded XML Filter expression (see example below), as specified in the OGC Filter Encoding Specification 1.0.

Request Parameter	Required/Optional	Description
SRSNAME=namespace:identifier	O	Spatial Reference System of the output features. Only since WFS 1.1.0.
BBOX=minx,miny,maxx,maxy	O	Bounding box corners (lower left, upper right) in the native SRS units, to be used in lieu of FEATUREID or FILTER.
Vendor-specific parameters	O	Optional experimental parameters.
SORTBY=property_name_list	O*	Comma-separated list of one or more property names, possibly suffixed with "+A" (ascending) or "+D" (descending). Allows to sort the result based on one or more property names, and a sorting order. The default sort order is ascending or "+A".
RESULTTYPE=response_type	O	Do we generate a complete response or only the count of features. Only since WFS 1.1.0. Possible values are "results" and "hits". The default value is "results".

A sample FILTER parameter value (not URL-encoded):

```
FILTER=<Filter><Within><PropertyName>ROADS.GEOMETRY<PropertyName><gml:Box><gml><coordinate
s>10,10 20,20</gml:coordinates></gml:Box></Within></Filter>
```

The second way to build a WFS GetFeature request is in HTTP-POST. The body of the request is encoded in XML. The <GetFeature> element contains one or more <Query> elements which contain the description of a query. The results of all queries contained in a GetFeature request are concatenated to produce the result.

The XML encoding of a WFS 1.0.0 GetFeature request is defined by the following XML Schema fragment:

```
<xsd:element name="GetFeature" type="wfs:GetFeatureType"/>
<xsd:complexType name="GetFeatureType">
 <xsd:sequence>
  <xsd:element ref="wfs:Query" maxOccurs="unbounded"/>
 </xsd:sequence>
 <xsd:attribute name="version" type="xsd:string" use="required" fixed="1.0.0"/>
 <xsd:attribute name="service" type="xsd:string" use="required" fixed="WFS"/>
 <xsd:attribute name="handle" type="xsd:string" use="optional"/>
 <xsd:attribute name="outputFormat" type="xsd:string" use="optional" default="GML2"/>
 <xsd:attribute name="maxFeatures" type="xsd:positiveInteger" use="optional"/>
</xsd:complexType>
<xsd:element name="Query" type="wfs:QueryType"/>
<xsd:complexType name="QueryType">
 <xsd:sequence>
  <xsd:element ref="wfs:SortBy" minOccurs="0" maxOccurs="1" />
```

```
<!-- see details below for the SortBy element -->
  <xsd:element ref="ogc:PropertyName" minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1"/>
 </xsd:sequence>
 <xsd:attribute name="handle" type="xsd:string" use="optional"/>
 <xsd:attribute name="typeName" type="xsd:QName" use="required"/>
 <xsd:attribute name="featureVersion" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:element name="SortBy" type="wfs:SortByType" />
```

Most of the attributes have the same meaning as in the HTTP-GET request and are explained in the table above.

Handle: The purpose of the handle attribute is to allow a client application to associate a mnemonic name with a request for error handling purposes. If a handle is specified and an exception is encountered, a Web Feature Service may use the handle to identify the offending element.

The <SortBy> element is not in the OGC-WFS 1.0.0 specification but is available in OGC WFS 1.1.0. ERDAS APOLLO supports it for both WFS 1.0 and 1.1. The SortBy clause allows a client to request that a WFS result be sorted by specific properties and in specific order. As with the Filter, a SortBy clause would be specified in a Query request. The following XML Schema fragment defines the <SortBy> element semantic.

```
<xsd:complexType name="SortByType">
 <xsd:sequence>
  <xsd:element ref="wfs:PropertyName" minOccurs="1" maxOccurs="unbounded" />
 </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PropertyName">
 <xsd:complexContent>
  <xsd:extension base="ogc:PropertyName">
   <xsd:attribute name="sortOrder" type="xsd:string" use="optional" />
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

Streaming Support

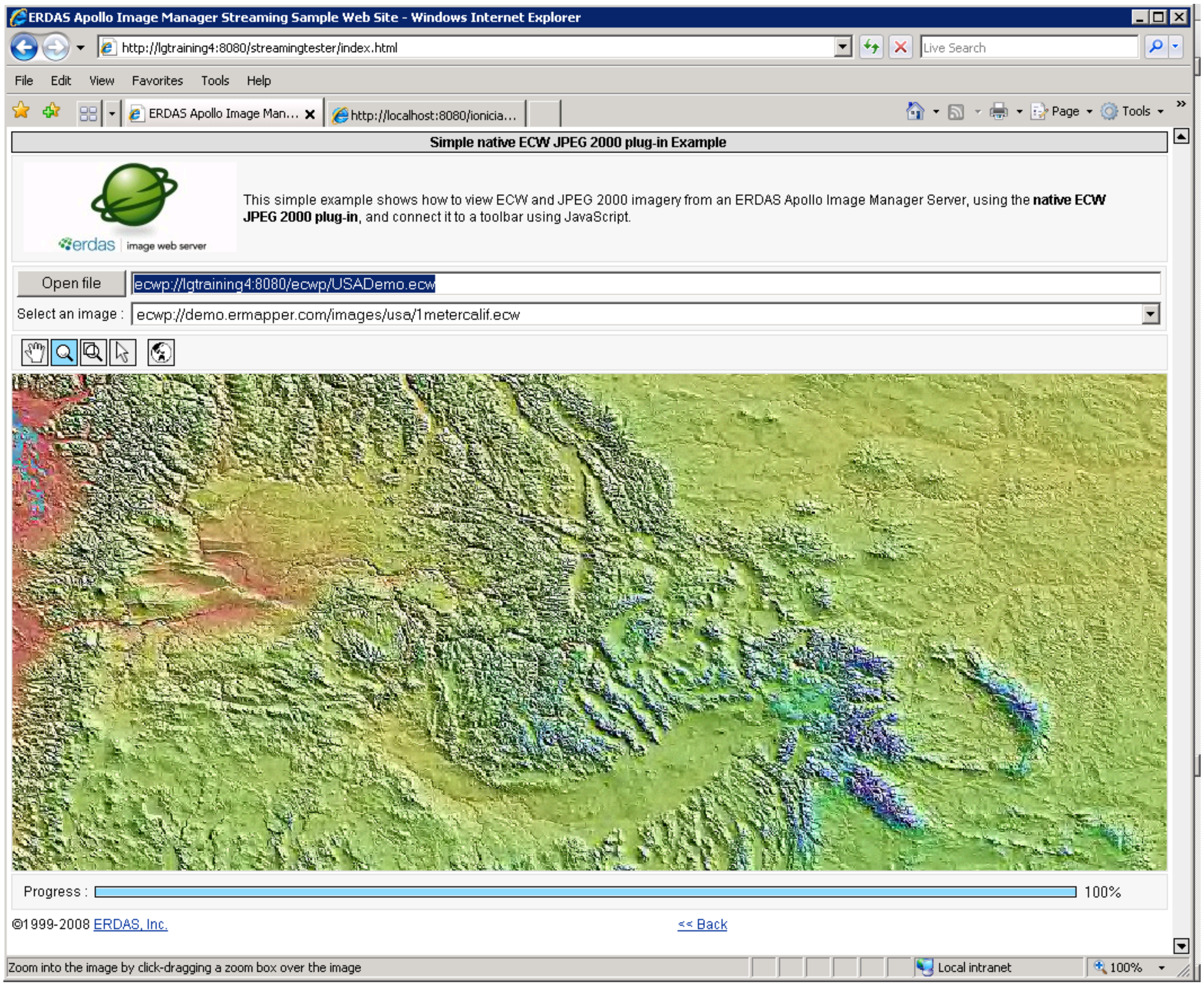
In this chapter:

Overview of Streaming Support

ERDAS APOLLO is able to stream some images using wavelet-based image compression. The ECW files from ERMapper can be streamed using the Enhanced Wavelet Compression Protocol (ECWP) and JPEG2000 files can be streamed using the JPEG 2000 Interactive Protocol (JPIP).

ERDAS APOLLO provides a web page that you can use to test streaming ability. That web page is: **http://[server]:8080/streamingtester/index.html.**

Figure 46: The Streaming Test Web Page


ERDAS Apollo Image Manager Streaming Sample Web Site - Windows Internet Explorer
http://lgtraining4:8080/streamingtester/index.html
Live Search
File
Edit
View
Favorites
Tools
Help
ERDAS Apollo Image Man...
http://localhost:8080/ionicia...
Page
Tools
erdas
image web server

Simple native ECW JPEG 2000 plug-in Example

This simple example shows how to view ECW and JPEG 2000 imagery from an ERDAS Apollo Image Manager Server, using the **native ECW JPEG 2000 plug-in**, and connect it to a toolbar using JavaScript.

Open file: ecwp://lgtraining4:8080/ecwp/USADemo.ecw

Select an image : ecwp://demo.ermapper.com/images/usa/1metercalif.ecw

Progress : 100%

©1999-2008 ERDAS, Inc. << Back

Zoom into the image by click-dragging a zoom box over the image

Local intranet 100%

ECWP Streaming

ERDAS APOLLO allows the ECW images from ERMapper to be streamed. Try to crawl an ECW image using the Data Manager, and enter the following URL in the Open File field of the streaming test web page:

```
ecwp://[server]:8080/ecwp/[datasetName].ecw
```

For example, if your ERDAS APOLLO Server is apollo.mycompany.com and you have added an image named world.ecw, enter:

```
ecwp://apollo.mycompany.com:8080/ecwp/world.ecw
```

JPIP Streaming

JPEG2000 images can be streamed using the JPIP protocol. Try to crawl a JPEG2000 image using the Data Manager, and enter the following URL in the Open File field of the streaming test web page:

```
jpip://[server]:8080/jpip/[datasetName].jp2
```

For example, if your ERDAS APOLLO server is apollo.mycompany.com and you have added an image named world.jp2, enter:

```
jpip://apollo.mycompany.com:8080/jpip/world.jp2
```

Accessing Secure Resources

You can access secure resources that require a user name and password using the same methods shown above.

The user name and password of the account must be appended to the URLs provided above, separated by pipe characters (|). For example, you would use the following URL to access the image world.jp2 with the user name ‘manager’ and the password ‘JFKE8uek3’ on the apollo.mycompany.com server:

```
jpip://apollo.mycompany.com:8080/jpip/world.jp2|manager|JFKE8uek3
```