ERDAS APOLLO Administrator’s Guide

Advantage/Professional Edition

August 2011

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.

Government Reserved Rights. MrSID technology incorporated in the Software was developed in part through a project at the Los Alamos National Laboratory, funded by the U.S. Government, managed under contract by the University of California (University), and is under exclusive commercial license to LizardTech, Inc. It is used under license from LizardTech. MrSID is protected by U.S. Patent No. 5,710,835. Foreign patents pending. The U.S. Government and the University have reserved rights in MrSID technology, including without limitation: (a) The U.S. Government has a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by U.S. Patent No. 5,710,835 and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations; (b) If LizardTech's rights in the MrSID Technology terminate during the term of this Agreement, you may continue to use the Software. Any provisions of this license which could reasonably be deemed to do so would then protect the University and/or the U.S. Government; and (c) The University has no obligation to furnish any know-how, technical assistance, or technical data to users of MrSID software and makes no warranty or representation as to the validity of U.S. Patent 5,710,835 nor that the MrSID Software will not infringe any patent or other proprietary right. For further information about these provisions, contact LizardTech, 1008 Western Ave., Suite 200, Seattle, WA 98104.

Table of Contents

Introduction

In this chapter:

What is Covered in This Guide?

This guide explains how to customize a standard installation of ERDAS APOLLO Server (Advantage/Professional).

The changes listed in the chapter **Post-Installation Tuning of ERDAS APOLLO** on page 3 chapter explain how to make the customizations that almost all of our customers want to make to their ERDAS APOLLO systems.

The changes listed in the chapter **Advanced Configuration of ERDAS APOLLO** on page 31 explain how to make customizations that many, though not all, of our customers usually want to make to their ERDAS APOLLO systems.

Audience

This guide is intended for people who will serve as system administrators for the ERDAS APOLLO system.

The instructions contained in this guide will assume that the reader can navigate the file structure of a computer and can perform basic tasks such as copying or moving files in a Windows or UNIX-type operating system.

The instructions for working with databases are intended for users with solid database administration skills.

Additional Help

- The ***ERDAS APOLLO Concepts Guide*** explains the terminology and technology that you need to know in order to understand the ERDAS APOLLO product. 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 of 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 and 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.

Conventions

Because you can choose where to install a program, the root directory where it is installed is referred to <*APPNAME_HOME*>, where APPNAME represents the name of the program.

Helpful Hints

A lot of the configuration tasks will require you to open different kinds of archive files, such as EAR, War, and JAR files.

ERDAS recommends that you use 7-zip, a free program that allows you to easily open archive files so you can work with their contents. You can download this program at **www.7-zip.org**.

Post-Installation Tuning of ERDAS APOLLO

In this chapter:

Introduction

The ERDAS APOLLO system is designed to work with the most common system configurations as soon as you purchase and install it. However, you may have a slightly different configuration and may need to make some small changes to adapt the ERDAS APOLLO system to it. We also find that many of our customers want to customize certain aspects of the ERDAS APOLLO system.

As ERDAS APOLLO runs, it gathers certain information about how it should run from different files that are located inside of its installation directory. By changing certain things inside these files, you can easily alter the way ERDAS APOLLO works.

The changes explained inside this chapter are some of the more basic configuration changes that you are very likely to require or want.

Specifying the Storage Directories for Metadata, Thumbnails, & Pyramids

The ERDAS APOLLO system uses files to store information about the thumbnails, pyramid layers for catalog items, metadata for catalog items, and the output from geoprocesses executed in the web client. By default, those files are located in the directories specified in the table below.

Table 1: Location of Metadata, Pyramid Layer, Thumbnail, and Geoprocess Output Files

Resource Name	Directory
Thumbnails	<APOLLO_HOME>\storage\legend\coverage\EAIM
Pyramid Layers	<APOLLO_HOME>\storage\pyramids\coverage\EAIM
Metadata	<APOLLO_HOME>\storage\metadata\coverage\EAIM
WPS Output	<APOLLO_HOME>\storage\wps\isms\process_output

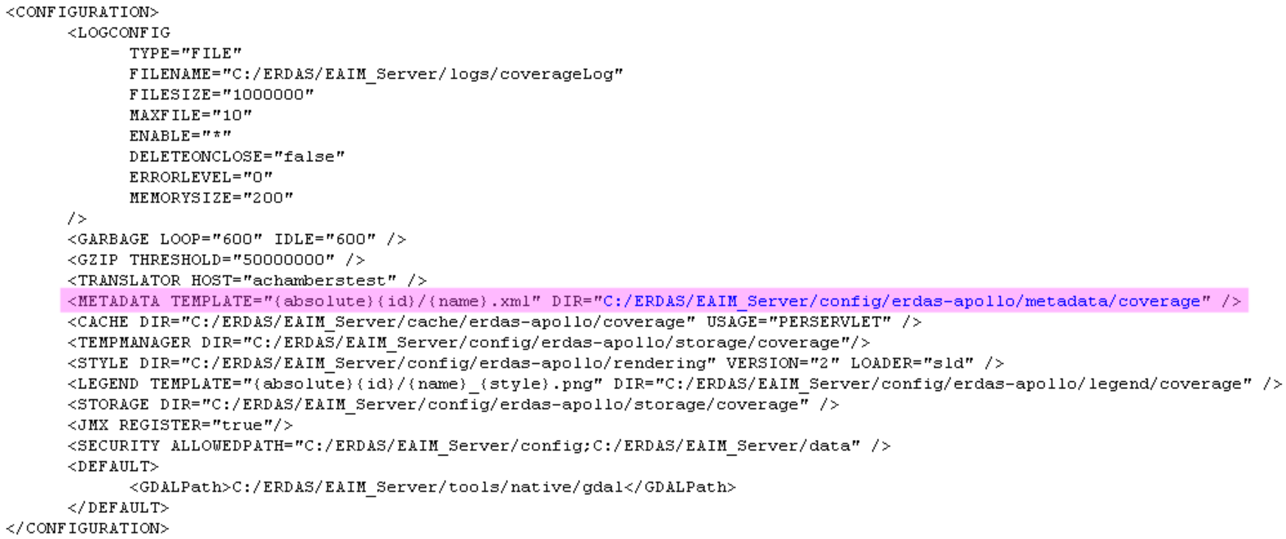
To store these files in different directories, you will need to edit some files in ERDAS APOLLO.

Changing the Storage Location for Metadata Files

1. Navigate to the directory `<APOLLO_HOME>\config\erdas-apollo\providers\coverage`.

2. Open the file `providers.fac`.

3. At the bottom of that file, you will find a block of configuration parameters.
   Find the one the one that says **METADATA TEMPLATE**
   .



```
<CONFIGURATION>
      <LOGCONFIG
            TYPE="FILE"
            FILENAME="C:/ERDAS/EAIM_Server/logs/coverageLog"
            FILESIZE="1000000"
            MAXFILE="10"
            ENABLE="*"
            DELETEONCLOSE="false"
            ERRORLEVEL="0"
            MEMORYSIZE="200"
      />
      <GARBAGE LOOP="600" IDLE="600" />
      <GZIP THRESHOLD="50000000" />
      <TRANSLATOR HOST="achamberstest" />
      <METADATA TEMPLATE="{absolute}{id}/{name}.xml" DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/metadata/coverage" />
      <CACHE DIR="C:/ERDAS/EAIM_Server/cache/erdas-apollo/coverage" USAGE="PERSERVLET" />
      <TEMPMANAGER DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/storage/coverage"/>
      <STYLE DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/rendering" VERSION="2" LOADER="sld" />
      <LEGEND TEMPLATE="{absolute}{id}/{name}_{style}.png" DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/legend/coverage" />
      <STORAGE DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/storage/coverage" />
      <JMX REGISTER="true"/>
      <SECURITY ALLOWEDPATH="C:/ERDAS/EAIM_Server/config;C:/ERDAS/EAIM_Server/data" />
      <DEFAULT>
            <GDALPath>C:/ERDAS/EAIM_Server/tools/native/gdal</GDALPath>
      </DEFAULT>
</CONFIGURATION>
```

4. Change the directory currently specified to the directory where you want ERDAS APOLLO to store your metadata files.

5. Save file `providers.fac` file and close it.

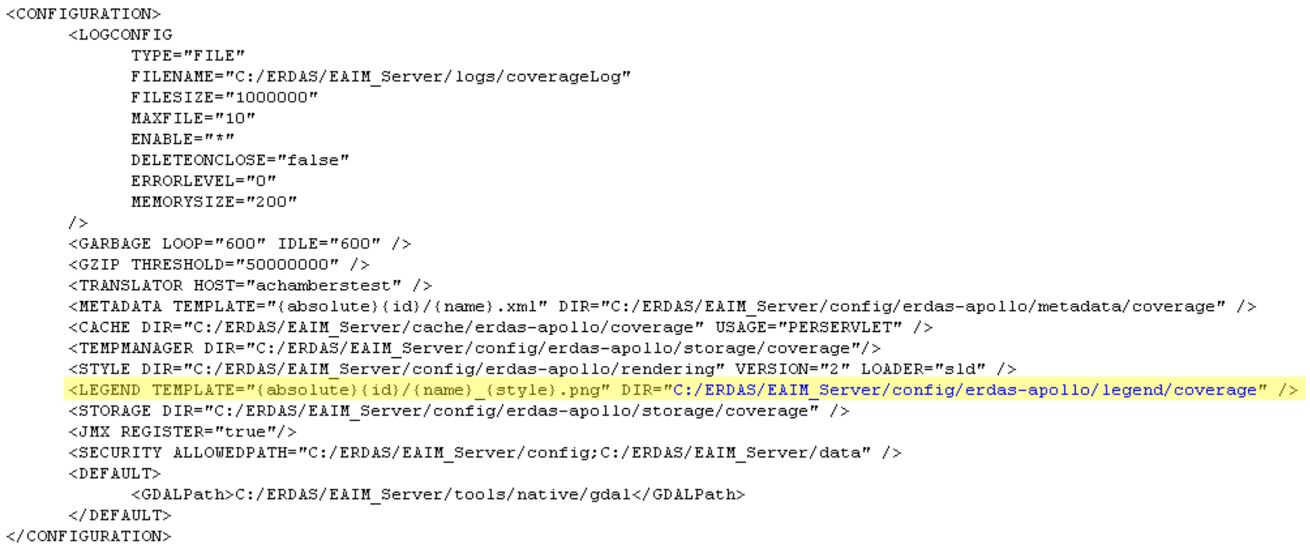
If you have already started your application server, you will need to restart it for the change you just made to take effect.

Changing the Storage Location for Thumbnail Files

1. Navigate to the directory `<APOLLO_HOME>\config\erdas-apollo\providers\coverage`.

2. Open the file `providers.fac`.

3. At the bottom of that file, you will find a block of configuration parameters.
   Find the one the one that says **LEGEND TEMPLATE**.



```
<CONFIGURATION>
      <LOGCONFIG
            TYPE="FILE"
            FILENAME="C:/ERDAS/EAIM_Server/logs/coverageLog"
            FILESIZE="1000000"
            MAXFILE="10"
            ENABLE="*"
            DELETEONCLOSE="false"
            ERRORLEVEL="0"
            MEMORYSIZE="200"
      />
      <GARBAGE LOOP="600" IDLE="600" />
      <GZIP THRESHOLD="50000000" />
      <TRANSLATOR HOST="achamberstest" />
      <METADATA TEMPLATE="{absolute}{id}/{name}.xml" DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/metadata/coverage" />
      <CACHE DIR="C:/ERDAS/EAIM_Server/cache/erdas-apollo/coverage" USAGE="PERSERVLET" />
      <TEMPMANAGER DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/storage/coverage"/>
      <STYLE DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/rendering" VERSION="2" LOADER="sld" />
      <LEGEND TEMPLATE="{absolute}{id}/{name}_{style}.png" DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/legend/coverage" />
      <STORAGE DIR="C:/ERDAS/EAIM_Server/config/erdas-apollo/storage/coverage" />
      <JMX REGISTER="true"/>
      <SECURITY ALLOWEDPATH="C:/ERDAS/EAIM_Server/config;C:/ERDAS/EAIM_Server/data" />
      <DEFAULT>
            <GDALPath>C:/ERDAS/EAIM_Server/tools/native/gdal</GDALPath>
      </DEFAULT>
</CONFIGURATION>
```

4. Change the directory currently specified to the directory where you want ERDAS APOLLO to store your thumbnail files.

5. Save the `providers.fac` file and close it.

If you have already started your application server, you will need to restart it for the change you just made to take effect.

Pyramid Files Location

The storage location for pyramid files is defined in three files. Change all locations to the new pyramid location.

- im-providers.fac
- rds.policy
- global-processmanager.properties

1. Navigate to the directory `<APOLLO_HOME>\config\erdas-apollo\providers\coverage`.
2. Open the file `im-providers.fac`.
3. Look for `PyramidDir` and change the path in the `Value` field.
4. Save the file.
5. Navigate to the directory `<APOLLO_HOME>\config\erdas-apollo\tools\native\nci`.
6. Open the file `rds.policy`.
7. Look for the line that contains `pyramids` in the path.
8. Change the path using the conventions where `${/}` equals a forward slash `/`.
9. Save the file.
10. Navigate to: `APOLLO_HOME\config\erdas-apollo\`
11. Open the file `global-processmanager.properties.`
12. Modify the rds.proxy.directory element to the new pyramid file location.
13. Save the file.
14. Restart your application server.

Specifying the Storage Directories for WPS Models and Outputs

The WPS models are created in ERDAS IMAGINE and are then published in ERDAS APOLLO so that those users can run the processes using data that is stored in their ERDAS APOLLO catalogs.

By default, the published models are stored in the directory `<APOLLO_HOME>\storage\wps\isms` and the WPS outputs are stored in the directory `<APOLLO_HOME>\storage\wps\isms\process_output`.

These locations should work for most organizations, but you can change the directories that are used to store the models or the outputs.

Changing the Storage Location for WPS Models

1. Navigate to the directory `<APOLLO_HOME>\config\erdas-apollo\providers\process`
2. Open the file `providers.fac`.
3. Find the **ISMSDIR** parameter.
4. Change the directory currently specified to the directory where you want ERDAS APOLLO to store the WPS models that people publish to it from ERDAS IMAGINE.
5. Save the `providers.fac` file and close it.
6. If you have already started your application server, you will need to restart it for the change you just made to take effect.

Changing the Storage Location for WPS Outputs

1. Navigate to the directory `<APOLLO_HOME>\config\erdas-apollo\providers\process`
2. Open the file `providers.fac`.
3. Look for the Configuration parameters. Within those parameters, find **TEMPMANAGER DIR**.
4. Change the directory currently specified to the directory where you want ERDAS APOLLO to store the outputs that are generated when you run a WPS process in the Web Client.
5. Save the `providers.fac` file and close it.
6. If you have already started your application server, you will need to restart it for the change you just made to take effect.

JBoss Tuning and Configuration

A special, preconfigured JBoss application server is provided with ERDAS APOLLO. If you want to use it, the ERDAS APOLLO product installer can install it and automatically deploy ERDAS APOLLO on it. This preconfigured version of JBoss will run on the majority of systems, however, there are some cases where it may need to be adjusted slightly to fit your system.

The JBoss JVM

The preconfigured JBoss that is provided by ERDAS APOLLO contains its own Java Virtual Machine (JVM). The settings that affect the performance of this JVM are stored in the `run_N86.bat`(32-bit) or `run_amd64.bat` (64-bit) file located in the `<APOLLO_HOME>\jboss\bin` directory.

ERDAS has performed extensive performance tests and has set this JVM so that it should provide optimal performance on most systems. It is generally best to retain those factory settings.

Memory Options

These options specify the memory used by the JVM. The values for each option include a number and m or M for megabytes, k or K for kilobytes, and g or G for gigabytes.

If you have a lot of available memory, you can increase the performance of ERDAS APOLLO by increasing the value of the -Xmx option, or by disabling the Windows paging file.

Table 2: Memory Options for the Java Virtual Machine

Name	32-bit Value	64-bit Value	Description
-Xms	1024m	1024m	Specify the initial size of the memory allocation pool
-Xmx	1024m	4096m	Specify the maximum size of the memory allocation pool
-XX:PermSize	128m	128m	Minimum size of the Permanent Generation
-XX:MaxPermSize	256m	256m	Maximum size of the Permanent Generation
-XX:NewSize	256m	N/A	Size of new generation
-Xss	64k	N/A	Set thread stack size

Behavioral Options

These options define the behavior of the JVM.

Table 3: Behavior Options for the Java Virtual Machine

Name	Description
-XX:+UseFastAccessorMethods	Use optimized versions of Get<Primitive>Field.
-XX:+UseTLAB	Use thread-local object allocation

JBoss Ports and a Firewall

JBoss comes with many socket based services that open listening ports. In this section we list the services that open ports that might need to be configured to work when accessing JBoss behind a firewall. The following table, shows the ports, socket type, associated service and link to the service configuration for the services in the default configuration file set.

Table 4: Ports Needed to Access JBoss behind a Firewall

Port/Type	Service Descriptor	Service Name	Attribute Name
1098 TCP	conf/jboss-service.xml	jboss:service=Naming	RmiPort
1099 TCP	conf/jboss-service.xml	jboss:service=Naming	Port
3873 TCP	deploy/ejb3.deployer/META-INF/jboss-service.xml	jboss.remoting:type=Connector,name=DefaultEjb3Connector,handler=ejb3	InvokerLocator
4444 TCP	conf/jboss-service.xml	jboss:service=invoker,type=jrmp (legacy 4.0.x invoker)	RMIObjectPort
4445 TCP	conf/jboss-service.xml	jboss:service=invoker,type=jrmp (legacy 4.0.x invoker)	ServerBindPort
4446 TCP	conf/jboss-service.xml	jboss.remoting:service=Connector,transport=socket (EJB2 beans in AS/EAP 4.2+)	serverBindPort on Configuration
8009 TCP	deploy/jboss-webdeployer/server.xml	jboss.web:service=WebServer	port on AJP Connector
8083 TCP	conf/jboss-service.xml	jboss:service=WebService	Port
8093 TCP	deploy/jms/uil2-service.xml	jboss.mq:service=InvocationLayer,type=UIL2	ServerBindPort
1100 TCP	deploy/cluster-service.xml	jboss:service=HAJNDI	Port
1101 TCP	deploy/cluster-service.xml	jboss:service=HAJNDI	RmiPort
1102 UDP	deploy/cluster-service.xml	jboss:service=HAJNDI	AutoDiscoveryGroup
4447 TCP	deploy/cluster-service.xml	jboss:service=invoker,type=jrmpha (legacy 4.0.x invoker)	RMIObjectPort
4448 TCP	deploy/cluster-service.xml	jboss:service=invoker,type=pooledha (legacy 4.0.x invoker)	ServerBindPort

Table 4: Ports Needed to Access JBoss behind a Firewall

Port/Type	Service Descriptor	Service Name	Attribute Name
49152 TCP	deploy/cluster-service.xml	jboss:service=${jboss.partition.name:DefaultPartition}	start_port on FD_SOCK

Clip, Zip, and Ship Configuration

The Clip, Zip, and Ship feature allows you to select multiple coverages for download in the ERDAS APOLLO Web Client. In response to the download request, a ZIP file containing the coverages is created on the server. Then, an e-mail with the URL to the location of the ZIP file is sent to the user.

You can download data of any type, as long as it is WCS-enabled.

Clip, Zip, and Ship supports the following output formats:

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

How Large Are the Output Images?

The GeoTIFF, IMAGINE, and National Imagery Transmission Format (NITF) are uncompressed. The size of an output image in these formats image is calculated using the following formula:

Image Size (*in bytes*) = [**Image Width** (*in pixels*) * **Image Height** (*in pixels*)] * **Number of Bands**

For example, if an image were 1000 pixels wide, 1000 pixels high, and consisted of 3 bands, the file size for that image would be 3,000,000 bytes.

And since 1024 bytes = 1 KB and 1024 KB = 1 MB, the image size would be about 2.86 MB.

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 5: 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

Download Zip Files from an FTP Server

By default, the zip files created by the Clip, Zip, and Ship feature are placed in the directory

```
<APOLLO_HOME>\storage\htdocs\provisioning
```

You can configure the ERDAS APOLLO system so that the zip files are placed on an FTP server instead.

Install FTP Server

We use the Internet Information Services (IIS) FTP component to serve the result of the clip, zip and ship work flow.

To install Internet Information Services (IIS) 6.0

1. Open the Windows Components Wizard (Start->Control Panel->Add or Remove Programs->Add/Remove Windows Components)
2. Under Components, click Application Server, and then click Details.
3. In the Application Server dialog box, under Subcomponents of Application Server, click Internet Information Services (IIS), and then click Details.
4. In the Internet Information Services (IIS) dialog box, under Subcomponents of Internet Information Services (IIS), add ftp optional component.
5. Click OK until you return to the Windows Component Wizard.
6. Click Next, and then click Finish.

Configure the FTP Service for Anonymous Login

Refer to http://support.microsoft.com/kb/323384 for more information.

To configure the FTP Service to allow only anonymous connections, follow these steps.

1. Start Internet Information Services Manager or open the IIS snap-in.
2. Expand Server_name, where Server_name is the name of the server.
3. Expand FTP Sites.
4. Right-click Default FTP Site, and then click Properties.
5. Click the Security Accounts tab.
6. Click to select the Allow Anonymous Connections check box (if it is not already selected), and then click to select the Allow only anonymous connections check box.

NOTE: When you select the Allow only anonymous connections check box, you configure the FTP Service to allow only anonymous connections. Users cannot log on using user names and passwords.

7. Click the Home Directory tab.
8. Click to select the Read and Log visits check boxes (if they are not already selected), and then click to clear the Write check box (if it is not already cleared).
9. Click OK.
10. Quit Internet Information Services Manager or close the IIS snap-in.

Configure the FTP Service for Security

This must be set up using proper domain user names

1. Start Internet Information Services Manager.
2. Expand Server_name, where Server_name is the name of the server.
3. Expand FTP Sites.
4. Right-click Default STP Side, and the click Stop.
5. Right-click on FTP Sites, and select New -> FTP Site to open the FTP Site Creation Wizard.

6. In the FTP Site Description field, enter the name you want to give the FTP Site in the Manager.

7. Leave the IP Address and Port Settings as the Default values.

8. On FTP User Isolation, select Isolate Users option.

9. On FTP Site Home Directory browse to the location of your created FTP home directory. Create the home directory inside <local host> C:\Inetpub example: C:\Inetpub\ftpsite

10. Leave FTP Site Access Permissions to default Read.

11. Finish the Wizard.

12. You may need to Right-click Start your new FTP Site.

13. In Windows Explorer create directories inside your FTP home directory that correspond to the domain and domain users and set Read permissions to those folders to that user Example: <DOMAIN>\rsmith.

14. In IIS, right-click your FTP Site and select New -> Virtual Directory to open the Virtual Directory Creation Wizard.

15. On the Virtual Directory Alias set the Alias to the user name Example: rsmith.

16. On the FTP Site Content Directory set the Path to the directory location. Example: C:\Inetpub\ftpsite\<DOMAIN>\rsmith

17. Leave Virtual Directory Access Permissions default set to Read.

18. Finish the Wizard.

19. Repeat these steps for multiple users as long as you have network passwords.

Configure Installation to use FTP

1. Configure the user

   - Add a user using Data Manager, use the same user name as the user name configured in FTP server.

   - Associate the appropriate role (esp_data_manager) to this user.

2. Configure provisioning properties

- Configure provisioning.output.directory from Data Manager, provisioning.output.directory=<FTP Home Location> Example: provisioning.output.directory=C:/Inetpub/ftpsite/<DOMAIN> (use the Output Directory field under Configuration/Clip,Zip,Ship in the Data Manager)

- Configure provisioning.delivery.protocol=ftp (use the Delivery Protocol field under Configuration/Clip,Zip,Ship in the Data Manager)

- Configure provisioning.output.repository.host= ftp server name (use the Delivery host field under Configuration/Clip,Zip,Ship in the Data Manager)

Request Data

- Log in to the Web Client with the user added to apollo-users.properties (For 2011 added in Data Manager). Requests will create the *.zip into your existing directory under the FTP Root.

- When you use the link to the zip file in the email notification you will be asked to log in. Log in to the Web Client using your domain username and password. The user must be added in the Data Manager.

- JBOSS impersonation needs FULL CONTROL to the FTP directory

Customize Clip, Zip, and Ship Emails

The ERDAS APOLLO system will email the user when the Clip, Zip, and Ship download request is received, and again when the files are ready for download. The ERDAS APOLLO installer collects information about the SMTP mail server you want to use for this task when you install ERDAS APOLLO Server. That information is stored in the global-server.properties file and you can modify it if necessary.

Change Subject and Body

The ERDAS APOLLO Server sends two different emails for the Clip, Zip, and Ship feature: one when it receives the download request, and a second when the download request has been processed and the zip file is ready to be downloaded from the server.

You can change the subject and the body of both of those emails.

To change the subject and body of the Clip, Zip and Ship emails:

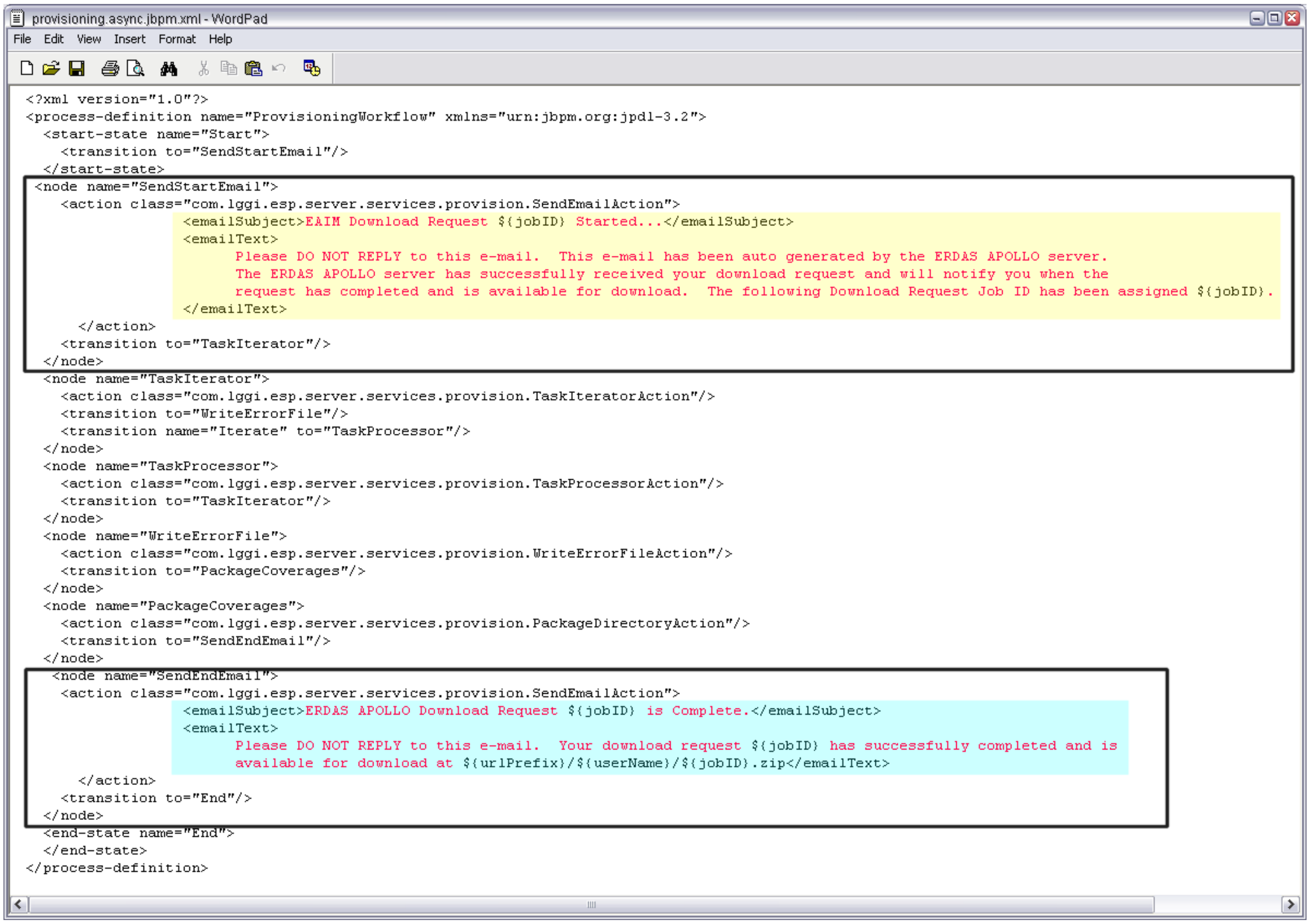
1. If you are using **JBoss**, navigate to the directory

   `<APOLLO_HOME>\jboss\server\default\deploy\`
   `erdas-apollo.ear\erdas-apollo.war\WEB-INF\classes`

   If you are using **WebLogic**, navigate to the directory

   `<APOLLO_HOME>\dist\weblogic\`
   `erdas-apollo.ear\erdas-apollo.war\WEB-INF\classes`

2. Open the file `provisioning.async.jbpm.xml` for editing.


provisioning.async.jbpm.xml - WordPad
File Edit View Insert Format Help

```
<?xml version="1.0"?>
<process-definition name="ProvisioningWorkflow" xmlns="urn:jbpm.org:jpdl-3.2">
  <start-state name="Start">
    <transition to="SendStartEmail"/>
  </start-state>
 <node name="SendStartEmail">
    <action class="com.lggi.esp.server.services.provision.SendEmailAction">
              <emailSubject>EAIM Download Request ${jobID} Started...</emailSubject>
              <emailText>
                    Please DO NOT REPLY to this e-mail.  This e-mail has been auto generated by the ERDAS APOLLO server.
                    The ERDAS APOLLO server has successfully received your download request and will notify you when the
                    request has completed and is available for download.  The following Download Request Job ID has been assigned ${jobID}.
              </emailText>
      </action>
    <transition to="TaskIterator"/>
  </node>
  <node name="TaskIterator">
    <action class="com.lggi.esp.server.services.provision.TaskIteratorAction"/>
    <transition to="WriteErrorFile"/>
    <transition name="Iterate" to="TaskProcessor"/>
  </node>
  <node name="TaskProcessor">
    <action class="com.lggi.esp.server.services.provision.TaskProcessorAction"/>
    <transition to="TaskIterator"/>
  </node>
  <node name="WriteErrorFile">
    <action class="com.lggi.esp.server.services.provision.WriteErrorFileAction"/>
    <transition to="PackageCoverages"/>
  </node>
  <node name="PackageCoverages">
    <action class="com.lggi.esp.server.services.provision.PackageDirectoryAction"/>
    <transition to="SendEndEmail"/>
  </node>
   <node name="SendEndEmail">
    <action class="com.lggi.esp.server.services.provision.SendEmailAction">
              <emailSubject>ERDAS APOLLO Download Request ${jobID} is Complete.</emailSubject>
              <emailText>
                    Please DO NOT REPLY to this e-mail.  Your download request ${jobID} has successfully completed and is
                    available for download at ${urlPrefix}/${userName}/${jobID}.zip</emailText>
      </action>
    <transition to="End"/>
  </node>
  <end-state name="End">
  </end-state>
</process-definition>
```

The **<emailSubject>** and **<emailText>** tags inside the **SendStartEmail** node control the subject and message text of the email that ERDAS APOLLO Server sends out when it receives the request. You can edit the content between these tags (shown in red in the figure) to change the subject and message text.

The variable **$ {jobID}** refers to a job number that the ERDAS APOLLO Server assigns to the download request. You don't have to use it in your subject or message, but it may be helpful if one of the end users needs to discuss the download request with you at a later time.

The **<emailSubject>** and **<emailText>** tags inside the **SendEndEmail** node control the subject and message text of the email that ERDAS APOLLO Server sends out when it is finished processing the request and the zip file is ready to be downloaded. You can edit the content between these tags (shown in red in the figure) to change the subject and message text.
The string **${urlPrefix}/${userName}/${jobID}.zip** composes the URL that the user needs to visit to access the zip file. This is essential information for the user, so you should leave it inside the message text.

Emails and a Firewall

Firewalls or virus protection software can block the Simple Mail Transfer Protocol (SMTP) on port 25. If the users who request downloads are not receiving Clip, Zip, and Ship emails, check the firewall and antivirus software on the computer on which the ERDAS APOLLO Server is running and the firewall and antivirus software on the mail host to make sure port 25 is not being blocked.

Change the Location of the Temporary Directory

The Clip, Zip, and Ship feature uses a temporary directory during the process of creating the zip file that will finally be placed in the output directory for download.

It is important that this temporary directory is located on a computer with enough space, especially if your system will be receiving a large number of download requests or it tends to receive very large download requests.

By default, ERDAS APOLLO will use the directory **<APOLLO_HOME>\storage\czs\temp** as the temporary directory for the intermediate files for Clip, Zip, and Ship**.**

To change the default location of the Clip, Zip and Ship temporary directory:

1. Open the `local-server.properties` file for editing.

   If you are using the JBoss application server, the file is located in

   ```
   <APOLLO_HOME>\jboss\server\default\deploy\
   erdas-apollo.ear\erdas-apollo.war\WEB-INF\classes
   ```

   If you are using WebLogic, the file is located in:

   ```
   <APOLLO_HOME>\dist\weblogic\
   erdas-apollo.ear\erdas-apollo.war\WEB-INF\classes
   ```

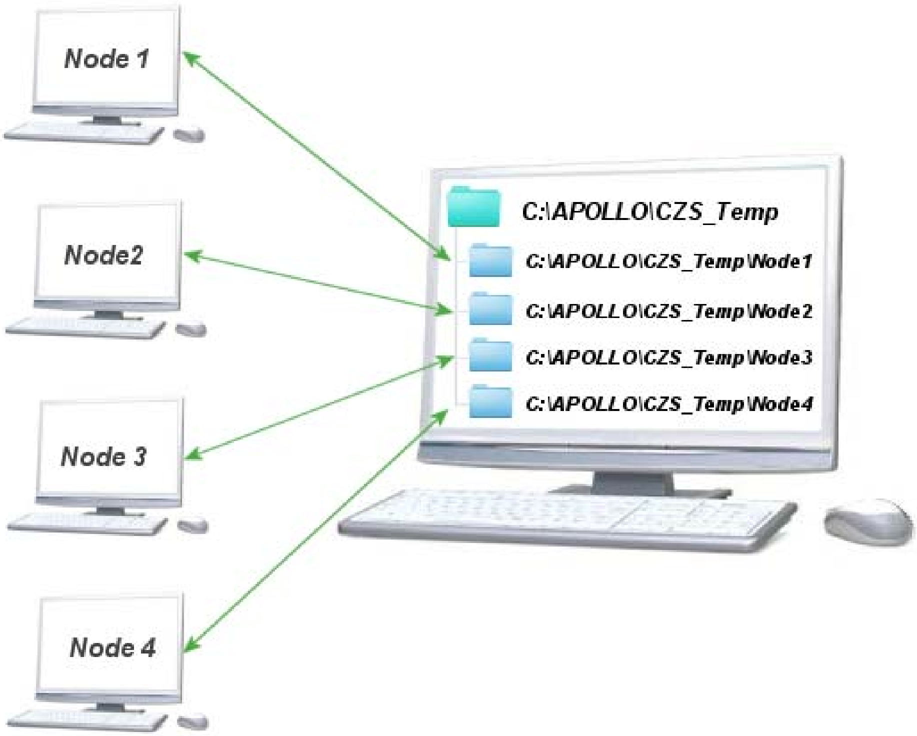
2. Find the property **czs.temp.directory**.

3. Find the value of the property. Change the value to the path of the location you would prefer to use as the temporary directory.

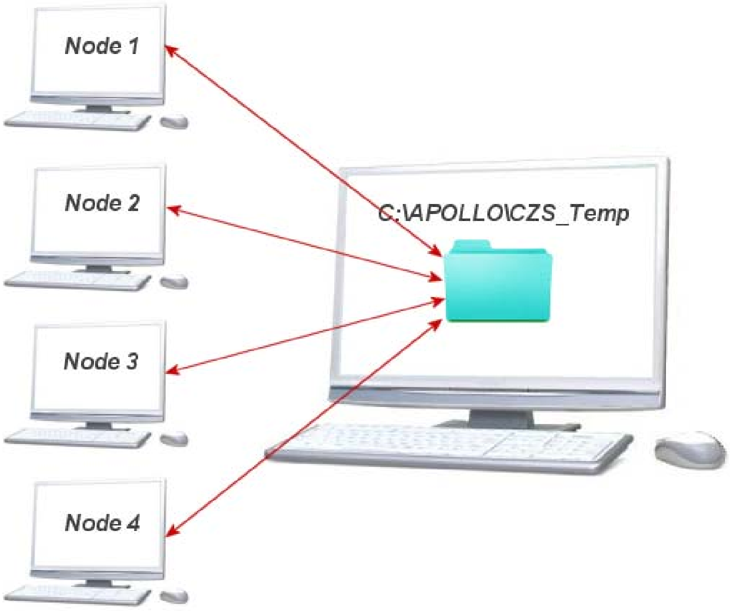
If you are using an ERDAS APOLLO Server cluster and would like to change the temp file for Clip Zip and Ship, you will need to change it in the local-server.properties file for the ERDAS APOLLO Server installation on each of the nodes in the cluster.

You can configure each of the nodes to use different directories, or you can configure them to each use a different subdirectory inside a common shared directory. Do not configure all the nodes to use the same exact temp directory.

THIS WILL WORK.



THIS WILL NOT WORK.



Custom Objects in the ERDAS APOLLO Catalog

The most common use of the ERDAS APOLLO catalog is to store imagery and service providers. However, you can also store other types of objects in it, such as spreadsheets, PDFs, movies, and text files.

In order for the ERDAS APOLLO Catalog to work with an additional type of object, each object needs to be accompanied by a Generic Item Metadata (GIM) file, which is a simple xml file that indicates what the object is (by its file name or URL), metadata about the object, and the Mime type of the object.

You can add custom objects using the ERDAS APOLLO Data Manager, the ERDAS APOLLO Catalog Web Interface, or the REST interface of the ERDAS APOLLO Catalog. Once you add a custom object, you will be able to locate it and view it in the Search tab of the ERDAS APOLLO Web Client or on the Browse page of the Catalog Web Interface tool.

Adding Custom Objects to the Catalog

Adding Custom Objects to the ERDAS APOLLO Catalog through Data Manager

You can use the Data Manager to add custom objects to your ERDAS APOLLO Catalog if the data is located somewhere on your network. You can add them one at a time, or you can add many at once using a Custom Object crawler.

Adding the custom objects to your catalog one at a time works well if you know where the file for the object is, and you know that object file does not have an associated GIM file. The Data Manager provides a dialog box that allows you to specify the object and some basic information about it. The system can generate a GIM file for you based on the information that you specify in the dialog box.

Adding custom objects with the crawler works well if you have multiple custom objects to add to the catalog and they each have a properly formatted GIM file. This can be the case if you create a program or automated system to generate these GIM files for certain objects so they can be easily incorporated into your ERDAS APOLLO Catalog.

See the *ERDAS APOLLO 2010 Data Manager Guide* for detailed instructions on adding custom objects using the Data Manager.

Adding Custom Objects to the ERDAS APOLLO Catalog through the Catalog Web Interface Tool

The Catalog Web Interface tool allows you to add objects using the object file, object URL, or the custom object harvester file.

Viewing Custom Objects

Viewing Custom Objects in the ERDAS APOLLO Web Client

Web Client users will be able to search for custom objects in the Search tab of the ERDAS APOLLO Web Client. Any custom objects that are returned by the search are shown in the Results panel with a Download Attachment button. When the user clicks this button, the Web Client uses the mime type of the custom object to launch the correct application so the user can view the object.

See the section "The Tabs Panel" in the "Web Client" chapter of the *ERDAS APOLLO 2010 User Guide* for more information.

Viewing Custom Objects in the ERDAS APOLLO Catalog Web Interface

Catalog Web Interface users will be able to search for custom objects on the Browse page of that tool by selecting "All Objects" in the dropdown box and entering the relevant search terms in the box beside it. The Catalog Web Interface displays all of the objects retuned by the search. When the user clicks the name of a custom object, the tool uses the mime type of the custom object to launch the correct application so the user can view the object.

See the "Catalog Web Interface" chapter of the *ERDAS APOLLO 2010 User Guide* for more information.

Structure of the Custom Objects XML Files

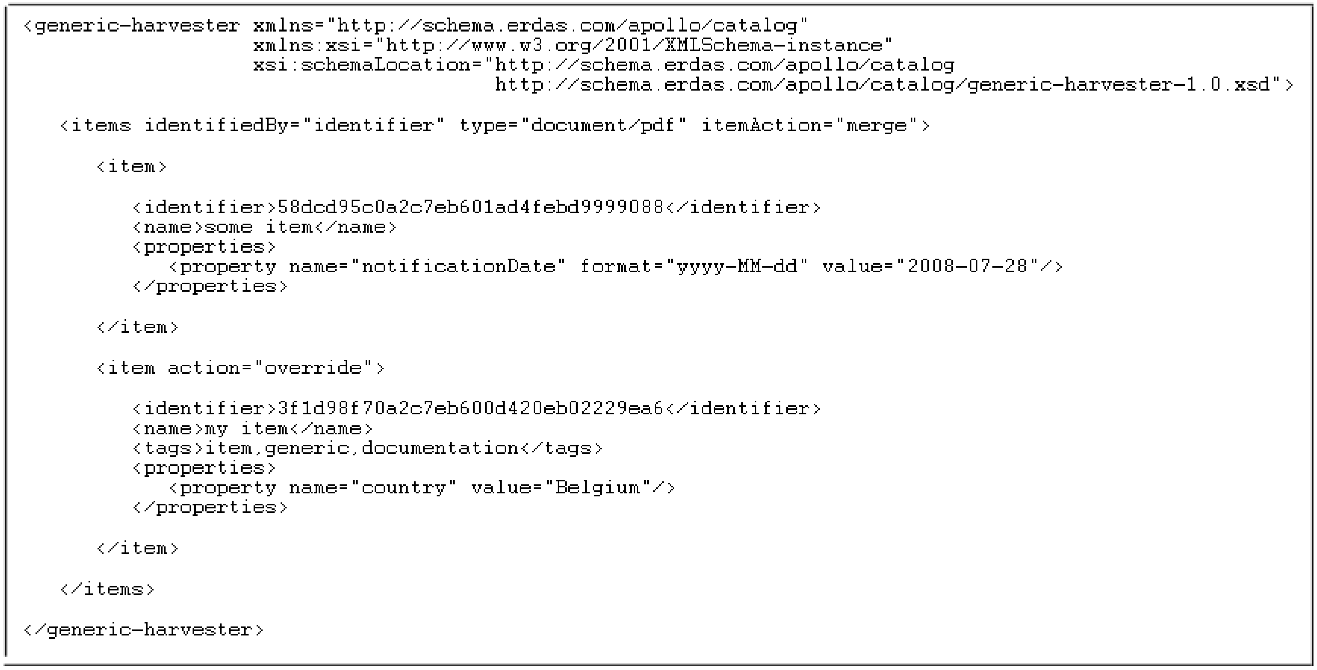
Custom Object Harvester File

The ERDAS APOLLO system is able to read and process a single file that indicates the changes to be made to custom objects in the Catalog. This ability allows you to speed up, or even automate the process of managing the custom objects in your ERDAS APOLLO Catalog.

This custom object harvester file is just a standardized XML file. For your convenience, a schema definition has been defined for this file. See **http://schema.erdas.com/apollo/catalog/generic-harvester-1.0.xsd**.

This file contains blocks of items that can share common settings such as the action to perform on the item (create/update/ignore/delete).

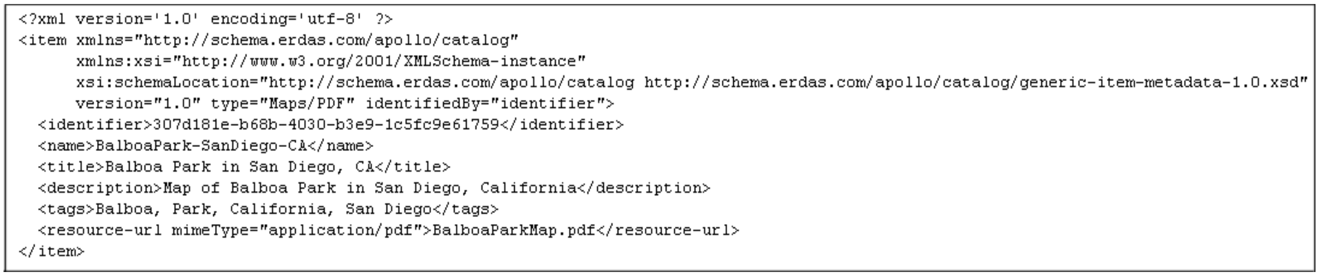
The following figure is a basic example of a custom object harvester file:



```
<generic-harvester xmlns="http://schema.erdas.com/apollo/catalog"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xsi:schemaLocation="http://schema.erdas.com/apollo/catalog
                                       http://schema.erdas.com/apollo/catalog/generic-harvester-1.0.xsd">

   <items identifiedBy="identifier" type="document/pdf" itemAction="merge">

      <item>

         <identifier>58dcd95c0a2c7eb601ad4febd9999088</identifier>
         <name>some item</name>
         <properties>
            <property name="notificationDate" format="yyyy-MM-dd" value="2008-07-28"/>
         </properties>

      </item>

      <item action="override">

         <identifier>3f1d98f70a2c7eb600d420eb02229ea6</identifier>
         <name>my item</name>
         <tags>item,generic,documentation</tags>
         <properties>
            <property name="country" value="Belgium"/>
         </properties>

      </item>

   </items>

</generic-harvester>
```

Custom Object GIM Document

The following is an example of the GIM file generated by the ERDAS APOLLO Data Manager when you add a new custom object to the ERDAS APOLLO Catalog:



```
<?xml version='1.0' encoding='utf-8' ?>
<item xmlns="http://schema.erdas.com/apollo/catalog"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://schema.erdas.com/apollo/catalog http://schema.erdas.com/apollo/catalog/generic-item-metadata-1.0.xsd"
      version="1.0" type="Maps/PDF" identifiedBy="identifier">
  <identifier>307d181e-b68b-4030-b3e9-1c5fc9e61759</identifier>
  <name>BalboaPark-SanDiego-CA</name>
  <title>Balboa Park in San Diego, CA</title>
  <description>Map of Balboa Park in San Diego, California</description>
  <tags>Balboa, Park, California, San Diego</tags>
  <resource-url mimeType="application/pdf">BalboaParkMap.pdf</resource-url>
</item>
```

The components of the GIM file are:

- **Object Type** (mime type) - the type of the object is a Concept that is identifier by a name. An object type can have a parent object type so it is possible to build a tree of object types. The object type is mandatory.

- **Identifier** - a user-identifier that must be unique within the whole catalog (and preferably across catalog, i.e. as a global unique identifier). If none is provided, one is generated automatically

- **Name** - the name of the item

- **Title** - the title of the item, that is a short human readable description of the item (255 chars max)

- **Description** - the full description of the item (also known as an abstract, 4000 chars max)

- **Tags** - a list of tags which are single words that describe/categorize the item without a formal classification (this can be seen as the tagging system in Web 2.0)

- **Footprint** - the footprint of the item expressed in WKT and in the standard 4326 EPSG

- **Resource URL** - a reference to a document or any other resource to link to the document (image, document, video, music, etc.)

- **Icon URL** - a reference to a thumbnail

- **Properties** - a set of arbitrary properties that you can create for the object type so that you can store additional metadata for objects of that type.

  The following data types are supported for the properties:

  - String
  - Integer
  - Long
  - Double
  - Float
  - Date (with the ability to specify a custom date format)

Adjusting Your Database Connection

Adjusting the Data Pool Size

ERDAS APOLLO is configured to connect to a database and work for a small number of users as soon as you install it. For optimal performance on a live customer installation, you will probably want to adjust the data pool size so that ERDAS APOLLO can accommodate a larger number of users.

To adjust the data pool size for ERDAS APOLLO:

1. Navigate to the directory `<APOLLO_HOME>\jboss\server\default\deploy\erdas-apollo.ear/erdas-apollo.war/WEB-INF`

2. Open file applicationContext.xml.

3. Find the following.

```
<bean id="dataSource"
   class="org.apache.commons.dbcp.BasicDataSource"
   destroy-method="close">
   <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
   <property name="url" value="jdbc:oracle:thin:@brunswickdevdb:1521:
      APODEV11"/>
   <property name="username" value="SNAG_EAIM_ML"/>
   <property name="password" value="APOLLO"/>
      <property name="initialSize" value="5"/>
      <property name="maxActive" value="10"/>
</bean>
```

4. Look for properties initialSize and maxActive and change them accordingly.

5. Edit the values of those properties as needed.
   The best values for those properties will vary from installation to installation, and are best determined by a knowledgeable system administrator.

6. When you are finished making your changes, save the file and restart the application server.

Using Windows Authentication for an APOLLO DB on the Microsoft SQL Server Platform

When you install an ERDAS APOLLO Server and connect it to a Microsoft SQL Server database, APOLLO is configured by default to expect that SQL Server is using *SQL Server Authentication* to provide access to the APOLLO database. This means that all of the security mechanisms that provide access to the APOLLO database will be managed inside of SQL Server by SQL Server itself.

If you prefer, you can use *Windows Authentication*, in which Windows user accounts on the database server computer will be used to gain access to the APOLLO database. To do this, you will need to adjust the database connection URL that is defined inside of the ERDAS APOLLO Server installation.

JBoss

To adjust the ERDAS APOLLO Server to work with a SQL Server database using Windows Authentication:

1. Navigate to the directory:
`<APOLLO_HOME>\jboss\server\default\deploy\apollo-server.ear/apollo-server.war/WEb-INF`

2. Open file applicationContext.xml.

3. Find the property named "url"

```
<bean id="dataSource"
    class="org.apache.commons.dbcp.BasicDataSource"
    destroy-method="close">
    <property name="driverClassName" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <property name="url" value="jdbc:sqlserver://yourserver:1433;DatabaseName=yourdbname"/>
    <property name="username" value="TESTER"/>
    <property name="password" value="APOLLO"/>
        <property name="initialSize" value="5"/>
        <property name="maxActive" value="10"/>
</bean>
```

4. Navigate to the directory
`<APOLLO_HOME>\jboss\server\default\deploy`.

5. Open the file `apollo-ds.xml` for editing.

6. Find the property called **xa-datasource-property**.

7. Change the value of the property (shown in blue in the figure above) by appending “**;integratedSecurity=true**” to the end of it.

8. Navigate to the directory:
`<APOLLO_HOME>\jboss\server\default\deploy`.

9. Open the file apollo-ds.xml for editing.

10. Next, find the property called **connection-url**.

11. Change the value of that property by appending “**;integratedSecurity=true**” to the end of it, also.

12. Save the file.

13. Restart the JBoss application server.

Raster Data Server (RDS) Configuration and Tuning Parameters

The Raster Data Server (RDS) is an external process that is generated by the ERDAS APOLLO ProcessManager service. The ProcessManager service is responsible for spawning a new instance of RDS or utilizing one from a pool of existing RDS process in order to fulfill imagery request. The datasets cached, size of the pool, memory allocated, and other options can be configured via the following files: `rds.properties`, `local-processmanager.properties`, and `server-processmanager.properties`.

The `rds.properties` file is located in the file `nci-rds.jar` that can be found in the directory <*APOLLO_HOME*>\tools\native\nci. To gain access to the file you can use a compression tool like WinZip, 7zip, WinRar, etc. Extract the `rds.properties` file, modify it and then replace the original contents of the jar file with the modified file.

⚠

It's important to understand that the application server must be stopped in order to modify the nci-rds.jar file. If the server is running you will encounter problems trying to save the modified rds.properties back into the jar as it will be in use by the RDS process. Failure to stop the server prior to modifying the jar may result in corruption of the jar and failure of the application.

Table 6: rds.properties Configuration Elements

Elements	**Description and examples**
cache-these-many-datasets	Limit the number of datasets in cache to this value.
cache-per-band(MB)	Limit the number of blocks in cache, in Megabytes, so that it doesn't exceed this value.
raster-heap-size(MB)	Limit the maximum heap the native raster system can use. This helps prevent out of memory issues especially when generating pyramids for stripped TIFFs (or similar file formats) during crawling or serving WMS requests. Set to -1 if you don't want to it to use maximum available memory.
warn-virtual-memory(MB)	This property is used by the native code when it reports the memory consumption of the RDS. Start freeing up resources (shrink the cached datasets) when the handle count reaches this value.
max-virtual-memory(MB)	This property is used by the native code when it reports the memory consumption of the RDS. Abort/quit RDS when the memory reaches this value.
warn-handle-count	This property is used by the native code when it reports the memory consumption of the RDS. Start freeing up resources (shrink the cached datasets) when the handle count reaches this value
max-handle-count	This property is used by the native code when it reports the memory consumption of the RDS. Abort/quit RDS when the handle count exceeds this value.

intf-impl	Add the interface-implementation fully classified names separated by hyphen ("-") as shown below: intf-impl=com.lggi.esp.coverage.decoder.raster.gio.GIORasterCoverageRemote-com.lggi.esp.coverage.decoder.raster.gio.GIORasterCoverageRemoteImpl

global-processmanager.properties:

This file is mainly for properties that are required to start RDS (Raster Data Server). The properties that exist in the file can be shared among multiple nodes in a cluster.This file exists in the <APOLLO_HOME>.

Table 7: global-processmanager.properties Configuration elements

Elements	Description and examples
rds.max.read.threads	Only for ImageX decoders for ECW/JP2 - maximum number of processing threads.Default is 2.
rds.debugrds	To enable debugging of the RDS process. Set to true to enable debugging. Default is false.
rds.haltonstart	Flag indicating if the RDS process should halt once it has started. Used for debugging. The default is false.
rds.max.pixel.request.size	Defines the maximum pixel request that GIORasterCoverageProxy can process. The default value is 25000000.
rds.proxy.directory	Defines the directory where proxy files are generated after pyramid generation of datasets.
processmanager.min.process.count	The minimum number of RDS processes to keep in the pool for servicing requests. The default is 1.
processmanager.max.process.count	The maximum number of RDS processes to keep in the pool for servicing requests. The default is 5.
processmanager.keepalivetime.inmins	The time in minutes that an RDS process should remain in the pool before being cleaned up if there is no activity. The default is 10.
processmanager.getprocess.timeout.inseconds	The time in seconds between the processmanager.getprocess.numentries properties to get a free process from the pool. The default is 30.
processmanager.getprocess.delay.inseconds	Deprecated

Table 7: **global-processmanager.properties Configuration elements** (Continued)

Elements	Description and examples
processmanager.getprocess.numretries	When a request is received by the ProcessManager, the number of times it will try to get a free process from the pool of RDS processes before failing and generating an exception. The default is 2.
wait.time.before.killing.rds.if.not.responding	Time in seconds that the ProcessMonitor waits for an ACK before it terminates an RDS process. The default is 10.
rds.gio.ecw.logging.level	The logging level for ECW running in GIO. Valid values are 0-3. Zero indicates to log only errors and exception; three indicates verbose logging. The default is 0.

local-processmanager.properties:

This file (like global-processmanager.properties) also contains properties required by RDS, but these properties are node specific. In a cluster, different nodes have their own set of properties. This file exists in <INSTALL_DIR>/jboss/server/{default} or {cluster}/deploy/erdas-apollo.ear/erdas-apollo.war/WEB-INF/classes.

Table 8: **local-processmanager.properties Configuration elements**

Elements	Description and examples
rds.classpath	Defines the classpath for the use by RDS when a new process is created. It references the .jar files located in the <APOLLO_HOME>/rds directory by default.
rds.security.policy	Defines the default java security policy for access to RDS resources.
rds.log4j.properties	Defines logger configuration used by RDS for logging errors. It is declarative by nature and can be customized to use any of the valid loggers provided by Log4J or a custom logger based on Log4J api.
rds.java.home	Specifies which jdk to use for rds gio processes.
rds.java64.home	Specifies which 64-bit jdk to use for rds, such as anything that can take advantage of 64-bit like ImageX decoder.
rds.jvm.options	Defines all JVM options used to fine tune the RDS process. Each option is separated by a space. Example: -Xms64m Xmx128m -XX:+AggressiveOpts

Table 8: **local-processmanager.properties Configuration elements** (Continued)

Elements	Description and examples
rds.gio.ecw.log.file	The log file for ECW running in GIO which logs any exceptions.

Input File Formats

ERDAS APOLLO 2010 uses the ERDAS GIO decoders (based on the ERDAS IMAGINE Raster SDK) and GDAL 1.7.2 to provide support for multiple imagery input formats. This section lists the formats that are fully tested and those that have not been tested but still might work out-of-the box.

Table 9: Tested imagery input formats

ALOS AVNIR-2 JAXA CEOS	ALOS PRISM JAXA CEOS
ALOS PRISM JAXA CEOS IMG	ALOS Palsar ERSDAC CEOS
ALOS Palsar ERSDAC VEXCEL	ALOS Palsar JAXA CEOS
BIL	BIL (Generic Binary)
BIP	BIP (Generic Binary)
BSQ	BSQa (Generic Binary)
Bitmap (BMP)	CIB (Controlled Image Base)
"DTED	DEM
Envisat	ERMapper formats (including ECW)
ERS	GeoTIFF
GIF	GRIB
HDR	HFA (IMG)
Ikonos	JFIF (JPEG)
JPEG 2000	Landsat
MrSID	NITF
PCX	PNG
PPN	QuickBird
RADARSAT-2	Spot
TGA	TIFF
WBMP	

Limitations

Some Database imagery formats are supported in ERDAS APOLLO Essentials - SDI with the following limitations.

- "ArcSDE Raster: requires the ArcSDE libraries
- "Oracle Georaster: support limited to 8-bit per band

Tested vector input formats

"OGC GML

"ESRI Shapefile

"MIF/MID

"Oracle Spatial

"PostGIS

"ArcSDE (requires the ArcSDE libraries)

Untested imagery input formats

The following tables list the untested imagery input formats.

Table 10: Formats available through the ERDAS GIO decoders

Arc Coverage	**ArcInfo & Space Imaging**
COSMO-SkyMed	FIT
GeoEye-1	GIS (Erdas 7.x)
LAN (Erdas 7.x)	Raster Product Format
TerraSAR-X	WorldView-1

Table 11: Formats available through GDAL 1.7.2

VRT: Virtual Raster	**ELAS: ELAS**
AAIGrid: Arc/Info ASCII Grid	HDF4
HDF5	HDF-EOS
MEM: In Memory Raster	XPM: X11 PixMap Format

Table 11: Formats available through GDAL 1.7.2

PCIDSK: PCIDSK Database File	PCRaster: PCRaster Raster File
ILWIS: ILWIS Raster Map	SGI: SGI Image File Format 1.0
SRTMHGT: SRTMHGT File Format	Leveller: Leveller heightfield
Terragen: Terragen heightfield	GMT: GMT NetCDF Grid Format
netCDF: Network Common Data Format	FIT: FIT Image
RMF: Raster Matrix Format	RST: Idrisi Raster A.1
NGR: Intergraph Raster	GSAG: Golden Software ASCII Grid (.grd)
GSBG: Golden Software Binary Grid (.grd)	PNM: Portable Pixmap Format (netpbm)
ENVI: ENVI .hdr Labelled	EHdr: ESRI .hdr Labelled
PAux: PCI .aux Labelled	MFF: Vexcel MFF Raster
MFF2: Vexcel MFF2 (HKV) Raster	BT: VTP .bt (Binary Terrain) 1.3 Format
"IDA: Image Data and Analysis	ADRG: ARC Digitized Raster Graphics
BLX: Magellan topo (.blx)	

Advanced Configuration of ERDAS APOLLO

This chapter contains information about performing some of the more advanced configurations of the ERDAS APOLLO system.

Before you perform any of the customizations, please back up the file that you will be changing.

General Server Configuration

Install Properties

The `install.properties` file contains a lot of the most basic options that control how your ERDAS APOLLO Server product will work. These options were initially set by the installer program when you installed ERDAS APOLLO.

This file is located in the `<APOLLO_HOME>` directory.

The following table shows the properties that you can change in the file and describes the behavior that the property controls. Once you make changes, you must rebuild the EAR/WAR files following these instructions.

1. Set environment variable ANT_HOME = <APOLLO_HOME>/tools/ant
2. Add %ANT_HOME%/bin to the PATH environment variable.
3. Open a command prompt and in the <APOLLO_HOME> directory, and run the applicable command.
    - ant tomcat6
    - ant jboss
    - ant tomcat5
    - ant weblogic
    - ant generic
4. Delete the erdas-apollo.ear and apollo-client.war files from the application server webapps/deploy directory. Be sure to back up the files if you have customizations.

5. Copy the contents of the appropriate <Iinstall_directory>/dist directory into the application server deploy directory (or deploy using the appropriate method for your application server).

Table 12: Customizable Parameters in the Install.Properties File

Property Name	Description
platform.home	The ERDAS APOLLO installation directory.
apollo.server.host	server host name
apollo.server.port	server port number
apollo.shutdown.port	server shutdown port number
apollo.server.admin. port	server administration port number
apollo.providers.vector.home	path to directory containing providers.fac for vector (WFS) offerings
apollo.providers.map.home	path to directory containing providers.fac for map (WMS) offerings
apollo.providers.coverage.home	path to directory containing providers.fac for coverage (WCS) offerings
apollo.providers.process.home	path to directory containing providers.fac for process (WPS) offerings
apollo.providers.admin.home	path to directory containing providers.fac for administration
apollo.im.home	home directory for image management components (usually installation directory
hibernate.connection.driver_class	type of database connection for Hibernate, e.g. org.postgresql.Driver
hibernate.connection.url	connect string to database for Hibernate
hibernate.connection.username	database connection username for Hibernate
hibernate.connection.password	database connection password for Hibernate
babel.home	home directory for Babel catalog components (usually installation directory)
apollo.server.home	
release.babel.hibernate.dialect	dialect for database connections, e.g. com.erdas.rsp.hibernate.postgis.PostgisDialect
release.babel.jdbc.driver.fqn	name of JDBC driver for Babel database connection
release.babel.db.user	user id for Babel database connection

Table 12: Customizable Parameters in the Install.Properties File

Property Name	Description
release.babel.db.password	password for Babel database connection
release.babel.db.url	connect string to database for Babel
babel.hibernate.dialect	dialect for Babel Hibernate database connection, e.g. com.erdas.rsp.hibernate.postgis.PostgisDialect
babel.db.user	user ID for Babel database connection
babel.db.password	password for Babel database connection
babel.db.url	connect string to database for Babel
babel.jdbc.driver.fqn	
babel.db.host	host name for Babel database connection
babel.db.port	port number for Babel database connection
babel.db.sid	database SID for Babel database connection
babel.log.home	
ionic.catalog.product.name	
apollo.java.home	JAVA Home for ERDAS APOLLO
eaim.server.home	home directory for image management components (usually installation directory)
eaim.server.host	host name for image management service
eaim.server.port	host port for image management service
eaim.server.smtp.host	host for SMTP connection
eaim.server.smtp.port	port for SMTP connection
eaim.server.smtp.user	user ID for SMTP connection
eaim.server.wcs.url	URL for WCS service
eaim.server.wrs.url	URL for WRS service
eaim.server.wfs.url	URL for WFS service
eaim.server.wps.url	URL for WPS service
eaim.server.wps.transientprovider.url	URL for transient WMS providers used by WPS
eaim.server.catalog.url	URL for ERDAS APOLLO Catalog service
eaim.server.clipzipship.url	URL for Clip/Zip/Ship service
eaim.server.quartzinterface.url	URL for Quartz interface
eaim.server.crawler.user	user ID for crawling

Table 12: Customizable Parameters in the Install.Properties File

Property Name	Description
eaim.server.crawler.pass	password for crawling
eaim.server.quartz.jdbc.delegate.class	class name for Quartz JDBC connection
apollo.im.home	installation directory
gio.home	path to GIO
platform.gio.home	also path to GIO
platform.gio.arch	processor architecture for GIO
platform.gio.arch.mode	
eaim.server.streamedraster.access.url	URL for streamed (ECWP) raster access
eaim.server.streamedraster.access.enabled	if True, ECWP access is enabled
eaim.server.wps.gio.exedir	executable directory for WPS GIO
eaim.server.wps.gio.ismsdir	ISMS directory for WPS GIO
eaim.server.wps.gio.name	name of WPS GIO
eaim.server.wps.gio.exe	executable for WPS GIO
eaim.server.iws.home	path to IWS components, usually something like {$platform.home}/tools/native/iws
ic.context.path	path to default context file
ic.logging.type	web client logging type (e.g. FILE)
apollo.client.components	name of properties file containing components information for apollo client (file must be in classpath). By default either apollo-im-components.properties (Professional) or apollo-im-components.properties (Essentials SDI)
apollo.client.contexts	name of properties file containing context information for apollo client (file must be in classpath)

Custom Metadata Elements

Queryables are custom metadata elements that you want to attach to the data in your catalog. When you set up the ERDAS APOLLO system to recognize these elements, the Data Manager will harvest them when it registers data in the catalog.

You can also configure the Web Client to allow you to see this metadata and conduct searches based on it.

Custom Queryables

To set up the system to recognize your queryables, you will need to open the file `queryables.xml`. It is located in the directory `<APOLLO_HOME>/config/erdas-apollo.`

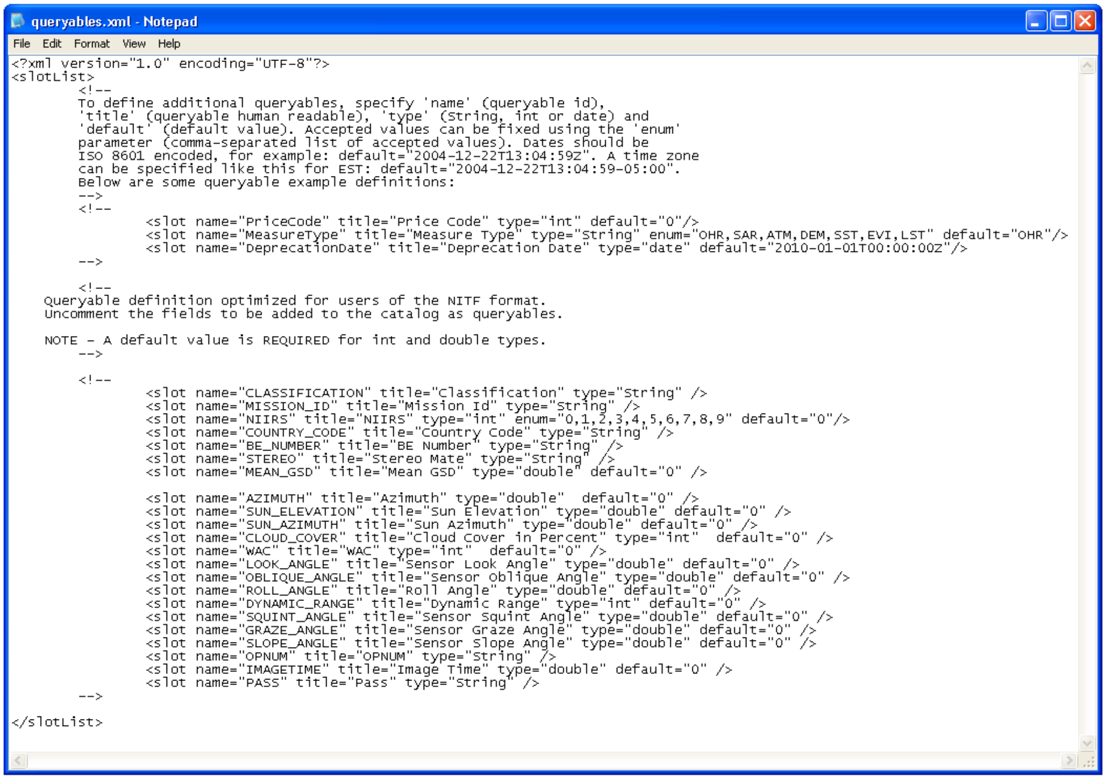
The file itself contains instructions about how to add the queryables so ERDAS APOLLO can recognize and use them. The file also includes a list of some commonly requested queryables. To use one of the queryables that is already defined in the file, you will need to uncomment it by relocating the comment delimiters **<!--** and **-->.**

The first figure below shows the `queryables.xml` file as it exists when you first install ERDAS APOLLO.

The second figure shows what the `queryables.xml` file would look like if you edited the file so you could use the queryables "CLASSIFICATION" and "MISSION_ID" that are predefined in the file.

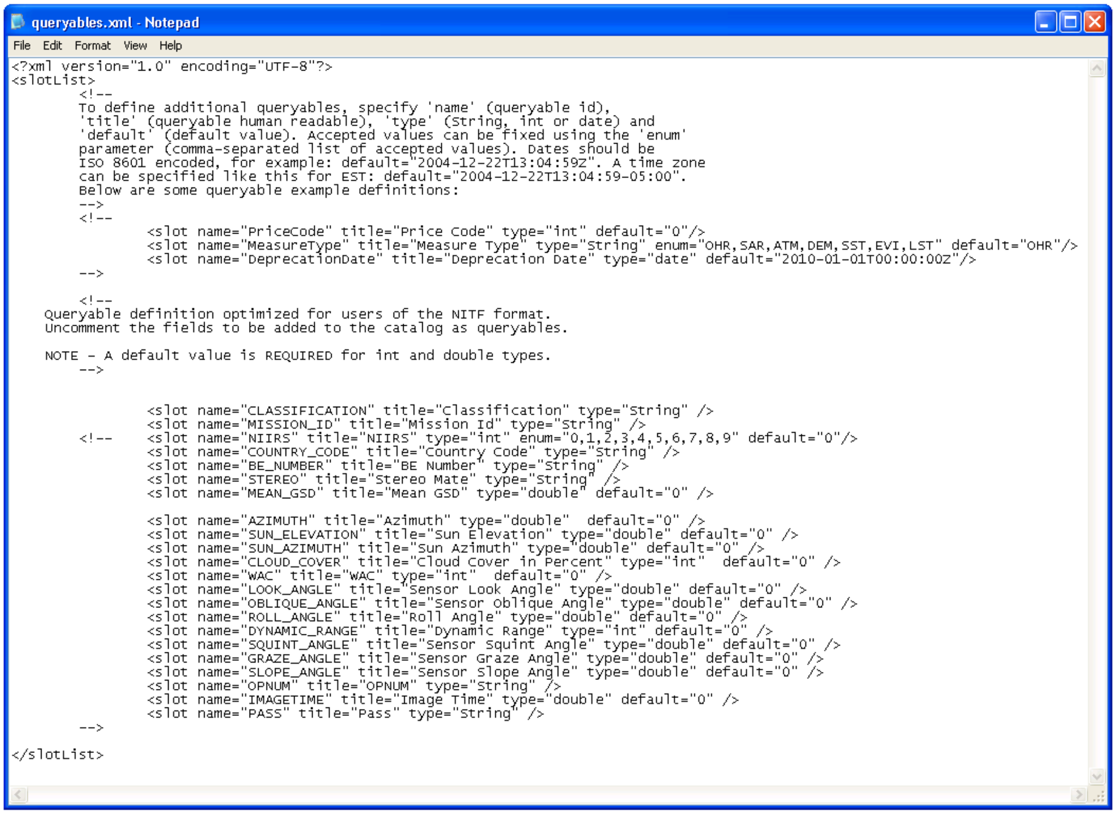
After you edit the `queryables.xml` file, do an Update Queryables, and then save the file and restart your application server for the changes to take effect.

Figure 1: An Example of the Default queryables.xml File



```
<?xml version="1.0" encoding="UTF-8"?>
<slotList>
        <!--
        To define additional queryables, specify 'name' (queryable id),
        'title' (queryable human readable), 'type' (String, int or date) and
        'default' (default value). Accepted values can be fixed using the 'enum'
        parameter (comma-separated list of accepted values). Dates should be
        ISO 8601 encoded, for example: default="2004-12-22T13:04:59Z". A time zone
        can be specified like this for EST: default="2004-12-22T13:04:59-05:00".
        Below are some queryable example definitions:
        -->
        <!--
                <slot name="PriceCode" title="Price Code" type="int" default="0"/>
                <slot name="MeasureType" title="Measure Type" type="String" enum="OHR,SAR,ATM,DEM,SST,EVI,LST" default="OHR"/>
                <slot name="DeprecationDate" title="Deprecation Date" type="date" default="2010-01-01T00:00:00Z"/>
        -->

        <!--
    Queryable definition optimized for users of the NITF format.
    Uncomment the fields to be added to the catalog as queryables.

    NOTE - A default value is REQUIRED for int and double types.
        -->

        <!--
                <slot name="CLASSIFICATION" title="Classification" type="String" />
                <slot name="MISSION_ID" title="Mission Id" type="String" />
                <slot name="NIIRS" title="NIIRS" type="int" enum="0,1,2,3,4,5,6,7,8,9" default="0"/>
                <slot name="COUNTRY_CODE" title="Country Code" type="String" />
                <slot name="BE_NUMBER" title="BE Number" type="String" />
                <slot name="STEREO" title="Stereo Mate" type="String" />
                <slot name="MEAN_GSD" title="Mean GSD" type="double" default="0" />

                <slot name="AZIMUTH" title="Azimuth" type="double"  default="0" />
                <slot name="SUN_ELEVATION" title="Sun Elevation" type="double" default="0" />
                <slot name="SUN_AZIMUTH" title="Sun Azimuth" type="double" default="0" />
                <slot name="CLOUD_COVER" title="Cloud Cover in Percent" type="int"  default="0" />
                <slot name="WAC" title="WAC" type="int"  default="0" />
                <slot name="LOOK_ANGLE" title="Sensor Look Angle" type="double" default="0" />
                <slot name="OBLIQUE_ANGLE" title="Sensor Oblique Angle" type="double" default="0" />
                <slot name="ROLL_ANGLE" title="Roll Angle" type="double" default="0" />
                <slot name="DYNAMIC_RANGE" title="Dynamic Range" type="int" default="0" />
                <slot name="SQUINT_ANGLE" title="Sensor Squint Angle" type="double" default="0" />
                <slot name="GRAZE_ANGLE" title="Sensor Graze Angle" type="double" default="0" />
                <slot name="SLOPE_ANGLE" title="Sensor Slope Angle" type="double" default="0" />
                <slot name="OPNUM" title="OPNUM" type="String" />
                <slot name="IMAGETIME" title="Image Time" type="double" default="0" />
                <slot name="PASS" title="Pass" type="String" />
        -->

</slotList>
```

Figure 2: An Example of queryables.xml, Edited to Include Some Predefined Queryables



```xml
<?xml version="1.0" encoding="UTF-8"?>
<slotList>
        <!--
        To define additional queryables, specify 'name' (queryable id),
        'title' (queryable human readable), 'type' (String, int or date) and
        'default' (default value). Accepted values can be fixed using the 'enum'
        parameter (comma-separated list of accepted values). Dates should be
        ISO 8601 encoded, for example: default="2004-12-22T13:04:59Z". A time zone
        can be specified like this for EST: default="2004-12-22T13:04:59-05:00".
        Below are some queryable example definitions:
        -->
        <!--
                <slot name="PriceCode" title="Price Code" type="int" default="0"/>
                <slot name="MeasureType" title="Measure Type" type="String" enum="OHR,SAR,ATM,DEM,SST,EVI,LST" default="OHR"/>
                <slot name="DeprecationDate" title="Deprecation Date" type="date" default="2010-01-01T00:00:00Z"/>
        -->

        <!--
    Queryable definition optimized for users of the NITF format.
    Uncomment the fields to be added to the catalog as queryables.

    NOTE - A default value is REQUIRED for int and double types.
        -->


                <slot name="CLASSIFICATION" title="Classification" type="String" />
                <slot name="MISSION_ID" title="Mission Id" type="String" />
        <!--    <slot name="NIIRS" title="NIIRS" type="int" enum="0,1,2,3,4,5,6,7,8,9" default="0"/>
                <slot name="COUNTRY_CODE" title="Country Code" type="String" />
                <slot name="BE_NUMBER" title="BE Number" type="String" />
                <slot name="STEREO" title="Stereo Mate" type="String" />
                <slot name="MEAN_GSD" title="Mean GSD" type="double" default="0" />

                <slot name="AZIMUTH" title="Azimuth" type="double"  default="0" />
                <slot name="SUN_ELEVATION" title="Sun Elevation" type="double" default="0" />
                <slot name="SUN_AZIMUTH" title="Sun Azimuth" type="double" default="0" />
                <slot name="CLOUD_COVER" title="Cloud Cover in Percent" type="int"  default="0" />
                <slot name="WAC" title="WAC" type="int"  default="0" />
                <slot name="LOOK_ANGLE" title="Sensor Look Angle" type="double" default="0" />
                <slot name="OBLIQUE_ANGLE" title="Sensor Oblique Angle" type="double" default="0" />
                <slot name="ROLL_ANGLE" title="Roll Angle" type="double" default="0" />
                <slot name="DYNAMIC_RANGE" title="Dynamic Range" type="int" default="0" />
                <slot name="SQUINT_ANGLE" title="Sensor Squint Angle" type="double" default="0" />
                <slot name="GRAZE_ANGLE" title="Sensor Graze Angle" type="double" default="0" />
                <slot name="SLOPE_ANGLE" title="Sensor Slope Angle" type="double" default="0" />
                <slot name="OPNUM" title="OPNUM" type="String" />
                <slot name="IMAGETIME" title="Image Time" type="double" default="0" />
                <slot name="PASS" title="Pass" type="String" />
        -->

</slotList>
```

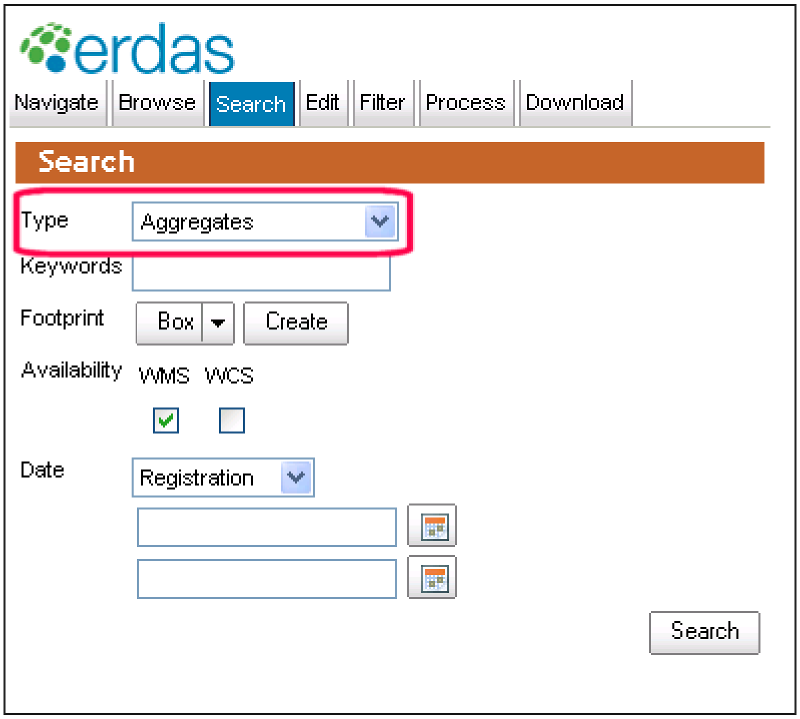
Custom Queryables in Searches

All of the Web Client searches are performed using the **Search** tab.

When you conduct a search on the Search tab, you first have to decide what you are searching for and select that in the **Type** box. A different set of search fields will appear depending on the type of search that you select.

In order to use the custom queryables in the searches, you first need to decide which type of searches will need to use these queryables. Then, you need to add these queryables fields to the fields that already appear for that type of search. Keep in mind that you can only add custom queryables to datasets and aggregates in your catalog. You cannot add custom queryables to any type of service providers, so you do not need to add queryable fields for those types of searches.

Figure 3: Search Tab in the Web Client, Out of the Box



Type: Aggregates

Keywords:

Footprint: Box

Availability: WMS ☑ WCS □

Date: Registration

All of the existing search fields are defined in the file `babel-types-infos.js`.

Before you begin working with this file, you should copy the `babel-types-infos.js` file and put the copy in a location somewhere outside of your `<APOLLO_HOME>` directory for safekeeping.

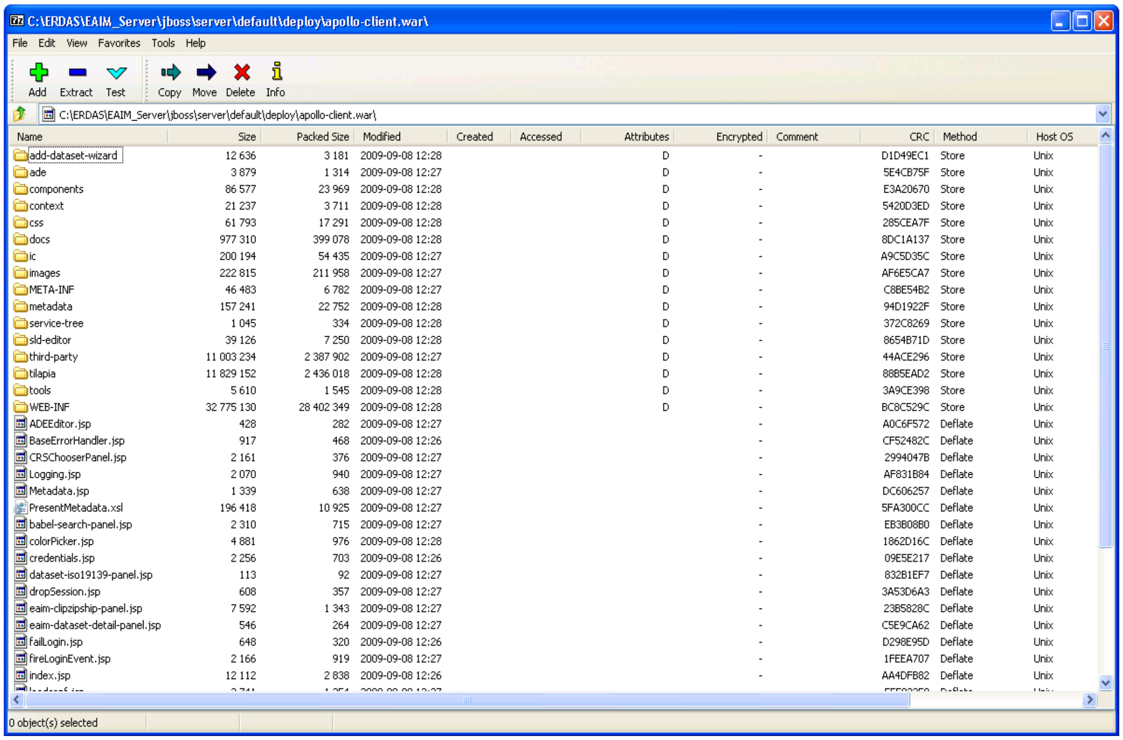
To open the babel-types-infos.js file:

1. If you are using JBoss as your application server, navigate to the directory `<APOLLO_HOME>/jboss/server/default/deploy/`

   If you are using WebLogic as your application server, navigate to the directory `<APOLLO_HOME>/dist/weblogic`

2. Find the archive file `apollo-client.war`.
   Right-click it and select **7-Zip > Open Archive** in the menu.

3. When 7-Zip opens the archive, you will see the following.



C:\ERDAS\EAIM_Server\jboss\server\default\deploy\apollo-client.war\
File
Edit
View
Favorites
Tools
Help
Add
Extract
Test
Copy
Move
Delete
Info

C:\ERDAS\EAIM_Server\jboss\server\default\deploy\apollo-client.war\

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted	Comment	CRC	Method	Host OS
add-dataset-wizard	12 636	3 181	2009-09-08 12:28			D	-		D1D49EC1	Store	Unix
ade	3 879	1 314	2009-09-08 12:27			D	-		5E4CB75F	Store	Unix
components	86 577	23 969	2009-09-08 12:28			D	-		E3A20670	Store	Unix
context	21 237	3 711	2009-09-08 12:28			D	-		5420D3ED	Store	Unix
css	61 793	17 291	2009-09-08 12:28			D	-		285CEA7F	Store	Unix
docs	977 310	399 078	2009-09-08 12:28			D	-		8DC1A137	Store	Unix
ic	200 194	54 435	2009-09-08 12:27			D	-		A9C5D35C	Store	Unix
images	222 815	211 958	2009-09-08 12:27			D	-		AF6E5CA7	Store	Unix
META-INF	46 483	6 782	2009-09-08 12:27			D	-		C8BE54B2	Store	Unix
metadata	157 241	22 752	2009-09-08 12:28			D	-		94D1922F	Store	Unix
service-tree	1 045	334	2009-09-08 12:28			D	-		372C8269	Store	Unix
sld-editor	39 126	7 250	2009-09-08 12:28			D	-		8654B71D	Store	Unix
third-party	11 003 234	2 387 902	2009-09-08 12:27			D	-		44ACE296	Store	Unix
tilapia	11 829 152	2 436 018	2009-09-08 12:28			D	-		88B5EAD2	Store	Unix
tools	5 610	1 545	2009-09-08 12:28			D	-		3A9CE398	Store	Unix
WEB-INF	32 775 130	28 402 349	2009-09-08 12:28			D	-		BC8C529C	Store	Unix
ADEEditor.jsp	428	282	2009-09-08 12:27				-		A0C6F572	Deflate	Unix
BaseErrorHandler.jsp	917	468	2009-09-08 12:26				-		CF52482C	Deflate	Unix
CRSChooserPanel.jsp	2 161	376	2009-09-08 12:27				-		2994047B	Deflate	Unix
Logging.jsp	2 070	940	2009-09-08 12:27				-		AF831B84	Deflate	Unix
Metadata.jsp	1 339	638	2009-09-08 12:27				-		DC606257	Deflate	Unix
PresentMetadata.xsl	196 418	10 925	2009-09-08 12:27				-		5FA300CC	Deflate	Unix
babel-search-panel.jsp	2 310	715	2009-09-08 12:27				-		EB3B08B0	Deflate	Unix
colorPicker.jsp	4 881	976	2009-09-08 12:28				-		1862D16C	Deflate	Unix
credentials.jsp	2 256	703	2009-09-08 12:26				-		09E5E217	Deflate	Unix
dataset-iso19139-panel.jsp	113	92	2009-09-08 12:27				-		832B1EF7	Deflate	Unix
dropSession.jsp	608	357	2009-09-08 12:27				-		3A53D6A3	Deflate	Unix
eaim-clipzipship-panel.jsp	7 592	1 343	2009-09-08 12:27				-		23B5828C	Deflate	Unix
eaim-dataset-detail-panel.jsp	546	264	2009-09-08 12:27				-		C5E9CA62	Deflate	Unix
failLogin.jsp	648	320	2009-09-08 12:26				-		D298E95D	Deflate	Unix
fireLoginEvent.jsp	2 166	919	2009-09-08 12:27				-		1FEEA707	Deflate	Unix
index.jsp	12 112	2 838	2009-09-08 12:26				-		AA4DFB82	Deflate	Unix

0 object(s) selected

4. Navigate to `tilapia/services/servicetypes/`.

5. Find the file `babel-types-infos.js`.
   Right-click it and select **Edit** in the menu.

Structure of the babel-types-infos.js File

Each search type is described in a different block within the file.

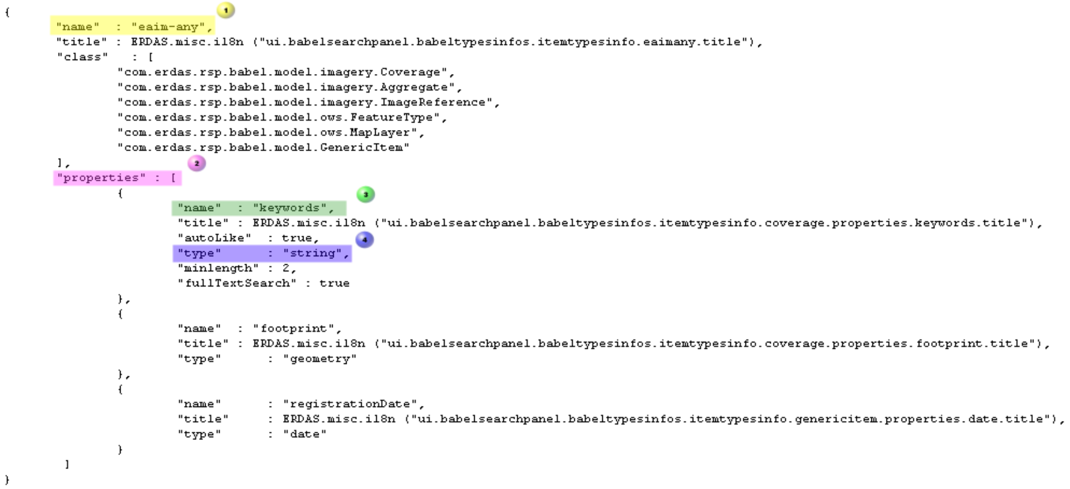
There are three main things you need to be able to locate and understand in each block:

1 The name of the ***search type***.

2 The header for the ***Properties*** block, which contains the search field definitions.

3 The name of a ***search field*** for that search type.

4 The ***data type*** of the search field.



```
{
        "name"   : "eaim-any",
        "title" : ERDAS.misc.i18n ("ui.babelsearchpanel.babeltypesinfos.itemtypesinfo.eaimany.title"),
        "class"   : [
                "com.erdas.rsp.babel.model.imagery.Coverage",
                "com.erdas.rsp.babel.model.imagery.Aggregate",
                "com.erdas.rsp.babel.model.imagery.ImageReference",
                "com.erdas.rsp.babel.model.ows.FeatureType",
                "com.erdas.rsp.babel.model.ows.MapLayer",
                "com.erdas.rsp.babel.model.GenericItem"
        ],
        "properties" : [
                {
                        "name"   : "keywords",
                        "title" : ERDAS.misc.i18n ("ui.babelsearchpanel.babeltypesinfos.itemtypesinfo.coverage.properties.keywords.title"),
                        "autoLike"  : true,
                        "type"        : "string",
                        "minlength" : 2,
                        "fullTextSearch" : true
                },
                {
                        "name"   : "footprint",
                        "title" : ERDAS.misc.i18n ("ui.babelsearchpanel.babeltypesinfos.itemtypesinfo.coverage.properties.footprint.title"),
                        "type"        : "geometry"
                },
                {
                        "name"        : "registrationDate",
                        "title"       : ERDAS.misc.i18n ("ui.babelsearchpanel.babeltypesinfos.itemtypesinfo.genericitem.properties.date.title"),
                        "type"        : "date"
                }
        ]
}
```

- The names of the ***search types*** shown in the `babel-types-infos.js` file do not match up exactly with the names of the search types shown in the Web Client because the `babel-types-infos.js` file uses internationalized keys for the titles. The following table shows which search type in the file represents a search type in the actual Web Client (using the English locale).

Search Type in the Web Client	Search Type in the babel-types-infos.js File
Any	eaim-any, eas-any
Datasets & Aggregates	eaim-coverage
Datasets	eaim-granule
Aggregates	eaim-aggregate
Vector Layers	wfs-feature-type
Map Layers	wms-map-layer

Search Type in the Web Client	Search Type in the babel-types-infos.js File
Coverage Layers	wcs-coverage
Custom Objects	generic-item

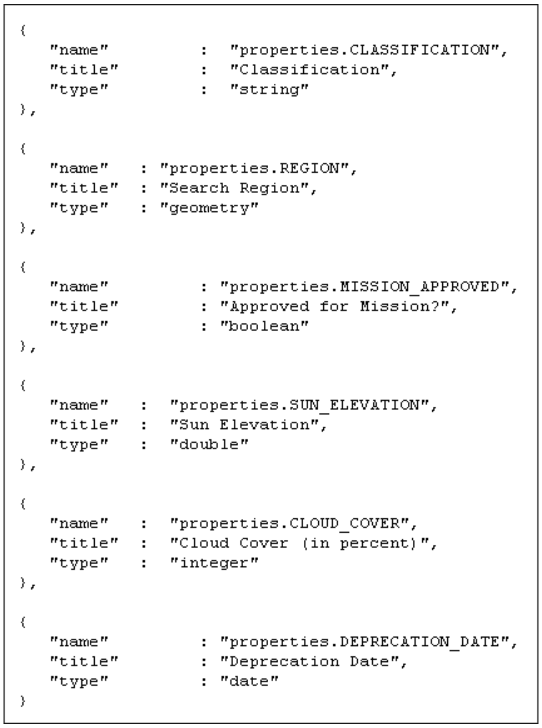
- The ***data types*** that can be used for a search field are:
  - string
  - geometry
  - integer
  - double
  - boolean
  - date

Creating a New Definition for the Custom Queryable Search Field

You will need to create a new definition for the search field for your custom queryable. Every entry must contain the following items:

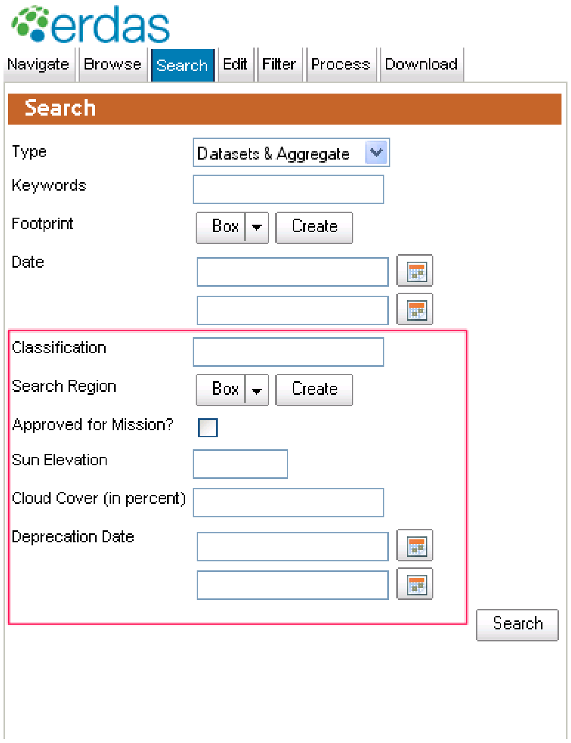
- **name** -the name of the custom queryable item in the format properties.<queryable_item_name>
  where <queryable_item_name> is the name of the custom queryable item in the `queryables.xml` file).
- **title** - the label that will be displayed next to the search field in the Web Client.
- **type** - the data type of the search field.

The following figure show some basic search field definitions that have been added to the babel-types-infos.js file.



```
{
   "name"         :  "properties.CLASSIFICATION",
   "title"        :  "Classification",
   "type"         :  "string"
},

{
   "name"   : "properties.REGION",
   "title"  : "Search Region",
   "type"   : "geometry"
},

{
   "name"         : "properties.MISSION_APPROVED",
   "title"        : "Approved for Mission?",
   "type"         : "boolean"
},

{
   "name"   :  "properties.SUN_ELEVATION",
   "title"  :  "Sun Elevation",
   "type"   :  "double"
},

{
   "name"   :  "properties.CLOUD_COVER",
   "title"  :  "Cloud Cover (in percent)",
   "type"   :  "integer"
},

{
   "name"         : "properties.DEPRECATION_DATE",
   "title"        : "Deprecation Date",
   "type"         : "date"
}
```

These search field definitions will create the search fields shown in the following figure.



Type: Datasets & Aggregate

Keywords

Footprint

Date

Classification

Search Region

Approved for Mission? □

Sun Elevation

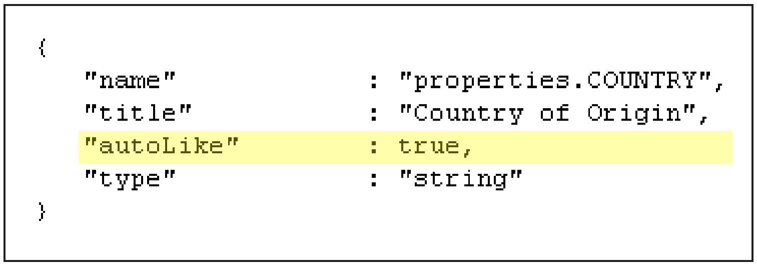
Cloud Cover (in percent)

Deprecation Date

You may also enter some additional properties to the search field definition in order to get the custom queryable search field to behave a bit differently.

autoLike

If you are creating a string search field, and you want the search to return all of the items that begin with the string entered in the field, add the autoLike property to the search field definition and set it to true, as shown in the example below. Do not place quotation marks around the value of the autoLike property.



```
{
   "name"                 : "properties.COUNTRY",
   "title"                : "Country of Origin",
   "autoLike"             : true,
   "type"                 : "string"
}
```

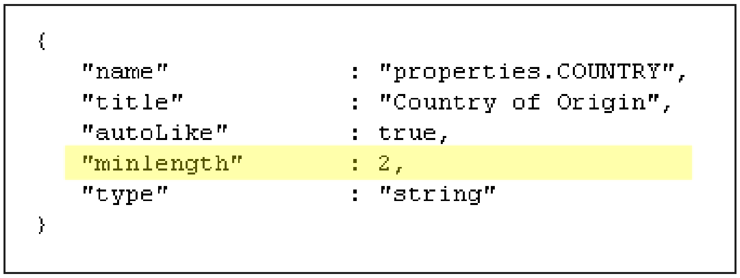
For example, if the user enters “United” in the search field created by the definition above, the search would return “United States of America” and “United Arab Emirates”. If the autoLike property were not present, the search would return nothing, because the Web Client would only return exact matches for the string.

If the autoLike property is included in the definition but its value is set to false, the search field behaves exactly as though the autoLike property were not included in the definition at all.

minlength

If you want the user to enter a minimum number of characters when using a string search field, add the minlength property to the search field definition and set the minimum string length. Do not place quotation marks around the string length number.

If the user types anything in this search field, the label will turn red if the field does not contain at least the minimum number of characters.

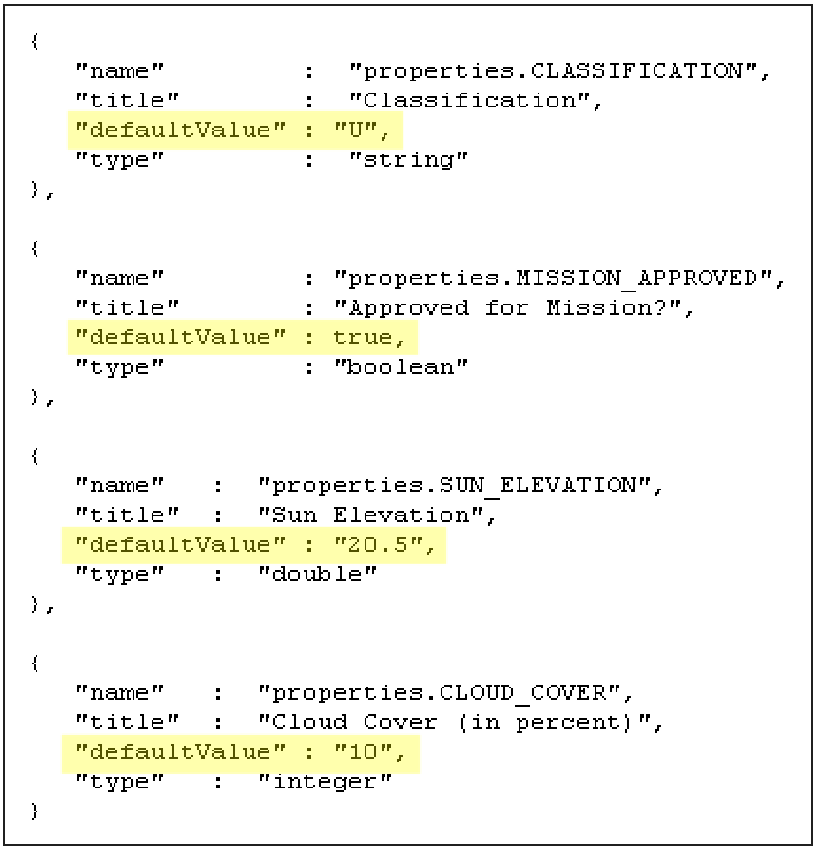


```
{
   "name"                 : "properties.COUNTRY",
   "title"                : "Country of Origin",
   "autoLike"             : true,
   "minlength"            : 2,
   "type"                 : "string"
}
```

defaultValue

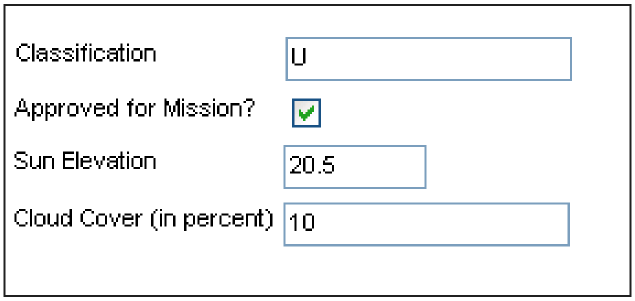
If you want to set a string, integer, double, or boolean search field to automatically display a certain value when it is created, add the defaultValue property to the search field definition.

If you are setting the defaultValue for a boolean search field, the value (true or false) must not be enclosed in quotation marks.



```
{
   "name"          :  "properties.CLASSIFICATION",
   "title"         :  "Classification",
   "defaultValue" : "U",
   "type"          :  "string"
},

{
   "name"          : "properties.MISSION_APPROVED",
   "title"         : "Approved for Mission?",
   "defaultValue" : true,
   "type"          : "boolean"
},

{
   "name"    :  "properties.SUN_ELEVATION",
   "title"   :  "Sun Elevation",
   "defaultValue" : "20.5",
   "type"    :  "double"
},

{
   "name"    :  "properties.CLOUD_COVER",
   "title"   :  "Cloud Cover (in percent)",
   "defaultValue" : "10",
   "type"    :  "integer"
}
```

The definition above will create the following search field in the Web Client.



Classification U

Approved for Mission? ☑

Sun Elevation 20.5

Cloud Cover (in percent) 10

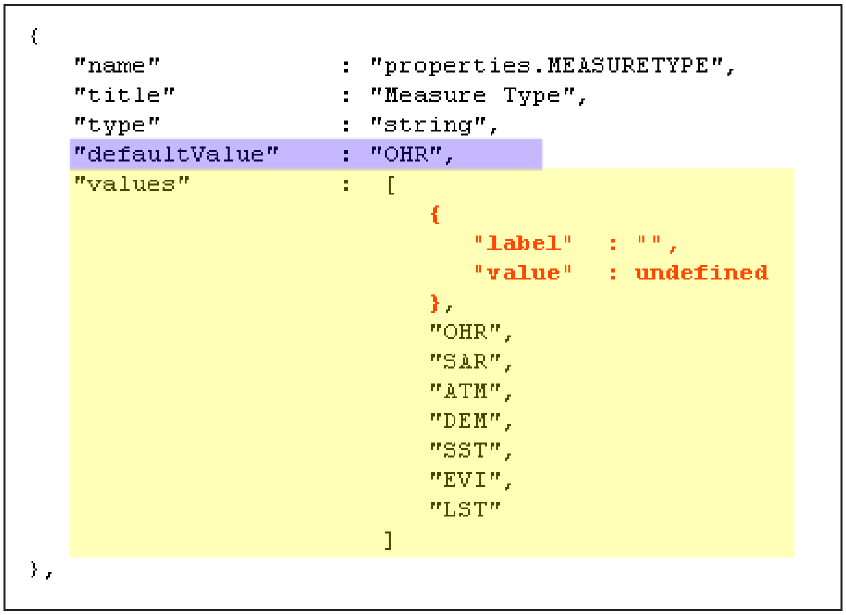
values

If you want to add a list of predefined options to a string, integer, or double search field, add the values property with your options defined in it.

It is important that you always include an “undefined” option. This allows the user to leave this search field blank so that it is not included in the search.

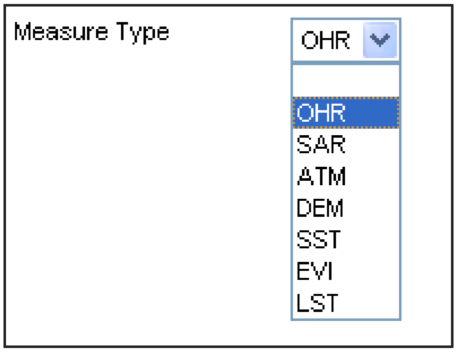
If you want one of the options to be selected by default, you can also add the defaultValue property and set its value to the option that should be preselected. If you just want the field to remain blank by default, you should not use the defaultValue property in the search field definition.

.



```
{
   "name"              : "properties.MEASURETYPE",
   "title"             : "Measure Type",
   "type"              : "string",
   "defaultValue"      : "OHR",
   "values"            :  [
                              {
                                 "label"   : "",
                                 "value"   : undefined
                              },
                              "OHR",
                              "SAR",
                              "ATM",
                              "DEM",
                              "SST",
                              "EVI",
                              "LST"
                          ]
},
```

The definition above will create the following search field the in the Web Client.

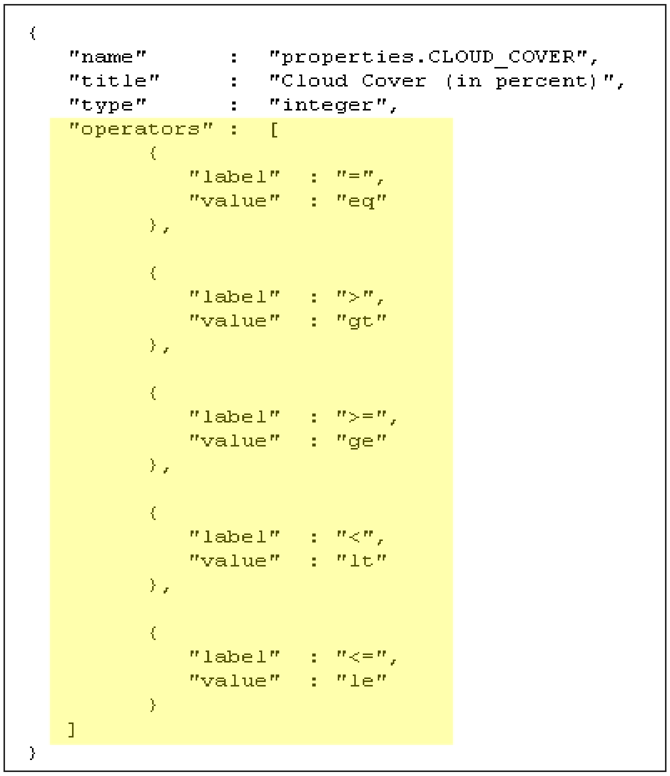


Measure Type OHR

operators

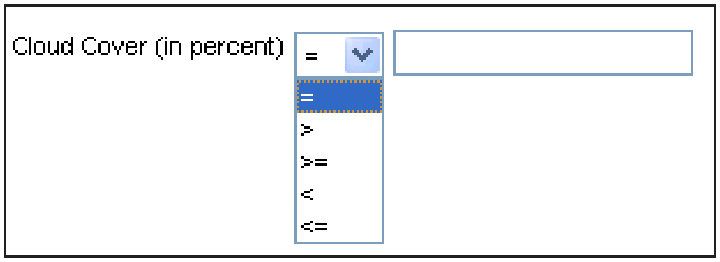
If you are creating a numerical field (type integer or double) and you want it to be accompanied by a dropdown list of operators, add the operators property to the search field definition as shown below. This allows the users to search for items whose value is equal to, less than, greater than, less than or equal to, greater than or equal to the value specified in the numerical field.

You can add this block to your definition exactly as you see it below; there is no need to change the values for the *label* or *value* properties.



```
{
   "name"       :  "properties.CLOUD_COVER",
   "title"      :  "Cloud Cover (in percent)",
   "type"       :  "integer",
   "operators" :  [
         {
            "label"   : "=",
            "value"   : "eq"
         },

         {
            "label"   : ">",
            "value"   : "gt"
         },

         {
            "label"   : ">=",
            "value"   : "ge"
         },

         {
            "label"   : "<",
            "value"   : "lt"
         },

         {
            "label"   : "<=",
            "value"   : "le"
         }
   ]
}
```

The definition above will create the following search field in the Web Client.

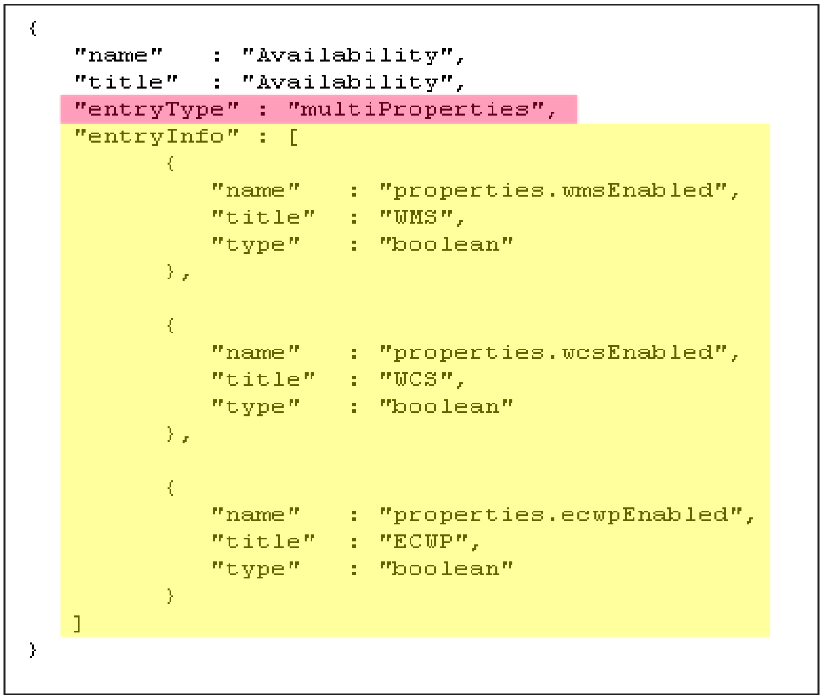


multiproperty

If you have certain small search fields that are related to each other, you can use the multiproperties field to place them on the same line.

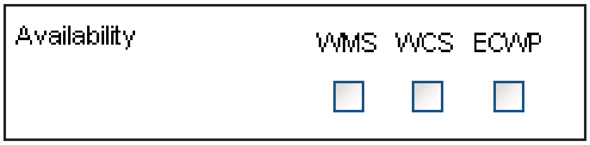
The following search field definition will create a group of checkboxes, with the entire group labeled “Availability” and each check box labeled “WMS”, “WCS”, and “ECWP”.

Notice that the name of each checkbox is preceded by “**properties.**” while the name of the checkbox cluster is not. This is because the values of each checkbox will be used to search for specific metadata values.



```
{
   "name"    : "Availability",
   "title"   : "Availability",
   "entryType" : "multiProperties",
   "entryInfo" : [
         {
            "name"    : "properties.wmsEnabled",
            "title"   : "WMS",
            "type"    : "boolean"
         },

         {
            "name"    : "properties.wcsEnabled",
            "title"   : "WCS",
            "type"    : "boolean"
         },

         {
            "name"    : "properties.ecwpEnabled",
            "title"   : "ECWP",
            "type"    : "boolean"
         }
   ]
}
```

The definition above will create the following search field in the Web Client.

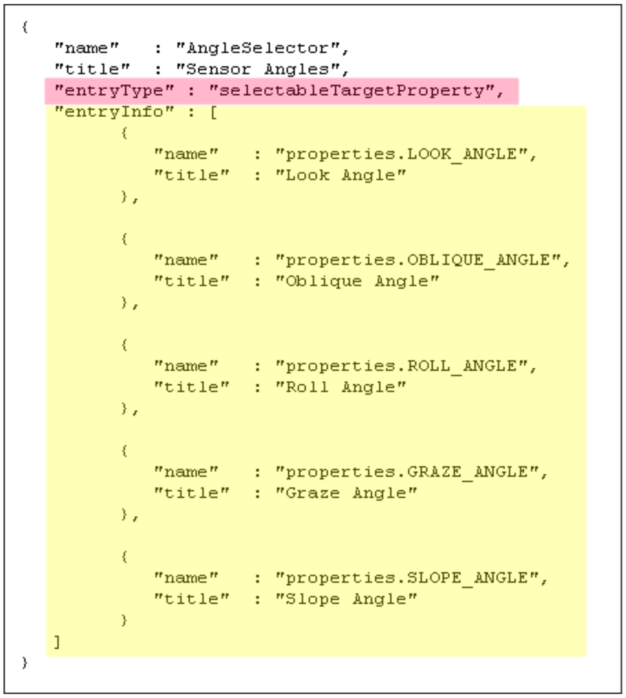


selectableTargetProperty

If you want to use the same set of controls to search for a custom queryable selected by the user, add the selectableTargetProperty to the search field definition.

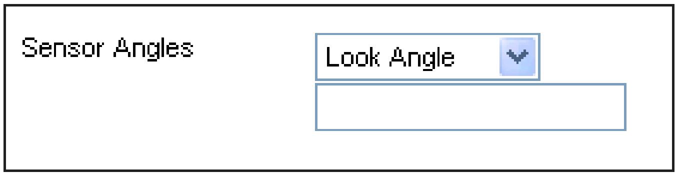
This allows you to provide a large number of search fields to the user without having so many fields shown on the tab that it becomes confusing for them. This also works well if you have a situation where you have a group of custom queryables fields that are mutually exclusive.

Notice that the name of each target property is preceded by “**properties.**” while the name of the dropdown box that lists them is not. This is because the values for the target properties will be used to search for specific metadata values.



```
{
   "name"   : "AngleSelector",
   "title"  : "Sensor Angles",
   "entryType" : "selectableTargetProperty",
   "entryInfo" : [
         {
            "name"    : "properties.LOOK_ANGLE",
            "title"   : "Look Angle"
         },

         {
            "name"    : "properties.OBLIQUE_ANGLE",
            "title"   : "Oblique Angle"
         },

         {
            "name"    : "properties.ROLL_ANGLE",
            "title"   : "Roll Angle"
         },

         {
            "name"    : "properties.GRAZE_ANGLE",
            "title"   : "Graze Angle"
         },

         {
            "name"    : "properties.SLOPE_ANGLE",
            "title"   : "Slope Angle"
         }
   ]
}
```

The definition above will create the following search field in the Web Client.

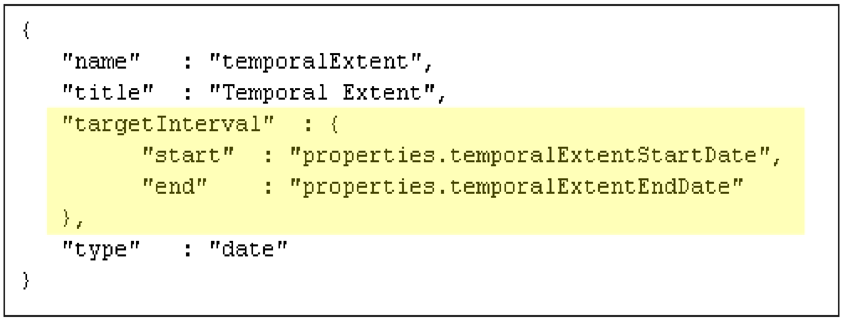


targetInterval

Normally, search fields of the date type allow you to search on a metadata element of the date type where the value will always be a single date. The search determines if the date value of that metadata element falls within a given range.

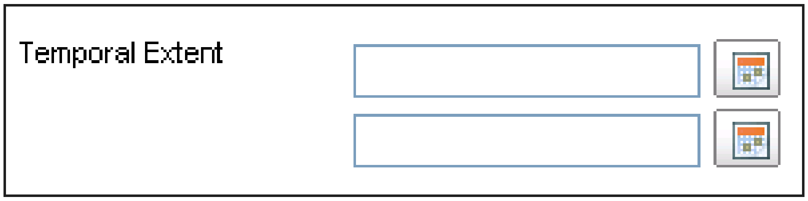
There are some metadata elements of the date type, such as a temporal extent, the where the value of the metadata element is a range of dates rather than just a single date.

In this case, you need to add the targetInterval property to the search fields definition so that the search will compare the range of dates in the metadata element to the range of dates provided by the user in the search fields.



```
{
   "name"   : "temporalExtent",
   "title"  : "Temporal Extent",
   "targetInterval"  : {
         "start"  : "properties.temporalExtentStartDate",
         "end"    : "properties.temporalExtentEndDate"
   },
   "type"   : "date"
}
```

The definition above will create the following search field in the Web Client.



Editing the babel-types-infos.js File

1. Find the block for the search type with which these search fields should be associated.

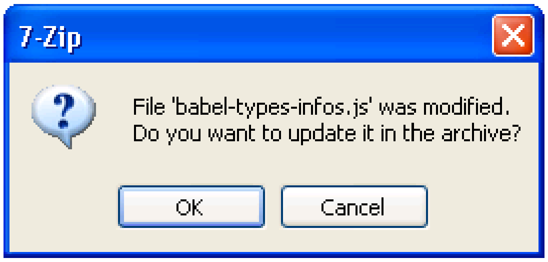
   The only items to which you can add custom queryables are datasets, aggregates, and custom objects.

2. Find the **Properties** block within that block. All of the existing search fields are defined there.

3. Place a comma after the last search field definition in the block.

4. Insert the entry for your custom queryable field.

5. Save and close the file.
   When you see the following message box, click **OK**.
   Your changes will not take effect in the Web Client if you do not update this file in the archive.



File 'babel-types-infos.js' was modified.
Do you want to update it in the archive?

6. Close 7-zip.

Your search fields should appear on the Search tab of the ERDAS APOLLO Web Client the next time it is opened.

Server Properties

The `global-server.properties` file contains a lot of the options that control how your ERDAS APOLLO Server product will work.

If you are using the JBoss application server that is bundled with the product, you will be able to find the `global-server.properties` file in the following location:

```
<APOLLO_HOME>/jboss/server/default/deploy/
erdas-apollo.ear/conf.
```

If you are using a WebLogic application server, you will be able to find the `global-server.properties` file in the following location:

```
<APOLLO_HOME>\dist\weblogic\erdas-apollo.ear\conf
```

If you change the `global-server.properties` file for any reason, you will need to restart your application server in order for the changes to take effect.

The following table shows the properties that you can change in the file and describes the behavior that the property controls.

Table 13: Customizable Parameters in the local-server.properties File

Property	Description
eaim.installation.home.directory	root folder where eaim/APOLLO is installed
gdal.path	path to gdall
rds.home.directory	rds/nci home directory
czs.temp.directory	Temporary directory for CZS (Clip, Zip, Ship) where intermediate files get created and destroyed
sms.execute.directory	Directory where the Imagine Spatial modeler EXE resides

Table 14: Customizable Parameters in the global-server.properties File

Property	Description
password.encryption.enabled	true to enable password encryption in the system, default is 'false'
com.lggi.esp.crawlers.login.userid	userid of the crawler user
com.lggi.esp.crawlers.login.password	password of the crawler user

Table 14: Customizable Parameters in the global-server.properties File

Property	Description
anonymous.access.enabled	if anonymous access is enabled in the system, default value is true
anonymous.login.userid	anonymous access user name
anonymous.login.password	anonymous access user password
com.lggi.esp.wcs.provider.path	location of WCS provider.fac file
com.lggi.esp.wcs.provider.name	name of WCS provider
com.lggi.esp.aggregate.null.property	defines the value to substitute for null values
com.lggi.esp.pyramid.decimation.factor	
com.lggi.esp.pyramid.level.width.cutoff.limit	
com.lggi.esp.pyramid.level.height.cutoff.limit	
com.lggi.esp.pyramid.aggregate.maximum.length	Get the maximum length in pixels that any side of an aggregate can be for pyramids to be allowed to be created.
com.lggi.esp.pyramid.outputformat.aggregate	
com.lggi.esp.thumbnail.default.width	The width of generated thumbnail images.
com.lggi.esp.thumbnail.default.height	The height of generated thumbnail images.
com.lggi.esp.footprint.size	The pixel size of the data subset on which the footprint will be calculated.
com.lggi.esp.footprint.maximum.points	The maximum number of points for the resulting footprint. If the footprint points defined in the global-server.properties is less than the minimum required, 5, the minimum points are returned.
com.lggi.esp.topic.name	JMS topic name
com.lggi.esp.topic.connection.factory	JMS connection factory name
ionic.ia.wcs.url	URL for WCS
ionic.ia.wrs.url	URL for WRS
ionic.ia.queryables	File URL to the queryables file
apollo.eaim.wps.url	URL for WPS
apollo.eaim.transient.url	URL for transient WMS providers used by WPS
file.browser.root.directory	The root directory for remote file browsing
provisioning.output.directory	The directory where zipped output files are stored.
provisioning.url.prefix	The URL prefix for zipped output files

Table 14: Customizable Parameters in the global-server.properties File

Property	Description
provisioning.max.request.size	The maximum size of provisioning requests in megabytes
provisioning.cleanup.schedule	schedule to run provisioning cleanup job, possible values are Hourly/Weekly/Daily/Monthly. Default is 'Daily'
provisioning.cleanup.start.time	Time of the day when the job executes, time format hh:mm:ss"
provisioning.output.expiry.duration.indays	files that are older than the value of this property will be deleted, property value should be greater than zero
security.util.class	application server specific security helper classes
jndi.ds.name	application server specific jndi names for datasources
java.naming.provider.url	application server specific java.naming.provider.url
jndi.port	application server specific
geoprocessing.cleanup.schedule	schedule to run geoprocessing cleanup job, possible values are Hourly/Weekly/Daily/Monthly. Default is 'Daily'
geoprocessing.cleanup.start.time	Time of the day when the job executes, time format "hh:mm:ss"
geoprocessing.output.expiry.duration.indays	WPS jobs that are older than the value of this property will be deleted along with the output the job generated, property value should be greater than zero
geoprocessing.aggregate.name	name of the aggregate under which WPS output will be catalogued
rds.home.directory	rds/nci home directory
spring.jaas.auth.login.config.file	Jaas auth login config file used by spring Jaas.
system.logging.enabled	if system logging is enabled in the system, default value is false. If enabled, all the relevant log events will be stored in the database.
data.manager.roles	comma separated role names that are allowed to use data manager

Security

ERDAS APOLLO uses a role-based security system where each role has a different level of access.

There are ***component-access*** roles that control the level of access to the components of the ERDAS APOLLO system itself. These roles are included in the standard installation of ERDAS APOLLO.

If you want to control the access to the data used by your ERDAS APOLLO system, you need to create a separate set of ***data-access*** roles and only use the included roles for their intended purpose, which is to provide access to the ERDAS APOLLO system components.

Table 15: Component-Access Roles (Included in ERDAS APOLLO)

Role	Description
esp_administrator	This role is for the overall administration of the ERDAS APOLLO system. Members of this role can add other users and set their level of access to the system. Membership in this role provides access to the ERDAS APOLLO Data Manager. The Data Manager will ***always*** provide this user with READ, UPDATE, DELETE, and MANAGE rights to all of the datasets and aggregates that are added to the ERDAS APOLLO Catalog.
esp_data_manager	This role is for the overall management of all the data used by the ERDAS APOLLO system. Membership in this role provides access to the ERDAS APOLLO Data Manager and ERDAS APOLLO Web Client. By default, members of this role receive READ, UPDATE, DELETE, and MANAGE rights to all of the datasets and aggregates that are added to the ERDAS APOLLO Catalog. If you need to provide users with access to the Data Manager component, but not READ, UPDATE, DELETE, and MANAGE rights to all of the data in it, you can do one of the following: • Use the security dialog boxes to change the rights that members of the esp_data_manager role will have to data on a case-by-case basis. These security dialog boxes appear when you use the Crawler Wizard or the Add Images Wizard to add data to your system, or if you right-click on data that is already in your catalog and select the Security option from the menu. The security dialog boxes will only allow you to change the rights for the esp_data_manager role if you are logged in to the Data Manager as a member of the esp_administrator or esp_data_manager roles. • Add an additional role that provides access to the Data Manager component, but who does not necessarily have READ, UPDATE, DELETE, and MANAGE rights by default. See **Creating an Additional Data Manager Component-Access Role** on page 77.
esp_data_analyst	*Reserved for future system use.*
esp_consumer	Members of this role can access the ERDAS APOLLO Web Client.

Table 15: Component-Access Roles (Included in ERDAS APOLLO)

esp_anonymous	Grants access to the ERDAS APOLLO Web Client as a guest user. Guest users will only be able to see the public data in the ERDAS APOLLO Catalog.

The role-based security system allows you to have individual users in your ERDAS APOLLO system as well. You grant access to a user by associating him with a role. Each user may be assigned to multiple roles. The standard installation of ERDAS APOLLO includes a few general user accounts so you can get the system running.

Table 16: Users (Included in ERDAS APOLLO)

User	Password	Role(s)
admin	apollo123	esp_administrator esp_consumer
dm	apollo123	esp_data_manager esp_consumer
da	apollo123	esp_data_analyst esp_consumer
consumer	apollo123	esp_consumer
public	public123	esp_anonymous

You can use the ***admin/apollo123*** and ***dm/apollo123*** to log in to the ERDAS APOLLO Data Manager and ERDAS APOLLO Web Client right after you have installed the products and started the application server.

You can use ***consumer/apollo123*** to log in to the ERDAS APOLLO Web Client right after you have installed the products and started the application server.

The user name/password combination ***da/apollo123*** is reserved for future system use. You may see it mentioned in configuration files as you customize your security, but you should not change it at this point.

The system uses the ***public/public123*** for users who are using the Web Client, but have not logged in to it. You should not try to explicitly log in to either the Data Manager or the Web Client with this user name/password combination because it is only designed for internal use by the system.

You need to change the password of the admin user as soon as possible after you install the software. The method for changing the password depends on how you are storing the security information for your ERDAS APOLLO system. The section **Security Information Location** on page 57 provides complete instructions on how to change your passwords.

Planning Your Changes to the Security Configuration

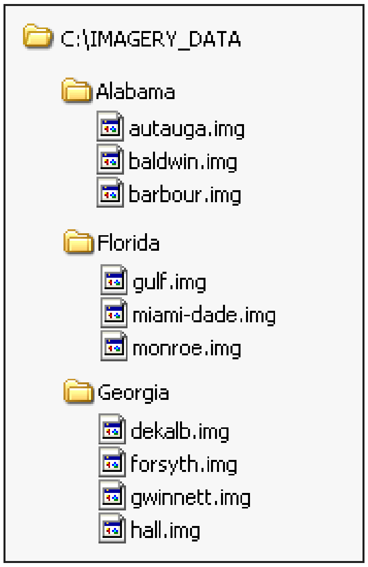
Most customers will want to add roles and users to the ERDAS APOLLO system. It is best to plan out what you need to add before you begin changing the system.

Example

You want to add individual user accounts for some of the staff members in your organization, John Doe, Jane Doe, and John Smith.

You will be working with data from Georgia, Florida, and Alabama, and you know that you only want users to see the data from a certain assigned state. You already have the imagery that you will be cataloging arranged in a directory structure so that all of the data is separated by state, as shown in the following diagram.

.



```
C:\IMAGERY_DATA
    Alabama
        autauga.img
        baldwin.img
        barbour.img
    Florida
        gulf.img
        miami-dade.img
        monroe.img
    Georgia
        dekalb.img
        forsyth.img
        gwinnett.img
        hall.img
```

John Doe is going to be your local ERDAS APOLLO administrator. He needs unrestricted access to the ERDAS APOLLO Data Manager and the ERDAS APOLLO Web Client. He also needs access to all of the data in the system, regardless of which state it comes from.

Jane Doe needs to be able to access to access the Data Manager and the Web Client, but she is only working to manage your data and will not be involved in the administration of the ERDAS APOLLO system.

John Smith is only going to be using the ERDAS APOLLO Web Client and only needs to be able to see data from Alabama.

To achieve this, you would need to:

1. Create a user account for each of your 3 users, using whatever account naming convention you like (JDoe, john_doe, etc.).

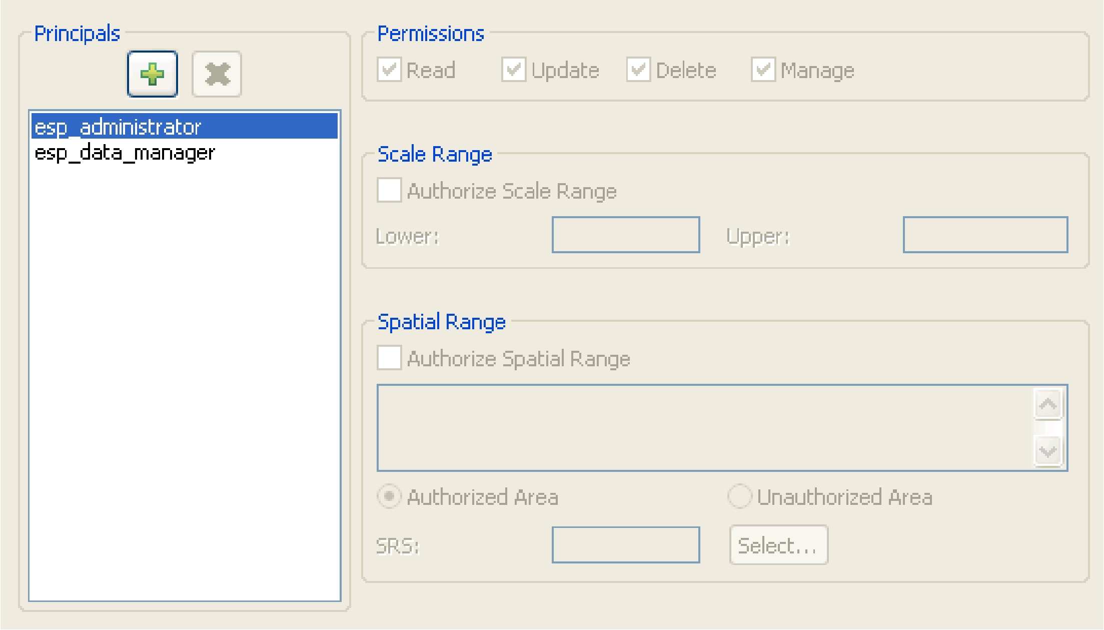
2. Create a role for each state, using whatever naming convention you like. (In this example, we will use the names DataAccess_GA, DataAccess_FL, and DataAccess_AL.)

3. Assign your users to the roles as shown in the table below.

User	Role(s)
John Doe	esp_administrator esp_consumer
Jane Doe	esp_data_manager esp_consumer
John Smith	esp_consumer DataAccess_AL

4. Use the ERDAS APOLLO Data Manager to set the proper security for the data itself.

   The ERDAS APOLLO 2010 Data Manager Guide explains all of the different ways that you can do this, but in every case, the controls you will be using will look something like the following figure.



esp_administrator
esp_data_manager

☑ Read ☑ Update ☑ Delete ☑ Manage

□ Authorize Scale Range

Lower: Upper:

□ Authorize Spatial Range

● Authorized Area ○ Unauthorized Area

SRS:

The **Principals** list should show every role that should have some type of access to this data, and only those roles.

By default, the Principals list always shows the esp_administrator and esp_data_manager roles.

You will need to use the ➕ and ✖ buttons to add any principals that need to have some kind of access to the data and to remove any that do not. The Data Manager will not allow you to remove the esp_administrator role.

Keep in mind that a user will have access to this data if even one of his roles has access to it. If he is in multiple roles that have access to it, then he has the same level of access as the role with the greatest level of access.

If this is a problem, you may need to change the security configuration for this data by giving that role with the greatest level of access a little less access, or you may need to remove that role from the list of principals for this data.

Security Information Location

The ERDAS APOLLO system allows you to store the information about roles and users in files, in the database, or you can use a Lightweight Directory Access Protocol (LDAP) service to access the user information you already have stored for your organization, such as the user information for e-mail or the local area network. Beginning with the 2011 release, the security information is stored and configured to use the database.

Storing Security Information in Files

In an ERDAS APOLLO installation, the system uses files to store the security definitions. These files cannot be managed using the Data Manager.

These files already include the standard roles and users that come with the system. If you want to continue using these files to store your security definitions, you will just need to add any additional roles and users that you want to use to these files. You also will want to change the default passwords that come with the system.

To add new users and roles to the system:

1. Navigate to the directory

   `<JBOSS_HOME>/server/default/conf/props/.`

2. Open the file `apollo-users.properties.`

```
# A sample users.properties file for use with the UsersRolesLoginModule
```

```
admin=apollo123
dm=apollo123
da=apollo123
consumer=apollo123
public=public123
John_Doe=03171952
Jane_Doe=07041982
John_Smith=10041971
```

3. Below the entries for the included users and passwords, type entries for your own users and passwords using the same format.

4. Save and close the `apollo-users.properties` file.

5. Open the file `apollo-roles.properties`.

   (It is also located in the directory

   `<JBOSS_HOME>/server/default/conf/props`.)

```
# A sample roles.properties file for use with the UsersRolesLoginModule
admin=esp_administrator,esp_consumer
dm-esp_data_manager,esp_consumer
consumer=esp_consumer
public=esp_anonymous
John_Doe=esp_administrator,esp_consumer
Jane_Doe=esp_data_manager,esp_consumer
John_Smith=esp_consumer, DataAccess_GA
```

6. Below the entries for the included users and roles, type entries for your own users and passwords using the same format.

   Even if a role has never been used before in this file, type it in this file just as you do with the ones that have been used before. The ERDAS APOLLO system will be able to recognize them.

7. Save and close the **apollo-roles.properties** file.

To change the default passwords that come with the system:

1. Navigate to the directory

   `<JBOSS_HOME>/server/default/conf/props/.`

2. Open the file `apollo-users.properties`.

```
# A sample users.properties file for use with the UsersRolesLoginModule
admin=apollo123
dm=apollo123
da=apollo123
consumer=apollo123
public=public123
```

3. Change the passwords that you want to change.

4. Save and close the `apollo-users.properties` file.

Storing Security Information in a Database

Security information is part of the APOLLO schema and by default the database login module is configured. Use the Data Manager > Configuration to add or change roles and users. You can also change the passwords that come with the system.

Using LDAP to Access Security Information Stored in an Active Directory Service

ERDAS APOLLO can use LDAP to communicate with an active directory service so that it can access the user information you already have stored for your organization, such as the user information for e-mail or the local area network. To set this up, you need to configure the active directory and ERDAS APOLLO.

- add the ERDAS APOLLO user roles to your active directory.
- add the users in your active directory to the appropriate role.
- change the `apollo-login-config.xml` file so that ERDAS APOLLO will recognize that you are using LDAP to perform security.

Before you begin configuring ERDAS APOLLO to use LDAP for the security domain, you need to obtain the following information:

- The URL and port number of the active directory service, which is generally in the format

```
  ldap://<domain_server>.<domain_name>.<top_level_domain>:389
```

  In the following example, we will use
  ldap://apollo.erdas.com:389

- The user name and password of a user who will be the administrator for the ERDAS APOLLO Data Manager.

  In the following example we will use
  user： **apolloadmin** password： **apolloadminpwd**

- The organizational unit containing the user who will be the administrator for the ERDAS APOLLO Data Manager.

  In the following example, we will use **Users**.

To configure your active directory:

1. Create these groups in the active directory: **esp_administrator, esp_consumer,** and **esp_anonymous**.
   Determine which users will need to be administrators for the system and which ones need to be able to use the ERDAS APOLLO Data Manager to crawl data. Add those users to the **esp_administrator** group.

2. Determine which users need to be able to log in to the ERDAS APOLLO Web Client to view data, but do not need to use the ERDAS APOLLO Data Manager. Add those users to the **esp_consumer** group.

3. Create a user account that the system can use for guest users to the system and add this user to the **esp_anonymous** group.

   If you create your guest user with the user name *public* and the password *public123*, it will be automatically recognized by the ERDAS APOLLO system.

   To use a public user account and password other than public/public123:

   1. Create that user account using the Data Manager.

To change the apollo-login-config file:

1. Navigate to the `<JBOSS_HOME>/server/default/conf` folder.

2. Change the name of the current `apollo-login-config.xml` file to `apollo-login-config.filebackup`.

3. Copy the `apollo-login-config.ldap` file and rename the copy `apollo-login-config.ldapbackup`

4. Open the original `apollo-login-config.ldap` file and find the lines shown below. Incorporate your information into those lines just as we did with our information in the example below. The parts that we changed are shown in blue.

```
<module-option
name="java.naming.provider.url">ldap://apollo.erdas.com:389/</module-option>
<module-option name="bindDN">CN=apolloadmin,CN=Users,DC=erdas,DC=com</module-
option>
<module-option name="bindCredential">apolloadminpwd</module-option>
<module-option name="baseCtxDN">CN=Users,DC=erdas,DC=com</module-option>
<module-option name="baseFilter">(sAMAccountName={0})</module-option>
<module-option name="rolesCtxDN">CN=Users,DC=erdas,DC=com</module-option>
```

5. Save the document as `apollo-login-config.xml`.

Hiding Clear Text Passwords in Configuration Files

The passwords that are used to gain access to the ERDAS APOLLO Server and the ERDAS APOLLO database are stored in the configuration files in clear text and are shown exactly as you typed them in.

If this is a security concern for your organization, you can hide these passwords.

Server Configuration Files

Some of the passwords are stored in ERDAS APOLLO Server configuration files for use by the system. Use the Data Manager to stored passwords under the Configuration > Security settings. You can encrypt the passwords in these files so that the system can still read them, but a human reader cannot.

The application server that you are using for ERDAS APOLLO Server determines which server configuration files contain passwords.

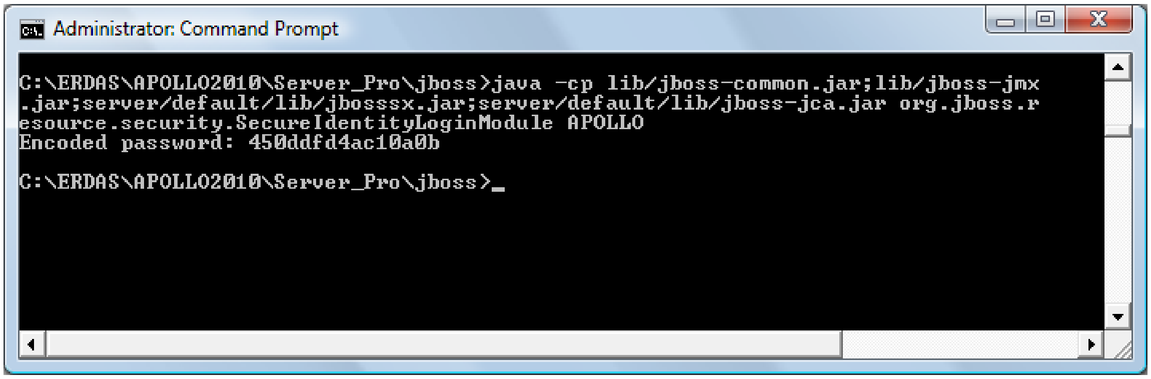
To encrypt the ERDAS APOLLO database password in the configuration file `apollo-login-config.xml`:

1. Navigate to the directory `<APOLLO_HOME>/jboss/server/default`.
2. Open the file `apollo-login-config.xml` for editing.
3. Open a command line window.
4. Navigate to the directory `<APOLLO_HOME>\jboss`.
5. Type in the following command at the prompt (or find it in the file `apollo-login-config.xml` and copy and paste it). Substitute the clear text database password for the <CLEAR_TEXT_PASSWORD> placeholder.

   This will invoke the JBoss secure identity login module, which will encrypt your clear text password.

```
java -cp lib/jboss-common.jar;lib/jboss-jmx.jar;server/default/lib/jbosssx.jar;
server/default/lib/jboss-jca.jar
org.jboss.resource.security.SecureIdentityLoginModule <CLEAR_TEXT_PASSWORD>
```

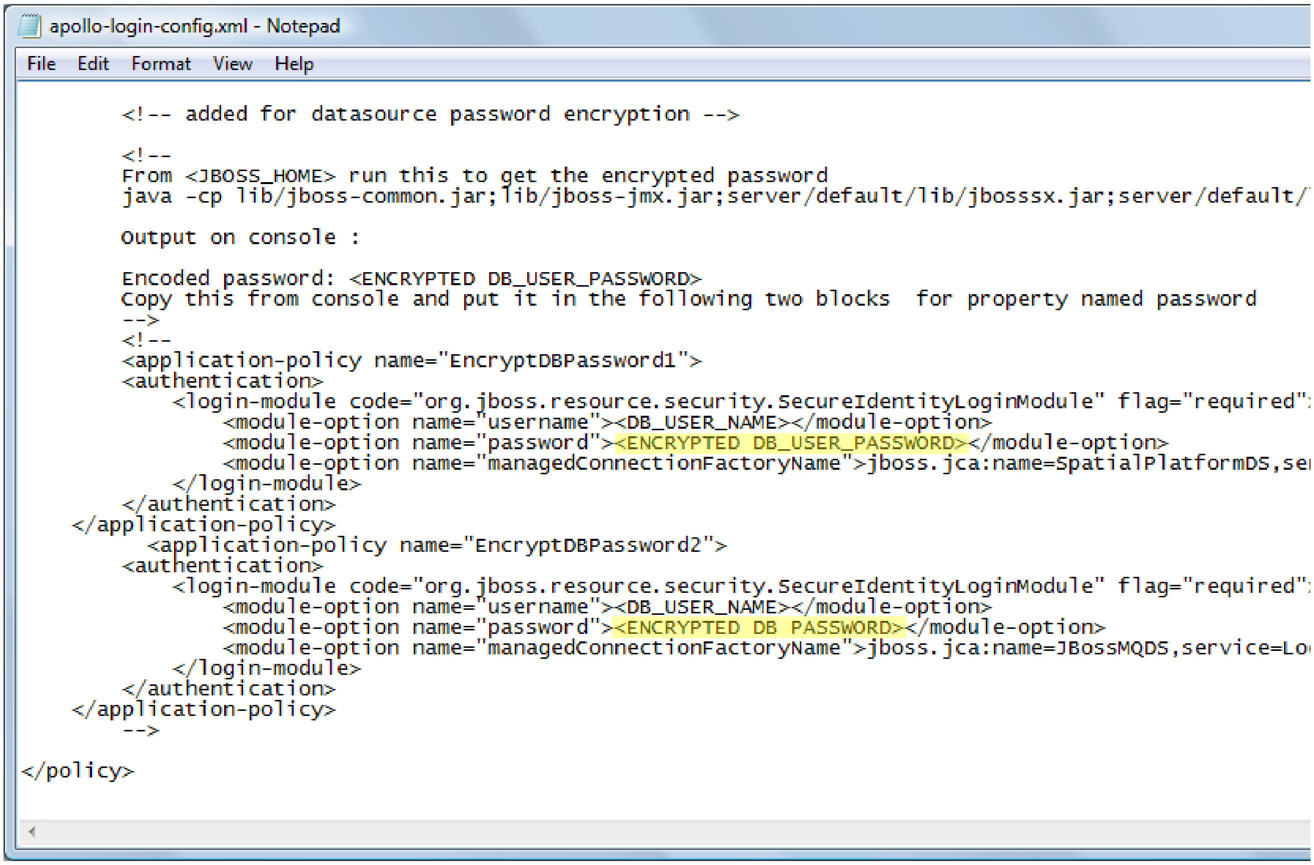
6. The secure identity login module displays the encrypted password in the command line window.



```
C:\ERDAS\APOLLO2010\Server_Pro\jboss>java -cp lib/jboss-common.jar;lib/jboss-jmx
.jar;server/default/lib/jbosssx.jar;server/default/lib/jboss-jca.jar org.jboss.r
esource.security.SecureIdentityLoginModule APOLLO
Encoded password: 450ddfd4ac10a0b

C:\ERDAS\APOLLO2010\Server_Pro\jboss>_
```

Leave the command line window open. You will need it again in one of the next steps.

7. Place this encrypted password inside the highlighted locations inside the `apollo-login-config.xml` file.



```
        <!-- added for datasource password encryption -->

        <!--
        From <JBOSS_HOME> run this to get the encrypted password
        java -cp lib/jboss-common.jar;lib/jboss-jmx.jar;server/default/lib/jbosssx.jar;server/default/

        Output on console :

        Encoded password: <ENCRYPTED DB_USER_PASSWORD>
        Copy this from console and put it in the following two blocks  for property named password
        -->
        <!--
        <application-policy name="EncryptDBPassword1">
        <authentication>
            <login-module code="org.jboss.resource.security.SecureIdentityLoginModule" flag="required"
                <module-option name="username"><DB_USER_NAME></module-option>
                <module-option name="password"><ENCRYPTED DB_USER_PASSWORD></module-option>
                <module-option name="managedConnectionFactoryName">jboss.jca:name=SpatialPlatformDS,se
            </login-module>
        </authentication>
    </application-policy>
          <application-policy name="EncryptDBPassword2">
        <authentication>
            <login-module code="org.jboss.resource.security.SecureIdentityLoginModule" flag="required"
                <module-option name="username"><DB_USER_NAME></module-option>
                <module-option name="password"><ENCRYPTED DB PASSWORD></module-option>
                <module-option name="managedConnectionFactoryName">jboss.jca:name=JBossMQDS,service=Lo
            </login-module>
        </authentication>
    </application-policy>
        -->

</policy>
```

8. Save and close the `apollo-login-config.xml` file.

Administrative Tools Configuration Files

In the `build.properties` file located in the directory `<APOLLO_HOME>/tools/schema-generator`, the passwords are stored so that if you ever need to use the ant tool to rebuild `erdas-apollo.ear` or `apollo-client.war`, the correct passwords will be included in the newly built files. To hide these passwords, you can simply save a copy of this file to a secure location and delete it from the directory `<APOLLO_HOME>/tools/schema-generator`. Restore it to its normal location if and when you need to use the ant tool to rebuild.

In the `build.properties` file located in the directory `<APOLLO_HOME>/tools/harvester-console`, the APOLLO admin password is stored for the process of harvesting services from an older version of ERDAS APOLLO and placing them in the catalog for ERDAS APOLLO 2010 or higher. After you initially upgrade to ERDAS APOLLO 2010, you probably will not need this file again, but you should still save it to a safe location in case something happens to your catalog and you need to rebuild it. After a copy of the file is saved in the safe location, you can delete it from the directory `<APOLLO_HOME>/tools/harvester-console` and restore it to its normal location if and when you need to rebuild the catalog.

Customizing Data Cataloging Behavior

Changing Default Security Settings

The Data Manager allows you to define the security for an aggregate or dataset in the catalog using the following panels or dialog boxes:

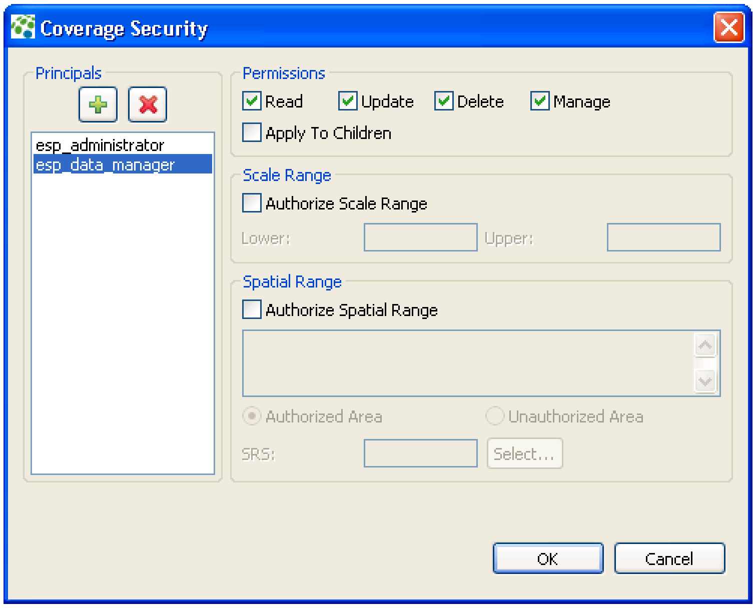
- **Coverage Security** dialog box- accessible by right-clicking an aggregate in Explorer View or a dataset in the Datasets View and selecting Security in the menu that appears.

- **Crawler Schedule Wizard** security settings panel- accessible when you are creating a crawler to register data in the catalog.

- **Add Images Wizard** security settings panel - accessible by right-clicking an aggregate in Explorer View and selecting Ad IMages... in the menu that appears.

- **Process Security** dialog box - accessible by right-clicking a process in the Explorer View and selecting Security in the menu that appears. (This dialog is only used in the Professional version of ERDAS APOLLO.)

On a fresh installation, the panel always shows the **esp_administrator** and **esp_data_manager** principals with all rights enabled. If you want to add additional principals or remove the esp_data_manager principal, use the Configuration settings in the Explorer pane of the Data Manager. You cannot use an asterisk (*) as the default principal.



Coverage Security

Principals

esp_administrator
esp_data_manager

Permissions

☑ Read ☑ Update ☑ Delete ☑ Manage

□ Apply To Children

Scale Range

□ Authorize Scale Range

Lower: Upper:

Spatial Range

□ Authorize Spatial Range

● Authorized Area ○ Unauthorized Area

SRS:

You can set the security panels so that they automatically include other principals. This makes it easier on Data Manager users if you have a certain user or role who frequently be given some type of access to the data in the ERDAS APOLLO Catalog.

Addition of Extra Metadata Parsers

How does ERDAS APOLLO Parse Metadata?

When the ERDAS APOLLO Data Manager first encounters data that it is going to register in the catalog, it uses an **imagery decoder** to read the data. This imagery decoder includes a basic, built-in metadata parser to detect the metadata that describes the data.

ERDAS APOLLO can then use a **metadata parser** to obtain any additional metadata from the data. The Metadata Parsers panels of the Crawler Schedule Wizard and Add Images Wizard show all of the metadata parsers that are available for use in your system and allow you to select which ones to run.

The ERDAS APOLLO Data Manager then registers the data in the catalog and also uses a metadata writer to create a file that contains all of the metadata information. ERDAS APOLLO comes with a built-in ISO 19139 **metadata writer** which allows it to generate an ISO 19139 metadata file for each piece of imagery data that is registered in the catalog.

The following table shows which attributes that the imagery decoder can read using the basic, built-in metadata parser and what the information for those attributes might look like.

Attribute Name	Sample Value
name	Center_atlanta_ll_wgs84.img
title	Downtown Atlanta 3 Band Color IKONOS Imagery
abstract	The IKONOS satellite sensor collected this Atlanta Downtown imagery in 6 strips. The source IKONOS imagery was mosaicked, stretched, and processed to create this 3-band single image.
node type	Dataset
data type	IMG
source name	Center_atlanta_ll_wgs84.img
source path	D:\ERDAS\Generation_4\APOLLO\data\IKONOS

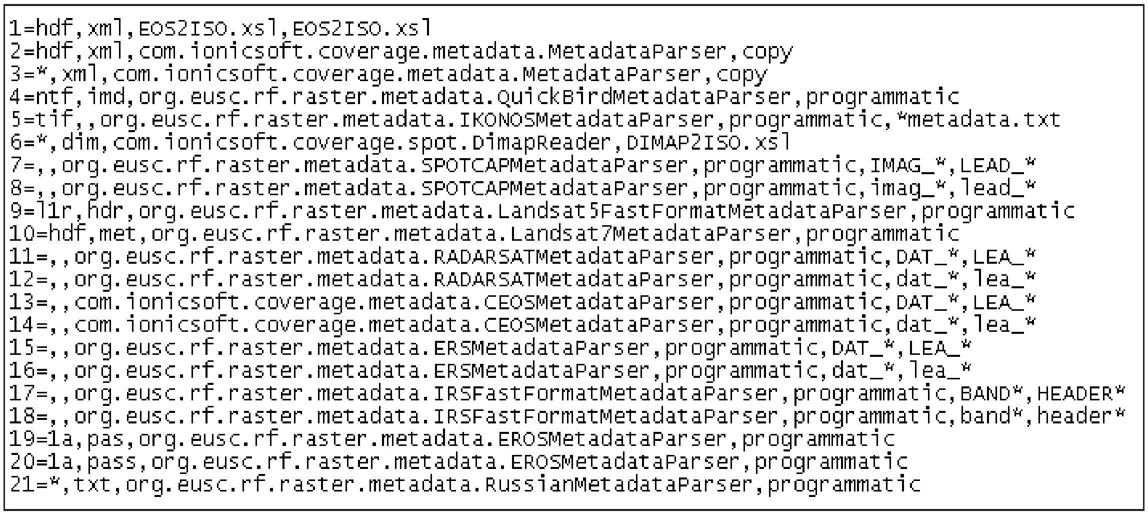
bounding box	-84.36064665742737, 33.81559696757166, -84.32653756651828, 33.84410605848075
srs	EPSG:4326
sizes	334*280
band axis	Band1,Band2,Band3

How Does ERDAS APOLLO Determine Which Metadata Parser to Use?

The metadata parsers and metadata writers that are available for use in your system are specified in the file `metadata.txt` that is located in the directory <APOLLO_HOME>/config/erdas-apollo/providers/coverage/.

The purpose of each line in this file is to tell ERDAS APOLLO which metadata parser and which metadata writer to use when it encounters a certain imagery format combined with a certain metadata file format. ERDAS APOLLO determines the imagery format by looking at either the file extension or the file name itself and it determines the metadata file format by looking at either the file extension or the file name itself.

The standard content of the file looks like the following:



```
1=hdf,xml,EOS2ISO.xsl,EOS2ISO.xsl
2=hdf,xml,com.ionicsoft.coverage.metadata.MetadataParser,copy
3=*,xml,com.ionicsoft.coverage.metadata.MetadataParser,copy
4=ntf,imd,org.eusc.rf.raster.metadata.QuickBirdMetadataParser,programmatic
5=tif,,org.eusc.rf.raster.metadata.IKONOSMetadataParser,programmatic,*metadata.txt
6=*,dim,com.ionicsoft.coverage.spot.DimapReader,DIMAP2ISO.xsl
7=,,org.eusc.rf.raster.metadata.SPOTCAPMetadataParser,programmatic,IMAG_*,LEAD_*
8=,,org.eusc.rf.raster.metadata.SPOTCAPMetadataParser,programmatic,imag_*,lead_*
9=l1r,hdr,org.eusc.rf.raster.metadata.Landsat5FastFormatMetadataParser,programmatic
10=hdf,met,org.eusc.rf.raster.metadata.Landsat7MetadataParser,programmatic
11=,,org.eusc.rf.raster.metadata.RADARSATMetadataParser,programmatic,DAT_*,LEA_*
12=,,org.eusc.rf.raster.metadata.RADARSATMetadataParser,programmatic,dat_*,lea_*
13=,,com.ionicsoft.coverage.metadata.CEOSMetadataParser,programmatic,DAT_*,LEA_*
14=,,com.ionicsoft.coverage.metadata.CEOSMetadataParser,programmatic,dat_*,lea_*
15=,,org.eusc.rf.raster.metadata.ERSMetadataParser,programmatic,DAT_*,LEA_*
16=,,org.eusc.rf.raster.metadata.ERSMetadataParser,programmatic,dat_*,lea_*
17=,,org.eusc.rf.raster.metadata.IRSFastFormatMetadataParser,programmatic,BAND*,HEADER*
18=,,org.eusc.rf.raster.metadata.IRSFastFormatMetadataParser,programmatic,band*,header*
19=1a,pas,org.eusc.rf.raster.metadata.EROSMetadataParser,programmatic
20=1a,pass,org.eusc.rf.raster.metadata.EROSMetadataParser,programmatic
21=*,txt,org.eusc.rf.raster.metadata.RussianMetadataParser,programmatic
```

If you want to add support for a metadata parser that you have obtained from a third-party or that you have created yourself, then you need to add it to the list in the `metadata.txt` file.

All of the items in the list must be in one of the following formats:

```
<Nb>=<Source_File_Extension>,<Metadata_File_Extension>,<Metadata_Parser>,<Metadata_Writer>
<Nb>=,<Metadata_File_Extension>,<Metadata_Parser>,<Metadata_Writer>,<Source_Filename_Pattern>
<Nb>=<Source_File_Extension>,,<Metadata_Parser>,<Metadata_Writer>,<Metadata_Filename_Pattern>
<Nb>=,,<Metadata_Parser>,<Metadata_Writer>,<Source_Filename_Pattern>,<Metadata_Filename_Pattern>
```

Where each line can consist of the following parameters:

- **<Source_File_Extension>** - the file extension of the imagery file.

  The extension can appear on several lines, as each imagery format may have several metadata extensions associated with it.

  The "*" means that any imagery file can be coupled with the described metadata extension.

  If the source file extension is not specified at all, the system will look at the <Source_Filename_Pattern> parameter.

- **<Metadata_File_Extension**> - the file extension of the metadata file.

  The same extension can appear on several lines because different imagery formats may use metadata files with the same file extension.

  If the metadata file extension is not specified at all, the system will look at the <Metadata_Filename_Pattern> parameter.

- **<Metadata_Parser>** - the metadata parser to be used when ERDAS APOLLO encounters the imagery format combined with the metadata file format that are specified in the line.

  The metadata parser specification can be a reference to an xsl file available at the "com.ionicsoft.coverage.metadata.resource" location in the class path, or it can reference any metadata parser class available in the class path as well. This defined somehow the way to ingest the metadata in the WCS framework.

- **<Metadata_Writer> -** the metadata writer to be used when ERDAS APOLLO encounters the imagery format combined with the metadata file format that are specified in the line.

  You can use any one of the following as your metadata writer specification:

  - A reference to an xsl file available at the "com.ionicsoft.coverage.metadata.resource" location in the class path, where the referenced xsl is intended to transform specific metadata into the ISO 19139 format.

  - The keyword **programmatic**. This specifies that the built-in ISO 19139 metadata writer should be used to generate an ISO 19139 metadata file.

  - The keyword **copy**. This specifies that any metadata file found along with the imagery file should just be copied and taken into the system "as is".

- **<Source_Filename_Pattern> -** this allows you to assign a metadata parser and writer based on the name of the imagery file instead of the file extension.

  This is only used if no <Source_File_Extension> parameter is defined.

  This parameter is helpful in instances where the imagery format does not use a file extension.

  This parameter allows you to define the filename pattern using "*" (multi-char) and "?" (single-char) wildcard characters.

- **<Metadata_Filename_Pattern>** - this allows you to assign a metadata parser and writer based on the name of the metadata file instead of the metadata file extension.

  This is only used if no <Metadata_File_Extension> parameter is defined.

  This parameter allows you to define the filename pattern using "*" (multi-char) and "?" (single-char) wildcard characters.

What Metadata Parsers Are Included with ERDAS APOLLO?

The following metadata parsers are included in the ERDAS APOLLO system:

- **Landsat5 (Fast Format rev. B)** - decodes Landsat5 (Fast Format rev. B) satellite images metadata. These metadata files usually have an `.hdr` extension.

- **RADARSAT** - decodes RADARSAT satellite images metadata. Such metadata filenames usually begin with *VDF_DAT*, *TRA* or *LEA* prefixes (like the CEOS format does).

- **EROS** - decodes EROS satellite images metadata. Such metadata files usually have a `.pass` extension.

- **CEOS -** This metadata parser is intended to decode CEOS satellite images metadata. Such metadata files usually begin with *VDF_DAT* or *LEA* prefixes.

- **DIMAP** - decodes SPOT DIMAP satellite images metadata. Such metadata files usually have a `.dim` extension.

- **SPOT (CAP/CEOS) Format** - decodes SPOT satellite images metadata (SPOT CAP/CEOS format only). Such metadata files usually begin with *VOLD* or *LEAD* prefixes.

- **ERS** - decodes ERS satellite images metadata. Such metadata files usually begin with *VDF_DAT* or *LEA* prefixes (like the CEOS format does).

- **QuickBird** - decodes QuickBird satellite images metadata. Such metadata files usually have a `.imd` extension.

- **IRS 1C-1D (Fast Format)** - decodes IRS 1C-1D (Fast Format) satellite images metadata. Such metadata files usually have a header file of exactly 4608 bytes. There is no name pattern defined for this format.

- **ISO 19115/19139 (v 1.0)** - decodes ISO-19115/19139 metadata files, based on the 1.0 version of the specification. Such metadata files usually have an `.xml` extension.

- **Landsat7** - decodes metadata for Landsat7 (Level 1) satellite images. Such metadata files usually end with a `_hdf.met` or `_mtl.txt` suffix.

- **IKONOS** - decodes metadata for IKONOS satellite images. Such metadata files usually end with a `_metadata.txt` suffix.

- **Russian Satellite** - decodes metadata for Russian satellite images. Such metadata files usually have a `.txt` extension (`info.txt` or `readme.txt` are good metadata file candidates).

How Do I Add My Own Metadata Parsers?

The *ERDAS APOLLO Solutions Toolkit Guide* provides instructions on creating your own metadata parsers so that they will be compatible with the ERDAS APOLLO system.

Web Client Configuration and Customization

Internationalization

When you install ERDAS APOLLO, the language of the ERDAS APOLLO Web Client defaults to the locale of the server computer from which the ERDAS APOLLO Web Client is served, and if that information is not present, it defaults to American English. The ERDAS APOLLO Web Client includes a drop-down box in the upper right corner that allows the user to explicitly choose the presentation language of the ERDAS APOLLO Web Client from the given choices of English, French, German, Polish, Dutch, Japanese, or Chinese. If you select another presentation language, that language will be used for the duration of the session.

You can configure the ERDAS APOLLO Web Client to open with the language you choose, and you can also add support for additional languages.

In order to perform these customizations, you need to know which files the ERDAS APOLLO Web Client uses to store the language information. All of the language files for the ERDAS APOLLO Web Client are located in the directory `<APOLLO_HOME>/webapps/apollo-client/default/WEB-INF/classes`.

The files are:

- `tilapia.properties` - contains the setting for the default language for the ERDAS APOLLO Web Client.

- `apollo-client.properties` - contains a list of all the languages supported by the ERDAS APOLLO Web Client.

- `tilapia-i18n.properties` (English)

  OR

  `tilapia-i18n`**_xx**`.properties`

  language file for the Toolkit where `xx` = fr (French), de (German), pl (Polish), nl (Dutch), ja (Japanese), or zh (Chinese).

- apollo-client-i18n.properties

  OR

  apollo-client-i18n_xx.properties

  Language file for the ERDAS APOLLO Web Client where xx = fr (French), de (German), pl (Polish), nl (Dutch), ja (Japanese), or zh (Chinese).

NOTE: “i18n” is an abbreviation for Internationlization denoted by the 18 letters between the first “I” and the last “n”.

Change the Default Language

ERDAS APOLLO reads a setting in a file to determine which files it should read to obtain the labels for the controls. You can change this setting and direct ERDAS APOLLO to use a different set of language files. If you want to use a language other than those listed above, create the language files before you change this setting.

1. Navigate to the directory `<APOLLO_HOME>/webapps/apollo-client/default/WEB-INF/classes`.

2. Open the file `tilapia.properties`.

3. Find the two lines shown below.

```
# # Set a default locale for the application
# locale.default=
```

4. Enter the desired locale code after `locale.default=`.

5. Uncomment (activate) the ***locale.default*** property by removing the # in front of the property.

6. Save and close `tilapia.properties`.

7. Restart the application server.

Additional Languages

You can customize your ERDAS APOLLO Web Client to support any language whose characters can be represented in Unicode.

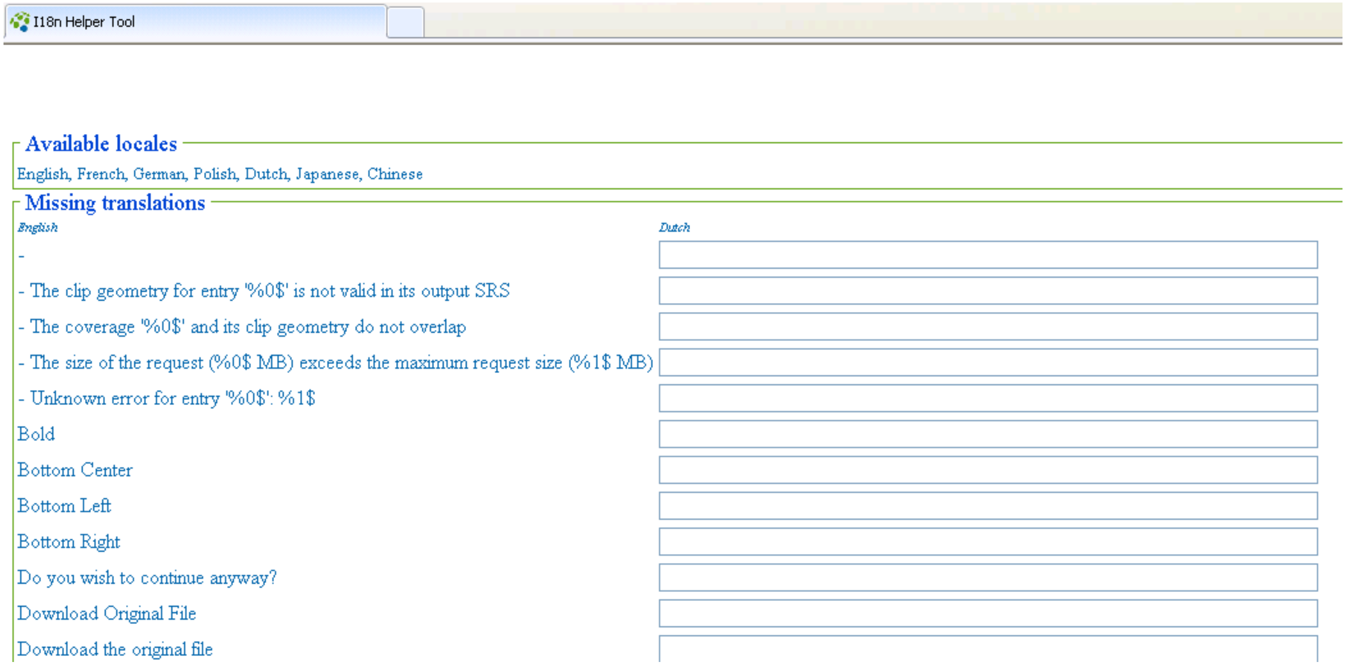
If you are using the ERDAS APOLLO Web Client, create new tilapia-i18n and apollo-client-i18n language files for your language and add this new language to the list of supported languages in the `apollo-client.properties` file.

If you are only using the Web Toolkit create a new tilapia language file. You will not need to create a new apollo-client language file, because it only contains labels for the ERDAS APOLLO Web Client. You also will not need to add this new language to the list of supported languages in `apollo-client.properties`.

After you create and edit the necessary files, you will need to rebuild the `apollo-client.war` file and redeploy it to your application server.

Create new language files

1. Navigate to the directory `<APOLLO_HOME>/webapps/apollo-client/default/WEB-INF/classes`.
2. Create an empty `tilapia_i18n_xx.properties` file (and a apollo-client-i18n_XX.properties file if you are translating the web client), where xx is a two-letter code that represents the name of the language. Look up the ISO 3166 codes for the translated language at `http://www.iso.org/iso/english_country_names_and_code_elements`.
3. Access `http://<APOLLO_HOME>/apollo-client/tools/i18n.jsp` and the I18n Helper Tool opens. The tool shows the strings that need a translation and the strings that are already written in the i18n.properties file for all loaded locales.



Available locales

English, French, German, Polish, Dutch, Japanese, Chinese

Missing translations

English	Dutch
-	
- The clip geometry for entry '%0$' is not valid in its output SRS	
- The coverage '%0$' and its clip geometry do not overlap	
- The size of the request (%0$ MB) exceeds the maximum request size (%1$ MB)	
- Unknown error for entry '%0$': %1$	
Bold	
Bottom Center	
Bottom Left	
Bottom Right	
Do you wish to continue anyway?	
Download Original File	
Download the original file	

4. Enter each translated text in the box to the right of the English version of the text. Duplicate any formatting place holders such as %s or %d. These place holders must remain intact to hold strings and numeric values.

5. When complete, click Save at the bottom of the page. The tool downloads your changes to a zip file with one or both of the following entries (depending on what strings you translated).

   - tilapia-i18n.changes (for the Web Toolkit)

   - apollo-client-i18n.changes (for the ERDAS APOLLO Web Client)

6. Copy the *.changes file(s) to your tilapia-i18n-xx.properties or apollo-client-i18n-xx.properties files accordingly.

Add to the list of supported languages

1. Navigate to the directory `<APOLLO_HOME>/webapps/apollo-client/default/WEB-INF/classes`.

2. Open the file `apollo-client.properties`.

3. Find the lines shown below.

```
# Locales
locale.info=EN|English
locale.info=FR|Fran\u00e7ais
locale.info=DE|Deutsch
```

Add another line with the ***locale.info*** property, and set the value of the property using the following format:

<LANGUAGE_CODE>|<LANGUAGE_NAME>

where <LANGUAGE_CODE> matches with the code you used to represent the language within the file name and <LANGUAGE_NAME> is the name of the language that appears in the dropdown box on the ERDAS APOLLO Web Client that will allow the user to select the presentation language.

Note that the name of the language must be expressed using only ASCII characters and Unicode escape (\uXXXX) characters.

If you added Spanish to the list, the list would look like the following:

```
# Locales
locale.info=EN|English
locale.info=FR|Fran\u00e7ais
locale.info=DE|Deutsch
locale.info=ES|Espa\u00a4ol
```

Rebuild/Redeploy the apollo-client.war file

1. Create an ANT_HOME system variable with the path to the directory `<APOLLO_HOME>/tools/ant/bin`.

2. After you have created the system variable, open a command line window and type:

   `cd <APOLLO_HOME>/tools/ant/bin` <press ENTER>
   `ant tomcat55` <press ENTER>

   The argument of the "ant" call should indicate the name of the application server that you are using.

   You can open and read the `build.xml` file located in the `<APOLLO_HOME>` directory to obtain the correct argument for your application server.

3. Wait until the build is successful, then go to the directory `<APOLLO_HOME>/dist/<APPSERVER_NAME>` and copy the file `apollo-client.war`.

4. Redeploy the new `apollo-client.war` file for your application server.

   For JBoss
   Paste into the directory
   `<APOLLO_HOME>/jboss/server/default/deploy`

   For Tomcat (5.5 and 6)
   Paste into the directory `<APOLLO_HOME>/tomcat/webapps`

5. Restart the application server.

Configuration

The files that make up the ERDAS APOLLO Web Client are placed together in the directory `<APOLLO_HOME>/webapps/apollo-client` when you install the ERDAS APOLLO Server.

Many customers like to customize their web clients. To do this, you will need to open the `<APOLLO_HOME>/webapps/apollo-client` directory, find the file or files that contain the properties you want to change, and make the required changes. After you have changed all of the files, you will need to run an Apache Ant script that will compress that directory into a **W**eb **AR**chive (WAR) file. That war file is deployed to the application server that you are using for the ERDAS APOLLO Server.

If you need to use the ERDAS TITAN client for WPS execution, do not remove the APOLLO-CLIENT.WAR from the ERDAS APOLLO Application Server.

Properties Files

The APOLLO Web Client uses a number of .properties files to store configuration parameters.

Anything you type in these properties files may potentially be viewable by anyone who can view the ERDAS APOLLO Web Client online. Do not put any information you want to keep confidential inside these files!

Default Hierarchy

The `root.properties` file is in WEB-INF/classes/tilapia.properties (this is hard-coded). All properties files need to be in the classpath (such as in WEB-INF/classes).

- `tilapia.properties`
    - `apollo-client.properties`
        - `apollo-client-contexts.properties` (specified in a tryimport in `apollo-client.properties`)
        - `apollo-im-components.properties` or `apollo-server-components.properties` (specified in a tryimport in `apollo-client.properties`)

Entries

Name	Description	Note
tryimport	look for the specified file in the classpath and if it is there, import it as an additional properties file	
bingmaps.key	False - Do not enable Microsoft Bing Maps; True - Enable Microsoft Bing Maps for internal use; <Customer's Bing Key> - Enable Microsoft Bing Maps for external use	set during installation
feature.panel.config	feature panel configuration as described in the *ERDAS APOLLO Solutions Toolkit Main Guide*, Section 3.6.5.2.4	
locale.default	Default locale - see **Internationalization** on page 70	
i18n.files	Location for internationalization files - see **Internationalization** on page 70	
layerinfohandlers.wfshandler.maxfeatures	Maximum number of features to display in the Layer Info tool	
metadata.tc211.stylesheet		
metadata.stylesheet.iso19139	The path to the XSL file for displaying ISO19139 metadata	
editors.timestameintervaleditor.dateoffset	The default date offset (in days) between the start and end date	

Name	Description	Note
context.startup	Context files that appear in the Context list	
context.overview.default	Path to the default context file	
jsonhelper.ident	Indentation of the JSON output in the logs	
log.type	Type of log (such as FILE, etc.)	
log.enable	Enable the log	
log.filename	Path to log file, such as /temp/mylog	
log.maxfile	Number of log files to create (the logger will create new log files as needed, cycling from mylog0 to mylogmaxfile -1)	
log.filesize	Maximum size of log file in bytes before a new log file is created	
service.ias.url	IAS URL	see eaim.server.wcs.url
service.catalog.url	Catalog URL	see eaim.server.catalog.url
service.wrs.url	WRS URL	see eaim.server.wrs.url
service.wps.url	WPS URL	see eaim.server.wps.url
service.clipzipship.url	Clip/Zip/Ship service URL (see eaim.server.clipzipship.url)	
service.quartzinterface.url	Quartz JSON Interface URL (see eaim.server.quartzinterface.url)	
service.wrs.type	WRS type: Babel or RSCatalog	
service.streamedraster.access.enabled	True: streaming (ECWP) access enabled	see eaim.server.streamedraster.access.enabled
service.streamedraster.access.url	URL for streaming (ECWP) access	see eaim.server.streamedraster.access.url
service.ias.iso19115Xslt	Path to the XSL file for displaying ISO19115 metadata (can be undefined)	
service.ias.queryables	Path to the XML file containing queryable information	
modules.search.layers.assumeErdas	If set to True, assume modules.search.layers references a WFS from ERDAS (allows optimizations)	

Name	Description	Note
modules.search.layers.useWms	If set to True, if the modules.search.layers WFS server exposes a WMS interface it be used for rendering	
thumbnail.width	Width of thumbnails in pixels	
thumbnail.height	Height of thumbnails in pixels	
thumbnail.create	If set to True, automatically create new thumbnails	
mail.smtp.host	SMTP host for Clip/Zip/Ship messages	see eaim.server.smtp.host
mail.smtp.port	SMTP host for Clip/Zip/Ship messages	see eaim.server.smtp.port
mail.smtp.user	SMTP host for Clip/Zip/Ship messages	see eaim.server.smtp.user
ui.objectinspector.defaultresultsperpage	Default number of results to display in object inspectors	
modules.search.max.nb.results	Maximum number of search results (-1 for unlimited)	
modules.search.thumbnail.popup	if set to True, display thumbnails in the popup when hovering over a result	
modules.search.thumbnail.details	if set to True, display thumbnails in the extended details panel	
scripts.bundles.desc	path to the file containing the description of custom modules	

Components

Components are discrete parts of the web client application that can be added or removed independently. The installer provides two files, `apollo-im-components.properties` for Professional and `apollo-server-components.properties` for Essentials-SDI. It may be desirable to remove components by deleting or commenting out their entries. Alternatively, a different properties file containing components information can by provided by providing a different tryimport entry in `apollo-client.properties`.

Entry	Usage
components.active	each entry will be added to the application on startup

Entry	Usage
components.search.typesfilter	the entries that appear in the "types" dropdown list in the Search panel
components.search.defaulttype	the "types" entry that is selected by default in the Search panel

Available components:

- **Browse**: The Browse tab, which allows the user to:
    - see all of the known services in tree form
    - add new services
    - create new transient services by uploading data from the local file system
- **Edit**: The Edit tab for viewing and editing features stored in a WFS
- **Filter**: The Filter tab for viewing and modifying the filter on a WFS layer
- **Search**: The Search tab allowing the user to search the catalog for available resources
- **OverviewMap**: The Overview Map displaying a large scale overview of the user's current map view
- **WPS**: The Process tab for managing WPS processes
- **DownloadImagery**: The Download tab for managing images for the Clip/Zip/Ship operation

Contexts

The ERDAS APOLLO Web Client provides a tool that allows users to pick from a list of predefined context files. By convention this list is populated from entries in `apollo-client-contexts.properties`. An example is provided in `apollo-client-contexts-samples.properties`. The first entry will be loaded at application startup by default. The format for these entries is pipe (|) delimited as follows:

- **context.startup**
  <path>|<title>|<documentation-page>|<overview-map-path>

- **path**
  the path to the context file, such as /context/default-basemap.xml

- **title**
  the title that will appear in the dropdown list, such as "Default Basemap"

- **documentation page**
  the path to an HTML page providing additional details of the context file (can be blank)

- **overview map path**
  the path to the context file that will be used in the overview map. If blank, defaults to **ic.context.path**

Apache HTTP

This section exalins information related to Apache products.

Configure a nonbundled Apache http Server

Out of the box, a node with JBoss and/or Tomcat application server works with mod-jk load balancer and Weblogic application server works with mod-proxy.

Mod-Jk setup

1. Copy `mod_jk.so` from `<APOLLO_HOME>\apache\modules` into `<APACHE_INSTALL>\modules` folder.

2. Check port number in httpd.conf if it matches with the one used during install.

```
# The value 80 should be replaced with the values used during installation.
Listen 80
```

3. Edit DocumentRoot setting in httpd.conf.

   - Modify DocumentRoot to `<APOLLO_HOME>/storage/htdocs` (This is required for CZS.)

   - Modify the security on `<APOLLO_HOME>/storage/htdocs`

```
#
# This should be changed to whatever you set DocumentRoot to.#
<Directory "<APOLLO_HOME>/storage/htdocs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
    # for more information.
    #
    Options FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   Options FileInfo AuthConfig Limit
    #
    AllowOverride None

    #
    # Controls who can get stuff from this server.
    #
    Order allow,deny
    Allow from all

</Directory>
```

4. Copy ApolloRedirect-jk.conf from `<APOLLO_HOME>\apache\conf` into `<APACHE_INSTALL>\conf.`

5. Add the following in httpd.conf.

```
#To work with mod-jk , pls add the following
Include conf/ApolloRedirect-jk.conf
```

6. Copy `<APOLLO_HOME>/apache/modules/mod_fastcgi-2.4.6_iwsAP22.dll` into `<APACHE_INSTALL>/modules`

7. Copy `ImageWebServer.conf` from `<APOLLO_HOME>\apache\conf` into `<APACHE_INSTALL>\conf`

8. Add the following in httpd.conf.

```
LoadModule fastcgi_module modules/mod_fastcgi-2.4.6_iwsAP22.dll

<IfModule fastcgi_module>
Include conf/ImageWebServer.conf
</IfModule>
```

9. Copy the property files namely uriworkermap.properties, workers.properties from `<APOLLO_HOME>/apache/con`f to `<APACHE_INSTALL>/conf.`

Mod-proxy Setup

1. Check port number in httpd.conf if it matches with the one used during install.

```
# The value 80 should be replaced with the values used during installation.
Listen 80
```

2. Edit DocumentRoot setting in httpd.conf.

   - Modify DocumentRoot to `<APOLLO_HOME>/storage/htdocs` (This is required for CZS.)

   - Modify the security on `<APOLLO_HOME>/storage/htdocs`

```
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "<APOLLO_HOME>/storage/htdocs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
    # for more information.
    #
    Options FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   Options FileInfo AuthConfig Limit
    #
    AllowOverride None

    #
    # Controls who can get stuff from this server.
    #
    Order allow,deny
    Allow from all

</Directory>
```

3. Copy `ApolloRedirect.conf` from `<APOLLO_HOME>\apache\conf` into `<APACHE_INSTALL>\conf`.

4. Add the following in httpd.conf .

```
#To work with mod-proxy , pls add -Djboss or -Dtomcat with the startup .
# Redirect requests from apache to appserver, pls add the following.
Include conf/ApolloRedirect.conf
```

5. Copy `<APOLLO_HOME>/apache/modules/mod_fastcgi-2.4.6_iwsAP22.dll` into `<APACHE_INSTALL>/modules`

6. Copy `ImageWebServer.conf` from `<APOLLO_HOME>\apache\conf` into `<APACHE_INSTALL>\conf`.

7. Add the following in `httpd.conf`

```
LoadModule fastcgi_module modules/mod_fastcgi-2.4.6_iwsAP22.dll.
```

8. Add the following after the `LoadModule` command.

```
<IfModule fastcgi_module>
Include conf/ImageWebServer.conf
</IfModule>
```

Trouble Shooting

This chapter gives specific information on trouble shooting issues.

WPS

GMD Model

When you publish from GMD to WPS, be sure you check the Delete if Exists field for the output Raster object. You must check this field or the WPS process fails for this model.

MDL Model

When you publish from MDL to WPS, be sure that the output raster object declaration is set to “DELETE_IF_EXISTING in the script.

```
Integer RASTER n5_PROMPT_USER FILE DELETE_IF_EXISTING
PUBOUT USEALL ATHEMATIC 8 BIT UNSIGNED INTEGER arg3;
```

Aggregate Styling

When you request a map for an aggregate, the styles applied to the data vary depending on where such styles or pyramids are defined. The result is sometimes surprising and might appear that the styles are not applied.

When a request is sent to an aggregate, it works in two steps.

9. builds a result coverage by fetching all the granule children (descending into the tree)

10. renders the result coverage by applying the style defined on the requested coverage.

Standard Styling Policy

The standard (default) styling policy is the following.

- If the aggregate can be directly rendered, it is rendered as a whole. An aggregate can be directly rendered if it has a decoder name or has a pyramid or has the style defined on itself or defined in the parent chain. It has a decoder name if all its children have the same type.

- If the aggregate cannot be directly rendered, each child is rendered separately (by applying the same logic) and then the final image is aggregated. Note that the order of the children is not guaranteed.

Child Styling Policy

An alternate styling policy, named "child", only renders the aggregate as a whole if it has a style defined or a pyramid associated. If it has none of those but has a decoder name (because all its children are homogeneous), then the children are rendered separately.

Deep Styling Policy

The Deep styling policy ignores the styles or pyramids on the aggregate. The children are always rendered separately.

PORTRAYALMODE

These options are activated through a service configuration parameter that lets you choose the rendering logic when datasets and aggregates have styles and/or pyramids. It is called PORTRAYALMODE. As listed above, it can have the following values.

- standard - old behavior (default)
- style - render an aggregate as a whole if it has a style defined or a pyramid associated, otherwise render each child separately
- deep - render each child separately
- default - same as standard (currently)

Example:

```
<PARAM NAME="PORTRAYALMODE" VALUE="standard" />
```

Appendix A - IMAGINE Projection System

This appendix discusses the following:

- **Spheroids and Datums** on page 3
- **Understanding Datums, Spheroids, and Projections** on page 5
- **Add a Custom SRS** on page 11

Spheroids and Datums

The ERDAS IMAGINE Projection System comes with an extensive library of built-in projections, spheroids, and datums.

Parametric Datums

A horizontal datum is a mathematical model of the Earth's surface that is used to calculate the coordinate components (for example, latitude and longitude) of a point on the surface of the Earth. This surface is defined by a spheroid and the position and orientation relationship of the spheroid to a reference mathematical model of the Earth. The georeferenced coordinates are unique only if qualified by a datum. If you go across two different datums during georeferencing without considering the coordinate shift between them, the potential error can be up to hundreds of meters.

There are three types of parametric datums supported in the IMAGINE Projection System:

- The first type of datum is defined by seven parameters referred to the reference ellipsoid WGS 84. The seven parameters are x-y-z translations, omega-phi-kappa rotations, and scale variations. The Standard Molodensky transformation and seven-parameter transformation are used for parametric datum shifts.

The seven-parameter datum corresponds to the EPSG Coordinate Frame Transformation.

- The second type of datum is defined by NADCON grids in which the coordinate shifts among datum NAD 27, NAD 83, and HARN are calculated by bilinear interpolation.
- The third type of datum is defined by MREs based on DMA documents. You can identify these datums by the labels ending with MRE on the Spheroid tab.

Most parametric datums applied to global areas are basically spheroids themselves without any position shift or rotations relative to WGS 84. They are assumed to have the same centers as that of WGS 84. The main reason to use the spheroid name as a global datum name is to make a smooth transition from older to newer projection versions. Avoid using them when other appropriate local datums are available.

It is not recommended that you use any datum if you are not sure what it is. Using the wrong datum can introduce significant geometric errors (up to a few hundred meters) when performing datum shift calculation.

For more information about the parametric datum shift, refer to the DMA TR 8350.2 document. For NADCON, please check with the National Geodetic Survey.

Surface Datums

Just as a parametric datum can be used to calculate the coordinate components (for example, latitude and longitude) of a point on the surface of the Earth, the surface datum is a reference surface to which heights and elevations are referred.

There are four types of surface datums:

- Constant surface datums provide a constant shift value to apply to the entire area.
- Raster (or grid) surface datums use information stored in raster data files to define shift surfaces.
- Multiregression surface datums use an MRE to define the shift surface.
- Vector (or point) surface datums are not currently supported in the IMAGINE Projection System.

For more information about the parameters for each of these datum transformations, see Grid Datum Example on page 53.

Understanding Datums, Spheroids, and Projections

The datums, spheroids, and projections are found in the <*APOLLO_HOME*>*tools*/native/raster/etc/spheroid.tab> file, which is used by the ERDAS IMAGINE Projection System.

The following listing shows part of the entry from the `spheroid.tab` file for the Australian National spheroid:

```
"Australian National" {
     15 6378160.0 6356774.719
     "Australian National" 0 0 0 0 0 0 0
     "Anna 1 Astro 1965" -491 -22 435 0 0 0 0
     "Australian Geodetic 1966" -133 -48 148 0 0 0 0
     "Australian Geodetic 1984" -134 -48 149 0 0 0 0
     "Australian Geodetic 1966 (MRE) Geoid" SURFACE
     {
          REGRESSION  -90 0 90 360    0.052359880    1.413716760    0.052359880   -
7.016223920
          HEIGHT =
               {
                     0  0 -4.258
                     0  1 2.740
                     0  2 70.479
                     1  1 -34.946
                     2  0 12.676
                     1  2 24.680
                     0  4 -348.586
                     2  2 -14.488
                     3  1 122.302
                     0  5 37.035
                     4  1 -17.307
                     5  0 -6.704
                     0  6 645.556
                     3  3 -80.304
                     5  1 -108.866
                     0  7 -19.475
                     3  4 -271.301
                     0  8 -430.089
                     8  0 -5.561
                     2  8 97.772
                     4  6 449.462
                     8  2 -69.354
                     4  7 -692.991
                     9  3 265.504
                     5  9 1311.842
                 }
     } DESCRIPTION = "Australian Geodetic 1966 (MRE) Geoid, from DMA TR 8350.2"

  .
  .
  .
```

"Australian National" is the name of the spheroid. The next line defines its sequence number in the spheroid.tab file, the semi-major axis, and the semi-minor axis (in meters). The general syntax is:

```
"Spheroid Name" {
      Sequence_Number Semi-Major_Axis Semi-Minor_Axis
      "Spheroid Name" 0 0 0 0 0 0 0
      Datums...
}
```

Seven-Parameter Ellipsoidal Transformation

Following the spheroid definition are datums associated with each spheroid. There are two types of datums that can be used by the IMAGINE Projection System: the ellipsoidal datum and the surface datum. The ellipsoidal datums are recorded in terms of the seven parameters required to calculate a shift to the WGS84 datum.

The seven-parameter ellipsoidal transformation corresponds to The EPSG coordinate frame transformation.

Any datums added to the spheroid.tab file must also be recorded as datum shift parameters to WGS84. The parameters are recorded in one of two ways, as shown in the two examples that follow.

Parametric Datum Example

The example below shows a parametric datum:

```
"Datum Name" [PARAMETRIC] dx dy dz rw rj rk ds [DESCRIPTION = string]
```

Where:

dx, **dy** and **dz** are the x,y,z translations to WGS84, in meters,

(or rw), (or rj), and (or rk) are the omega, phi, kappa rotations to WGS84, in radians and scientific notation,

and, **ds** is the scale change to WGS84 in scientific notation.

The keyword **ELLIPSOIDAL** is optional.

The text **DESCRIPTION** of the datum is also optional.

Most parametric datums applied to global areas are basically spheroids themselves without any position shifts or rotations relative to WGS 84. They are assumed to have the same centers as that of WGS 84. The use of the "global datum" syntax below is no longer required by the IMAGINE Projection System.

Grid Datum Example

The example below shows a grid datum:

```
"Datum Name" GRID gridfilename [DESCRIPTION = Description of the datum]
```

Where:

The keyword **GRID** is mandatory.

The **gridfilename** is the raster file used to define the datum. The raster files used for the transformations must reside in the <*APOLLO_HOME*>/native/raster folder. No paths can be used to point to these files.

The text **DESCRIPTION** of the datum is optional.

Spheroid Example

The example below shows a spheroid:

```
"Spheroid Name" 0 0 0 0 0 0 0
```

The main reason to use the spheroid name as a global datum name is to make a smooth transition from older to newer projection versions. Avoid using them when other appropriate local datums are available.

You can add new datums to an existing spheroid by editing the spheroid.tab file with any text editor and adding a new line for each datum in the section for that spheroid. You can also add a new spheroid with associated datums by adding a block of text to the end of the file using the following syntax:

```
"Spheroid Name" {
   Sequence_Number Semi-Major_Axis Semi-Minor_Axis
   "Datum Name 1" dx dy dz rw rj rk ds
   "Datum Name 2" dx dy dz rw rj rk ds
   "Datum Name 3" dx dy dz rw rj rk ds
   "Datum Name n" dx dy dz rw rj rk ds
}
```

Surface Datum Types

In addition to the ellipsoidal datum, the IMAGINE Projection System supports surface datums. There are four types of surface datums:

- Constant

- Raster (or Grid)
- Vector (or Point)
- Multiregression

Of these, the IMAGINE Projection System currently supports the constant, raster, and multiregression datums.

Each one of the three supported datums has its own parameters.

CONSTANT datums provide a constant shift value to apply to the entire area.

```
"Datum Name" SURFACE [BASEDATUM="DatumName"]
{
CONSTANT minlat minlon maxlat maxlon latshift lonshift htshift
}
DESCRIPTION = "transformation description"
```

Where:

SURFACE indicates the type of surface datum.

BASEDATUM indicates the datum that is being used as the base for the shift surface. If no BASEDATUM is declared, the BASEDATUM defaults to the datum with the same name as the spheroid. Any other base datum must be declared and predefined based upon the same base spheroid. The BASEDATUM can be either an ellipsoidal or surface datum.

CONSTANT indicates that a constant shift is performed on the BASEDATUM.

minlat provides the minimum latitude value of the bounding box in decimal degrees.

minlon provides the minimum longitude value of the bounding box in decimal degrees.

maxlat provides the maximum latitude value of the bounding box in decimal degrees.

maxlon provides the maximum longitude value of the bounding box in decimal degrees.

latshift the amount of shift in the latitude direction in decimal seconds.

lonshift the amount of shift in the longitude direction in decimal seconds.

htshift the amount of shift in the elevation in meters.

DESCRIPTION provides a description about the transformation, including the base datum and the source for the shift constants.

RASTER surface datums use information stored in raster data files to provide the transformation information.

The raster files used for the transformations must reside in the <*APOLLO_HOME*>/native/raster folder. No paths can be used to point to these files.

```
"Datum Name" SURFACE [BASEDATUM="DatumName"]
{
RASTER RESAMPLE="Resample_Method"
noDataValue
LATITUDE = "LatFile"
LONGITUDE = "LonFile"
HEIGHT = "HtFile"
}
DESCRIPTION = "transformation description"
```

Where:

SURFACE indicates the type of surface datum.

BASEDATUM indicates the datum that is being used as the base for the shift surface. If no BASEDATUM is declared, the BASEDATUM defaults to the datum with the same name as the spheroid. Any other base datum must be declared and predefined based upon the same base spheroid. The BASEDATUM can be either an ellipsoidal or a surface datum.

RASTER indicates that a raster shift surface based on the BASEDATUM is defined.

RESAMPLE defines the resampling method used on the raster files. The BILINEAR resampling method is the default and is optional. The BICUBIC SPLINE resampling method must be declared.

noDataValue defines the value to substitute for null values in the shift surface files.

LATITUDE indicates the raster file to use as the latitude shift surface. This file is optional.

LONGITUDE indicates the raster file to use as the longitude shift surface. This file is optional.

HEIGHT indicates the raster file to use as the elevation shift surface. This file is optional.

While LAT, LON, and HEIGHT are all optional parameters, at least one of these three must be specified to define a valid surface datum.

DESCRIPTION provides a description about the surface datum, including a more descriptive datum name and the source for the shift files.

MULTIREGRESSION surface datums use an MRE to represent the datum shift surface.

```
"Datum Name" SURFACE [BASEDATUM="DatumName"]
{
REGRESSION minlat minlon maxlat maxlon
A     B     C     D
LATITUDE =  {Vexp_1     Uexp_1     Coeff_1
Vexp_2     Uexp_2     Coeff_2
Vexp_n    Uexp_n    Coeff_n
}
LONGITUDE = {Vexp_1     Uexp_1     Coeff_1
Vexp_2     Uexp_2     Coeff_2
Vexp_n    Uexp_n    Coeff_n
}
Height = {Vexp_1     Uexp_1     Coeff_1
Vexp_2     Uexp_2     Coeff_2
Vexp_n    Uexp_n    Coeff_n
}
} DESCRIPTION = "transformation description"
```

Where:

SURFACE indicates the type of surface datum.

BASEDATUM indicates the datum that is being used as the base for the shift surface. If no BASEDATUM is declared, the BASEDATUM defaults to the datum with the same name as the spheroid. Any other base datum must be declared and predefined based upon the same base spheroid. The BASEDATUM can be either an Ellipsoidal or a surface datum.

REGRESSION indicates that a multiregression transformation will be performed on the BASEDATUM.

minlat provides the minimum latitude value of the bounding box in decimal degrees.

minlon provides the minimum longitude value of the bounding box in decimal degrees.

maxlat provides the maximum latitude value of the bounding box in decimal degrees.

maxlon provides the maximum longitude value of the bounding box in decimal degrees.

A, **B** are the coefficients for the longitude shift. If v is the latitude, V = A * v + B

C, **D** are the coefficients for the latitude shift. If u is the latitude, U = C * u + D

LATITUDE lists the exponents and coefficients in decimal seconds for the MRE. Each row gives the V exponent, the U exponent, and the coefficient for one term in the MRE. The sum of all of these terms is the result of the MRE. This surface is optional.

LONGITUDE lists the exponents and coefficients in decimal seconds for the MRE. Each row gives the V exponent, the U exponent, and the coefficient for one term in the MRE. The sum of all of these terms is the result of the MRE. This surface is optional.

HEIGHT lists the exponents and coefficients in meters for the MRE. Each row gives the V exponent, the U exponent, and the coefficient for one term in the MRE. The sum of all of these terms is the result of the MRE. This surface is optional.

While LAT, LON, and HEIGHT are all optional parameters, at least one of these three must be specified to successfully perform a shift surface transformation.

DESCRIPTION provides a description about the surface datum, including a more descriptive datum name and the source for the shift files.

NOTE: For detailed information on how the ERDAS IMAGINE Projection System works, please see the ERDAS IMAGINE documentation.

Add a Custom SRS

Only datasets with a valid SRS can be crawled in the Data Manager. If a dataset has a custom SRS that is not supported in ERDAS APOLLO, you can add it and then successfully crawl the data.

Follow these steps to add a custom SRS.

1. Add to ERDAS IMAGINE projection system
   - **Modify epsg.plb** on page 16
2. Add to ERDAS APOLLO
   - **Create usersref.xml** on page 18
   - **Modify coordinate_system_category.xml** on page 20
   - **Rebuild and Deploy the Webapps** on page 20
   - **Test the Custom SRS in the Data Manager** on page 21
   - **Test the Custom SRS in the Web Client** on page 22

All of the SRS-related files are located in the cots.srs.jar file located in <APOLLO_HOME>webapps\erdas-apollo\profiles\advantage\WEB-INF\lib\cots-srs-1.3.jar. Unzip the file and browse to \com\ionicsoft\sref\impl\resource\ and the following SRS files are included.

- sref.xml - contains a factory reference location - do not modify this file
- factorysref.xml - default ERDAS SRS file - contains the information for all ERDAS-supported SRSs - do not modify this file

- ionicsref.xml - contains additional SRSs requested by ERDAS customers

- usersref.xml - contains any custom SRSs - this file is not included in the installation so you must create it when you add a custom SRS (see **Create usersref.xml** on page 18)

Be sure to back up all files that you modify.

Projection System Information

All of the SRS information that the ERDAS IMAGINE Projection System needs is contained in the following files.

- `epsg.plb`
- `mapprojections.dat`
- `spheroid.tab`
- `units.dat`
- `sptable.tab`

The first file, `epsg.plb`, is located in this directory:

```
<APOLLO_HOME>\tools\native\raster\etc\Projections
```

The last four files are located in this directory:

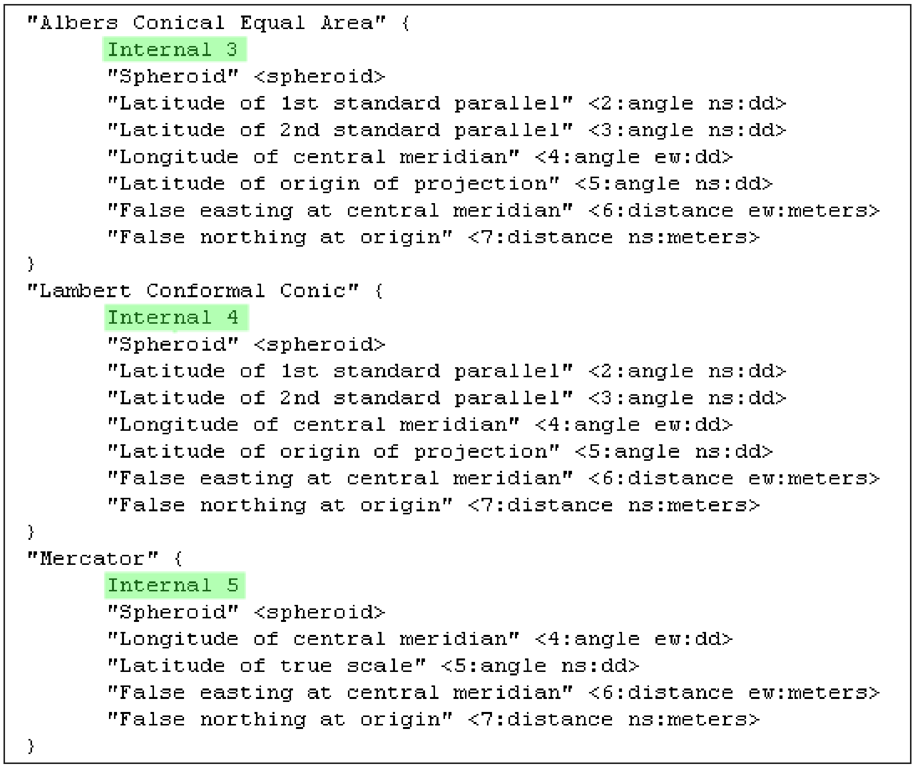
```
<APOLLO_HOME>\tools\native\raster\etc\
```

mapprojections.dat

This file contains definitions of all the most commonly used map projections.

The entries in the `epsg.plb` file must be linked to one of the projection definitions in `mapprojections.dat` using the projection identifiers. You do not need to modify this file to add a custom SRS. This is for advanced procedures not outlined in this chapter.

Figure 4: The Projection Identifiers in MapProjections.Dat

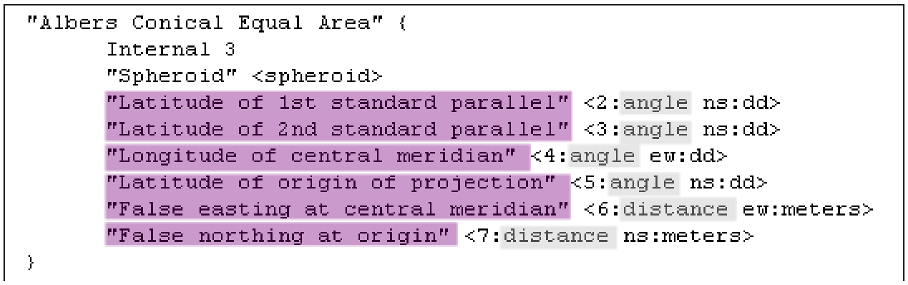


```
"Albers Conical Equal Area" {
       Internal 3
       "Spheroid" <spheroid>
       "Latitude of 1st standard parallel" <2:angle ns:dd>
       "Latitude of 2nd standard parallel" <3:angle ns:dd>
       "Longitude of central meridian" <4:angle ew:dd>
       "Latitude of origin of projection" <5:angle ns:dd>
       "False easting at central meridian" <6:distance ew:meters>
       "False northing at origin" <7:distance ns:meters>
}
"Lambert Conformal Conic" {
       Internal 4
       "Spheroid" <spheroid>
       "Latitude of 1st standard parallel" <2:angle ns:dd>
       "Latitude of 2nd standard parallel" <3:angle ns:dd>
       "Longitude of central meridian" <4:angle ew:dd>
       "Latitude of origin of projection" <5:angle ns:dd>
       "False easting at central meridian" <6:distance ew:meters>
       "False northing at origin" <7:distance ns:meters>
}
"Mercator" {
       Internal 5
       "Spheroid" <spheroid>
       "Longitude of central meridian" <4:angle ew:dd>
       "Latitude of true scale" <5:angle ns:dd>
       "False easting at central meridian" <6:distance ew:meters>
       "False northing at origin" <7:distance ns:meters>
}
```

The definitions in `mapprojections.dat` also indicate which parameters you need to supply in the `epsg.plb` for an SRS associated with the projection.

In the figure below, the items highlighted in purple are the parameters you must provide in an `epsg.plb` entry for an SRS associated with the Albers Conical Equal Area projection. The items highlighted in silver indicate the type of measurement for each item.

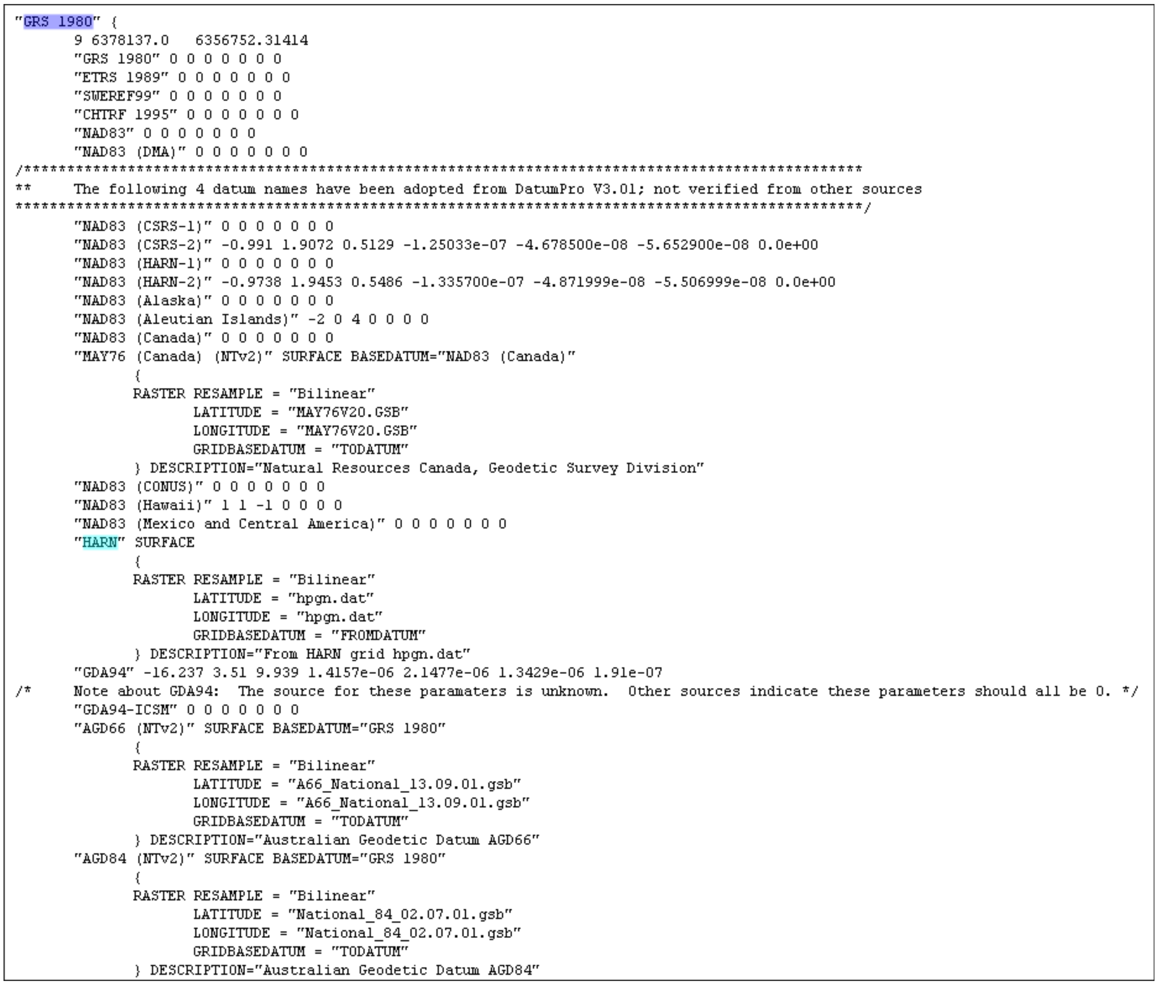
Figure 5: The Projection Parameters in MapProjections.dat



```
"Albers Conical Equal Area" {
       Internal 3
       "Spheroid" <spheroid>
       "Latitude of 1st standard parallel" <2:angle ns:dd>
       "Latitude of 2nd standard parallel" <3:angle ns:dd>
       "Longitude of central meridian" <4:angle ew:dd>
       "Latitude of origin of projection" <5:angle ns:dd>
       "False easting at central meridian" <6:distance ew:meters>
       "False northing at origin" <7:distance ns:meters>
}
```

spheroid.tab

This file contains a mathematical definitions of the most commonly used spheroids, along with definitions of the datums that are most commonly used with the spheroid. You do not need to modify this file to add a custom SRS. This is for advanced procedures not outlined in this chapter.



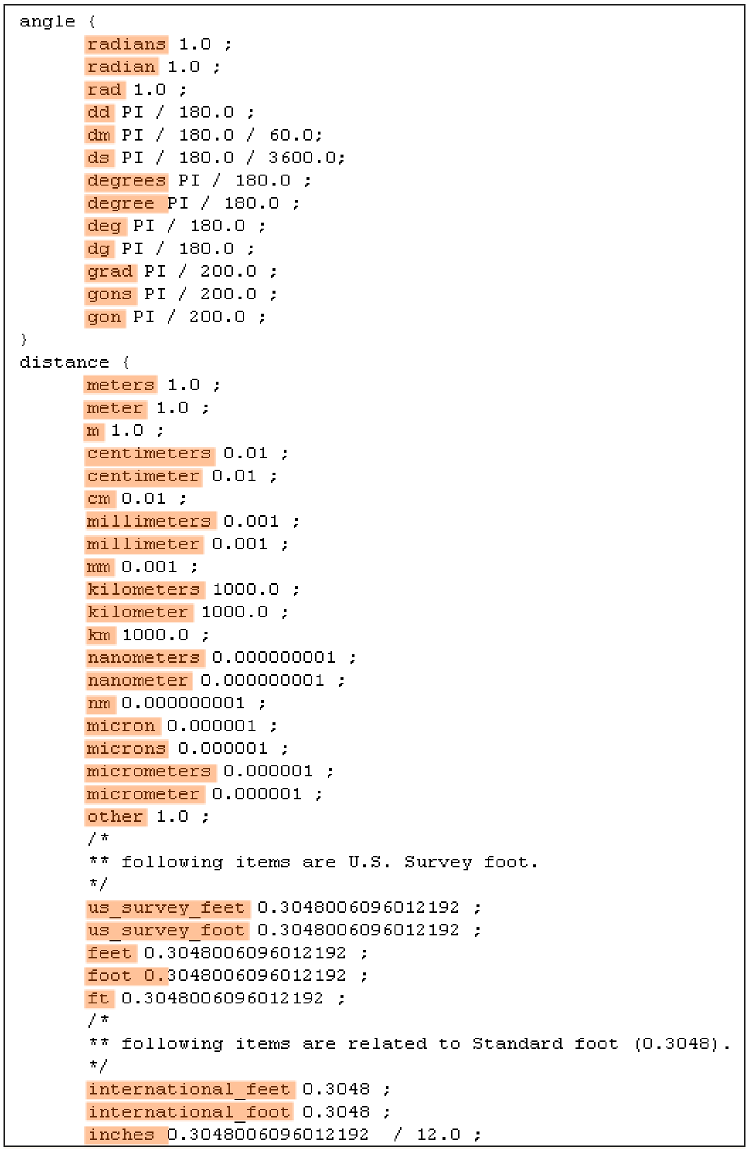
```
"GRS 1980" {
        9 6378137.0   6356752.31414
        "GRS 1980" 0 0 0 0 0 0 0
        "ETRS 1989" 0 0 0 0 0 0 0
        "SWEREF99" 0 0 0 0 0 0 0
        "CHTRF 1995" 0 0 0 0 0 0 0
        "NAD83" 0 0 0 0 0 0 0
        "NAD83 (DMA)" 0 0 0 0 0 0 0
/**************************************************************************************************
**      The following 4 datum names have been adopted from DatumPro V3.01; not verified from other sources
**************************************************************************************************/
        "NAD83 (CSRS-1)" 0 0 0 0 0 0 0
        "NAD83 (CSRS-2)" -0.991 1.9072 0.5129 -1.25033e-07 -4.678500e-08 -5.652900e-08 0.0e+00
        "NAD83 (HARN-1)" 0 0 0 0 0 0 0
        "NAD83 (HARN-2)" -0.9738 1.9453 0.5486 -1.335700e-07 -4.871999e-08 -5.506999e-08 0.0e+00
        "NAD83 (Alaska)" 0 0 0 0 0 0 0
        "NAD83 (Aleutian Islands)" -2 0 4 0 0 0 0
        "NAD83 (Canada)" 0 0 0 0 0 0 0
        "MAY76 (Canada) (NTv2)" SURFACE BASEDATUM="NAD83 (Canada)"
                {
                RASTER RESAMPLE = "Bilinear"
                        LATITUDE = "MAY76V20.GSB"
                        LONGITUDE = "MAY76V20.GSB"
                        GRIDBASEDATUM = "TODATUM"
                } DESCRIPTION="Natural Resources Canada, Geodetic Survey Division"
        "NAD83 (CONUS)" 0 0 0 0 0 0 0
        "NAD83 (Hawaii)" 1 1 -1 0 0 0 0
        "NAD83 (Mexico and Central America)" 0 0 0 0 0 0 0
        "HARN" SURFACE
                {
                RASTER RESAMPLE = "Bilinear"
                        LATITUDE = "hpgn.dat"
                        LONGITUDE = "hpgn.dat"
                        GRIDBASEDATUM = "FROMDATUM"
                } DESCRIPTION="From HARN grid hpgn.dat"
        "GDA94" -16.237 3.51 9.939 1.4157e-06 2.1477e-06 1.3429e-06 1.91e-07
/*      Note about GDA94:  The source for these paramaters is unknown.  Other sources indicate these parameters should all be 0. */
        "GDA94-ICSM" 0 0 0 0 0 0 0
        "AGD66 (NTv2)" SURFACE BASEDATUM="GRS 1980"
                {
                RASTER RESAMPLE = "Bilinear"
                        LATITUDE = "A66_National_13.09.01.gsb"
                        LONGITUDE = "A66_National_13.09.01.gsb"
                        GRIDBASEDATUM = "TODATUM"
                } DESCRIPTION="Australian Geodetic Datum AGD66"
        "AGD84 (NTv2)" SURFACE BASEDATUM="GRS 1980"
                {
                RASTER RESAMPLE = "Bilinear"
                        LATITUDE = "National_84_02.07.01.gsb"
                        LONGITUDE = "National_84_02.07.01.gsb"
                        GRIDBASEDATUM = "TODATUM"
                } DESCRIPTION="Australian Geodetic Datum AGD84"
```

units.dat

This file contains all of the units of measure most commonly used for coordinate systems and projections.

The file is divided into different blocks for all of the things that need to be measured, such as angles, distance, and area.

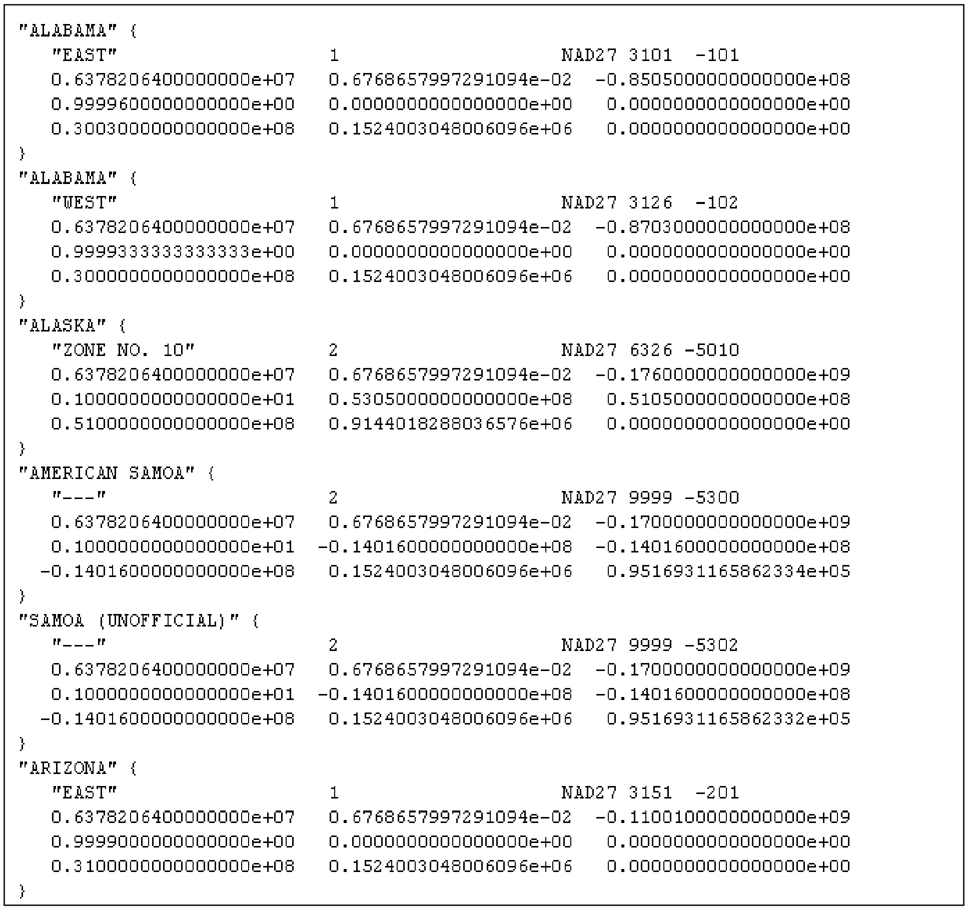
Each block contains a list of units that ERDAS IMAGINE can recognize, along with a conversion factor that is used internally by the software. You do not need to modify this file to add a custom SRS. This is for advanced procedures not outlined in this chapter.



```
angle {
        radians 1.0 ;
        radian 1.0 ;
        rad 1.0 ;
        dd PI / 180.0 ;
        dm PI / 180.0 / 60.0;
        ds PI / 180.0 / 3600.0;
        degrees PI / 180.0 ;
        degree PI / 180.0 ;
        deg PI / 180.0 ;
        dg PI / 180.0 ;
        grad PI / 200.0 ;
        gons PI / 200.0 ;
        gon PI / 200.0 ;
}
distance {
        meters 1.0 ;
        meter 1.0 ;
        m 1.0 ;
        centimeters 0.01 ;
        centimeter 0.01 ;
        cm 0.01 ;
        millimeters 0.001 ;
        millimeter 0.001 ;
        mm 0.001 ;
        kilometers 1000.0 ;
        kilometer 1000.0 ;
        km 1000.0 ;
        nanometers 0.000000001 ;
        nanometer 0.000000001 ;
        nm 0.000000001 ;
        micron 0.000001 ;
        microns 0.000001 ;
        micrometers 0.000001 ;
        micrometer 0.000001 ;
        other 1.0 ;
        /*
        ** following items are U.S. Survey foot.
        */
        us_survey_feet 0.3048006096012192 ;
        us_survey_foot 0.3048006096012192 ;
        feet 0.3048006096012192 ;
        foot 0.3048006096012192 ;
        ft 0.3048006096012192 ;
        /*
        ** following items are related to Standard foot (0.3048).
        */
        international_feet 0.3048 ;
        international_foot 0.3048 ;
        inches 0.3048006096012192  / 12.0 ;
```

sptable.tab

This file contains definitions for the State Plane coordinate systems that are used in the Unites States of America. You do not need to modify this file to add a custom SRS. This is for advanced procedures not outlined in this chapter.



```
"ALABAMA" {
   "EAST"                  1                   NAD27 3101  -101
   0.6378206400000000e+07  0.6768657997291094e-02  -0.8505000000000000e+08
   0.9999600000000000e+00  0.0000000000000000e+00   0.0000000000000000e+00
   0.3003000000000000e+08  0.1524003048006096e+06   0.0000000000000000e+00
}
"ALABAMA" {
   "WEST"                  1                   NAD27 3126  -102
   0.6378206400000000e+07  0.6768657997291094e-02  -0.8703000000000000e+08
   0.9999333333333333e+00  0.0000000000000000e+00   0.0000000000000000e+00
   0.3000000000000000e+08  0.1524003048006096e+06   0.0000000000000000e+00
}
"ALASKA" {
   "ZONE NO. 10"           2                   NAD27 6326 -5010
   0.6378206400000000e+07  0.6768657997291094e-02  -0.1760000000000000e+09
   0.1000000000000000e+01  0.5305000000000000e+08   0.5105000000000000e+08
   0.5100000000000000e+08  0.9144018288036576e+06   0.0000000000000000e+00
}
"AMERICAN SAMOA" {
   "---"                   2                   NAD27 9999 -5300
   0.6378206400000000e+07  0.6768657997291094e-02  -0.1700000000000000e+09
   0.1000000000000000e+01 -0.1401600000000000e+08  -0.1401600000000000e+08
  -0.1401600000000000e+08  0.1524003048006096e+06   0.9516931165862334e+05
}
"SAMOA (UNOFFICIAL)" {
   "---"                   2                   NAD27 9999 -5302
   0.6378206400000000e+07  0.6768657997291094e-02  -0.1700000000000000e+09
   0.1000000000000000e+01 -0.1401600000000000e+08  -0.1401600000000000e+08
  -0.1401600000000000e+08  0.1524003048006096e+06   0.9516931165862332e+05
}
"ARIZONA" {
   "EAST"                  1                   NAD27 3151  -201
   0.6378206400000000e+07  0.6768657997291094e-02  -0.1100100000000000e+09
   0.9999000000000000e+00  0.0000000000000000e+00   0.0000000000000000e+00
   0.3100000000000000e+08  0.1524003048006096e+06   0.0000000000000000e+00
}
```

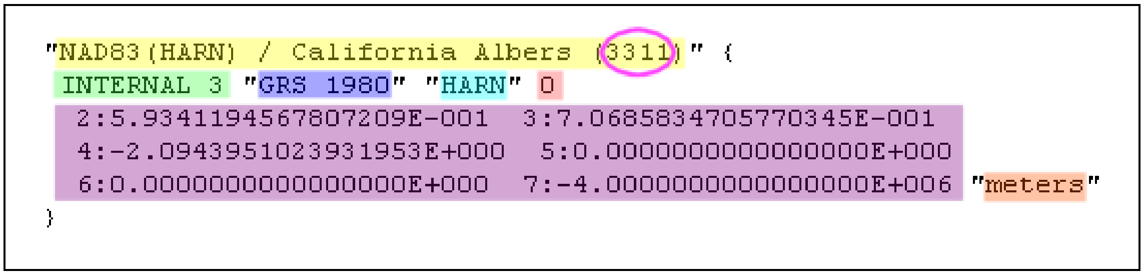
Modify epsg.plb

The file `epsg.plb` contains all of the SRS definitions for the ERDAS IMAGINE Projection System. To add a new SRS to your ERDAS IMAGINE Projection System, create an entry for it in this file.

This entry will make use of information that is specified in the other four files (see above) that are used by the ERDAS IMAGINE Projection System.

The following figure shows an example of an SRS definition in the `epsg.plb` file. The following table describes the elements of the definition.

Figure 6: Example of an Entry in the File epsg.plb



```
"NAD83(HARN) / California Albers (3311)" {
 INTERNAL 3 "GRS 1980" "HARN" 0
  2:5.9341194567807209E-001  3:7.0685834705770345E-001
  4:-2.0943951023931953E+000  5:0.0000000000000000E+000
  6:0.0000000000000000E+000  7:-4.0000000000000000E+006 "meters"
}
```

Element Color	Element Name	Description
1	SRS Name	The name of the SRS as it will be displayed in the metadata.
2	EPSG Code	The EPSG code for this SRS. You can check the EPSG Geodetic Parameter Registry at http://www.epsg-registry.org/ to verify that your number is unique.
3	Projection Identifier	Identifies the projection in the **sptable.tab** file that matches the SRS you are adding.
4	Spheroid Name	The name of the spheroid used for this projection. This spheroid must have an entry in the **spheroid.tab** file.
5	Datum Name	The name of the datum used for this projection. This datum must have en entry in the **spheroid.tab** file. In the spheroid's definition in the spheroid.tab file, there is a list of the commonly associated datums. Find the name of the datum used for this SRS. Replace **DATUM** in the `epsg.plb` entry with the name of that datum.
6	Zone Number	This is applicable only to the UTM and State Plane projections and is specified in **sptable.tab**. For everything else, this should be zero.
7	Parameters	The parameters for this SRS, such as false easting, false northing, longitude of the central meridian, etc. The file **sptable.tab** indicates which parameters must be specified for this type of projection.
8	Units	The unit of measurement used in the coordinate system. This unit must have an entry in the file **units.dat**.

1. Navigate to <APOLLO_HOME>\tools\native\raster\etc\projections\epsg.plb

2. Open the file and add the projection parameters according to the structure described in **Figure 6**. The parameters will vary depending on which projection the SRS is associated with. To find out which parameters are required for your SRS, open the file **sptable.tab**.

   - Each parameter in the `epsg.plb` file is preceded by a number and a colon, and the parameters are always numbered starting with **2**.

   - The parameter values must be entered in (scientific) E notation and require 17 significant digits and 3 digits for the exponent.

     For example:
     .006789247 should be entered as 6.7892470000000000E-003
     12,450,000 should be entered as 1.2450000000000000E+007

   - If the parameter represents an angle measurement, the value must be given in RADIANS, not degrees.

Create usersref.xml

This file is not included with the installation. You must create it to add a custom SRS. This file defines the datum, spheroid, and the custom projection parameters.

1. Open any text editor and type the following lines into the new file.

```
<?xml version="1.0" encoding="utf-8" ?>

  <SREF>



  </SREF>
```

2. Save the new file as `usersref.xml`.

3. Open the file `ionicsref.xml` and find an entry that uses the same projection as the SRS you want to create.

4. Copy the entire entry beginning with a **<PROJCS ID..** tag and ending with the tag **</PROJCS>**.

5. Paste the entry into the `usersref.xml` file, between the **SREF** and **/SREF** tags. Use this as a template to define your custom SRS.

6. Change the **PROJCS ID** number to the EPSG number you are using for this custom SRS.

7. Change the **NAME** value to the name you are using for the custom SRS.

8. The **UNIT ID** is a code that represents the units used in the coordinate system (feet, metres, kilometers, etc.) The valid codes are located at the top of the `factorysref.xml` file.

9. The **GEOCS ID** represents the datum or spheroid used for the custom SRS. The valid codes are located in the `factorysref.xml` file.

10. The parameters vary depending on the type of projection you are adding. You can get the proper values for the parameters by referring back to the parameters that you entered in the `epsg.plb` file.

The values for the parameters are always given in RADIANS in the `epsg.plb` file. They must always be indicated in ARC SECONDS in the `usersref.xml` file.

9. Save and close this `usersref.xml` file.

If the usersref.xml file will be used with another program such as ERDAS APOLLO Style Editor or the command-line tool, copy it to a location with the com\ionicsoft\sref\impl\resource structure and add the root of that structure to CLASSPATH.

Integrate usersref.xml

Because ERDAS APOLLO is an enterprise software system with different components in different locations, you add the usersref.xml in several different places.

Use any advanced zip utility to open the archive files.

1. Copy the newly created usersref.xml to the .jar file in the following location.

```
<APOLLO_HOME>\webapps\erdas-apollo\profiles\advantage\WEB-INF\lib\cots-srs-
1.3.jar\com\ionicsoft\sref\impl\resource\
```

2. For the Web Client, copy the newly created usersref.xml to the .jar file in the following location.

```
<APOLLO_HOME>\webapps\apollo-client\default\WEB-INF\lib\cots-srs-
1.3.jar\com\ionicsoft\sref\impl\resource\
```

3. For the Data Manager, the archive file is in another archive file. First open the following archive file.

```
<APOLLO_HOME>\configuration\org.eclipse.osgi\bundles\7\1\.cp\lib\cots-srs.jar
```

4. Copy the newly created usersref.xml to the second .jar file in the following location.

```
cots-srs.jar\com\ionicsoft\sref\impl\resource\
```

Modify coordinate_system_category.xml

Next, add the new SRS to a category in the coordinate_system_category file.

Use any advanced zip utility to open the archive files.

1. Edit the coordinate_system_category xml file in three places. Add the references to the custom projection (two custom SRSs are shown in the example shown below) at the end of the file in each location.

```
<crs:category name="Custom">
   <crs:code name="MGI AT-Styria 31 / Bessel" value="EPSG:44431" />
   <crs:code name="MGI AT-Styria 34 / Bessel" value="EPSG:44434" />
</crs:category>
```

2. Navigate to:

```
<APOLLO_HOME>\webapps\erdas-apollo\profiles\advantage\WEB-INF\lib\cots-srs-
1.3.jar\com\ionicsoft\sref\impl\resource\
```

open the archive file and make the changes to coordinate_system_category xml as shown in the example in **step 1.**

3. For the Web Client, navigate to:

```
<APOLLO_HOME>\webapps\apollo-client\default\WEB-INF\lib\cots-srs-
1.3.jar\com\ionicsoft\sref\impl\resource\
```

open the archive file and make the changes to coordinate_system_category xml as shown in the example in **step 1.**

4. For the Data Manager, the archive file is in another archive file. First open the following archive file.

```
\configuration\org.eclipse.osgi\bundles\7\1\.cp\lib\cots-srs.jar
```

5. Open the second archive file and make the changes to coordinate_system_category xml as shown in the example listed in **step 1.**

```
cots-srs.jar\com\ionicsoft\sref\impl\resource\
```

Rebuild and Deploy the Webapps

After you create and modify the files as described above, you must rebuild the ERDAS APOLLO webapps and deploy them in JBoss.

1. Stop JBoss.
2. Ant jboss. Open a command line windows and type:

```
Cd <APOLLO_HOME> [enter]

<APOLLO_HOME>\tools\ant\bin\ant jboss
```

3. Wait until the build completes.

4. Deploy the new webapps in JBoss.

   - Navigate to <APOLLO_HOME>\dist\jboss and copy erdas-apollo.ear

   - Navigate to `<APOLLO_HOME>\jboss\server\default\deploy` and back up erdas-apollo.ear

   - Copy erdas-apollo.ear from the \dist folder to the default/deploy folder.

5. Delete the JBoss cache by deleting the `<APOLLO_HOME>\jboss\server\default\work` folder.

6. Start JBoss.

Test the Custom SRS in the Data Manager

Follow these instructions to make sure that your custom SRS is added.

1. Launch the ERDAS APOLLO Data Manager.

2. In the Explorer View, right-click on the ROOT node and create an aggregate.

3. Locate the property **Default SRS** and click on it.

4. Click the [...] button that appears in the **Value** column. The Spatial Reference Systems dialog box opens.

5. Locate your custom SRS in the list on this dialog, highlight it, and click **OK**.

6. Click Save 💾 from the tool bar.

7. Click the Crawl 🕷 icon to create a crawler that will load data into your ERDAS APOLLO Catalog.

8. When you are asked to specify the directory to crawl, select the directory that contains your custom SRS imagery.

   If the crawler still does not recognize the dataset, check the parameters in the `usersref.xml` file or in the imagery.

Test the Custom SRS in the Web Client

Follow these instructions to make sure that your custom SRS is added.

1. Launch the ERDAS APOLLO Web Client.
2. Click the SRS box on the Info Bar above the map.
3. Select **-Select-**. The Spatial Reference Systems dialog box opens.
4. Look for your custom SRS in the list on the Spatial Reference Systems dialog.