



# Upgrading InQuira

---

## *A Guide to Upgrading InQuira Components*

---

InQuira Version 8.2.3

Document Number UP82-AG-02

January 7, 2011

**InQuira, Inc.**

900 Cherry Ave., 6<sup>th</sup> Floor  
San Bruno, CA 94066

### **COPYRIGHT INFORMATION**

Copyright © 2002 - 2010 InQuira, Inc.  
Product Documentation Copyright © 2003 - 2010 InQuira, Inc.

### **RESTRICTED RIGHTS**

This document is incorporated by reference into the applicable license agreement between your organization and InQuira, Inc. This software and documentation is subject to and made available only pursuant to the terms of such license agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy, modify, disassemble or reverse engineer the software and documentation, except as specifically allowed in the license agreement and InQuira will take all necessary steps to protect its interests in the software and documentation. To the extent certain third party programs may be embedded into the InQuira software, you agree that the licensors for such third party programs retain all ownership and intellectual property rights to such programs, such third party programs may only be used in conjunction with the InQuira software, and such third party licensors shall be third party beneficiaries under the applicable license agreement in connection with your use of such third party programs.

The information in this document is subject to change without notice and does not represent a commitment on the part of InQuira, Inc. The documentation is provided "AS IS" without warranty of any kind including without limitation, any warranty of merchantability or fitness for a particular purpose. Further, InQuira, Inc. does not warrant, guarantee, or make any representations regarding the use, or the results thereof. Although reasonable measures have been taken to ensure validity, the information in this document is not guaranteed to be accurate or error free.

### **TRADEMARKS AND SERVICE MARKS**

InQuira, Inc., InQuira 8, InQuira 7, InQuira 6, InQuira 5, InQuira Natural Interaction Engine, Intelligent Search, Information Manager, iConnect, and Call Center Advisor are trademarks or registered trademarks of InQuira, Inc. Siebel 7, Siebel Call Center 7, and all other Siebel products are the property of Siebel Systems, Inc. Sentry Spelling-Checker Engine Copyright © 2000 Wintertree Software, Inc.

All other trademarks and registered trademarks contained herein are the property of their respective owners.

---

# Contents

---

<b>Preface</b>	<b>About This Guide</b> .....	<b>1</b>
	In This Guide .....	1
	Contacting InQira .....	2
	InQira Product Documentation .....	2
	InQira Analytics Documentation .....	3
	InQira iConnect Documentation .....	3
	InQira Information Manager Documentation .....	3
	InQira Intelligent Search Documentation .....	5
	Screen and Text Representations .....	6
	References to World Wide Web Resources .....	6
<b>Chapter 1</b>	<b>Upgrading Intelligent Search</b> .....	<b>7</b>
	Upgrading from InQira 8.0.4.22 to InQira 8.2.3 .....	7
	Upgrade from 8.0.x to 8.0.4.22 .....	8
	Upgrade from 8.0.4.22 to 8.2 .....	13
	Upgrading from InQira 8.2.x to InQira 8.2.3 .....	28
	Upgrading from InQira 8.1.x to InQira 8.2.3 .....	29
	Post Upgrade Instructions for Windows OS .....	30
	Post Upgrade Instructions for 8.1.3.x Upgrades .....	30
<b>Chapter 2</b>	<b>Upgrading Information Manager</b> .....	<b>31</b>
	Before Upgrading from 8.2.2 to 8.2.3 .....	31
	Upgrading from a Release Prior to InQira 7.3 .....	35
	Upgrading from InQira 7.3 and Later Releases .....	36
	Completing the Upgrade Process .....	37
	Post Upgrade Notes .....	37
	Change Notification Update .....	38
	OpenJMS Removal .....	38
	Jgroups Configuration .....	39

InQuira Client Library API Interface .....	39
Reinstall Services on Windows OS .....	40
Information Manager Collection Configuration Files .....	40
InfoCenter Upgrade .....	40
Post-Upgrade Security Modification .....	42
IMUpgrader Utility Overview .....	42
OpenJMS Uninstallation Tasks .....	43
IMUpgrader GUI Operation .....	44
IMUpgrader Command Line Operation .....	44
Limitations .....	46
IMUpgrader Troubleshooting .....	46
IMUpgrade Fails to Upgrade Database on Linux .....	47
Resolve Conflicting .jar Files in IM Web Applications .....	47
<b>Chapter 3 Upgrading Analytics .....</b>	<b>50</b>
Upgrading Analytics from Version 7.x to 8.0 .....	51
Upgrading Intelligent Search Analytics from 7.x to 8.0 .....	51
Upgrading Information Manager Analytics from 7.x to 8.0 .....	53
Upgrading Analytics from Version 8.0.x to 8.2.3 .....	54
Schema Tables .....	55
Processing ETL Log Files Created in Previous Releases .....	61

---

---

# About This Guide

This guide is intended for technical staff who are responsible for upgrading InQuira from a prior release to 8.2.3. This guide covers only the process for upgrading a “plain vanilla” installation of InQuira. It does not cover processes for upgrading any custom work done to your installation.

**Important!** All changes should be applied and thoroughly tested in a development environment prior to being implemented in production. Also be sure to back up your database prior to upgrading to mitigate risk of data loss.

More information about specific features and defect resolution for each release can be found in Release Notes for your point release.

## In This Guide

The *InQuira Upgrade Guide* is divided into the following sections:

[Chapter 1, Upgrading Intelligent Search](#)

This chapter describes the upgrade process for the InQuira Intelligent Search component.

[Chapter 2, Upgrading Information Manager](#)

This chapter describes the upgrade process for the InQuira Information Manager component.

[Chapter 3, Upgrading Analytics](#)

This chapter describes the upgrade process for the InQuira Analytics component.

---

## Contacting InQira

You can contact InQira by mail, telephone, fax, and email.

<b>Address:</b>	900 Cherry Ave. 6th Floor San Bruno, CA 94066
<b>Telephone:</b>	(650) 246-5000
<b>Fax:</b>	(650) 264-5036
<b>Email:</b>	For sales information, send email to <a href="mailto:sales@inquire.com">sales@inquire.com</a> . For product support, send email to <a href="mailto:support@inquire.com">support@inquire.com</a> .
<b>World Wide Web:</b>	Learn more about InQira products, solutions, services, and support on the world wide web at: <a href="http://www.inquire.com">www.inquire.com</a> .

## InQira Product Documentation

InQira documentation is available only to licensed users of our software products and may not be redistributed in any form without express permission from InQira, Inc.

The InQira documentation is available in PDF format. Customers can download the PDF files from:

<http://documentation.inquire.com/>

**Note:** You need a PDF reader application installed on each processor on which you plan to view the InQira product documentation. The Adobe Acrobat reader is available from Adobe Systems at: <http://www.adobe.com>.

If you encounter a problem, need help using the documentation, or want to report an error in the content, please contact InQira Customer Support.

If you need help obtaining InQira product documentation, or want to obtain permission to redistribute a portion of the contents, please contact your InQira account representative.

Detailed information about each product document set is available in:

- “InQira Analytics Documentation” on page 3
- “InQira iConnect Documentation” on page 3
- “InQira Information Manager Documentation” on page 3
- “InQira Intelligent Search Documentation” on page 5

## InQira Analytics Documentation

InQira Analytics is distributed with the following documentation.

Document	Number	Description
<b>InQira Analytics Installation Guide</b>	IA80-IG-00	This guide is intended for technical staff who are responsible for installing InQira Analytics. It provides detailed information on installing and configuring the InQira Analytics product for use with an InQira application.
<b>Analytics User Guide</b>	IA80-CA-00	This guide is intended for systems and application administrators who need to configure the Intelligent Search and Information Manager Analytics components to report on InQira application performance.

## InQira iConnect Documentation

The InQira iConnect products are distributed with the following documentation.

Document	Number	Description
<b>iConnect Developers Guide</b>	CA80-IG-01	This guide is intended for application developers and systems administrators who need to plan for and integrate the InQira iConnect with an InQira application and a supported CRM application.
<b>iConnect for Siebel Contact Center Integration Guide</b>	CA80-IG-00	This guide is intended for application developers and systems administrators who need to plan for and integrate the InQira Contact Center Advisor with an InQira application and a supported CRM application.

## InQira Information Manager Documentation

InQira Information Manager is distributed with the following documentation.

Document	Number	Description
<b>Information Manager Installation Guide</b>	IM80-IG-00	This guide is intended for technical staff who are responsible for installing InQira Information Manager. It provides detailed information on installing and configuring the Information Manager product.

---

Document (Continued)	Number	Description (Continued)
<b>Information Manager Administration Guide</b>	IM80-CA-00	This guide is intended for systems and application administrators who need to configure and administer an InQira Information Manager application, and integrate it with an InQira application. It also contains information for general business users who need to use the Information Manager to create and manage content.
<b>Information Manager Content Authoring Guide</b>	IM80-AG-00	This guide is intended for technical staff who are responsible for authoring content in InQira Information Manager. It provides detailed information on creating content and managing workflow tasks in the Information Manager console.
<b>Information Manager Developer's Guide</b>	IM80-WSR-00	This guide is intended for application developers who need to integrate Information Manager content, content category, and user and security functions with external applications. It contains reference information and examples for all packages, classes, methods, and interfaces of the Information Manager Web Services API.

## InQuira Intelligent Search Documentation

Intelligent Search is distributed with the following documentation.

<b>Document</b>	<b>Number</b>	<b>Description</b>
<b>Intelligent Search Installation Guide</b>	IS80-IG-00	This guide is intended for technical staff who are responsible for installing InQuira. It provides detailed information on installing InQuira and configuring the application on a single processor using the Installation Configuration Environment facility.
<b>Intelligent Search Administration Guide</b>	IS80-CA-00	This guide is intended for system and application administrators who need to configure an InQuira application in an enterprise environment. It describes InQuira integration, development, configuration, and maintenance processes and tasks.
<b>Intelligent Search Language Administration Guide</b>	IS80-LA-00	This guide is intended for business users and subject matter experts who need to create and maintain the language processing elements of a InQuira application using the System Manager. This book provides usage information about the System Manager, conceptual information about the InQuira language objects, and task information about the process of managing the user experience provided by the InQuira application.
<b>Intelligent Search Language Tuning Guide</b>	IS80-LD-00	This guide is intended for application developers who need to create and maintain advanced InQuira language-processing elements using the Dictionary and other InQuira Language Workbench applications.
<b>Intelligent Search Optimization Guide</b>	IS80-AG-00	This guide is intended for application developers who need to implement InQuira advanced features, including Personalized Navigation and Process Wizards.
<b>Intelligent Search Application Development Guide</b>	IS80-API-00	This guide provides information about integrating and customizing the InQuira Personalized Response User Interface.
<b>Intelligent Search Language Reference</b>	IS80-LRG-00	This guide is for language developers implementing InQuira applications that utilize the intent libraries and advanced language processing functions. These guides are published as separate documents that provide reference information for each industry-specific intent library. Each reference also contains complete descriptions of InQuira Match Language and Variable Instantiation Language.

---

Document (Continued)	Number	Description (Continued)
<b>Intelligent Search User Interface Guide</b>	IS80-UI-00	This guide is intended for application developers who need to customize the InQuira Personalized Response User Interface, and integrate it with a production web application. It contains information about the elements and features of the User Interface, and provides guidelines for integrating it into an enterprise web architecture, customizing its appearance and functionality, and implementing various special features.

## Screen and Text Representations

The product screens, screen text, and file contents depicted in the documentation are examples. We attempt to convey the product's appearance and functionality as accurately as possible; however, the actual product contents and displays may differ from the published examples.

## References to World Wide Web Resources

For your convenience, we refer to Uniform Resource Locators (URLs) for resources published on the World Wide Web when appropriate. We attempt to provide accurate information; however, these resources are controlled by their respective owners and are therefore subject to change at any time.

# Upgrading Intelligent Search

This chapter provides instructions for:

- **Upgrading from InQuira 8.0.4.22 to InQuira 8.2.3**
- **Upgrading from InQuira 8.2.x to InQuira 8.2.3**
- **Upgrading from InQuira 8.1.x to InQuira 8.2.3**
- **Post Upgrade Instructions for Windows OS**
- **Post Upgrade Instructions for 8.1.3.x Upgrades**

## Upgrading from InQuira 8.0.4.22 to InQuira 8.2.3

This chapter provides instructions for the supported method of upgrading a dictionary from version 8.0.4.22, the SLD (Single-Language Dictionary), to version 8.2.3, the MLD (Multi-Language Dictionary). This procedure ensures that dictionary objects are converted to the new formats, and that concept references remain valid. This section includes the following three steps and associated substeps:

- 1 Upgrade from 8.0.x to 8.0.4.22**
  - a Upgrade Dictionary Objects to 8.0.4.22**
  - b Export Desired Elements from 8.0.4**
- 2 Upgrade from 8.0.4.22 to 8.2**
  - a Install InQuira 8.2**
  - b Prepare the 8.2 Dictionary**
  - c Import Concepts**
  - d Check Concept and Cluster Domains**
  - e Perform First Dictionary Validation**
  - f Import Taxonomy**
  - g Import Process Wizards**
  - h Import Non-Ontology Objects**

- i Perform Second Dictionary Validation**
  - j Conduct Concept Name Cleanup**
  - k Perform Third Dictionary Validation**
  - l Conduct Concept Relationship Checks**
  - m Update Spell Check Changes**
  - n Update Quality Monitor Test Profiles**
  - o Add Functional Improvements**
- 3 Upgrading from InQuira 8.2.x to InQuira 8.2.3**

## Upgrade from 8.0.x to 8.0.4.22

Use the information in the following sections to:

- **Upgrade Dictionary Objects to 8.0.4.22**
- **Export Desired Elements from 8.0.4**

### Upgrade Dictionary Objects to 8.0.4.22

The dictionary to be upgraded must be on version 8.0.4.22, as this version has the necessary export formats available. It is not necessary to move the taxonomy to 8.0.4.22; this can be upgraded directly from earlier versions.

If the dictionary is currently on another 8.0.x version, follow these steps to move the dictionary objects to 8.0.4:

- 1** Install 8.0.4, choosing the appropriate industry.
- 2** Replace the 8.0.4 vroot directory in <installation-root>\base\development\content\data with the vroot from your existing 8.0.x installation.
- 3** Copy over any custom context variables (optional).
  - a** Open the latest numbered config file from the 8.0.x installation, found in <installation-root>\instances<instance>\development\content\data\config\default.
  - b** Find the custom context variables; each is contained in a tag like this one: <preferences name="My Context Variable">.
  - c** Open the latest configuration file from the 8.0.4 installation and copy these context variables in before the </preferenceService> tag.

- d Find the custom portlets; <answer-purpose> entries with an index greater than 9.
  - e In the 8.0.4 configuration, insert the custom portlets after the last defined <answer-purpose>. Make certain not to skip any index numbers. Also, be certain that if the custom portlet has a <purpose-template> entry that it points to a valid reference.
- 4 Copy over any process wizards (optional).
    - a Start the 8.0.4 workbench. This creates an applications directory under <installation-root>\instances\<instance>\development\content.
    - b In the dictionary manager, click the process wizard icon; this creates the processWizards directory.
    - c Replace the 8.0.4 processWizard directory in <installation-root>\instances\<instance>\development\content\applications\default\dictionary with the processWizard directory in the existing 8.0.x installation.

**Note:** Copying over context variables, custom portlets, and process wizards in steps 3 and 4 is optional. The advantage is that if you wish to open any rules referencing these objects in the 8.0.4 dictionary during the export process, you will not get an error warning you that they do not exist.

If your dictionary is currently on 7.3, follow the instructions above. Then do the following:

- 1 Open and then close the workbench to create the applications directory.
- 2 From ICE prompt, run this command:

```
run com.inquiria.intents.tools.Upgrade <install-dir>\instances\<instance>\development\content\applications\default\dictionary\rules\users\internal\<user>\rules
```

This command changes the format of the intent response templates in 7.3 to make them compatible with 8.0. Now you should be able to open the workbench again and view your dictionary objects.

## Export Desired Elements from 8.0.4

**Important!** The Search upgrader does not support migrating concepts on an upgrade. There are three options for handling concepts:

- Keep the legacy (pre-upgrade) concepts. See “Export Concepts” on page 10 and “Import Concepts” on page 15 for details on this process.
- Move to the upgraded concepts, losing the legacy (pre-upgrade) concepts. This requires accepting the concepts as delivered with the install/upgrader.

- Use the Language Workbench import/export functionality to migrate specific legacy (pre-upgrade) concepts. Generally this requires Professional Services language expert assistance.

## PERFORM DICTIONARY VALIDATION

For each domain group in use, perform dictionary validation on all objects, using the **Validate Dictionary** option from the Tools menu. Select all objects for validation, but uncheck “list objects not in target domain group”. Ignore any messages relating to installed rules. Fix all other issues, e.g. concept loops and null references within custom rules and intents.

Note that concept validation alerts you when concepts have the same name except for case, e.g. <noun.mycompany.MyWidget> and <noun.mycompany.mywidget>. These are likely mistakes, and should be cleaned up before the upgrade. If two such concepts are exported, the second concept in the file is ignored, which may not produce the desired results. Validation also warns you if a rule or intent references a concept which does not have synonyms. Either add synonyms or remove the concept reference to fix this.

## IDENTIFY AND EXPORT CUSTOM DICTIONARY OBJECTS

The first step is to determine which dictionary objects should be transferred to the new dictionary.

### Export Concepts

Concepts may be exported one of two ways:

- You may select the concepts in the results of a search via the find tool, and use the right-click “Export” option.
- You may create a text file with one concept name per line, using the format <pos>. <domain>:<conceptName>. (Note that the third part of the concept name should be preceded by a colon rather than a dot.) Using the Tools menu, choose the option “Export Concepts”, and select the file of concept names, then the destination directory.

For **concepts**, find the ones which have been either created or modified for the installation by searching on author or last edit date. Do not attempt to export all of the concepts in the dictionary. There is also no need to export any which have been dispreferred. In the find tool, you can choose to retrieve only preferred concepts. If there is some reason you need to export dispreferred concepts, they will be imported to 8.2.x as disabled concepts.

**Note:** Some concepts may not have an edit date. To make sure that you get all desired concepts, search on (1) last edit date, (2) custom author, and (3) custom concept domain name.

After you have found the desired concepts, select all in the find results, right-click, and choose “Export”. (If there are objects other than concepts in your find results, the “Export” option does

not appear in the right-click menu.) Choose the format “MLD format”, and save the files to a folder. If you have multiple exports, you can export to the same folder, and choose to merge rather than overwrite the existing contents. You can also choose to save the log by appending it to a previous log. This makes it easier to track multiple exports.

When you export concepts that have children, those relationships will be lost; only parent relationships are retained in the export files. For this reason, you will be given the option of exporting the children along with the parents, if those children are not already being exported explicitly. If you choose not to do so, the log will contain a list of these child concepts. At this point you may choose to export all or some of the concepts in this list by creating a text file with the concept names and exporting them as described above.

If you try to export a concept with no synonyms, or a reference to a concept with no synonyms, you see a message like the following:

```
WARNING: concept 'verb.contact:delete' typeOf 'verb.change:take_away'  
ignored -- no synonyms
```

Concepts with no synonyms have been “deleted”, so there is no need to make any fixes based on these messages.

For non-English dictionaries:

If you have both English and non-English installations, these will be merged in 8.2.x. You will want to merge concepts which have the same name in the different installations, since concepts in 8.2.x may have multiple synsets in different languages. To do this, identify the concepts whose synonyms you wish to export. Select all, and right-click, choosing “Export Synonyms”, and then the language of those synonyms. You may then export these to the same directory as for English, and they will be imported along with the English synonyms.

### Export Process Wizards

For **process wizards**, import to 8.2.x involves simply pointing directly to the old process wizard directory, `development\content\applications\default\dictionary\processWizards`, so no export is needed.

### Export Users

For workbench users, export as follows:

- 1 Open the User Manager.
- 2 Select the Users to be exported.
- 3 Right-click selection and select “Export to MLD” from popup menu.
- 4 Select Directory for export.

**Note:** Standard Users 'Administrator' and 'im\_synch' are ignored. Any exported or ignored users are logged to the console.

**Note:** Imported dictionary objects other than concepts retain their original timestamp and author information. If you wish to search on these objects by author in the 8.2.x dictionary, you must add those users. For this reason, you want to keep users who have edited objects even if they are no longer using the workbench.

### Export All Other Dictionary Objects

For all other objects, search for each of the six types of non-concept objects one at a time, each time searching only for those edited after the date of the original installation. For each result set, determine which ones have been created or modified for the installation, and of that set, which are still actively needed for the installation.

If there are installed intents or rules being used, it is recommended that you not export them, but instead use the 8.2.x versions of these. Most of these have been improved by changing the IML to be language-neutral and by retesting their accuracy. Also, be aware that many installed search component rules have been removed because they were found to not consistently improve search results. Some of those may have been modified in your installation, but you may find that you do not need them at all.

After you determine which of the objects in the result set you want to export, select them, right-click, and choose “Export for MLD”. (If there are concepts in your find results in addition to the non-concept objects, the “Export for MLD” option does not appear in the right-click menu.) Export each of the result sets to the same directory.

Objects are stored in subdirectories as follows:

- ALIAS\_LIST: Alias Lists
- ANSWER: Intent Responses
- HIERARCHY: Intent Hierarchies
- INTENT: Intents
- PROFILE: User Profiles
- RULE: Rules
- USER: Users

**Note:** If the same object is exported twice, it overwrites the previous version of that object.

## Upgrade from 8.0.4.22 to 8.2

Use the information in the following sections to:

- **Install InQuira 8.2**
- **Prepare the 8.2 Dictionary**
- **Import Concepts**
- **Conduct Ontology Import**
- **Check Concept and Cluster Domains**
- **Perform First Dictionary Validation**
- **Import Taxonomy**
- **Import Process Wizards**
- **Import Non-Ontology Objects**
- **Perform Second Dictionary Validation**
- **Conduct Concept Name Cleanup**
- **Perform Third Dictionary Validation**
- **Conduct Concept Relationship Checks**
- **Update Spell Check Changes**
- **Update Quality Monitor Test Profiles**
- **Add Functional Improvements**

### Install InQuira 8.2

See the *InQuira Intelligent Search Installation Guide* for installation instructions.

### Prepare the 8.2 Dictionary

Make the following customizations, as applicable:

- 1 Copy the `autoimport.xml` file from the `Upgrade.zip` file to each instance folder before restarting the instance.  
As each instance is restarted, the `autoimport.xml` file, which includes the configuration changes, will be overlaid on the current `#.xml` for each instance, creating a new `#.xml`. The file will then be renamed to `autoimport.xml.bak`.
- 2 Open the dictionary manager, and add the users you exported from 8.0, using the “Import Dictionary Objects” option on the Tools menu.

**Note:** The Administrator user has the password "Administrator" so this should be reset if necessary.

- 3 Create at least one concept domain for your custom concepts using the "Edit Domains" option on the Tools menu; it should follow the naming convention "<Site Name> Concepts". If you want only certain concepts to be available to certain domain lists, you can create multiple concept domains, but for most applications a single custom concept domain suffices. Remember that concepts are multilingual, so you do not want to create a language-specific concept domain (e.g. "MyCompany English Concepts").
- 4 If your installation requires the set of approximately 23,000 US city concepts, then import these to 8.2 at this time. Choose "Import Ontology Objects" from the Tools menu, browse to:

```
<install-root>\resources\Dictionary_Import_Files\  
US_City_Concept_Import
```

Click **Import**. Do not import US City concepts from 8.0 unless you have already prepared the 8.2 dictionary with the US city concepts from the 8.2 import sheets, as it will cause errors in the import.

- 5 Recreate any custom domains and domain lists that are still needed. Add your custom domain(s), including the new concept domain(s) to the appropriate domain list(s). (Concept domains do not need to be ordered with respect to other domains.) For example, if you are upgrading an English installation from a Computer dictionary and you previously used the Standard domain list, you now use the English Computer domain list, with the addition of your custom domains.

If you are upgrading dictionaries with multiple languages and/or multiple domain lists, use the following guidelines. Suppose in 8.0 you have an English Telecom dictionary with the following domain lists and custom domains:

- MyTelco Shopping (contains custom domains "MyTelco" and "Shopping")
- MyTelco Business (contains custom domains "MyTelco" and "Business")
- Standard (contains custom domain "MyTelco")

These would correspond to the following domain lists in 8.2.x:

- English MyTelco Shopping
- English MyTelco Business
- English Telecom

To create these domain lists, follow these steps:

- a Create the concept domain "MyTelco Concepts".
- b Create the non-concept domains "MyTelco", "Shopping", and "Business".

- c Edit the domain list "English Telecom", and add the domain "MyTelco" to the end of the list; add the domain "MyTelco Concepts" anywhere in the list (concept domains are not ordered with respect to other domains). Save your changes.
- d Choose "Save as" on the English Telecom domain list, and give it the new name "English MyTelco Shopping". Add the domain "Shopping" to the end of the list.
- e Repeat step d for English MyTelco Business, adding the domain "Business" at the end of the list.
- f If you also had a Spanish Telecom dictionary, you would follow the same steps, adding MyTelco Concepts to each of the Spanish domains, along with any custom non-concept domains, using the "Spanish Telecom" domain list as the basis for the other domain lists.
- g In the Dictionary Manager, add subject maps for your new domain lists; from the Tools menu select "Edit Subject Maps".

Following the previous example, you would add subject maps for "MyTelco Shopping" and "MyTelco Business", with the mappings below:

```
MyTelco Shopping
  English is mapped to English MyTelco Shopping
  Spanish is mapped to Spanish MyTelco Shopping

MyTelco Business
  English is mapped to English MyTelco Business
  Spanish is mapped to Spanish MyTelco Business
```

## Import Concepts

### IMPORT STEPS

Exporting the concepts in MLD format generates three files: concepts.csv, synsets.csv, and clusters.csv. If you have multiple sets of concept files to import, you may import them one at a time, or you may combine the files before import.

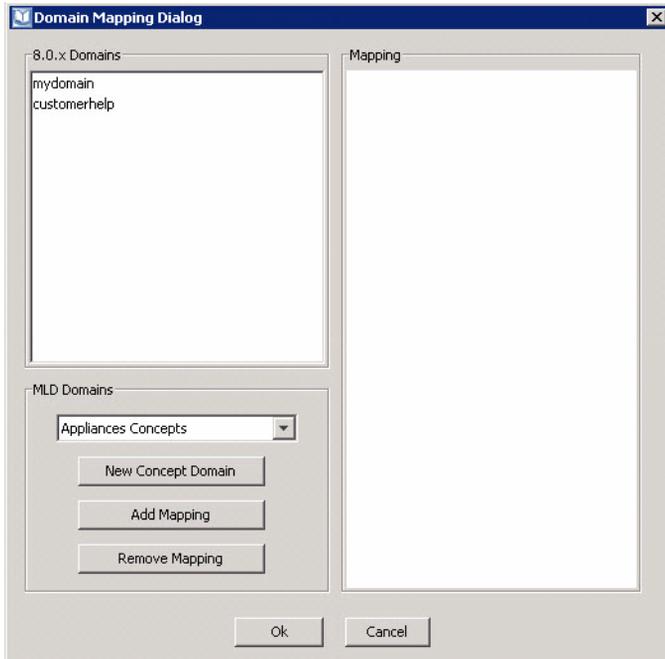
**Important!** You must import all concepts before importing non-ontology objects. Importing the concepts establishes a mapping between old concept names and new concept names. Any non-ontology objects imported from 8.0 (such as rules, intents, alias lists, and the taxonomy) may reference concepts using their old names. Upon import, these concept references are updated using the mapping established during the concept import. Therefore all concepts should be imported first in order to ensure that the mapping is fully established so that references in non-ontology objects are updated correctly. See "Further Explanation of the Concept Update Process" below for more details.

When you first import your concept files, you might find that some adjustments need to be made. Therefore the first iterations of the concept import are test runs.

To import the concept files, go to the Tools menu of the Dictionary Manager, choose "Import Ontology Objects", and point to their directory. The first dialog screen will ask if you only want to check for errors, rather than doing a complete import. Choose "yes". After this you will be prompted to "Select Subject". Choose the industry of the original dictionary you are upgrading.

### Domain Mapping

If there are custom concepts in the import, you see a dialog box prompting you to map old concept domains to new concept domains. For example:



This indicates that there are concepts being imported which are not found in the mapping file, and that these concepts have the 8.0 domains "mydomain" and "customerhelp" (i.e. these are the middle of their three-part names). In general, you want to find the concept domain you created in section 2, and map all of the unrecognized concepts in your import file to that domain.

To do this, select all of the domains in the 8.0.x Domains window, and choose your custom concept domain from the drop down list. (If you have not yet created this domain, you can choose "New Concept Domain" and create it now.) Now click "Add Mapping" and then "OK".

## ERROR MESSAGES

After the domain mapping, the import files are validated. If there are any error messages relating to invalid objects, it is recommended that you fix them before conducting the final concept import.

Below are descriptions of common error messages:

```
ERROR: Cluster 'authorization' -- the following clustermates don't exist
in the ontology or in the Import File: [noun.mycompany.deauthorization]
```

This error occurs if there is a reference (e.g. in a cluster or parent/child relationship) to a concept which cannot be found in the 8.2.x dictionary or in the import files. At the end of the validation, all of these references will be summarized in a sorted list:

The following missing concepts were referenced in the import files:

```
noun.mycompany:deauthorization
noun.mycompany:myWidget
```

Depending on whether the missing concept is important to your installation, you may either (1) remove the reference to the missing concept in the 8.0 concept being exported, or (2) export the missing concept from 8.0, adding it to the existing export directory. If you choose the latter option, you may copy this list of missing concepts to a text file and use that for the export.

```
ERROR: The following concepts were not added, because at least 1 of their
parents were not added.
```

Typically the problem is that the import files contain a loop between concepts, for example, Concept A is a child of itself, or a child of Concept B which is in turn a child of Concept A. These loops are reported during dictionary validation in 8.0, and should be fixed before exporting concepts.

```
ERROR: concepts.csv, line 868 concept 'noun.mycompany.my_credential:8_0
Concepts' was previously defined in this file!
```

If a concept is defined twice in the same file, the second occurrence will be ignored. Be sure that the file contains the intended concept.

## OTHER LOG MESSAGES

### Deprecated concepts

As the ontology has been refined from version to version of InQuira, many concepts have been deprecated. This may be because the concept represents obscure words that are not likely to appear in our customers' content, or because it is redundant with another concept.

If the import files contain references to relationships with concepts which have been deprecated prior to 8.1, then the concepts in those relationships are ignored, and you see a message like the following:

```
Concept 'verb.social.rent1' type of parents -- deprecated concept
'verb.communication.contract' IGNORED.
```

The deprecated concept is not ignored if it is also being imported. That is, if `<verb.communication.contract>` was exported from 8.0, it is also imported into 8.2.x and the relationships specified with other concepts are retained. At the end of the validation, the log will show a list of these deprecated concepts:

```
The following deprecated concepts were referenced in the import files:
noun.act:Coral_Sea
noun.object:Coral_Sea
```

If any of these concepts are important to your installation, you may copy them to a text file and use that to export them from 8.0. Otherwise these concept references will be ignored during the final import to 8.2.x.

Concepts deprecated between 8.1 and 8.2.x have not been removed from 8.2.x; instead, they have been placed in a special domain for deprecated concepts. See the section below on concept domains for details.

### Ignored concepts

In some cases the concept you import may not actually be different from the 8.2.x version. If you choose to replace or merge the import concept with the existing 8.2.x. concept, and the concepts are found to be identical, then you will see the following log message:

```
Concept IPP:Computer Concepts was ignored; it is the same as an existing
concept of same name
```

### Single clustermate

If the import files contain a cluster which has only one clustermate, it is ignored:

```
Cluster 'daemon' will not be added because it has only 1 clustermate.
```

### Renamed Concepts

When concepts are imported from 8.0 to 8.2.x, their names must be converted from the old three-part format (pos.domain:name) to the new one-part format. These names are generated in the following way:

#### **1 Recognized Concepts**

If the three-part concept name is recognized by being found in the mapping file for the industry you have chosen, then it is given a new name based on that mapping.

In this case, you will see a log message like the following:

```
Known SLD concept renamed via map: noun.location:Aberdeen ==> Aberdeen
```

## 2 Unrecognized Concepts

If the three-part concept name is not found in the mapping file, then a new mapping is added to the file, with the new name generally corresponding to the third part of the three-part name. Namespace conflicts are resolved automatically (see "Further Explanation of the Concept Update Process" below for more details).

In this case, you will see a log message like the following:

```
Unknown SLD concept renamed automatically: noun.mycompany:myWidget ==>
myWidget
```

### Merged concepts

Some concepts have been merged in 8.2.x because they have basically the same meaning as another concept. For example, when constructing the installed 8.2.x dictionary, the 8.0 concepts <noun.computer.page> and <noun.communication.page> were merged into a single 8.2.x concept, <page>, because they were redundant with each other. If you have made changes to each of the original concepts, and are importing them to 8.2.x, the features of those concepts will also be automatically merged to the single concept <page>. You will be alerted to this with the following log message:

```
Known SLD concept renamed via map: noun.computer.page ==> page
Merged concepts noun.computer.page and noun.communication.page into a
single concept, page
```

```
Known SLD concept renamed via map: noun.communication.page ==> page
```

Check these concepts to make sure that the results of the merge make sense with your customizations.

## Conduct Ontology Import

After you have finalized your import files, and verified that they no longer cause any errors, you may conduct the actual import. Select the subject and the domain mapping as described above. You will then choose how to handle any concepts which were found to already exist in the 8.2.x dictionary. You are asked to either (1) ignore the concept in the import file, (2) replace the concept in the dictionary with the one in the import file, (3) add a suffix to the name of the concept being imported to differentiate it from the one in the dictionary, or (4) merge the two concepts.

See *Intelligent Search Language Tuning Guide*, Chapter 2: Dictionary Manager, "Import Dictionary Objects, Ontology Objects, and Process Wizards" for more information.

When you choose "replace", it replaces the features of the concept (score, domain, and relationships), and any synsets that are explicitly imported. As of 8.2, it will also replace any synsets that exist for other languages. In other words, if you import a concept with an English synset and choose "replace", the English synset will be replaced, and the synsets for all the other languages will disappear.

In order to preserve the non-English synsets, the best choice is usually to merge the existing concepts. After the import is done, you have a chance to compare the installed 8.2.x version with your imported concept. Before committing changes, find the concepts which have been changed and choose “Diff” from the right-click menu on the find results. This gives you a side by side comparison with the differences highlighted. You can then choose to revert to the installed 8.2.x version or make any other desired changes.

**Important!** Do not make any changes to concept names until you have finished the import of non-ontology objects!

### **BACKING OUT OF A CONCEPT IMPORT**

At any point during the concept import, you may decide that you need to start over.

To do this, complete the following steps:

- 1 Find all of the concepts last edited on today’s date.
- 2 Select all concepts in the find results, right click, and choose “revert”.
- 3 When you do the next import, you see a message, “Mapping file from previous import available. Do you want to start over?” Choose “yes”.

This resets the dictionary to its original state, and you can start the concept import over from the beginning.

### **IMPORTING MULTIPLE SETS OF CONCEPTS**

You may decide that it is easier to import certain sets of concepts separately. For example, you may wish to map one set of concepts to a particular 8.2 concept domain, regardless of the domain in their three-part name.

If you import a second set of concepts, the import remembers your settings from the previous set. You see a message, “Mapping file from previous import available. Do you want to start over?” Choose “no”, and the same mapping file for concept names are used. Any custom concepts that were previously imported have been added to the custom mapping file, and so they are renamed the same way. The log displays these renamed concepts with the message “Known SLD concept renamed via map...”.

If there are concepts whose names are unrecognized, you are shown the domain mapping dialog again, with your previous domain mapping choices. At this point you may change any mapping choices before proceeding. Be sure to import all concepts before you proceed to the import of other dictionary objects.

## FURTHER EXPLANATION OF THE CONCEPT UPDATE PROCESS

### Concept Names

When the dictionaries were migrated to 8.1, part of the process was to both update concept names from three-part names (<noun.state.account>) to one-part names (<account>). For the most part, the one-part name is simply the third part of the three-part name, but in some cases this resulted in duplicate names, and so a more specific name was given to one of the concepts. Furthermore, consolidating all of the industries into a single dictionary resulted in some duplicate names, so some name updates are specific to a particular industry. This is why it is necessary to choose the right industry when first importing the concepts.

The mappings from old three-part names to new one-part names are maintained in text files under <installation>\resources\8.0\_to\_8.1\_concept\_mapping. When you choose an industry for an import, the contents are copied to a new file called inquiramapping-file. As you import your concepts it adds new mappings for custom concepts not yet in the mapping file.

In general, custom concepts are mapped as follows:

- 1 Only the third part of the three-part name is kept.
- 2 Underscores convert to spaces.
- 3 Illegal characters convert to underscores. The exception is for three-part names ending with a double quote, e.g. <noun.mycompany.myWidget2000">. In previous versions, a double quote was frequently appended to a concept ending in a numeral. During the concept name conversion these double quotes will be deleted rather than replaced with an underscore.
- 4 If the above steps result in two concepts in the import file having the same name, they are appended with \_1 and \_2. For example, if the import file contains <noun.customerhelp.widget> and <noun.mydomain.widget>, then those concepts are mapped to <widget\_1> and <widget\_2> respectively.
- 5 Similarly, if the above steps result in the concept having the same name as a concept that already exists in the 8.1 dictionary, then the concept being imported is appended with \_1. For example, if the import file contains <noun.mydomain.account>, and the dictionary has a concept <account>, then <noun.mydomain.account> is mapped to <account\_1>.

Since installed concept names do not contain underscores in 8.2, automatically adding underscores to resolve namespace conflicts between distinct concepts then allows you to search on the underscore when the concept import is done in order to find names which may need to be changed. Note that when you conduct this search, you must escape the underscore with a backslash, i.e. search on "\\_" (no quotes), not "\_". You do this as the last step of the dictionary upgrade, so that references in non-ontology objects are also changed.

### Concept Synonyms

When you import synonyms, they are automatically lower cased. This is to preserve the case matching behavior in pre-8.1 versions. Previously, English synonyms were completely case-insensitive. If a synonym had upper case, it still annotated lower case strings. For example, the synonym "MyCompany Widget" would still annotate "mycompany widget".

As of 8.1 an English synonym with upper case only annotates a string with the same case. This way if there is a synonym which could be confused with a common word when it is not capitalized, it can be restricted to strings with upper case. Non-English synonyms are still case-insensitive, but you can force them to be case-sensitive by adding the capitalized synonym to the stem exception list. See the "Concept Synonyms" section of "Functional Improvements" below for further details.

### Check Concept and Cluster Domains

When you have finished importing ontology objects, use the Find Tool to retrieve them all. Sort by Domain and check to see whether there are any which have been assigned a domain not included in your domain list(s). Change the domain of these concepts to a domain appropriate for your domain list(s) or add the domain of these concepts to your domain list(s).

If you wish to change all of the domains at once, you may use the following steps:

- 1 Export the concepts with the undesired domains.
- 2 In the concepts.csv file, change the domain name to the desired domain.
- 3 Re-import the files, choosing "Replace all" in order to replace the domain name.

Do the same for clusters if necessary.

This situation can arise because concepts were removed from dictionaries for certain industries in previous versions, but were retained in other industries. Therefore if you are importing concepts installed as part of a previous version, those concepts may be imported in the domain for another industry.

This situation may also arise if a concept or cluster has been deprecated between 8.1 and 8.2.x. These objects will be in a deprecated domain, i.e. Deprecated Computer Concepts or Deprecated Base Concepts. You may find that you no longer need these concepts, as they have been found to be outdated or redundant with other concepts.

### Perform First Dictionary Validation

Run dictionary validation from the Tools menu. For each of the domain list(s) you intend to use, check the Concepts, Clusters, and Synsets boxes. Make sure that no issues have been introduced. You may need to fix issues in the import files, back out your changes, and reimport, as described in section 3.2.

**Note:** The installed dictionary contains disabled concepts, so it is normal to see the message “Found Disabled Concepts”.

## Import Taxonomy

To import the 8.0 taxonomy, use the following steps:

- 1 Select the Tools/Import SLD Taxonomy menu action in the Dictionary Manager.
- 2 Use the browser to select the taxonomy.xml file to be imported. Be sure to take the taxonomy from the original installation being upgraded.
- 3 On completion a dialog will appear to report success or failure. If any errors were found, a follow-up dialog will appear, listing the errors, which can be copied and saved. When you import, any concept references in the taxonomy convert to the new format based on the mapping file. If there are any concept references not found in the mapping file, either because the concept was deprecated or is a custom concept that was not imported, then the import does not complete. You must either (1) export and import the missing concepts, or (2) change those references to concepts which have been imported or which were retained in 8.2.x.

## Import Process Wizards

In the 8.2.x dictionary, choose “Import Process Wizards” from the Tools menu, and find the directory where the 8.0 process wizards are stored. (The default location is development\content\applications\default\dictionary\processWizards in the 8.0 dictionary.)

This imports the process wizards to the 8.2.x directory

<instance>\development\content\applications\default\userdata\users\internal\<user>\domain\PROCESSWIZARD.

## Import Non-Ontology Objects

As with the concept import, the first import of the non-ontology objects should be done in test-only mode. This will give you a chance to fix any errors, and add any further concepts to the dictionary should they be needed.

The first step in the import is to resolve any unrecognized domains. When you import non-ontology objects, they retain the domain they had in 8.0 unless that domain is not recognized. In that case, you are asked if you want to recreate the domain in 8.2.x, or map the unrecognized domain to an existing 8.2.x domain.

Most installed domains have the same name in 8.2.x as in previous versions, but a few have changed. You may map the older installed domains below to existing 8.2.x domains as follows:

- Intent Library Telecom Multilingual --> Intent Library Telecom

- Intent Library Automotive Multilingual --> Intent Library Automotive
- Intent Library Finance Multilingual --> Intent Library Finance
- Intent Library Xena CH --> Intent Library Customerhelp
- Intent Library Telecom-Consumer --> Intent Library Telecom

References to concepts with their three-part names automatically update to the correct one-part name using the mapping file generated as part of the concept import. If the three part concept name is not found in the mapping file, then it does not convert.

The log file notes this; for example:

```
Alias List 'My Alias List' key[1] -- 'noun.mycompany.myWidget' not found
in SLD to MLD concept name map
```

Find these lines in the log file (search on “not found in SLD to MLD concept name map”), and determine whether these concepts need to be added. These concepts are also summarized in the log:

```
The following missing concepts were referenced in the import files:
noun.mycompany.deauthorization
noun.mycompany.myWidget
```

The log will also note references to concepts which have been deprecated prior to 8.1, for example:

```
Alias List 'Test deprecated concept' key[9] -- InQuira CONCEPT
'noun.food.prune_whip' HAS BEEN DEPRECATED -- FIX!
```

These concepts are also summarized at the end of the validation:

```
The following deprecated concepts were referenced in the import files:
noun.food.prune_whip
noun.person.jimdandy
```

Determine whether these missing concepts need to be included in your installation. If so, you may copy the desired concepts from the log and create a text file to use in exporting them from 8.0. The best practice is to export them to a new folder, and follow the same steps importing them as with the first set of concepts. However, when you are prompted to use the existing mapping file, choose “yes”, as described in the section “Importing Multiple Sets of Concepts”.

After you are able to validate the non-ontology import files without any errors, you may go ahead with the actual import.

### Perform Second Dictionary Validation

Choose the domain list(s) that contain the imported objects and run dictionary validation. Fix any issues found.

## Conduct Concept Name Cleanup

When you imported concepts, there may have been namespace conflicts automatically resolved by adding `_1`, `_2`, etc. to the end of the concept name. Also, if there were any illegal characters in the imported concept names, they would have been replaced by underscores. (See section 3.4 above.) You can find these concepts by searching on the underscore in concept names, but ***you must escape the underscore by preceding it with a backslash***, i.e. the search string is: `\_`. (If you search just on the underscore, the find tool attempts to find all concepts. If this happens, you must shut down the dictionary and restart, since this search takes a long time.)

There is no technical need to change the names of these concepts, but it makes them easier to use if you give them more descriptive names. When you change the name of a concept, all references to that concept in other dictionary objects automatically update.

## Perform Third Dictionary Validation

Run dictionary validation and check for any issues.

## Conduct Concept Relationship Checks

### PARENT/CHILD RELATIONSHIPS

In addition to running validation, you need to check any important relationships between concepts that existed in the 8.0 dictionary. When concepts are exported, there is a column for “TypeOfParent” and “PartOfParent”, but no similar column for children. This means that when you migrate a concept that had a certain child in the 8.0 dictionary, but do not also migrate that child, the relationship is lost. Since you had the chance to add children to your export to retain those relationships, this should not be an issue.

However, there may be other concepts in the 8.2.x dictionary which refer to an imported concept as a parent, and this relationship will be retained. If you are concerned about the possibility of additional unwanted children on your modified concepts, you can check this by searching on the CHILDREN operator in your custom rules and intents; any concepts expected to have children should have the same ones they did in 8.0.

Also, be aware that if you choose to replace an existing concept, the replacement still has all the children of the original concept, since that relationship is defined on the children rather than the parent.

Similarly, any installed concept modified on import will continue to belong to any existing clusters, since clusters are also defined independently of concepts. If you wish to remove a concept from a cluster, you will need to edit that cluster directly.

## CLUSTER RELATIONSHIPS

Search on your custom rules and intents for the SIMILAR operator, and check to see if the referenced concept belongs to the expected cluster.

**Note:** In 8.2.x, clusters can be referred to in IML by name using square brackets, e.g. [cash]. The SIMILAR operator still functions as before, so the IML expression SIMILAR(<cash>) will match on all concepts that belong to a cluster that includes <cash> as a member.

## Update Spell Check Changes

If there have not been any changes to the installed spell check list, then this step is not necessary. If you have added custom corrections, then find them in the `correct.tlx` file in your 8.0.4 installation in `<INSTALL_DIR>/inquire/data/lexicons/correct.tlx`.

The `correct.tlx` is a text file, whose lines have the format:

```
<CORRECTION_CANDIDATE><TAB>A<CORRECTION>
```

for example:

```
deisel Adiesel
```

Your custom corrections appear at the top, before the line “developorsAdevelopers”. Find the spell check list in the 8.2.x dictionary and add your custom corrections to it, but without the “A” in front of the correction.

## Update Quality Monitor Test Profiles

Configure the Quality Monitor to point to the existing database used to store tests and results. You must update the Test Profiles by hand, as follows:

- 1 Update references to the 8.0.4 domain list with references to the name of the 8.2.x domain list.
- 2 For English tests, change the Language setting in the Test Profiles from ‘Default’ to ‘English’.

Remember that in the Quality Monitor, if there are any test results tied to a Test Profile, then that Test Profile can’t be edited, as it would invalidate the environment for the saved results. If you want to be able to compare the 8.0.x results with the 8.2.x results, you need to create new Test Profiles by copying and modifying the existing ones.

## Add Functional Improvements

### CONCEPT SYNONYMS

As part of the upgrade, you have the opportunity to improve annotation for any custom concepts with these types of synonyms. For example, the concept for a product with a name like “Word” should have that synonym capitalized so that it is less likely to annotate the common noun “word”. Acronyms and abbreviations are also good candidates for capitalized synonyms, e.g. <Maine> has the synonym “ME”, which does not annotate the lower case pronoun “me”.

**Note:** This automatic case sensitivity works only for English. For other languages you must both (1) have the synonym in the desired case, and (2) add the synonym in the desired case as both stem and token in the stem exception list for that language.

### ADDING LANGUAGES

If you wish to add custom translations to a set of concepts, export the synonyms (find the set of concepts with the find tool, select all, right-click and choose “export synonyms”). Translate the synonyms, in the resulting synsets.csv file, change the Language from English to the desired language, and re-import the file.

### OPENCLASS CONCEPTS

If you rely on any installed open class concepts in your pre-8.1 installation, you must enable these concepts in 8.2.x. Out of the box, these concepts, e.g <date Openclass> are disabled because in some installations they can lead to performance issues.

## Upgrading from InQuira 8.2.x to InQuira 8.2.3

Run the following command from the ICE prompt:

```
upgrade.bat/sh
```

In order to preserve dictionary customizations made in 8.2, the best practice is to move the dictionary from the 8.2 installation to the 8.2.3 installation. To do this:

- 1 Find the vcroot directory in your 8.2 installation under  
`<install-root>\base\development\content\data.`
- 2 Move this vcroot to your 8.2.3 installation, replacing the existing vcroot under  
`<install-root>\base\development\content\data.`

Now you can open the dictionary as any of the users that were defined in 8.2. You can also take advantage of any of the synonym translations that were added in 8.2.3. To do this:

- 1 Choose "Import Ontology Objects" from the Tools menu, and browse to the appropriate directory.
  - a For new French, German, Italian, Japanese, or Spanish synonyms for Base Concepts look under:

```
<install-root>\resources\Dictionary_Import_Files\Synonym_
translations_for_8.2.3\Base_Concept_synonym_translations_for
_8.2.3
```

- b For new French, German, or Spanish synonyms for Computer Concepts look under

```
<install-root>\resources\Dictionary_Import_Files\Synonym_
translations_for_8.2.3\Computer_Concept_synonym_translations
_for_8.2.3
```

## Upgrading from InQuira 8.1.x to InQuira 8.2.3

Use the following instructions to upgrade InQuira:

- 1 Open an ICE window.
- 2 Stop all InQuira services running on all instances.
- 3 Extract the upgrade archive into a temporary directory.
- 4 CD into the temporary directory.
- 5 CD into the `verify` directory.

- 6 Execute the script:

```
verifyFrom.{bat,sh} <version>
```

where `<version>` is one of 8.1.3.x, 8.2.x, or 8.2.2.x for the respective OS (Windows or Linux).

This script compares the installation with the expected set of files. If there are any MISMATCHES, zip up the mismatched files and send them to InQuira Customer Support to evaluate if there will be any problems upgrading. If no mismatches occur, continue with step 7.

- 7 Execute the search upgrade script: `upgrade.{bat,sh}`.
- 8 Follow the instructions on screen after the upgrade completes.
- 9 Copy the `autoimport.xml` from the upgrade folder to:  
`<INSTALL_HOME>/instances/<INSTANCE_NAME>/`
- 10 Open an ICE window and restart InQuira services.

**Note:** Note that between 8.1 and 8.2.3, many concepts in the Base Concepts and Computer Concepts domains were deprecated due to being outdated, obscure, or redundant. Therefore these imports will not cover all of the concepts that were in those domains as of 8.1.

## Post Upgrade Instructions for Windows OS

InQira 8.2.x uses a newer version of the Java Service Wrapper, which requires the reinstallation of all of the InQira services on Windows systems.

After the upgrade has been completed, each service must be uninstalled and then reinstalled, as follows:

For each default (scheduler) instance:

- 1 Create an ICE window for the instance.

```
cd <install_dir>\instances\<<default_instance_name>
.\setenv.bat
```

- 2 Enter the following:

```
inquiraservice -uninstall
inquiraservice -install
```

**Note:** If Information Manager has been added to this default instance, wait until after running the IM Upgrader installer before reinstalling the IM services.

For each Search Runtime instance:

- 1 Create an ICE window for the instance.

```
cd <install_dir>\instances\<<runtime_instance_name>
.\setenv.bat
```

- 2 Enter the following:

```
inquiraservice -uninstall
inquiraservice -install
```

## Post Upgrade Instructions for 8.1.3.x Upgrades

Customers upgrading from InQira 8.1.3.x releases may need to manually remove facet CMS-CATEGORY using the following steps:

- 1 Launch Personalized Navigation.
- 2 Navigate to the CMS-CATEGORY facet.
- 3 Delete the CMS-CATEGORY facet.
- 4 Save and commit.

---

---

# Upgrading Information Manager

This chapter describes upgrading Information Manager from any version between 7.2.x and 8.0.3 to Version 8.2.3.

This chapter describes:

- **Before Upgrading from 8.2.2 to 8.2.3**
- **Upgrading from a Release Prior to InQuira 7.3**
- **Upgrading from InQuira 7.3 and Later Releases**
- **Post Upgrade Notes**
- **InfoCenter Upgrade**
- **IMUpgrader Utility Overview**
- **Limitations**

## Before Upgrading from 8.2.2 to 8.2.3

If you are upgrading from 8.2.2 to 8.2.3, you must modify `registeredapps.xml` prior to performing the upgrade to avoid potential issues that may arise during the upgrade.

**Important!** This issue occurs only when upgrading from 8.2.2 to 8.2.3.

Before upgrading to 8.2.3, review the current `registeredapps.xml` under

```
IM_HOME\config\registeredapps.xml
```

for the following:

- Confirm the path to the web application is correct for each application.
- Confirm the `app.type` is correctly defined for the iConnect and Self-Service Portal applications.

Clean up the `registeredapps.xml` prior to upgrade as follows:

- 1 Stop Information Manager.
- 2 Find any duplicate /redundant entries in the `registeredapps.xml` for your deployed applications.

```

<registeredapps>
  <app>
    <context><![CDATA[IMWebServices]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\imws]]></path>
    <url><![CDATA[http://ESF44//IMWebServices/WebObjects/IMWebServices.woa]]></url><repository><![CDATA[IMWEBSERVICES]]></rep
    <version><![CDATA[8.2.2.0]]></version><instance.type><![CDATA[IMWEBSERVICES]]></instance.type>
  </app>

  <app>
    <context><![CDATA[InfoManager]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\InfoManager]]></path>
    <url><![CDATA[http://ESF44//InfoManager/WebObjects/InfoManager.woa]]></url>
    <repository><![CDATA[IMADMIN]]></repository>
    <version><![CDATA[8.2.2.0]]></version><instance.type><![CDATA[IMADMIN]]></instance.type></app>

  <app>
    <context><![CDATA[test_infocenter]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps]]></path>
    <url/>
    <instance.type/>
    <app.type><![CDATA[InfoCenter]]></app.type>
    <version><![CDATA[8.2.2.0]]></version>
    <repository><![CDATA[TEST]]></repository></app>

  <app><context><![CDATA[IMTagLibrary]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_infocenter]]></path>
    <url><![CDATA[http://ESF44//IMTagLibrary/WebObjects/IMTagLibrary.woa]]></url><repository><![CDATA[TEST]]></repository>
    <version><![CDATA[8.2.2.0]]></version>
    <instance.type><![CDATA[INFOCENTER]]></instance.type>
  </app>

  <app><context><![CDATA[test_iconnect]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps]]></path>
    <url/><instance.type/><app.type><![CDATA[iConnect]]></app.type>
    <version><![CDATA[8.2.2.0]]></version>
    <repository><![CDATA[TEST]]></repository>
  </app>

  <app>
    <context><![CDATA[IMTagLibrary]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_iconnect]]></path>
    <url><![CDATA[http://ESF44//IMTagLibrary/WebObjects/IMTagLibrary.woa]]></url>
    <repository><![CDATA[TEST]]></repository><version><![CDATA[8.2.2.0]]></version>
    <instance.type><![CDATA[INFOCENTER]]></instance.type>
  </app>

  <app>
    <context><![CDATA[test_ssp]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps]]></path>
    <url/><instance.type/><app.type><![CDATA[SSP]]></app.type>
    <version><![CDATA[8.2.2.0]]></version>
    <repository><![CDATA[TEST]]></repository>
  </app>

  <app>
    <context><![CDATA[IMTagLibrary]]></context>
    <path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_ssp]]></path>
    <url><![CDATA[http://ESF44//IMTagLibrary/WebObjects/IMTagLibrary.woa]]></url>
    <repository><![CDATA[TEST]]></repository><version><![CDATA[8.2.2.0]]></version>
    <instance.type><![CDATA[INFOCENTER]]></instance.type>
  </app>
</registeredapps>

```

### 3 Copy the path of this redundant entry and paste it to the web application entry that precedes it.

```

<registeredapps>
<app>
<context><![CDATA[IMWebServices]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\imws]]></path>
<url><![CDATA[http://ESF44//IMWebServices/WebObjects/IMWebServices.woa]]></url><repository><![CDATA[IMWEBSERVICES]]></re
<version><![CDATA[8.2.2.0]]></version><instance.type><![CDATA[IMWEBSERVICES]]></instance.type>
</app>

<app>
<context><![CDATA[InfoManager]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\InfoManager]]></path>
<url><![CDATA[http://ESF44//InfoManager/WebObjects/InfoManager.woa]]></url>
<repository><![CDATA[IMADMIN]]></repository>
<version><![CDATA[8.2.2.0]]></version><instance.type><![CDATA[IMADMIN]]></instance.type></app>

<app>
<context><![CDATA[test_infocenter]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps]]></path>
<url/>
<instance.type/>
<app.type><![CDATA[InfoCenter]]></app.type>
<version><![CDATA[8.2.2.0]]></version>
<repository><![CDATA[TEST]]></repository></app>

<app><context><![CDATA[IMTagLibrary]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_infocenter]]></path>
<url><![CDATA[http://ESF44//IMTagLibrary/WebObjects/IMTagLibrary.woa]]></url><repository><![CDATA[TEST]]></repository>
<version><![CDATA[8.2.2.0]]></version>
<instance.type><![CDATA[INFOCENTER]]></instance.type>
</app>

<app><context><![CDATA[test_iconnect]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps]]></path>
<url/><instance.type/><app.type><![CDATA[Iconnect]]></app.type>
<version><![CDATA[8.2.2.0]]></version>
<repository><![CDATA[TEST]]></repository>
</app>

<app>
<context><![CDATA[IMTagLibrary]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_iconnect]]></path>
<url><![CDATA[http://ESF44//IMTagLibrary/WebObjects/IMTagLibrary.woa]]></url>
<repository><![CDATA[TEST]]></repository><version><![CDATA[8.2.2.0]]></version>
<instance.type><![CDATA[INFOCENTER]]></instance.type>
</app>

<app>
<context><![CDATA[test_ssp]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps]]></path>
<url/><instance.type/><app.type><![CDATA[SSP]]></app.type>
<version><![CDATA[8.2.2.0]]></version>
<repository><![CDATA[TEST]]></repository>
</app>

<app>
<context><![CDATA[IMTagLibrary]]></context>
<path><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_ssp]]></path>
<url><![CDATA[http://ESF44//IMTagLibrary/WebObjects/IMTagLibrary.woa]]></url>
<repository><![CDATA[TEST]]></repository><version><![CDATA[8.2.2.0]]></version>
<instance.type><![CDATA[INFOCENTER]]></instance.type>
</app>
</registeredapps>

```

#### 4 Remove the redundant entry.

```

:registeredapps>
jp>
intext><![CDATA[IMWebServices]]></context>
ith><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\imws]]></path>
`l><![CDATA[http://ESF44//IMWebServices/WebObjects/IMWebServices.woa]]></url><repository><![CDATA[IMWEBSERVICES]]></reposito
:rsion><![CDATA[8.2.2.0]]></version><instance.type><![CDATA[IMWEBSERVICES]]></instance.type>
ipp>

jp>
intext><![CDATA[InfoManager]]></context>
ith><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\InfoManager]]></path>
`l><![CDATA[http://ESF44//InfoManager/WebObjects/InfoManager.woa]]></url>
:pository><![CDATA[IMADMIN]]></repository>
:rsion><![CDATA[8.2.2.0]]></version><instance.type><![CDATA[IMADMIN]]></instance.type>
ipp>

jp>
intext><![CDATA[test_infocenter]]></context>
ith><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_infocenter]]></path>
`l/>
instance.type><![CDATA[INFOCENTER]]></instance.type>
ip.type><![CDATA[InfoCenter]]></app.type>
:rsion><![CDATA[8.2.2.0]]></version>
:pository><![CDATA[TEST]]></repository>
ipp>

jp><context><![CDATA[test_iconnect]]></context>
ith><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_iconnect]]></path>
`l/>
instance.type><![CDATA[INFOCENTER]]></instance.type>
ip.type><![CDATA[iConnect]]></app.type>
:rsion><![CDATA[8.2.2.0]]></version>
:pository><![CDATA[TEST]]></repository>
ipp>

jp>
intext><![CDATA[test_ssp]]></context>
ith><![CDATA[C:\apps\InQuira_8.2.2.0\instances\InfoManager\appserverim\webapps\test_ssp]]></path>
`l/><instance.type/><app.type><![CDATA[SSP]]></app.type>
:rsion><![CDATA[8.2.2.0]]></version>
instance.type><![CDATA[INFOCENTER]]></instance.type>
:pository><![CDATA[TEST]]></repository>
ipp>

`registeredapps>

```

- 5 Repeat steps 2 - 4 for each deployed web application, confirming:
  - the path to the web application is correct for each application.
  - the `app.type` is correctly defined for the iConnect and Self-Service Portal applications.
- 6 Proceed with the upgrade using the instructions in this chapter.

## Upgrading from a Release Prior to InQuira 7.3

This section describes how to upgrade Information Manager from a release prior to 7.3.

- Create a file called `registeredapps.xml` in the `<IM_HOME>/config` folder with the structure shown below. The valid repository for the IM management console is `IMADMIN` and the valid repository for web services is `IMWEBSERVICES` for all web services WARs. The entries are case sensitive and should be embedded in CDATA sections.

Supply the specified information inside the CDATA sections.

```
<registeredapps>
  <app>
    <context><![CDATA[<context the web app is deployed under>]]></context>
    <path><![CDATA[ <fully qualified path to the expanded web services folder>]]></
path>
    <url><![CDATA[ http://<host>:<port>/<context>]]></url>
    <repository><![CDATA[IMWEBSERVICES]]></repository>
    <version><![CDATA[8.1.1.0.3]]></version>
  </app>
  <app>
    <context><![CDATA[<context the web app is deployed under>]]></context>
    <path><![CDATA[<fully qualified path to the expanded web services folder>]]></
path>
    <url><![CDATA[ http://<host>:<port>/InfoManager]]></url>
    <repository><![CDATA[IMADMIN]]></repository>
    <version><![CDATA[8.1.1.0.3]]></version>
  </app>
  <app>
    <context><![CDATA[<context the web app is deployed under>]]></context>
    <path><![CDATA[<fully qualified path to the expanded web services folder>]]></
path>
    <url><![CDATA[ http://<host>:<port>/<context>]]></url>
    <repository><![CDATA[<REPOSITORY REFERENCE KEY>]]></repository>
    <version><![CDATA[8.0.3.0.15]]></version>
  </app>
</registeredapps>
```

**Important!** InQuira strongly recommends that you view the IMUpgrader upgrade logs closely for issues and errors. If individual scripts or data migration classes fail, they must be fixed before trying to run the upgraded application.

- Complete the upgrade procedure by following the procedure described in “Upgrading from InQuira 7.3 and Later Releases” on page 36.

---

## Upgrading from InQuira 7.3 and Later Releases

This section describes how to upgrade an existing Information Manager application from 7.3 or later to 8.2.3. This section also describes how to complete the upgrade procedure described in “Before Upgrading from 8.2.2 to 8.2.3” on page 31.

- Make a backup of the IM database schema and content resource files prior to attempting this upgrade process.

It is important to shut down access to the Information Manager database schema and file system during the upgrade process to ensure a smooth upgrade cycle. It is only necessary to update the database schema only once during the upgrade process, the binary files for each web application on each server/instance must be replaced during the upgrade process.

- If using an Oracle database, set the maximum number of open cursors to at least 600.
- Perform the following steps on each server where Information Manager applications are running:
  - a.) Stop all Information Manager applications.
  - b.) Run the IMUpgrader utility. This utility can also be run from the ICE command prompt if necessary. If performing the upgrade for the database make sure to pass the `updatedb=true` parameter into the IMUpgrader utility. See “IMUpgrader Command Line Operation” on page 44 for more options on running the IMUpgrader from the ICE environment.

For older installations, the IMUpgrader will not be able to automatically replace the binary files for the IM JSP Tag library or the IM Web Services. These applications must be updated manually using the instructions below. There are special update instructions for InfoCenter based applications.

---

## Completing the Upgrade Process

**Important!** The following mandatory steps must be completed by someone with “super user”, “super support”, or “site admin” privileges to complete the IM upgrade:

- 1 From within the IM Console, Rebuild XML for each channel:
  - a Click the "Repository" tab.
  - b Under "Channels" in the left Action Bar, click "List".
  - c For each channel in Channel List, click "Rebuild XML".
- 2 From within the IM Console, Reindex the Full Text Index:
  - a Click the "Tools" tab.
  - b Under "Full Text Search" in the left Action Bar, click "Reindex".
  - c Click the "Index Repository" link at the bottom of the page.

**Note:** The amount of time it takes for each of these steps to complete depends on the size of the IM database. Each step could take several hours to complete, particularly on exceptionally large databases.

## Post Upgrade Notes

Validate that the IM Upgrade was successful using the validation steps outlined below.

- Review the upgrade logs located in `<IM_HOME>/logs/upgrade` for any errors that are not marked as info or warning.
- Verify that each web application deployed in the `appserverim/webapps` folder has been updated. In the cases of the WAR files, the WAR should have been replaced. For IM tag library applications, the files inside the `WEB-INF` folder should have been updated. The `WEB-INF/version.txt` file should reflect the newest upgraded version.
- Verify that the `<IM_HOME>/wars` folder has the latest war files copied to it.
- Verify that the `<IM_HOME>/install/taglib` folder has the latest InfoCenter JSP and binaries installed.
- Verify the `INSTALLHISTORY` table in the database reflects all of the correct upgrades as delineated in the `version.xml`, located in `<IM_HOME>\upgrade\version.xml`. If something is missing it should be reflected in the upgrader log file in the `<IM_HOME>/logs/upgrader` folder.
- Verify that the `<IM_HOME>/jms` folder has been deleted and the Windows OpenJMS service has been removed.

- Resolve potential conflicting .jar files in Information Manager Web Applications. For more information, see “Resolve Conflicting .jar Files in IM Web Applications” on page 47.

After validating the installation and restarting IM, review the start up log for errors relating to the installation. A common error looks similar to:

```
ERROR
com.inquiraservices.notification.templates.NotificationTemplateManager -
COULD NOT FIND CONTENTS FOR TEMPLATE = ../InQira_8.0/InfoManager/config/
SYSTEM/taskconfig/contentinlimbo_en.xml -- MAKE SURE THE TEMPLATE FILES ARE
LOADED FOR ALL LOCALES
```

If this error occurs, the missing templates can be manually copied from the `<IM_HOME>/upgrade/files/deploy/InfoManager/config/SYSTEM/taskconfig` into the `<IM_HOME>/config/SYSTEM/taskconfig` folder of the system being upgraded. Once the templates have been copied, restart Information Manager and the errors should go away.

## Change Notification Update

InQira Information Manager uses a change notification mechanism to notify all running IM instances when a database change is made to keep the data fresh in all of the instances. In 8.1.2.4 InQira Information Manager was updated to replace the JMS change notification system with a new change notification system based on Jgroups multicast communication framework. This new mechanism does not rely on a single server to broadcast the changes like the JMS based solution and is hooked into the Information Manager database access framework in a much more robust manner.

## OpenJMS Removal

As of versions 8.0.4.20 and 8.1.2.4, Information Manager no longer utilizes OpenJMS for change notification. During the upgrade to this version previous installations of OpenJMS provided with Information Manager should be removed along with the supporting files. The following files/folders can be safely removed after upgrading to 8.0.4.20 or 8.1.2.4 or newer:

- `<IM_HOME>/jms`
- On Windows platforms, remove the OpenJMS service from the control panel Services
- In each `<IM_HOME>/config/<repository>/application.properties` – remove the following properties:
  - `changenotification.topicconnectionfactory`
  - `changenotification.jndiurl`
  - `changenotification.jms.topicName`
  - `changenotification.initialcontextfactory`

- In each deployed web application remove the following files:
  - WEB-INF/lib/openjms\*.jar
  - WEB-INF/lib/spice-jndikit\*.jar
  - WEB-INF/lib/jms\*.jar

## Jgroups Configuration

There are new configuration options available in the `<IM_HOME>/config/<repository>/application.properties` to control the Jgroups configuration:

- `jgroups.eofsynchronizer.group.identifier = InQuiraEOFGroup`
- `jgroups.configsynchronizer.group.identifier = InQuiraConfigGroup`
- `jgroups.eofsynchronizer.enabled = true`
- `jgroups.configsynchronizer.enabled = true`

These new parameters are set to default values OOTB. Jgroups utilizes a multicast UDP broadcast and will discover other jgroups enabled applications on the same network based on the group identifier. In certain circumstances it may be necessary to change the group identifiers to segregate groups of servers to prevent messages being broadcast between the server groups. All servers/instances in the same group must have the same group identifier. There are two Jgroups defined OOTB – the InQuiraEOFGroup handles change notification messages related to database changes between instances. The InQuiraConfigGroup handles configuration changes made thru the IM Management console to the config.properties and sends a notification out to other IM applications on the network to re-read the config.properties file. This notification does NOT transmit the values that changed only the fact that something changed. It is still necessary to make the changes in each IM\_HOME. The **enabled** parameters are used to disable the change notification functionality. These should be left on to ensure proper operation of Information Manager. Disabling either service may require IM to be restarted in order to view specific changes.

## InQuira Client Library API Interface

The previous builds of Information Manager provided an earlier version of the IM web services that were deployed as `imws.war` and `IMWOWebservices.war`. These files have been consolidated into the new `IMWS.war` deployed with 8.1.2.4. The older style XML based interface available prior to 8.1.2.4 will be *supported* but will not be *extended*. The preferred interface is now the InQuira Client Library API. All code using the older style XML based interface should be replaced with the newer client library API calls in order to take advantage of the latest functionality.

**Important!** During upgrade, remove the older IM web services providers to avoid conflicts.

---

## Reinstall Services on Windows OS

InQira 8.2.2 uses a newer version of the Java Service Wrapper, which requires the reinstallation of all of the InQira services on Windows systems.

After the upgrade has been completed, the IM service must be uninstalled and then reinstalled, as described below.

For a Standalone Information Manager Installation:

- 1 Create an ICE window for the instance.

```
cd <install_dir>\instances\InfoManager
.\setenv.bat
```

- 2 Enter the following:

```
inquiraimservice -uninstall
inquiraimservice -install
```

For an Add-on Information Manager Installation:

- 1 Create an ICE window for the default (scheduler) instance.

```
cd <install_dir>\instances\<default_instance_name>
.\setenv.bat
```

- 2 Enter the following:

```
inquiraimservice -uninstall
inquiraimservice -install
```

## Information Manager Collection Configuration Files

Check the IM collection configuration file **Force Public** setting following the upgrade. Make certain that **Force Public** is set to `Off`. If set to `On`, all IM content for this collection will be public regardless of individual document settings.

## InfoCenter Upgrade

In version 8.0.1.3.x, InfoCenter was updated to provide a more flexible and higher performance design for managing the configurable properties. Starting in 8.0.1.3, the properties that were originally stored in the `i_configure.jsp` were migrated to `WEB-INF/infocenter.properties`. This change required updates to the underlying InfoCenter JSP pages in a manner similar to the following:

The old way to get a configuration parameter:

```
<%@ include file="/apps/infocenter/i_configure.jsp" %>
```

After including the `i_configure.jsp` page all of the variables that were defined there would be available to the current page.

The new way to get configuration parameters using `infocenter.properties`:

```
<%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8"
language="java" errorPage="" import="java.text.*,
java.util.*,com.inquiria.client.configuration.Configuration" %>
int days = Configuration.getIntForKey("daysToShowAlerts");
```

Where `daysToShowAlerts` is a value defined in the `WEB-INF/infocenter.properties` file. It will be necessary to update all customized JSP pages that use the old `i_configure.jsp` to use the newer `infocenter.properties` file.

To upgrade an older version of InfoCenter do the following:

- 1 Replace the JSP pages located in `<YOUR_WEB_APP>/apps/InfoCenter/system` with the new InfoCenter JSP files located in the `<IMUPGRADER_HOME> /IMTagLibrary/apps/infocenter/system` folder
- 2 Update any custom pages to read the `infocenter.properties` file.
- 3 The IMUpgrader utility, described in “IMUpgrader Utility Overview” on page 42, **does not** upgrade the existing deployed InfoCenter JSP page installations at this time. After running the IMUpgrader, you must manually update existing pre-deployed InfoCenter implementations by doing the following:
  - a Back up the existing deployed InfoCenter application. By default, the copy can be placed in the `<IM_HOME>/upgrade` folder by version number.
  - b Copy the files from `<IM_HOME>/install/taglib/apps/infocenter/system` and replace the files in the same location in the deployed InfoCenter installation.
  - c Validate that the files from the `<IM_HOME>/install/taglib/WEB-INF` match the ones in your web application. If they do not match, they should be copied to your web application. The `WEB-INF/version.txt` should match the expected version of your installation.
  - d Review the customized JSP pages and compare them to the newly updated InfoCenter system pages. Apply any necessary changes. In most cases, the changes are going to include localizations, JSP logic fixes, and potentially changes in URLs that could affect analytic reporting.

## Post-Upgrade Security Modification

You must manually update the `validation.properties` for each deployed Web Application on the Information Manager side. To update:

- 1 Stop all IM services.
- 2 For each deployed IM Web Application, go to  
`<IM_HOME>\install\<webapp_type>`
- 3 Edit the file `WEB-INF/classes/resources/validation.properties`
- 4 Find the entry for `Validator.URIStrng`, update the entry to:

```
Validator.URIStrng=^[\\p{L}\\p{P}\\p{Nd}+=&amp;%\\\\\\\\\\-$#_?:.!@/ ]+$
```

- 5 Find the entry for `Validator.URIStrng2`, update the entry to:

```
Validator.URIStrng2=^[\\p{L}\\p{P}\\p{Nd}+=&amp;%\\\\\\\\\\-$#_?:.!@*'() ,/ ]+$
```

- 6 Save the changes.

## IMUpgrader Utility Overview

The IMUpgrader is a java based command line application that can be used to apply the required SQL scripts and data migration commands to an Information Manager database schema to update it to the current version.

For convenience the IMUpgrader is packaged into a GUI installation program that will execute the basic steps needed to upgrade a single server installation of Information Manager (IM). For more complex installations it may be required to run the IMUpgrader from the ICE environment to customize the upgrade steps. After the IMUpgrader runs using the GUI utility, it leaves behind the upgrade files in the `<IM_HOME>/upgrade` folder. This folder can be copied to the other IM servers that need to be upgraded manually.

Prior to 8.x, the version information for each upgrade of IM was stored in the `DBVERSION` table of the IM database schema. Starting in 8.x, the version upgrade information is now stored in the `INSTALLHISTORY` table. Data from the `DBVERSION` table will be automatically moved to the new table and the old `DBVERSION` table will be removed during the upgrade process.

Prior to performing any modifications to the files during the upgrade process, the IMUpgrader makes a backup of each of the web applications and stores it in the `<IM_HOME>/backup` folder. There is a backup folder for each version upgrade that occurs.

A record of the upgrade process is also stored in the `<IM_HOME>/receipts` folder.

The IMUpgrader tries to determine the version of the installed application by examining the database, and looking for previous upgrade information. The required steps needed for the version upgrade are then looked up in the version.xml that is located in the upgrade folder. The IMUpgrader will execute the required upgrade steps in the following order to move from the installed version to the upgrade version.

Replace the binary files for each IM based application detected (this information is stored in the <IM\_HOME>/config/registeredapps.xml file, which is created/updated each time an IM application starts up):

- Execute all pre-migrate SQL scripts for the specified supported database
- Execute each listed data migration class
- Execute all post data migration SQL scripts

The output of the upgrade operation is stored in the <IM\_HOME>/logs/UPGRADE folder. Each time the IMUpgrader is run a new log file is created to record the activity. It is important to review these files after each upgrade to ensure that everything succeeded correctly.

## OpenJMS Uninstallation Tasks

As of 8.0.4.20 Information Manager no longer uses OpenJMS (or other JMS providers) to perform change synchronization. During the upgrade to either of these builds, the previously installed OpenJMS software will be uninstalled.

After the IMUpgrader completes, it may be necessary to manually remove portions of the JMS installation as noted below. Ensure that IM is not running prior to performing these steps.

- 1 (Windows only) Remove the OpenJMS service that is installed by using the `inquiraimservice.bat` script.

```
inquiraimservice.bat -uninstall
```

After verifying that both the InfoManager service and the OpenJMS services have been successfully removed, reinstall the InfoManager service using the `inquiraimservice.bat` script:

```
inquiraimservice.bat -install
```

- 2 (Windows only) Verify that the newly installed InfoManager service no longer has a dependency on the OpenJMS service.
- 3 There are configuration files located in <INSTALL\_DIR>/conf that can be deleted for the JMS service configuration.
- 4 For each IM application in the <IM\_HOME>/config folder edit the application.properties file and remove the following parameters:

# Change Notification Topic Name	changenotification.jms.topicName = infomanager
# Jndi URL for the JMS notifications	changenotification.jndiurl = tcp:// localhost:3035
# JMS initial context factory	changenotification.initialcontextfactory = org.exolab.jms.jndi.InitialContextFactory
# JMS topic Connection Factory	changenotification.topicconnectionfactory = JmsTopicConnectionFactory

- Restart IM after all of the applications have been updated.

## IMUpgrader GUI Operation

In order to use the IMUpgrader it is necessary to download or obtain the IMUpgrader for your platform from the InQuira support group.

Prior to running the IMUpgrader – make sure to have a backup of your IM database schema – the changes made are not reversible.

If running the IMUpgrader for the first time or if it is necessary to update the IM database schema – make sure that the **-updatedb=true** parameter is set. The IMUpgrader uses the database configuration information from the <IM\_HOME>/config/IMADMIN/application.properties file to connect to the database to be updated.

## IMUpgrader Command Line Operation

The IMUpgrader can be run from the ICE command prompt to perform customized upgrade operations. See “Required Parameters” and “Optional Parameters” below.

When running the IMUpgrader from the command line on Windows platforms, it is necessary to surround the arguments to the batch script with double quotes in order for the batch script to work properly. The command processor on Windows only accepts 10 arguments. For example:

```
C:\InQuira_8.0\InfoManager\upgrade > imupgrader.bat "-im_home=
C:\InQuira_8.0\InfoManager" "-updatedb=true"
```

### Required Parameters

- im\_home=<full path to InfoManager configuration files>
- updatedb=<true|false>

Flag to indicate whether or not to update the database. Defaults to false.

## Optional Parameters

Parameter	Description
-updatefiles=<true false>	Flag to indicate whether or not to update/replace the web application, IMWS, or InfoManager files. Defaults to true.
-user=<db user to update>	Defaults to IMADMIN property JDBCUserid if not specified.
-passwd=<db user passwd>	Defaults to IMADMIN property JDBCPassword if not specified.
-jdbcurl=<db JDBC URL to update>	Defaults to IMADMIN property JDBCURL if not specified.
-jdbcdriver=<db driver class>	Defaults to IMADMIN property JDBCDRIVER if not specified. <sup>1</sup>
-version=<version to apply>	Installs only the specified version. <sup>1</sup>
-script=<SQL script to apply>	Runs only the specified SQL script. The path needs to be relative to the IMUpgrader folder. <sup>1</sup>
-class=<full java class name of migration class>	Executes the specified java migration class to perform.
-migrateDBVersion=<true false>	Creates installhistory table during upgrade. Defaults to true. Use when there is no installhistory table present in the database.
-infocenter	Does the following: <ul style="list-style-type: none"> <li>• Upgrades InfoCenter config file (infocenter.properties). Compares the current config and the original installed config (under \InfoManager\install\taglib). Any changes are considered customer modifications. The changes are moved to a custom config file named infocenter_custom.properties. This new custom config overrides any entry in the infocenter.properties after the upgrade.</li> <li>• Upgrades InfoCenter resource files (ApplicationResource.properties). Compares the current resource file and the original installed resource file (under \InfoManager\install\taglib). Any changes are considered customer modifications. The changes are moved to a custom config file named ApplicationResource_custom_locale.properties. This new custom config overrides any entry in the corresponding locale of the ApplicationResource_locale.properties after upgrade.</li> </ul>

1. Some options are assumed to be mutually exclusive of the others. Specifically only one of version, script, or class can be specified at a time.

---

## Limitations

Please note the following limitations.

### IMUpgrader Troubleshooting

The IMUpgrader is designed to upgrade MOST of the things within IM but it does not upgrade everything at this time.

The IMUpgrader does not replace any JSP pages that are deployed with a web application. The IMUpgrader only replaces files inside the WEB-INF folder. Any changes to InfoCenter JSP pages must be manually applied at this time.

Older installations of IM, IM web services, and JSP web applications did not do a good job of identifying the installed version. In recent builds, a file called registeredapps.xml located in <IM\_HOME>/config contains an entry for each web application. This file is updated each time the web application launches and contains the version of the application as well as the location of the application. If the web application has not registered itself in this file the IMUpgrader will not update it. These applications will need to be updated manually.

The IMUpgrader log file contains a record of all actions taken during the course of the upgrade. At the top of the log, there is a section that identifies the version currently installed and lists the unapplied updates that it finds in the version.xml. If the current version cannot be found in the version.xml the upgrader will not do anything.

The upgrade folder contains all of the required SQL scripts and files needed to upgrade an IM installation. In the event that it is necessary to do a manual upgrade, the files can be copied from the upgrade folder. The version.xml file contains the list of operations required for each step of the upgrade. If it is necessary to run the upgrade manually, make sure to execute all of the <PRE> SQL scripts first, followed by the <SCRIPT> migration classes and finally execute the <POST> SQL scripts last after verifying that the upgrade steps were successful by inspecting the logs.

To run a SQL script manually using the IMUpgrader execute the following command:

```
imupgrader.sh im_home=<path to $IM_HOME> -script=sql/oracle/BUILD_80200.sql
```

To run a Java migration class manually, execute the following command:

```
imupgrader.sh im_home=<path to $IM_HOME> -class=  
com.inqira.infomanager.migration.IM80200
```

If upgrading multiple servers or instances of an IM installation, it is only necessary to run the DB migration scripts once for each database environment being upgraded. After updating the files on the first server, copy the updated web applications to the other servers/instances. It is not necessary to run the IMUpgrader on the other servers.

If it is necessary to run the IMUpgrader on other instances, it will be necessary to pass in `-updatedb=false`. It will also be necessary to delete the entries in the `INSTALLHISTORY` that shows the files being copied.

The IMUpgrader stores a backup of the web apps it updates in the `<IM_HOME>/backup` folder. It does NOT keep a backup of the repository data before the update starts.

## IMUpgrade Fails to Upgrade Database on Linux

The `imupgrader.sh` script calls `java` without specifying its full path, which should be `<install_dir>/jre/bin/java`. If the IM Upgrade Installer is run in a new shell, where the `<install_dir>/instances/<instance_name>/setenv.sh` script has not been run previously to set up an ICE Environment, then the `<JAVA_HOME>` hasn't been set and the `PATH` has not been set to include `<JAVA_HOME>/bin`. The `imupgrader.sh` script should be modified to determine and set the full path for `java`, as it should not be required to set up an ICE Environment to run the IM Upgrade installer.

The SH script code for getting the full path to the `java` program is:

```
DIR=`pwd`
cd `dirname $0`
SCRIPTPATH=`pwd`
cd "$SCRIPTPATH/../../..
INSTALL_DIR=`pwd`
cd "$DIR"
$INSTALL_DIR/jre/bin/java ...
```

This code

- Saves the current working directory in `DIR`
- Sets `SCRIPTPATH` to the directory path for the `imupgrader.sh` script (`<install_dir>/infoManager/upgrade`)
- Changes Working Directory to the grandparent directory of the script (`<install_dir>`)
- Sets `INSTALL_DIR` to this directory
- Changes the Working Directory back to whatever it was originally
- Executes the upgrader using the full path to the InQuira installed 'java' program.

## Resolve Conflicting .jar Files in IM Web Applications

After upgrading to 8.2, inspect the `WEB-INF/lib` folder of each deployed IM web application for `jar` files that may conflict with the newer `.jar` files provided with the installation. The installed `WAR` files should be correct. This only applies to IM JSP tag library based web applications (InfoCenter and custom UIs).

The following table provides a complete list of the .jar files that should be deployed with 8.2. If the installation directory contains .jar files other than those listed below, delete the extra .jar files, *unless* they are being used by custom code in the deployed web applications.

### .jar files

• activation.jar	• immodel.jar	• lucene-1.5-rc1-dev.jar
• ajax-wrapper-comp-1.0.1.jar	• imservices.jar	• mail.jar
• ant.jar	• imservletfilter.jar	• mysql-connector-java-5.0.8-bin.jar
• axis-ant.jar	• imtaglibrary.jar	• nekohtml.jar
• axis.jar	• infra-1.0.jar	• NetComponents.jar
• bsf.jar	• inquirea-infra-1.1.jar	• ojdbc14.jar
• castor-1.2-anttasks.jar	• iqflexjson.jar	• orai18n.jar
• castor-1.2-codegen.jar	• j2eeclient.jar	• org.json-0.0.1.jar
• castor-1.2-ddlgen.jar	• javacsv.jar	• PDFBox-0.7.1.jar
• castor-1.2-jdo.jar	• javafoundation.jar	• poi-2.5-final-20040302.jar
• castor-1.2-xml-schema.jar	• javajdbcadaptor.jar	• quartz.jar
• castor-1.2-xml.jar	• javawebobjects.jar	• saaj.jar
• castor-1.2.jar	• javawebservicesclient.jar	• saxpath.jar
• cayenne.jar	• javawebservicesupport.jar	• serializer.jar
• commons-beanutils.jar	• JavaWOExtensions.jar	• slf4j-api-1.5.0.jar
• commons-codec-1.3.jar	• JavaWOJSPServlet_client.jar	• slf4j-log4j12-1.5.0.jar
• commons-collections-3.1.jar	• javawojspServlet.jar	• soap.jar
• commons-configuration-1.4.jar	• JavaXML.jar	• ssce.jar
• commons-dbcp-1.1.jar	• jaxrpc.jar	• standard.jar
• commons-discovery-0.2.jar	• jcommon-0.8.0.jar	• stax-1.2.0.jar
• commons-fileupload-1.2.jar	• jdbc-se2.0.jar	• stax-api-1.0.1.jar
• commons-io-1.3.2.jar	• jericho-html-3.1.jar	• swing-layout-1.0.jar
• commons-lang-2.3.jar	• jettison-1.0-RC2.jar	• Tidy.jar
• commons-logging-1.0.4.jar	• jgroups-all.jar	• tlddoc.jar
• commons-net-ftp-2.0.jar	• jms-1.1.jar	• tm-extractors-0.4.jar
• commons-pool-1.1.jar	• js.jar	• trove-2.0.1.jar
• concurrent-1.3.4.jar	• jsonorg.jar	• velocity-1.5.jar
• derby.jar	• jsse.jar	• webserviceutils.jar
• derbyclient.jar	• jta.jar	• wsdl4j-1.5.1.jar
• ERExtensions.jar	• jta1.0.1.jar	• xalan.jar
• ERJars.jar	• jtids-1.2.jar	• xercesImpl.jar

**.jar files (Continued)**

• ERJGroupsSynchronizer.jar	• jug-lgpl-2.0.0.jar	• xml-apis.jar
• erxservletadaptor.jar	• junit-4.4.jar	• xmlParserAPIs.jar
• erxwo54.jar	• junit.jar	• xmlunit-1.1.jar
• httpunit.jar	• ldap.jar	xpp3_min-1.1.4c.jar
• icu4j-3_8_1.jar	• log4j-1.2.14.jar	• xstream-1.3.jar

# Upgrading Analytics

You upgrade InQira Analytics applications using the following processes to upgrade the Intelligent Search Analytics and Information Manager Analytics projects, as appropriate for your installation.

**Note:** Before upgrading to InQira Analytics 8.2.3, you will need to upgrade to MicroStrategy to 8.1.1 and obtain a new license key from MicroStrategy. It can take up to 7 days to receive a new license key from MicroStrategy.

This chapter describes two types of upgrade procedures:

- **Upgrading Analytics from Version 7.x to 8.0**
- **Upgrading Analytics from Version 8.0.x to 8.2.3**

and presents additional information on:

- **Schema Tables**
- **Processing ETL Log Files Created in Previous Releases**

---

## Upgrading Analytics from Version 7.x to 8.0

This section covers:

- **Upgrading Intelligent Search Analytics from 7.x to 8.0**
- **Upgrading Information Manager Analytics from 7.x to 8.0**

### Upgrading Intelligent Search Analytics from 7.x to 8.0

Question: Is this correct? Do we have a way to upgrade Search Analytics from a 7.x release?

Use the following process to upgrade InQuira Analytics 2.1 and 7.x installations to InQuira Analytics 8.0.1. The upgrade process results in two MicroStrategy projects:

- a project for the existing (InQuira Analytics 2.1 or greater) application
- a new InQuira Analytics 8.0.1 project

**Note:** For InQuira Analytics versions previous to 2.1, contact <CustSupp> for assistance in upgrading.

To upgrade InQuira Analytics:

- 1 Load and transform any existing log files into your existing application prior to the upgrade.  
By design, ETL does not process old log files. For example, version 8.2.3 ETL does not process 8.1.2.3 IM ETL log files.  
When upgrading from 8.1.x.x to 8.2.3, process all 8.1.x.x ETL log files, using 8.1.x.x, prior to upgrading to 8.2.3.  
If it is not possible to process existing ETL log files, see “Processing ETL Log Files Created in Previous Releases” on page 61.
- 2 Delete the InQuira Analytics errors directory.
- 3 Install InQuira Analytics as described in the InQuira Analytics Installation Guide.
- 4 Create the InQuira Analytics database schema by executing the following commands from the InQuira 8 ICE environment:  

```
- scheduler run DBInterface -u ods  
- scheduler run DBInterface -u analytics  
- scheduler run DBInterface -u hierarchy
```
- 5 Create the new InQuira Analytics metadata schema using the DBInterface as described in the InQuira Analytics Installation Guide.

- 6 Install MicroStrategy 8.01 as described in the InQuira Analytics Installation Guide. Refer to the MicroStrategy documentation for additional information if necessary.
- 7 Open the InQuira Project for your existing application Analytics 2.1.
- 8 Merge the MicroStrategy projects.  
For installations that do not include customization, the project merging process consists of the following steps:
  - setting up the rules for the merge process
  - completing the mergeFor installations that include customization, the project merging process consists of the following steps:
  - setting up the rules for the merge process
  - completing the merge
  - testing the merged projects

We recommend that persons defining the merge process be familiar with this process. The merge process merges the existing application into the new application, so that you can maintain the current application for use in the event of errors. The actual merge process can take less than 10 minutes.

Testing the projects consists of comparing the new projects with the previous version by manually paging through and drilling within the reports.

If the testing process reveals errors, <CustSupp> can assist by re-merging the project or by making manual repairs. The time required depends on the amount of customization and whether there are conflicts between the two projects.

**Note:** The source project source is Analytics <current\_version> metadata. The destination project source is Analytics 8.0.1 metadata.

- Access the MicroStrategy Project Merger Wizard:  
Start->Programs->MicroStrategy->Administrator->ObjectManager->Project Merger Wizard
- Select “Next” to advance to Select the Source Project Source, and select the InQuira Analytics <current\_version> metadata
- Select “Next” to advance to Select Destination Project Source and Project, and select the InQuira Analytics 8.0.1 metadata
- Specify the merge rules at the object category level
- Specify the merge rules for the object types. You can use the default settings.
- Select “Next” to specify the rules for specific objects
- Perform the merge process

- Update the destination project schema using the Schema->Update Schema option in the MicroStrategy Desktop application
- Test the destination project after merging. You may need assistance from <CustSupp> to modify some of the objects.

## Upgrading Information Manager Analytics from 7.x to 8.0

To upgrade InQuira Information Manager Operational Reports and Analytics Version 7.2.2 and 7.3.

- 1 Load and transform any existing log files into your existing application prior to the upgrade.
- 2 Install InQuira Information Manager Analytics as described in the InQuira Analytics Installation Guide.
- 3 Create the InQuira Analytics database schema by executing the following commands from the InQuira 7 ICE environment:
  - scheduler run DBInterface -u imoltp
  - scheduler run DBInterface -u imanalytics
  - scheduler run DBInterface -u imods
- 4 Create the new InQuira Information Manager Operational Reports and Analytics metadata schema using the DBInterface as described in the InQuira Analytics Installation Guide
- 5 Install MicroStrategy 8.02 as described in the InQuira Analytics Installation Guide. Refer to the MicroStrategy documentation for additional information if necessary.
- 6 Open the InQuira Project for the existing Analytics application
- 7 Merge the MicroStrategy projects

## Upgrading Analytics from Version 8.0.x to 8.2.3

General steps:

- 1 Load all the remaining logs.

By design, ETL does not process old log files. For example, version 8.2.3 ETL does not process 8.1.2.3 IM ETL log files.

When upgrading from 8.1.x.x to 8.2.3, process all 8.1.x.x ETL log files, using 8.1.x.x, prior to upgrading to 8.2.3.

If it is not possible to process existing ETL log files, see “Processing ETL Log Files Created in Previous Releases” on page 61.

- 2 Backup the following files:

<install\_home>\archive\inquirawb.war

<install\_home>\inquiralib\inquira\inquira.jar

- 3 Install InQuira 8.2.3 Analytics and point the analytics database to the existing analytics database on 8.0.x.

- 4 Translate the schemas from the previous two datasources (Analytics datasource and Information Manager Star Schema datasource) to the current single datasource (Analytics datasource)

In InQuira 8.2.3, one datasource is used for all the analytics schemas. If you are upgrading from a release that uses two different datasources, dump all the tables from the two datasources into a single datasource. Please do this step with assistance from your DBA and then confirm the tables listed under “Upgrading Analytics from Version 7.x to 8.0” on page 51 are included in your datasource.

- 5 Run the following commands from an ICE window to update schemas and metadata:

Command	Description
<code>scheduler run DBInterface -u analytics</code>	Upgrades search ODS schema and IM analytics ODS and star schema.
<code>scheduler run DBInterface -u imoltp</code>	Upgrades IM Operational schema.
<code>scheduler run DBInterface -u odsarchive</code>	Upgrades Search ODS archive schema.
<code>scheduler run DBInterface -c metadata</code>	Upgrades analytics metadata.

**Note:** This command will recreate the metadata, thus all the customizations of the reports will lost.

---

## Schema Tables

**Table 3.1 Search ODS tables**

REP\_DRILL\_ONLY  
REP\_GLBL\_ACTIVITY\_TXT  
REP\_GLBL\_ACTIVITY\_TYPE\_TXT  
REP\_GLBL\_ARCHIVE\_LOG  
REP\_GLBL\_CLUSTER\_TXT  
REP\_GLBL\_CONCEPT\_TXT  
REP\_GLBL\_CONFIDENCE\_LEVEL\_TXT  
REP\_GLBL\_CONFIDENCE\_SCORE  
REP\_GLBL\_CONTEXT\_MAPPING  
REP\_GLBL\_CONTEXT\_VALUE\_TXT  
REP\_GLBL\_FACET\_TXT  
REP\_GLBL\_FIELD\_TXT  
REP\_GLBL\_HOUR  
REP\_GLBL\_I\_QUESTION\_TXT  
REP\_GLBL\_IASSIST\_TXT  
REP\_GLBL\_INTENT\_PARAM\_TXT  
REP\_GLBL\_INTENT\_RESPONSE\_TYPE  
REP\_GLBL\_INTENT\_TXT  
REP\_GLBL\_LANG\_TXT  
REP\_GLBL\_LOG\_EVENT  
REP\_GLBL\_LOG\_EVENT\_COMPLETE  
REP\_GLBL\_LOG\_FILE\_STATUS  
REP\_GLBL\_MAX\_ID  
REP\_GLBL\_MICROSITE\_TXT  
REP\_GLBL\_N\_QUESTION\_TXT  
REP\_GLBL\_PAGE\_TXT  
REP\_GLBL\_PROCESSW\_STEP\_TXT  
REP\_GLBL\_PROCESSW\_TEMP\_TXT  
REP\_GLBL\_PRODUCT\_HIERARCHY  
REP\_GLBL\_PURPOSE\_TXT

**Table 3.1 Search ODS tables (Continued)**

REP\_GLBL\_QUERY\_SOURCE\_TXT  
REP\_GLBL\_QUESTION\_TXT  
REP\_GLBL\_RATING  
REP\_GLBL\_RESPONSE\_TXT  
REP\_GLBL\_RESPONSE\_TYPE\_TXT  
REP\_GLBL\_SEQ\_VIRPAGE\_TXT  
REP\_GLBL\_SESS\_TXT  
REP\_GLBL\_TEMP\_HOUR  
REP\_GLBL\_TIME  
REP\_GLBL\_U\_FACET\_TXT  
REP\_GLBL\_U\_QUESTION\_TXT  
REP\_GLBL\_UNKNOWN\_TXT  
REP\_GLBL\_USER\_SEGMENT\_TXT  
REP\_GLBL\_USER\_TXT  
REP\_GLBL\_VIRTUALPAGE\_TXT  
REP\_ODS\_VERSION  
REP\_SESS\_ACTIVITY  
REP\_SESS\_CLICK\_THRU  
REP\_SESS\_CLUSTERS  
REP\_SESS\_CONCEPT\_DIRECT\_MATCH  
REP\_SESS\_CONCEPT\_MATCH  
REP\_SESS\_CONTEXT1  
REP\_SESS\_CONTEXT2  
REP\_SESS\_FACET\_ACTIVITY  
REP\_SESS\_FACET\_SELECT  
REP\_SESS\_FEEDBACK  
REP\_SESS\_FORM\_FIELD\_FACT  
REP\_SESS\_FORM\_PAGE\_FACT  
REP\_SESS\_FORM\_VIR\_PAGE\_FACT  
REP\_SESS\_IASSIST\_STD\_VALUES  
REP\_SESS\_IASSIST\_TMP\_QUESTION

**Table 3.1 Search ODS tables (Continued)**

REP\_SESS\_INTENT\_MATCH  
REP\_SESS\_INTENT\_PARAMS  
REP\_SESS\_NO\_RESPONSE  
REP\_SESS\_PROC\_WIZARD\_STEP  
REP\_SESS\_PROCESS\_TEMP  
REP\_SESS\_QUESTION  
REP\_SESS\_QUESTION\_ACTIVITY  
REP\_SESS\_RATING  
REP\_SESS\_RESPONSE  
REP\_SESS\_UNKNOWN\_WORD  
REP\_SESS\_WORK\_SORT  
REP\_SESSION

**Table 3.2 IM ODS Tables**

REP\_ACCESS\_ACTIVITY\_FACT  
REP\_ANSWER\_DIM  
REP\_CASELINK\_DIM  
REP\_CAT\_CCHAN\_DIM\_HELPER  
REP\_CAT\_CONTENT\_DIM\_HELPER  
REP\_CATEGORY\_DIM  
REP\_CATEGORY\_DIM\_HELPER  
REP\_CCHAN\_CONTENT\_DIM\_HELPER  
REP\_CONT\_STATUS\_DIM\_HELPER  
REP\_CONTENT\_DIM  
REP\_CONTENT\_GROUP\_DIM\_HELPER  
REP\_CONTENT\_TEAM\_DIM\_HELPER  
REP\_CONTENTAUDITLOG  
REP\_CONTENTCHANNEL\_DIM  
REP\_DISCUSSIONBOARD\_DIM  
REP\_EXPIRE\_ACTIVITY\_FACT

**Table 3.2 IM ODS Tables (Continued)**

REP\_FORUM\_DIM  
REP\_FORUM\_RATING\_FACT  
REP\_IM\_ODS\_VERSION  
REP\_LOADED\_LOG\_FILE  
REP\_LOCALE\_CONT\_DIM\_HELPER  
REP\_LOCALE\_DIM  
REP\_MESSAGE\_DIM  
REP\_PUB\_ACTIVITY\_FACT  
REP\_QUESTION\_DIM  
REP\_RATING\_DIM  
REP\_REPOS\_CAT\_DIM\_HELPER  
REP\_REPOS\_CCHAN\_DIM\_HELPER  
REP\_REPOS\_CONTENT\_DIM\_HELPER  
REP\_REPOSITORY\_DIM  
REP\_REPOSITORY\_DIM\_HELPER  
REP\_SURVEY\_DIM  
REP\_SURVEY\_RESULT\_DIM  
REP\_TEAM\_DIM  
REP\_TEAM\_DIM\_HELPER  
REP\_TOPIC\_ACTIVITY\_FACT  
REP\_TOPIC\_DIM  
REP\_USER\_ACTIVITY\_FACT  
REP\_USER\_DIM  
REP\_USER\_GROUP\_DIM\_HELPER  
REP\_USER\_TEAM\_DIM\_HELPER  
REP\_USERACTIVITYLOG  
REP\_USERGROUP\_DIM  
REP\_VIEW\_CONTENT\_DIM\_HELPER  
REP\_VIEW\_DIM  
REP\_WORKFLOW\_DIM  
REP\_WORKFLOW\_STEP\_DIM

**Table 3.3 IM Star Procedures**

CAT\_TEAM\_HIERARCHY\_UNKNOWN  
MS\_IM\_POPULATE\_TIME\_DIM  
MS\_IM\_UPDATE\_TIME\_DIM

**Table 3.4 IM Star Tables**

MS\_IM\_ACCESS\_ACTIVITY\_FACT  
MS\_IM\_ANSWER\_DIM  
MS\_IM\_CASELINK\_DIM  
MS\_IM\_CAT\_CCHAN\_DIM\_HELPER  
MS\_IM\_CAT\_CONTENT\_DIM\_HELPER  
MS\_IM\_CATEGORY\_DIM  
MS\_IM\_CATEGORY\_DIM\_HELPER  
MS\_IM\_CCHAN\_CONTENT\_DIM\_HELPER  
MS\_IM\_CONT\_STATUS\_DIM\_HELPER  
MS\_IM\_CONTENT\_DIM  
MS\_IM\_CONTENT\_GROUP\_DIM\_HELPER  
MS\_IM\_CONTENT\_TEAM\_DIM\_HELPER  
MS\_IM\_CONTENTCHANNEL\_DIM  
MS\_IM\_CONTENTSTATUS\_DIM  
MS\_IM\_DISCUSSIONBOARD\_DIM  
MS\_IM\_EXPIRE\_ACTIVITY\_FACT  
MS\_IM\_FORUM\_DIM  
MS\_IM\_FORUM\_RATING\_FACT  
MS\_IM\_LOCALE\_CONT\_DIM\_HELPER  
MS\_IM\_LOCALE\_DIM  
MS\_IM\_LOCALIZED\_CONTENT\_DIM  
MS\_IM\_MESSAGE\_DIM  
MS\_IM\_PUB\_ACTIVITY\_FACT  
MS\_IM\_QUESTION\_DIM

**Table 3.4 IM Star Tables (Continued)**

MS\_IM\_RATING\_DIM  
MS\_IM\_REPOS\_CAT\_DIM\_HELPER  
MS\_IM\_REPOS\_CCHAN\_DIM\_HELPER  
MS\_IM\_REPOS\_CONTENT\_DIM\_HELPER  
MS\_IM\_REPOSITORY\_DIM  
MS\_IM\_REPOSITORY\_DIM\_HELPER  
MS\_IM\_STAR\_VERSION  
MS\_IM\_SURVEY\_DIM  
MS\_IM\_SURVEY\_RESULT\_DIM  
MS\_IM\_TEAM\_DIM  
MS\_IM\_TEAM\_DIM\_HELPER  
MS\_IM\_TIME\_DIM  
MS\_IM\_TOPIC\_ACTIVITY\_FACT  
MS\_IM\_TOPIC\_DIM  
MS\_IM\_USER\_ACTIVITY\_FACT  
MS\_IM\_USER\_DIM  
MS\_IM\_USER\_GROUP\_DIM\_HELPER  
MS\_IM\_USER\_TEAM\_DIM\_HELPER  
MS\_IM\_USERGROUP\_DIM  
MS\_IM\_VIEW\_CONTENT\_DIM\_HELPER  
MS\_IM\_VIEW\_DIM  
MS\_IM\_WORKFLOW\_DIM  
MS\_IM\_WORKFLOW\_STEP\_DIM

## Processing ETL Log Files Created in Previous Releases

**Important!** This section provides a workaround to allow for running multiple versions of IS Log files into a single version of ETL Search Analytics. Use this **ONLY** if you are unable to process ETL log files at the time of upgrade.

Each of the Search logs generated contains information from the version they were created from. For Example, below is the top part of a log coming out of Search -

```
[1 WrapperStartStopAppMain(1)] Event(Code=INSTANCE_EVENT, id=#0B0I1P-1)
started at 7/28/10 6:13 AM: Instance MyCompanyrt1 in runtime/Development
mode with uuid c017b15f-9a30-11df-b7ec-889aa0ea362b created. Release MLD
Build 7

[2 WrapperStartStopAppMain(2)] Event(Code=START_INSTANCE, id=#0B0I2P-1)
started at 7/28/10 6:13 AM: Instance MyCompanyrt1 in runtime/Development
mode with uuid c017b15f-9a30-11df-b7ec-889aa0ea362b is initializing.

[3 WrapperStartStopAppMain (2)]
#####

[4 WrapperStartStopAppMain (2)] Initializing instance with name
MyCompanyrt1 in development/runtime mode at Wed Jul 28 03:13:20 PDT 2010.

[5 WrapperStartStopAppMain (2)] Release 8.1.x.xx

[6 WrapperStartStopAppMain (2)]
#####
```

Note that the version shows 8.1.x.xx.

In the InQuira Indexer instance, there is a derby database file named `logevent-n.n.n.nn.db` - where `n.n.n.nn` would need to be `8.1.x.xx` in order to read the `8.1.x.xx` files coming out of Search. So in its entirety, this file's name would look like `logevent-8.1.x.xx.db`. This file is located under -

```
<INQUIRA_HOME>\instances\<Indexer Name>\development\content\data\log\def
```

### PROCEDURE

In order to have other versions of log files be processed by Search ETL, look at what the version indicates inside those log files and then do the following:

Copy the existing `logevent-n.n.n.nn.db` file to the same location as it currently exists but rename this copy based on the header information in the Search Log file. For example, an upgrade was performed from 8.1.2.0 to 8.2.3. This would mean that there would be a derby db file in the location already specified above called `logevent-8.1.2.0.db`. Copy this file, and name it `logevent-8.2.3.0.db`. Keep it in the same location as the original `logevent` file. Now, it should be possible to process both the old log files generated from 8.1.2.0 and new log files being generated from 8.2.3.