



Cisco Workload Automation Informatica Adapter Guide

Version 6.3

First Published: August, 2015
Last Updated: September 6, 2016

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

© 2016 Cisco Systems, Inc. All rights reserved.

Contents 3**Preface 5**

- Audience 5
- Related Documentation 5
- Obtaining Documentation and Submitting a Service Request 5
- Document Change History 5

Introducing the Informatica Adapter 7

- Overview 7
- Prerequisites 8
- Software Requirements 8

Configuring the Informatica Adapter 9

- Overview 9
- Installing and Configuring Informatica Libraries 9
- Licensing an Adapter 12
- Securing the Informatica Adapter 13
 - Defining Informatica Users 13
 - Authorizing Schedulers to Work with Informatica Jobs 14
- Configuring the Master to Support Informatica TLS 17
 - Configuring the Master for Informatica 9.6.1 TLS for Windows 17
 - Configuring the Master for Informatica 9.6.1 TLS for Unix 18
- Defining an Informatica Adapter Connection 18
 - Adding an Informatica Adapter Connection 19
 - Connecting to the Integration Service 22

Using the Informatica Adapter 23

- Overview 23
- Defining Informatica Jobs 23
 - Informatica Job Definition 23
 - Working with the Tree View 26
 - Start Workflow from a Task 26
 - Run a Specific Task 27
 - Specifying Informatica Job Parameters 28
 - Checking the Informatica Job Status 30
 - Controlling the Workflow 31
- Monitoring Informatica Jobs 33
- Controlling Informatica Tasks and Workflows 38
 - Recovering a Workflow 39
 - Recovering a Task 39
 - Restarting a Workflow from a Specific Task 40
- Controlling Adapter and Agent Jobs 40
 - Holding a Job 40
 - Aborting a Job 41
 - Rerunning a Job 41
 - Making One Time Changes to an Adapter or Agent Job Instance 41
 - Deleting a Job Instance before It Has Run 41

Troubleshooting the Informatica Adapter 43

- Overview 43

- Troubleshooting 43
 - Enabling Debugging 43
 - Informatica Service Fails to Start 43
 - Informatica Connection Is Red 44
 - Testing Using Command Line Mode 44
 - Monitoring with JConsole 45

Configuring service.props 47

- About Configuring service.props 47
- service.props Properties 47



Preface

This guide describes the installation, configuration, and usage of the Informatica Adapter with Cisco Workload Automation (CWA).

Audience

This guide is for administrators who install and configure the Informatica Adapter for use with CWA, and who troubleshoot CWA installation and requirements issues.

Related Documentation

See the *Cisco Workload Automation Documentation Overview* for your release on cisco.com at:

<http://www.cisco.com/c/en/us/support/cloud-systems-management/tidal-enterprise-scheduler/products-documentation-roadmaps-list.html>

...for a list of all CWA guides.

Note: We sometimes update the documentation after original publication. Therefore, you should also review the documentation on [Cisco.com](http://cisco.com) for any updates.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Document Change History

The table below provides the revision history for the *Cisco Workload Automation Informatica Adapter Guide*.

Version Number	Issue Date	Reason for Change
6.1.0	October 2012	New release.
6.2.1	June 2014	Documentation available in online Help only.

Document Change History

Version Number	Issue Date	Reason for Change
6.2.1 SP2	June 2015	Configuration provided in the <i>TES Installation Guide</i> ; usage provided in online Help only.
6.2.1 SP3	May 2016	Consolidated all Informatica Adapter documentation into one document.
6.3	August 2016	<ul style="list-style-type: none">■ Rebranded “Cisco Tidal Enterprise Scheduler (TES)” to “Cisco Workload Automation (CWA)”.■ Miscellaneous edits for the 6.3 release.



1

Introducing the Informatica Adapter

This chapter covers these topics:

- [Overview, page 7](#)
- [Prerequisites, page 8](#)
- [Software Requirements, page 8](#)

Overview

With the CWA Adapter for Informatica, organizations can add complex scheduling functions to their Informatica solution and consolidate multiple schedulers into one centralized console. The enterprise adapter, in conjunction with CWA, can be used to define, launch and monitor Informatica workflows within an Informatica PowerCenter. The adapter enhances Informatica automation and integrates seamlessly in an enterprise scheduling environment.

To use CWA's Informatica adapter, you first define connections to your Informatica environments. You then create Informatica jobs similar to other Scheduler jobs, with the distinction that the **Informatica Job Definition** dialog provides an easy to use screen that allows for selection of folders, workflows and sessions, and processing options. You monitor job execution in real time from the **Job Activity** pane.

For purposes of this document, Informatica and PowerCenter are used interchangeably.

The CWA Adapter for Informatica integrates with PowerCenter using Informatica's Load Manager SDK (a set of application programming interfaces/APIs that allows interaction with the PowerCenter Server for workflow management). Via this programming interface, the Informatica Adapter communicates with the Load Manager component of the PowerCenter to run and monitor workflows. To provide for user access to Repository data such as Folder, Workflow and Workflow Task definitions, the Informatica Adapter also requires a database connection to the PowerCenter Repository Database. Database connectivity is provided via Java Database Connectivity (JDBC) programming interface.

An Informatica job definition refers to workflows retrieved from the PowerCenter repository and provides for parameter value overrides that can refer to Scheduling variables. Based on defined scheduling criteria, the Informatica Adapter submits a PowerCenter workflow to be run directly by PowerCenter's Integration Service. Job execution is authenticated against PowerCenter through associations between CWA runtime users and PowerCenter namespaces, roles and users. Once a workflow has been submitted, the Adapter monitors the workflow and its tasks until completion. During execution, real-time workflow and task status details are visible from the **Job Activity** pane. When the job has completed, the Adapter returns its final status (used to control downstream jobs) as well as details associated with the run, to the **Job Activity** pane. Additionally, the Adapter lets users cancel, abort, recover or rerun workflows from CWA, a feature that provides significantly more control over the environment than afforded by other non-integrated scheduling solutions.

Prerequisites

The Cisco Workload Automation Adapter for Informatica integrates with PowerCenter via:

- Informatica's C-Load Manager API
- Metadata Exchange (MX), a set of relational database views for access to the PowerCenter metadata repository

Through use of this programming interface and database access, the Adapter establishes a connection with PowerCenter's Integration Service in order to manage workflow, task and workflow/session log requests. The Load Manager API includes a set of C-libraries which are platform specific and therefore requires installation and configuration of the libraries that is specific to the user environment. (Refer to [Installing and Configuring Informatica Libraries, page 9](#), later in this document for more information). Additionally, to provide Repository access to Folders, Workflows and Workflow Tasks, the Informatica Adapter also establishes a database connection to the PowerCenter Repository database associated with the Repository Service. This database connection is established on demand and is not a perpetual connection. (Refer to [Defining an Informatica Adapter Connection, page 18](#), later in this document for further details).

Cisco Workload Automation Adapters require Java 7. (Refer to *Cisco Workload Automation Compatibility Guide* for further details).

Software Requirements

CWA supports the following Informatica environment:

- PowerCenter versions 9.0 and above on Microsoft SQL Server and Oracle databases. Other databases are supported given the **Custom JDBC Driver** option. See the CWA Compatibility Matrix for information about specific version support.

Note: You will need the Database JDBC Drivers appropriate for your configuration for connectivity to the PowerCenter Repository database. Obtain the JDBC jar files from the vendor as needed. See Step 5 of [To install and configure the Informatica Libraries:, page 9](#).

For Windows installations: The Informatica Libraries requires Microsoft Visual C++ runtime components. If your system does not include these components and there is an error starting the Informatica Adapter, you may need to install the runtime components from Microsoft. You can check for the existence of file: mfc71u.dll (a component of the C++ runtime components), on your system to verify this prerequisite. The file needs to be in the system library path accessible to the adapter (e.g. either in the windows system path or accessible via the PATH environment).

Download the installer from Microsoft, as follows for the version of the Informatica Adapter being installed:Download "Microsoft Visual C++ 2008 Redistributable Package (x86)" or "Microsoft Visual C++ 2008 Redistributable Package (x64)" as appropriate. These downloads can be found from Microsoft's site given an internet search from your browser.



2

Configuring the Informatica Adapter

Overview

The Informatica Adapter software is installed as part of a standard installation of CWA. However, you must perform the following steps to license and configure the adapter before you can schedule and run Informatica jobs:

- [Installing and Configuring Informatica Libraries, page 9](#)
- [Licensing an Adapter, page 12](#)
- [Securing the Informatica Adapter, page 13](#)
- [Configuring the Master to Support Informatica TLS, page 17](#)
- [Defining an Informatica Adapter Connection, page 18](#)

See [Configuring service.props, page 47](#) for information about general and Informatica-specific properties that can be set to control things like logging and connection properties. Many of these are described in this chapter.

Note: The Informatica Adapter interacts with the Load Manager component of Power Center given a set of Power Center libraries. It interacts with the Power Center Repository metadata given database access via JDBC. [Installing and Configuring Informatica Libraries, page 9](#) describes how to install and configure the necessary libraries.

Installing and Configuring Informatica Libraries

Note: For Unix, add the following entries in user's profile located in the user's home directory. For example: `.profile` or `.bash_profile` (Linux). You need to source the profile after applying all profile updates. For example, `././profile`. Once the following steps are performed, you must restart the master for the configurations to take affect.

To install and configure the Informatica Libraries:

1. Copy the Informatica Library archive appropriate for your configuration from the CWA **InformaticaBridge** folder.

Notes:

- For 64-bit library versions, you must also run with 64-bit Java. If your Java release is not 64-bit, use the 32-bit versions of the libraries.
- To support Informatica Power Center 9.6.2 connections on a 64-bit Solaris platform, choose the 9.6.1 upgraded Solaris 64-bit library version.

2. Extract the library from the **infalib** archive to the master machine:

- a. Create a directory under the master services directory called **infa**.

Windows:

```
C:\Program Files\TIDAL\master\services\infa\
```

Unix:

```
/opt/tidal/master/services/infa
```

- b. Extract the archive to this location. The archive distribution contains directories: lib and locale. The system will be configured to refer to these locations in the next steps.

3. Configure the System Path to include the Informatica Library Path (i.e. **lib** directory):

Windows:

```
C:\Program Files\TIDAL\master\services\infa\lib
```

For Windows, include the library path in the "Path" Environment Variable.

Unix:

```
/opt/master/services/infa/lib or master/services/infa/lib
```

For Solaris/Linux, include the library path in "LD_LIBRARY_PATH".

For AIX, include the path in "LIBPATH". For 64-bit also include "LD_LIBRARY_PATH".

For HP-UX, include the path for "SHLIB_PATH". For 64-bit also include "LD_LIBRARY_PATH".

4. Create or update the **INFA_DOMAINS_FILE** Environment Variable to the location of the Informatica *domains.infa* file for the Power Center configuration.

This requires that the *domains.infa* file be local to the Master machine; copy it from your Power Center installation as needed. Put this file in the infa directory created in step 2a above).

Note: To configure connections to multiple Power Center servers, modify the local *domains.infa* file that was copied to the Master machine. Add values for the vector xml tag corresponding to each Server that will be configured as an Informatica Adapter.

The following example includes server information to two Power Center servers, one to Dev and another for Prod. These are referred to as **dev-infa** and **prod-infa**, respectively in the sample *domains.infa* file.

```
<Portals xmlns:common="http://www.informatica.com/pcsf/common"
xmlns:usermanagement="http://www.informatica.com/pcsf/usermanagement"
xmlns:domainservice="http://www.informatica.com/pcsf/domainservice"
xmlns:logservice="http://www.informatica.com/pcsf/logservice"
xmlns:domainbackup="http://www.informatica.com/pcsf/domainbackup"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:metadata="http://www.informatica.com/pcsf/metadata"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:domainconfigservice="http://www.informatica.com/pcsf/domainconfigservice"
xmlns:alertservice="http://www.informatica.com/pcsf/alertservice"
xmlns:licenseusage="http://www.informatica.com/pcsf/licenseusage"
xmlns:webserviceshub="http://www.informatica.com/pcsf/webserviceshub" xsi:type="common:PCSFVector"
objVersion="1.1.19">
<vector xsi:type="domainservice:Portals" objVersion="1.1.19">
<domainName>Domain_dev-infa</domainName>
<address xsi:type="metadata:NodeRef" objVersion="1.1.19">
<host>dev-infa</host>
<port>6001</port>
</address>
</vector>
<vector xsi:type="domainservice:Portals" objVersion="1.1.19">
<domainName>Domain_prod-infa</domainName>
<address xsi:type="metadata:NodeRef" objVersion="1.1.19">
<host>prod-infa</host>
```

Installing and Configuring Informatica Libraries

```
<port>6001</port>
</address>
</vector>
</Portals>
```

Example on Windows:

```
INFA_DOMAINS_FILE=C:\Program Files\TIDAL\master\services\infa\domains.infa
```

Example on Unix:

```
export INFA_DOMAINS_FILE=/opt/TIDAL/master/services/infa/domains.infa
```

5. Configure the Locale Path for the Informatica Library by setting the **TDLINFA_LOCALE** *service.props* value of the Informatica Adapter as described below. See [Configuring service.props, page 47](#) below for more details.

In the **config** directory located under the Adapter's GUID directory, create or update the *service.props* file, create both the directory and file if it does not yet exist. Include an entry for **TDLINFA_LOCALE** that points to the **Load Manager Library** locale directory.

Windows:

```
C:\Program
Files\TIDAL\Scheduler\master\services\{7640B420-5530-11DE-8812-7B8656D89593}\config\service.props
```

Unix:

```
/opt/tidal/master/services/{7640B420-5530-11DE-8812-7B8656D89593}/config/service.props
```

service.props entry example:

Windows:

```
TDLINFA_LOCALE=C:\\Program Files\\TIDAL\\Scheduler\\master\\services\\infa\\lib\\locale
```

Unix:

```
TDLINFA_LOCALE=/opt/tidal/master/services/infa/lib/locale
```

6. You need access to the Database JDBC Drivers for connectivity to the Power Center Repository database. Obtain the JDBC jar files from the vendor as needed and copy the corresponding *.jar* files to the services **lib** directory.

Windows:

```
C:\Program Files\TIDAL\Scheduler\master\services\{7640B420-5530-11DE-8812-7B8656D89593}\lib
```

Unix:

```
/opt/tidal/master/services/{7640B420-5530-11DE-8812-7B8656D89593}/lib
```

7. Edit the adapter *service.props* file to help control logout issues:

- a. In a text editor, open the **service.props** file in the `\config` directory where the CWA files are installed.

- b. Optionally, add one or more of these parameters on separate line(s) in the file:

DISCONN_ON_LOSTCONN=Y	Setting this flag to Y avoids an unnecessary logout call to the Informatica server when the connection is lost. This logout call usually hangs.
OUTPUT_ASYNC_LOGOUT=Y	Setting this flag to Y avoids jobs getting stuck in Gathering Output status.
SCANFOR_SESSIONSTATS=N	Setting this parameter to N turns off the default behavior of Informatica jobs collecting the session statistics during the job run.
SCANFOR_SESSIONSTATS_AFTER_WF_ENDS=Y	Setting this parameter to Y turns off the gathering of session statistics during each poll for the status of Informatica jobs.

- c. Save the changes and close the service.props file.

Note: See [Configuring service.props, page 47](#) for documentation about these and other properties that can be set using service.props.

8. Reboot the Master machine on Windows as needed. Source the profile file as needed on Unix.
9. Restart the Master.

Licensing an Adapter

Each CWA Adapter must be separately licensed. You cannot use an Adapter until you apply the license file. If you purchase the Adapter after the original installation of CWA, you will receive a new license file authorizing the use of the Adapter.

You might have a Demo license which is good for 30 days, or you might have a Permanent license. The procedures to install these license files are described below.

To license an Adapter:

1. Stop the master:

Windows:

- a. Click on **Start** and select **All Programs>Cisco Workload Automation>Scheduler>Service Control Manager**.
- b. Verify that the master is displayed in the **Service** list and click on the **Stop** button to stop the master.

UNIX:

Enter **tesm stop**

2. Create the license file:

- For a Permanent license, rename your Permanent license file to *master.lic*.
- For a Demo license, create a file called *demo.lic*, then type the demo code into the *demo.lic* file.

3. Place the file in the **C:\Program Files\TIDAL\Scheduler\Master\config** directory.

4. Restart the master:

Windows:

Click **Start** in the Service Control Manager.

UNIX:

Enter **tesm start**

The master will read and apply the license when it starts.

5. To validate that the license was applied, select **Registered License** from **Activities** main menu.

Securing the Informatica Adapter

There are three types of users associated with the Informatica Adapter, Database Users, Runtime Users and Schedulers. You maintain user definitions from the **Users** pane.

■ Database Users

To access Power Center metadata such as folder and workflow definitions, the Informatica Adapter connects to the Power Center Repository database as a database user. There is only one database user per Informatica Adapter Connection, and this user must have the appropriate permissions to access the Power Center metadata definitions. In particular, this user must have read access to the following Power Center database views: REP_SUBJECT, REP_WORKFLOWS, REP_TASK_INST, REP_WORKFLOW_DEP, REP_TASK_INST_RUN. (Refer to [Defining an Informatica Adapter Connection](#), page 18 for more details).

■ Runtime Users

Runtime users, in the context of Informatica jobs, represent those namespaces, users and passwords required for authentication. Informatica operations require authentication against a valid Informatica user as defined by an Informatica administrator. Refer to the Power Center documentation for more information on managing object access.

■ Schedulers

Schedulers are those users who will define and/or manage Informatica jobs. To secure scheduling operations by user, please use CWA's security features as explained later in this document. There are three aspects of a user profile that grant and/or limit access to scheduling jobs that effect Informatica:

- Security policy that grants or denies add, edit, delete and view capabilities for Informatica jobs.
- Authorized runtime user list that grants or denies access to specific authentication accounts for use with Informatica jobs.
- Authorized agent list that grants or denies access to specific Informatica Adapter connections for use when defining Informatica jobs.

Note: If you want the CWA to use FTP to retrieve Informatica workflow and session logs, you will also need to provide an FTP user.

Defining Informatica Users

To define a Informatica user:

1. From the **Navigator** pane, expand the **Administration** node and select **Runtime Users** to display the defined users.
2. Right-click **Runtime Users** and select **Add Runtime User** from the context menu (*Insert* mode).

-or-

Click the **Add** button  on the CWA menu bar.

The **User Definition** dialog displays.

3. Enter the new user name in the **User Name** field.
4. For documentation, enter the Full Name or description associated with this user.
5. In the **Domain** field, select a Windows domain associated with the user account required for authentication, if necessary.
6. To define this user as a runtime user for Informatica jobs, click **Add** on the **Passwords** tab.

The **Change Password** dialog displays.

7. Select **Informatica** from the **Password Type** list.
8. Enter a password (along with confirmation) in the **Password/Confirm Password** fields.

Only those users with a password specified for Informatica will be available for use with Informatica jobs. The password might be the same as the one specified for Windows/FTP jobs.

9. Click **OK** to return to the **User Definition** dialog.

The new password record displays on the **Passwords** tab.

10. Click **OK** to add or save the user record in the CWA database.

Note: For more information about the User Definition dialog, see Users in your *Cisco Workload Automation User Guide*.

Authorizing Schedulers to Work with Informatica Jobs

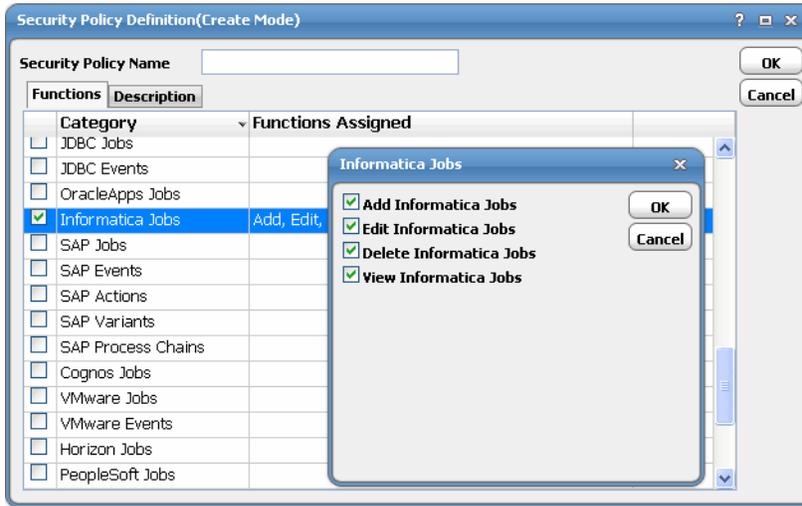
Authorizing schedulers involves two tasks:

- Defining a security policy that authorizes user access to Informatica jobs.
- Defining Scheduler users to work with Informatica jobs.

To define a Security Policy that authorizes access to Informatica jobs:

1. From the **Navigator** pane, select **Administration>Security Policies** to display the **Security Policies** pane.
2. Right-click **Security Policies** and select **Add Security Policy** from the context menu. You can also right-click a defined security policy in the **Security Policies** pane and select **Edit Security Policy**.

The **Security Policy Definition** dialog displays.



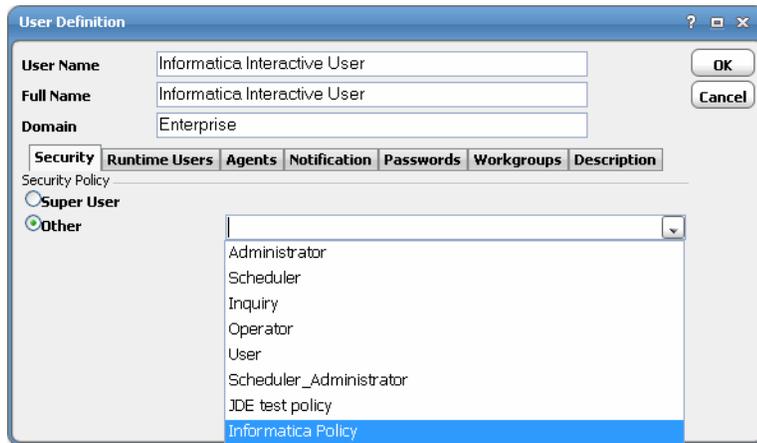
Note: Refer to the *Cisco Workload Automation User Guide* for a general discussion on setting up security policies that you associate with Scheduler Users.

3. In the **Security Policy Name** field, enter a name for the policy.
4. On the **Functions** page, scroll to the **Informatica Jobs** category, double-click the Informatica Job section, then click the check boxes next to the functions that are to be authorized under this policy (**Add**, **Edit**, **Delete** and **View Informatica Jobs**).
5. Click **OK** on the **Informatica Jobs** dialog.
6. Click **OK** to save the policy.

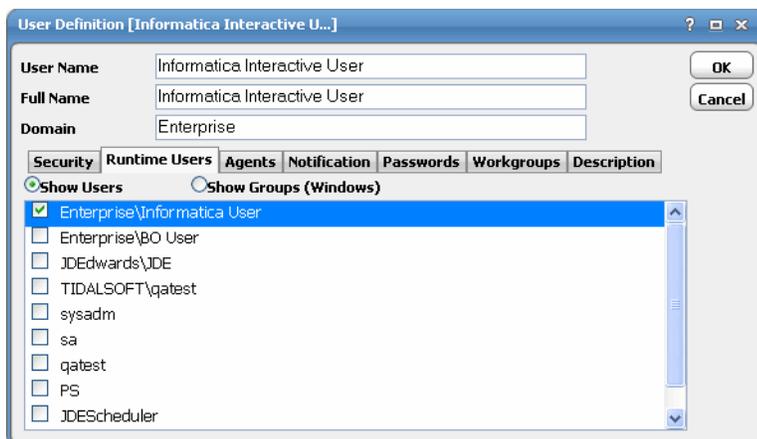
To define a Scheduler user to work with Informatica jobs:

1. From the **Navigator** pane, expand the **Administration** node and select **Interactive Users** to display the defined users.
2. Right-click **Interactive Users** and select **Add Interactive User** from the context menu (*Insert* mode). You can also right-click a user in the **Interactive Users** pane and select **Edit Interactive User** from the shortcut menu (*Edit* mode).

The **User Definition** dialog displays.



3. If this is a new user definition, enter the new user name in the **User Name** field.
4. For documentation, enter the **Full Name** or description associated with this user.
5. In the **Domain** field, select a Windows domain associated with the user account required for authentication, if necessary.
6. On the **Security** page, select the **Other** option and then select the security policy that includes authorization for Informatica jobs.
7. Click the **Runtime Users** tab.



8. Select the Informatica users that this scheduling user may use for Informatica authentication in Informatica jobs.
9. Click the **Agents** tab.
10. Select the check boxes for the Informatica connections that this scheduling user can access when scheduling jobs.
11. Click **OK** to save the user definition.

Configuring the Master to Support Informatica TLS

The Informatica Adapter in CWA 6.3 supports the Informatica 9.6.1 TLS-enabled environment for secure communication between the Informatica Adapter and the Informatica 9.6.1 Power Center.

Warning: 9.6.1 Informatica libraries are not backward compatible. These libraries can only connect to Informatica 9.6.1 Power Center with or without enabling TLS.

To accommodate this feature, you must configure the CWA Master for Windows and Unix environments as described in these sections:

- [Configuring the Master for Informatica 9.6.1 TLS for Windows](#)
- [Configuring the Master for Informatica 9.6.1 TLS for Unix](#)

Configuring the Master for Informatica 9.6.1 TLS for Windows

To configure the Master to support Informatica 9.6.1 TLS for Windows:

1. Stop the Master.
2. Copy the truststore files from the Informatica server's security directory to the Master c:\infa [Informatica libraries in the CWA Master]. Create a folder truststore and place the copied truststore files infa_truststore.jks and infa_truststore.pem in it.

```
<INFA_HOME>\9.6.1\services\shared\security
```

to

```
C:\infa\truststore
```

3. Create an environment variable:

```
INFA_TRUSTSTORE value: C:\infa\truststore
```

4. Optionally, if custom truststore files are used, then perform the following steps:

- a. If a custom truststore with a custom SSL certificate is used, then copy the custom infa_truststore.jks and infa_truststore.pem files.
- b. Generate the encrypted Informatica truststore password for the infa_truststore.jks with the pmpasswd utility as follows:

```
<INFA_HOME>/server/bin/pmpasswd <password_for_infa_truststore.jks_file> -e CRYPT_SYSTEM
```

Or..

```
<INFA_CLIENT_HOME>\clients\PowerCenterClient\client\bin\pmpasswd.exe  
<password_for_infa_truststore.jks_file> -e CRYPT_SYSTEM
```

- c. Set the following environment variables along with the INFA_DOMAINS_FILE variable so that it is available to the Informatica Adapter process:

```
INFA_TRUSTSTORE // variable value with the directory that contains the truststore files named  
infa_truststore.jks and infa_truststore.pem  
INFA_TRUSTSTORE_PASSWORD // password encrypted with the pmpasswd utility
```

5. Add <tlsEnabled>true</tlsEnabled> in domains.infa xml, corresponding to the TLS-enabled Server that will be configured as an Informatica Adapter connection.

6. Run the test.bat in c:\infa.
7. Restart the Master.

Configuring the Master for Informatica 9.6.1 TLS for Unix

To configure the Master to support Informatica 9.6.1 TLS for Unix:

1. Stop the Master
2. Copy the truststore files from the Informatica server's security directory to the Master /opt/infa [Informatica libraries in the CWA Master]. Create a folder truststore and place the copied truststore files infa_truststore.jks and infa_truststore.pem in it.

```
<INFA_HOME>\9.6.1\services\shared\security
```

to

```
/opt/infa/truststore
```

3. Create an environment variable in the .profile:

```
INFA_TRUSTSTORE value: /opt/infa/truststore
```

4. Optionally, if custom truststore files are used, then follow these steps:

- a. If a custom truststore with a custom SSL certificate is used, then copy the custom infa_truststore.jks and infa_truststore.pem files.
- b. Generate the encrypted informatica truststore password for the infa_truststore.jks with the pmpasswd utility as follows:

```
<INFA_HOME>/server/bin/pmpasswd <password_for_infa_truststore.jks_file> -e CRYPT_SYSTEM
```

Or...

```
<INFA_CLIENT_HOME>\clients\PowerCenterClient\client\bin\pmpasswd.exe  
<password_for_infa_truststore.jks_file> -e CRYPT_SYSTEM
```

- c. Set the following environment variables along with the INFA_DOMAINS_FILE variable so that it is available to the Informatica Adapter process:

```
INFA_TRUSTSTORE // variable value with the directory that contains the truststore files named  
infa_truststore.jks and infa_truststore.pem  
INFA_TRUSTSTORE_PASSWORD // password encrypted with the pmpasswd utility
```

5. Add <tlsEnabled>true</tlsEnabled> in domains.infa.xml, corresponding to the TLS-enabled Server that will be configured as an Informatica Adapter connection.
6. Run the test.sh in /opt/infa.
7. Restart the Master.

Defining an Informatica Adapter Connection

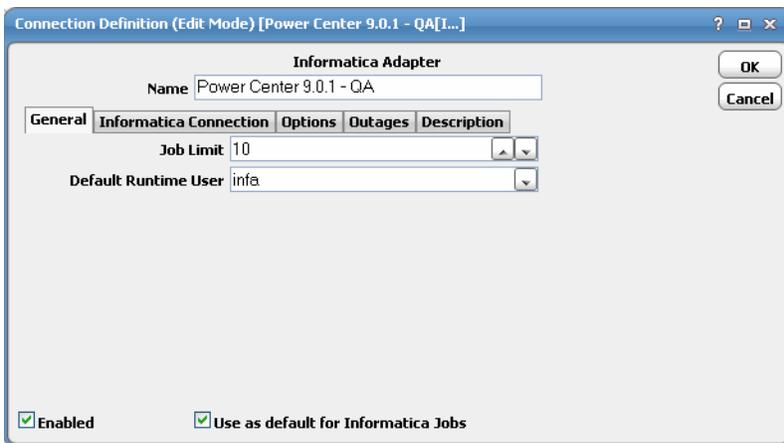
You must create at least one Informatica adapter connection to an integration service before CWA can run your Informatica jobs. These connections must also be licensed before CWA can use them. A connection is created using the **Connection Definition** dialog.

Adding an Informatica Adapter Connection

To add a connection:

1. From the **Navigators** pane, navigate to **Administration>Connections** to display the **Connections** pane.
2. Right-click **Connections** and select **Add Connection>Informatica Adapter** from the context menu.

The **Informatica Adapter Connection Definition** dialog displays.



3. On the **General** page, enter a name for the new connection in the **Name** field.
4. In the **Job Limit** field, select the maximum number of concurrent active processes that CWA should submit to the Informatica server at one time.
5. (Optional) From the **Default Runtime User** drop-down list, select the name of a default runtime user automatically assigned to a job when this connection is selected during job definition.

When a job has a runtime user assignment, the job is executed using the security profile of that user. In order to run workflows, the runtime user must be a Power Center user who has execute privileges for RunTime Objects as well as execute privileges for the folder containing the workflow. Refer to the Power Center documentation for more information on managing object access.

6. (Optional) Select the **Use as default for Informatica Jobs** option if you wish for this connection to be used as the default whenever creating Informatica jobs.

7. Click the **Informatica Connection** tab.

The **Repository Database** tab specifies configuration information needed to connect to the Power Center Repository Database associated with the specific Repository Service that is used to submit workflows. Some of this information is available via the Informatica Administration Console. See your Power Center Administrator for further information.

Note: The Repository Database configuration is required in order to save the connection definition, however a successful database connection does not prevent a user from defining an Informatica job. As previously mentioned, the Repository Database information is used to connect to the Power Center Repository Database for access to Folder and Workflow definitions, used when defining Informatica jobs. This is a convenience feature that allows for ease of use during job definition, by providing picklists for Folder and Workflow selections. However, a user may supply their own inputted values for these fields, if known. Furthermore, Folder and Workflow values may be a user-defined variable that is resolved at runtime. While the Repository Database connection fields are required in order to save an Informatica Connection, they are not required for job definition or execution. Therefore a connection may appear healthy, denoted as green, even if a database connection can not be established.

8. From the **Database Type** list, select the database type for the Repository Database. (Current database support is for Microsoft SQLServer and Oracle.)
9. Select the **Custom Jdbc Driver** option to connect to a Power Center repository that is not MSSql or Oracle. Once selected, the Database Type field allows you to input the appropriate Jdbc driver. The **Override JDBC URL** checkbox is then selected where input for the connection url is required. Following are examples of common Jdbc driver and connection strings:

Database	Jdbc Driver	Connection URL
MSSql	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<HOST>:<PORT>;databaseName=<DATABASENAME>
Oracle Thin	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<HOST>:<PORT>:<DATABASENAME>
Sybase (jConnect 4.2 and earlier)	com.sybase.jdbc.SybDriver	jdbc:sybase:Tds:<HOST>:<PORT>?SERVICENAME=<DATABASENAME>
DB2	COM.ibm.db2.jdbc.app.DB2Driver	jdbc:db2://<HOST>:<PORT>/<DATABASENAME>

10. In the **Database Host Name** and **Database Port** fields, enter the database host name and port for the Repository Database.

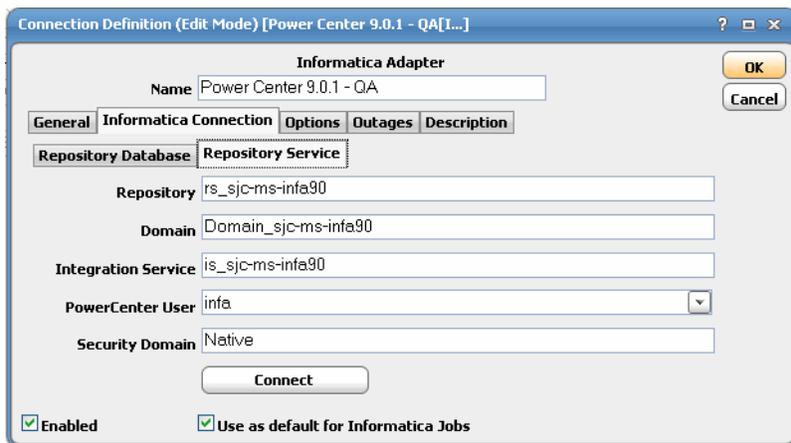
Defining an Informatica Adapter Connection

11. From the **Database User** list, select the Informatica Database user that is used to connect to the Repository Database. This user should have been created in an earlier step.
12. In the **Database Name** field, supply the database name or SID (for Oracle). You can find the SID name in your *tnsnames.ora* file.
13. Once the database fields have been supplied, a jdbc connection URL will be automatically generated, however this URL may be overridden via the **Override Jdbc URL** text field.
14. Click the **Connect** button to test the connection to the Repository database.

Note: The Connect button activates when all required fields are populated. When a successful connection has been made, the button will switch to Connected and will be grayed out. If the connection fails an Information dialog such as the following displays showing the connection url and user account that had failed to establish a successful connection.



15. Click the **Repository Service** tab to specify the Repository, Domain, Integration Service and User for workflow submissions.

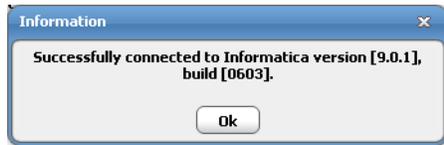


16. In the **Repository** field, supply the name of the Power Center Repository Service.
17. In the **Domain** field, supply the name of the Power Center domain.
18. In the **Integration Service** field, provide the name of the Integration Service that will be running workflows.
19. From the **Power Center User** list, select the Power Center user account authorized for submitting workflows. Refer to the Power Center documentation for more information on management of users and roles. This user should have been created in an earlier step.
20. (Optional) In the **Security Domain** field, enter the security domain of the user account used to log into Power Center.

Note: If the user's security domain is " Native" , no value is required.

21. Once all required fields have been supplied the **Connect** button becomes enabled. Use the **Connect** button to test the connection to the repository service. If successfully connected an information dialog displays specifying the Power Center version and build and the **Connect** button becomes disabled.

Upon successful connection the following dialog displays:



Connecting to the Integration Service

There are two ways to connect to the Integration Service. The recommended approach, as described in the proceeding step, is to use the Integration Service Name and Domain Name. Connecting in this manner requires the definition of the INFA_DOMAINS_FILE environment variable as specified in a previous step under installation and configuration. This connection approach is the current approach and strongly advised as per Informatica.

An alternate approach, and for backwards compatibility, is to provide the Power Center Host Name and Integration Service Port. This method of connecting may be necessary if you are using the adapter to connect to two or more Power Center environments, but the Power Center domain is not uniquely identified in the INFA_DOMAINS_FILE. For example, both DEV and PROD Power Center environments have the same domain name. Connecting in this manner does not require the definition of the INFA_DOMAINS_FILE environment variable, but is not recommended as it requires the Integration Service Port, and starting with Power Center 8.x, this port changes each time the service is restarted. The port number can be determined by viewing the log entries of the Integration Service from the Power Center Administration Console and looking for the entry:

```
"Opened Service Port [xxxx] to listen for Client connections."
```

However, when using this method to connect, we want to ensure that the Integration Service will always start on a specific port for consistency in connectivity. In Power Center 8.1.x and above, the "ServerPort" custom property can be configured for this purpose.

To define a constant port number of the Integration Service:

1. From the Power Center Administration Console select the **Domain** tab.
2. Highlight the Integration Service.
3. On the **Properties** tab, select the **Custom Properties** section and click **Edit**.
4. Add the `ServerPort` property and the set the port value to any available port.
5. To define a connection using this alternate method, specify the following Configuration Parameters in the Connection definition, **Options** tab.
 - `SERVER_NAME` - Enter the Power Center Host Name
 - `SERVER_PORT` - Enter the Integration Service Port

If there is a failure connecting the repository service, an **Error** dialog displays.



3

Using the Informatica Adapter

Overview

This chapter guides you through using the features of the Informatica Adapter with CWA, including:

- [Defining Informatica Jobs, page 23](#)
- [Monitoring Informatica Jobs, page 33](#)
- [Controlling Informatica Tasks and Workflows, page 38](#)
- [Controlling Adapter and Agent Jobs, page 40](#)

Defining Informatica Jobs

This section provides instructions for defining an Informatica job in CWA.

Informatica Job Definition

To define an Informatica job:

1. In the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click **Jobs** and select **Add>Informatica Job** from the context menus.

The **Informatica Job Definition** dialog displays. The **Run** tab is selected by default.

The screenshot shows the 'Informatica Job Definition' dialog box with the 'Run' tab selected. The 'Informatica Job Name' is 'wf_NestedWorklets'. The 'Job Class' is empty. The 'Parent Group' is '\PowerCenter 8.6.1 Group'. The 'Owner' is 'Schedulers'. The 'Agent/Adapter Name' is 'Power Center 9.0.1 - QA'. The 'Agent List' is empty. The 'Runtime User' is 'infa'. The 'Duration' section shows 'Estimated' as 1:07, 'Minimum' as 1:00, and 'Maximum' as 1:00. The 'Exclude Completed Abnormally' checkbox is checked. The 'Inherited' checkbox is also checked. The dialog is titled 'Informatica Job Definition [wf_NestedWorklets[\PowerC...]' and has 'OK' and 'Cancel' buttons.

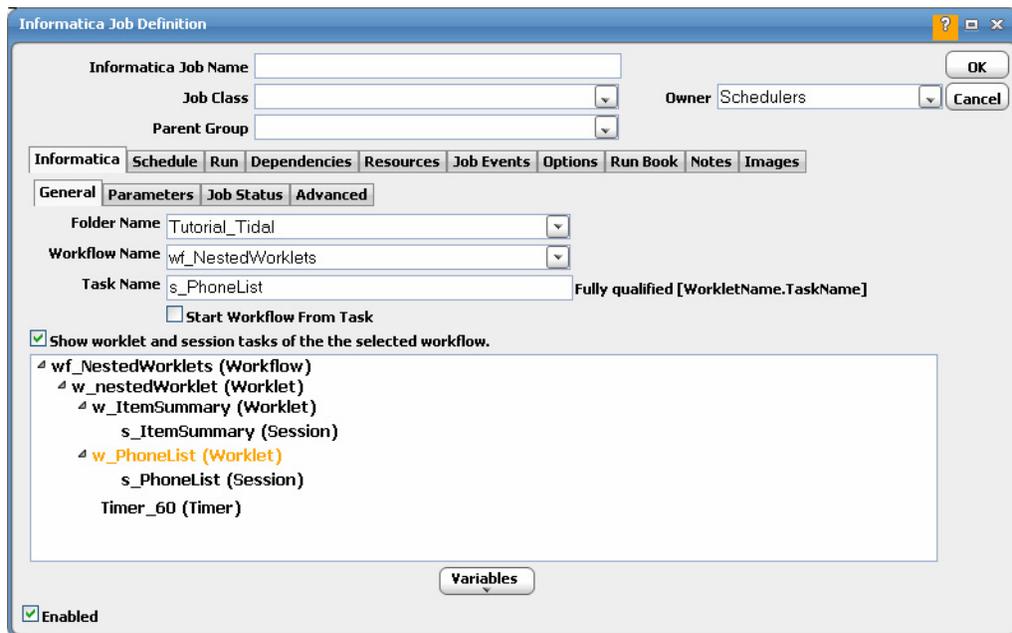
3. In the upper portion of the dialog, specify the following information to describe the job:

- **Job Name** – Enter a name that describes the job.
- (Optional) **Job Class** – If you want to assign a defined job class to this job, select it from the drop-down list.
- **Owner** – Select the user or group name from the drop-down list for the person who owns this job.
- **Parent Group** – If this job should be inside a job group, select the name of the parent group from the drop-down list. All properties in the Agent Information section are inherited from its parent job group.

4. On the **Run** tab, specify the following connection information in the **Agent/Adapter Information** section:

- **Agent/Adapter Name** – Select the Informatica adapter connection to be used for this job from the drop-down list.
- **Runtime User** – Select a valid runtime user with the appropriate Informatica authority for the job from the drop-down list. The user must have the appropriate Informatica authority for the operation.

Specify the appropriate **Tracking** and **Duration** information for the job. Refer to the *Cisco Workload Automation User Guide* for information on these options.

5. Click the **Informatica** tab.6. On the **General** tab, select the folder containing the workflow from the **Folder Name** list.

Alternatively, you can enter your own value if known or supply a user-variable by clicking the **Variables** button and selecting a variable from the list.

7. From the **Workflow Name** list, select the workflow to schedule.

Alternatively, you can enter your own value if known or supply a user-variable by clicking the **Variables** button and selecting a variable from the list.

Once a folder/workflow selection has been made, to display the worklets and sessions for the workflow in the tree view, select the **Show worklet and session tasks of the selected workflow** checkbox. This is selected by default, but if you do not wish for tasks to be loaded the next time this job is viewed, you may deselect it. Refer to the PowerCenter documentation for more information on workflow task types.

Note: When using Scheduler variables for either the Folder Name or Workflow Name, the Show worklet and session tasks of the selected workflow option will be deselected and disabled and not workflow task details displayed. This is because in order to display workflow tasks information, both the Folder and Workflow name needs to resolved to their true values, which occurs only during job runtime.

8. (Optional) If you want to schedule this workflow to run a single task or to start from a task, the **Task Name** field is required.

Enter the fully qualified task name to run or start from. Alternatively, you can select the task from the tree view and it will automatically populate. If you are running the entire workflow then leave this field blank.

Note: The fully qualified task name includes the hierarchy structure. If the task is part of a workflow, then include the task name only. If the task is part of a worklet, include the worklet and the task name separated by periods. Include the entire hierarchy separated by “.” For example, <WorkletName.TaskName>.

Select the **Start Workflow From Task** checkbox if you want to run the workflow starting from a specific task. If you are running a single workflow task, leave this deselected. See [Working with the Tree View, page 26](#) for information about starting a workflow from a specific task or running a single task of the workflow.

Working with the Tree View

Two task options are available when selecting a worklet or task and using the context menu associated with the tree view:

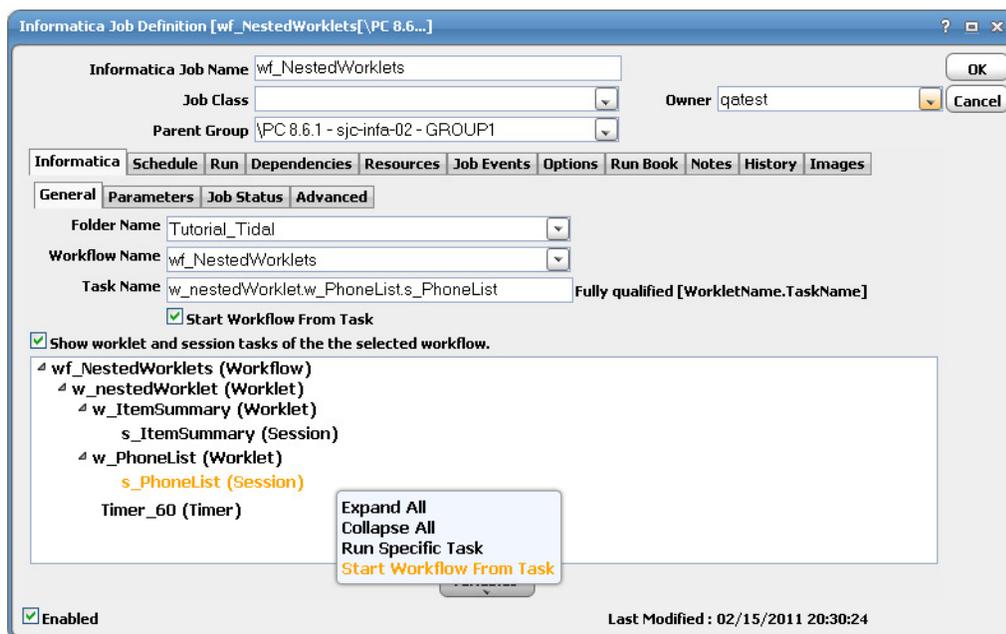
- [Start Workflow from a Task, page 26](#) - The workflow begins with the specified task and continues with the remaining tasks.
- [Run a Specific Task, page 27](#) - Run a single task of the workflow.

Start Workflow from a Task

The workflow begins with the specified task and continues with the remaining tasks.

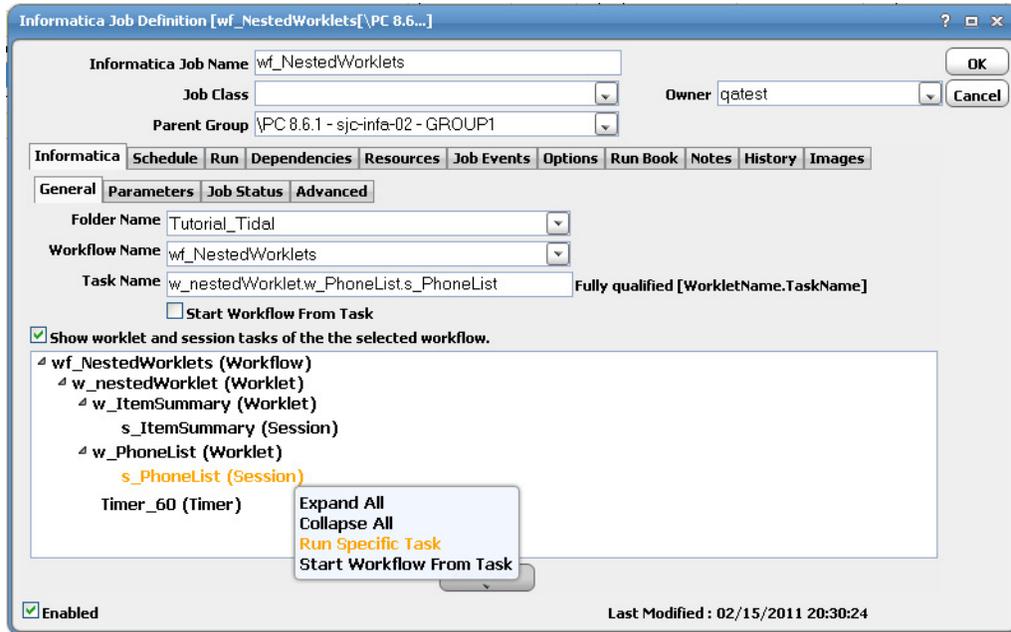
When selected through the tree view, the fully-qualified **Task Name** field is automatically populated and the **Start Workflow From Task** option is automatically selected, as applicable.

The figure below is an example of the **Start Workflow From Task** option. When this workflow runs, it will run from session task `s_PhoneList` and continue with any remaining tasks.



Run a Specific Task

The figure below is an example of the **Run Specific Task** option. When this workflow runs, it will run only the session task `s_PhoneList`.



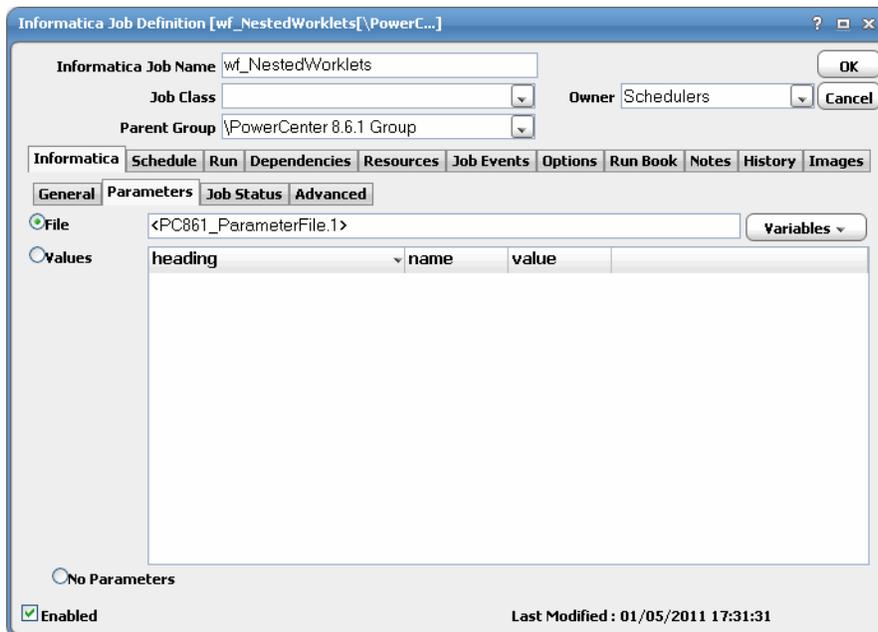
The following options are also available through the corresponding popup menu as shown above:

- **Expand All** - All items of the tree are displayed with the individual tasks that comprise the workflow. Expanded workflows/tasks display a minus icon in front of the workflow/task name.
- **Collapse All** - Workflows are displayed without displaying the individual tasks that comprise workflow. Collapsed workflows display a plus icon in front of the workflow name.

Specifying Informatica Job Parameters

To specify Informatica job parameters:

1. Click the **Parameters** tab.



2. (Optional) Override parameters by selecting one of the following options:

- **File** – Enter the name of the parameter file containing the workflow parameters. The file should be an absolute path name appropriate to the operating system where the Integration Service runs. This file should adhere to the parameter file format defined in the PowerCenter documentation.

- **Values** – Specify individual parameter value overrides.

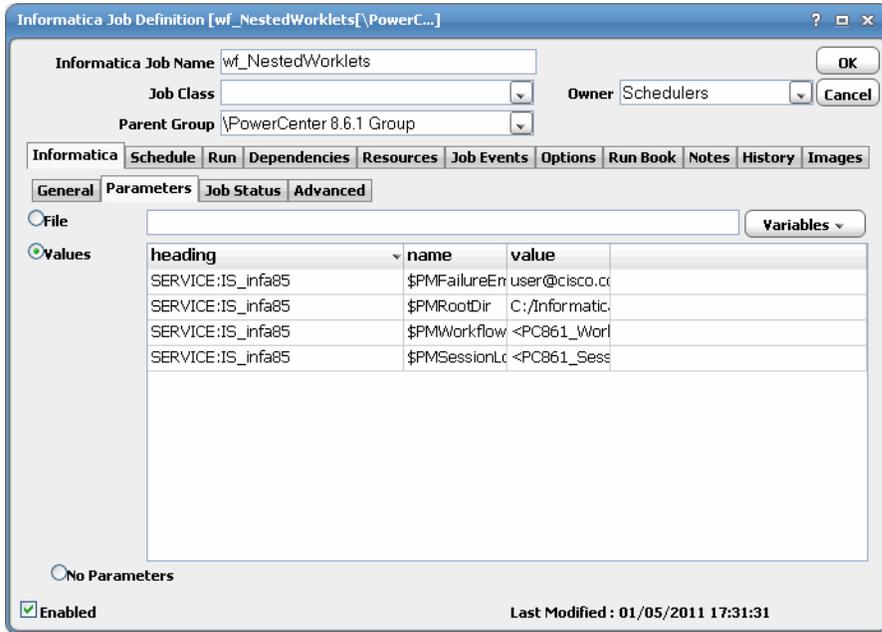
Note: You can also insert Scheduler variables as needed by clicking the Variables button and selecting the variable from the list.

- **No Parameters** – Select this option if you do not want to specify any parameters. This is the default.

The previous figure is an example of parameter overrides of the workflow's parameter file whose file name value is resolved given a CWA variable definition.

Defining Informatica Jobs

The following figure is an example of parameter overrides of individually specified parameter values, which include both static definitions and CWA Variables.



Just as with a parameter file, when specifying parameter values, parameters are specified in a section or heading, with name/value pairs. Refer to the PowerCenter documentation for more information on Parameter File Structure.

Checking the Informatica Job Status

To specify how job status is determined:

1. Click the **Job Status** tab.

The screenshot shows the 'Informatica Job Definition [389_infa3]' dialog box. The 'Job Status' tab is selected. The 'Job Status Determination Method' section has several radio button options: 'Workflow Status Only' (selected), 'Any Error Code Encountered', 'Session Status Only', 'Both Workflow and Session Statuses', 'Source, Target or Transformation Errors', 'Zero Source or Target Rows Processed', and 'Match Target Rows Processed against Source Rows Processed'. There is also a checked checkbox for 'Treat Workflow status of TERMINATED as final'. At the bottom left, there is a checked checkbox for 'Enabled'. At the bottom right, it says 'Last Modified : 12/09/2015 17:20:47'. The dialog box has 'OK' and 'Cancel' buttons in the top right corner.

The default is **Workflow Status Only**.

For the following options, an exit code is generated that specifies the type of error encountered and the job completes abnormally:

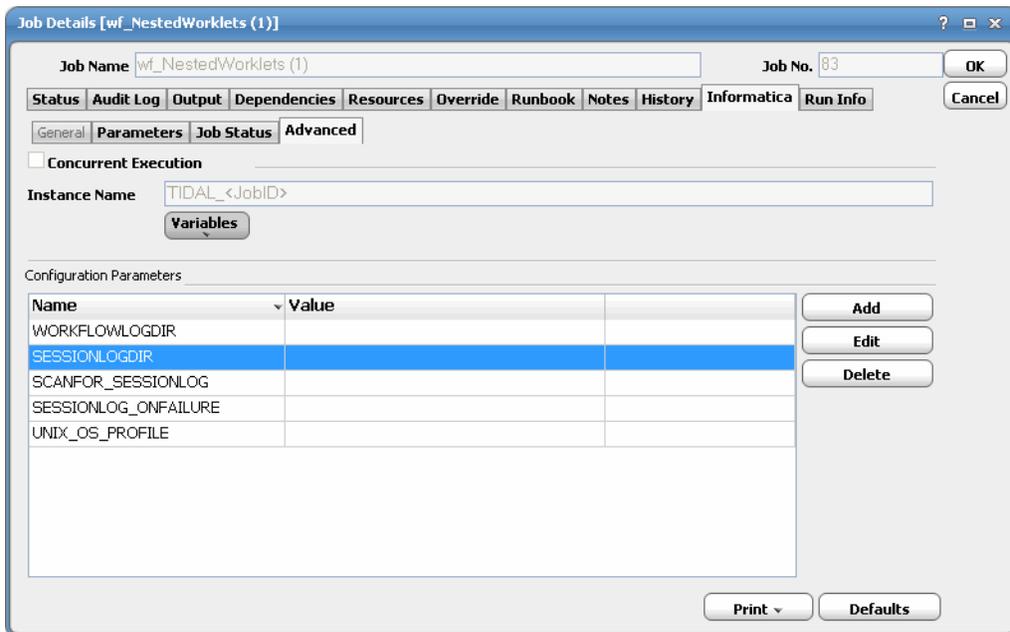
- (Default) **Workflow Status Only**– Job status is the workflow status
- **Any Error Code Encountered** – Job’s exit code is the first error encountered and job is set to abnormal if non-zero.
- **Session Status Only** – Job status is based on the status of the workflow’s sessions. If any session fails, the job completes abnormally.
- **Both Workflow and Session Statuses** – Job status is determined given both workflows and sessions. The job completes abnormally if either the workflow fails or any of its sessions fail.
- **Source, Target or Transformation Errors** – Job status is abnormal given source, target or transformation errors.
- **Zero Source or Target Rows Processed** – Job status is abnormal if zero source or target rows specified.
- **Match Target Rows Processed against Source Rows Processed** – Job status is abnormal if target rows processed is less than source rows.
- **Treat Workflow status of TERMINATED as final** – If checked, if any one of the workflow statuses is terminated, then the terminated status is considered to be the final job run status and the job execution ends. By default, this option is checked.

Controlling the Workflow

You can control the Informatica job workflow. For example, you can specify that jobs run concurrently or specify parameters that override the connection definition.

To specify optional advanced features for running the workflow:

1. Click the **Advanced** tab.



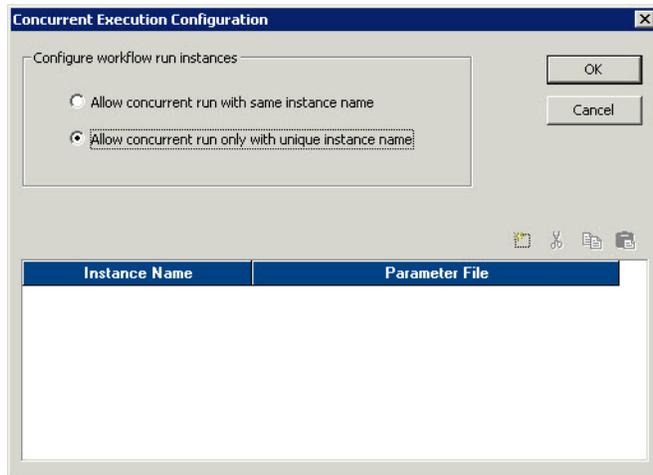
2. When scheduling a concurrent workflow from the Informatica Adapter, select the **Concurrent Execution** checkbox and ensure the **Instance Name** provided is unique, as recommended by Informatica.

You may specify a static value or use the **Variables** button to specify a variable. You will not be able to specify an Instance Name unless you select the checkbox. If you want the Informatica Adapter to automatically generate a unique Instance Name, select the **Concurrent Execution** option and leave the **Instance Name** field blank. When the job is saved, the Instance Name of **TIDAL_<JobID>** is replaced with the jobrun id upon job execution.

If the Instance Name is not **TIDAL_<JobID>**, then during the job save, an **Information** dialog displays reminding you of the recommendation pertaining to concurrent workflows.

3. Click **OK** to dismiss the dialog and continue with the save or click **Cancel** to verify a unique Instance Name.

Note: For Workflows configured for concurrent execution, it is recommended that the option "Allow concurrent run only with unique instance name" be used (see figure below). Refer to PowerCenter documentation for more information.



4. (Optional) In the **Configuration Parameters** section, specify additional job parameters as shown in this example.

Configuration Parameters	
Parameter Name	Value
WORKFLOWLOGDIR	y
SESSIONLOGDIR	\\hou-infa-02\infa_shared\SessLogs
SCANFOR_SESSIONLOG	\\hou-infa-02\infa_shared\WorkflowLogs
UNIX_OS_PROFILE	
*	

If included, these options override the corresponding options that had been specified in the connection definition.

For performance, you might configure the Informatica Adapter to retrieve workflow and session logs using FTP (See [Adding an Informatica Adapter Connection, page 19](#)) or read them directly on the PowerCenter server rather than requesting them from PowerCenter.

The following parameters are available for this purpose:

Informatica Connection Parameters	Description
IGNORE_CODES	This parameter can be set in service.props, job configuration and connection configuration parameters. The order of precedence is service.props (applicable for all jobs running in all connections), job level (only for that particular job), and connection (applicable for all jobs in the connection). This parameter is used to specify Informatica-specific error codes, separated by commas (,), that you want to ignore while running a job.
LAUNCH_DELAY (in milliseconds)	This parameter can be set in service.props, job configuration and connection configuration parameters. The order of precedence is service.props (applicable for all jobs running in all connections), job level (only for that particular job), and connection (applicable for all jobs in the connection). If a non-zero value is set for this parameter, then the jobs are delayed for the x milliseconds before being submitted to Informatica.
SCANFOR_SESSIONLOG	A value of <code>y</code> or <code>y</code> specifies that Session Log file names will be determined by scanning for their values in the workflow log. By default, the Informatica Adapter does not read the log files directly, but requests them from PowerCenter (currently not recommended). You must also specify <code>SESSIONLOGDIR</code> to use this option.

Informatica Connection Parameters	Description
SESSIONLOGDIR	Specifies a UNC or NFS path to the location of the session logs (relative to the master) if not using FTP for output. If you are going to use FTP, the path is an absolute or relative path based on the home directory of the chosen FTP logon. Because PowerCenter may return the wrong session log given a request, configure the Informatica Adapter to scan for the session log name from the workflow log. The following option is designed for this purpose.
SESSIONLOG_ONFAILURE	A value of \mathbf{Y} or \mathbf{y} indicates that session logs for failed sessions will be included in job output even if "Summary Only" is in effect for job output.
SKIP_SERVERSTATS	Set this parameter to \mathbf{Y} to prevent reading of server statistics from the Server during connect and during each connection poll.
UNIX_OS_PROFILE	Enter the Unix operating system profile used to run the workflow. This applies to PowerCenter for Unix. If the Integration Service uses operating system profiles, it runs workflows with the settings of the operating system profile assigned to the workflow or to the folder that contains the workflow. Refer to the PowerCenter documentation for more information on managing Operating System Profiles.
WORKFLOWLOGDIR	Specifies a UNC or NFS path to the location of the workflow logs (relative to the master) not using FTP for output. If you are going to use FTP, the path is an absolute or relative path based on the home directory of the chosen FTP logon.

Note: These options will override the same options specified at the connection level.

5. Click the **Options** tab of the **Job Definition** dialog to configure output settings.

Save Output Options provides for output configuration. By default, the job is set to "Discard" output. You may change this option to "Append" or "Replace". For Informatica jobs, output consists of the workflow details including its workflow log. To include the workflow's session details, deselect the "Summary Only" checkbox, which is selected by default.

6. Click **OK** to save the job.

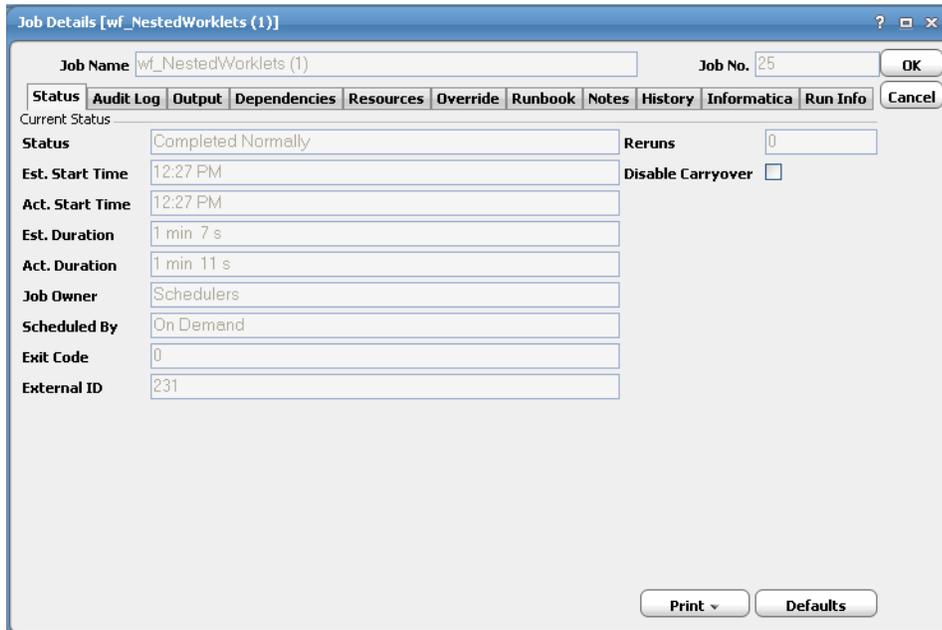
Monitoring Informatica Jobs

As Informatica tasks run as pre-scheduled or event-based jobs, you can monitor the jobs as you would any other type of job in CWA using the **Job Details** dialog. You can also use Business Views to monitor job activity and view when the jobs are active (see the *Cisco Workload Automation User Guide* for instructions on using Business Views).

To monitor job activity:

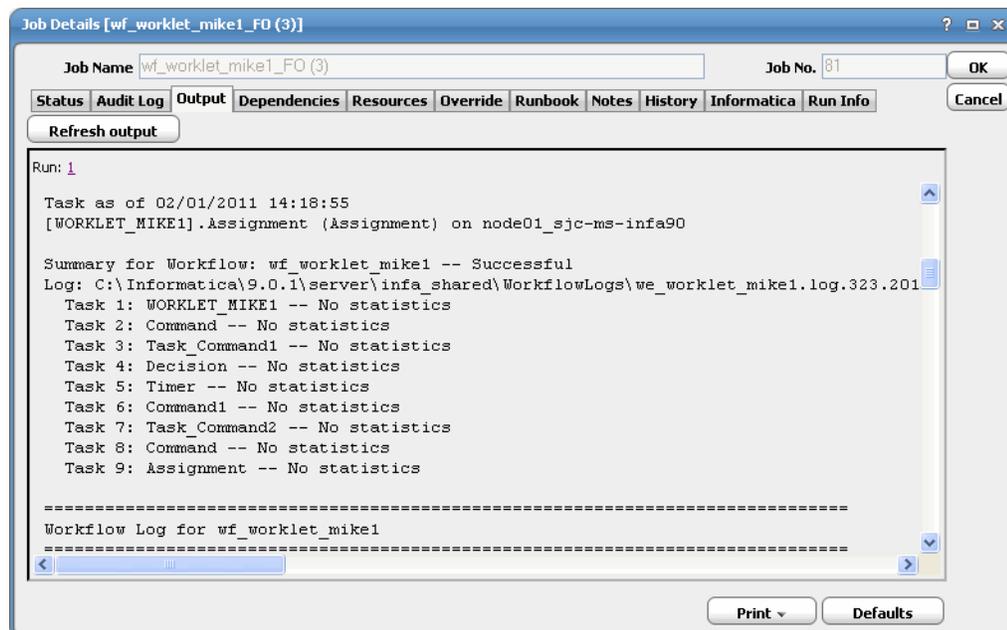
1. In the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** console.
2. Right-click job and select **Details** from the context menu.

The **Job Details** dialog displays.



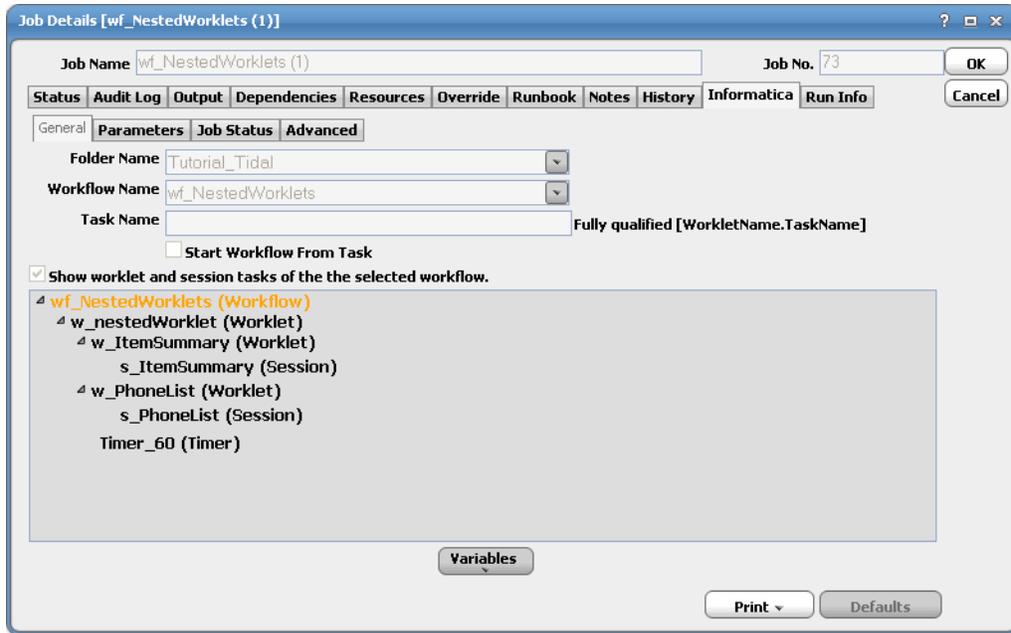
The **Status** page displays by default. You can view the status of the job, the start and end time, how long it ran, and how it was scheduled. The External ID contains the workflow run id.

3. Click the **Output** tab to view the output for the PowerCenter job.



The output consists of summary data for the workflow including all its tasks. If the **Summary Only** checkbox is selected for the job definition, no session log output is included. If the **Summary Only** checkbox is not selected, it includes the workflow log as well as any session logs for session tasks of the workflow.

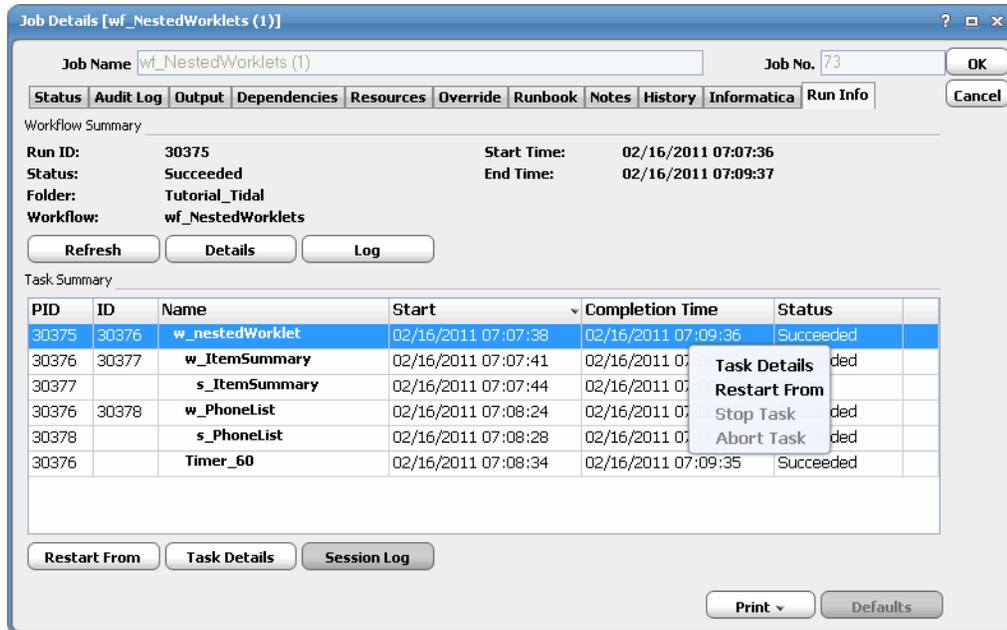
- Click the **Informatica** tab to view the job data for the run instance as defined during job definition. If the job is not currently running, you can override certain attributes of the definition prior to its scheduled run or rerun. Changes only affect the current instance of the job.



When you override values, the new values are saved for this run instance only.

- Click the **Parameters** tab to override parameters on the run instance.
- Click the **Job Status** tab to override the Job Status determination.
- Click the **Advanced** tab to override the job options.

8. Click the **Run Info** tab to view the runtime status of the workflow and its corresponding tasks. This tab provides data similar to what is shown in PowerCenter's Workflow Monitor.



The previous figure displays the run details of the workflow and its tasks. In this example, the workflow `wf_nestedWorklets` (with workflow run id 1586) contains a single worklet task: `w_nestedWorklet` (with worklet run id 1587). This worklet in turn contains 2 worklet tasks: `w_ItemSummary` (with worklet run id 1588) and `w_PhoneList` (with worklet run id 1589). These two worklet tasks in turn contain session and timer tasks.

The **Run Info** tab contains the following elements:

- Workflow Summary

Run ID- The workflow Run ID.

- **Folder** - The name of the folder containing the workflow.
- **Workflow** - The name of the scheduled workflow.
- **Status** - The status of the workflow.
- **Start Time** - The time the workflow began.
- **Completion Time** - The time the workflow completed.

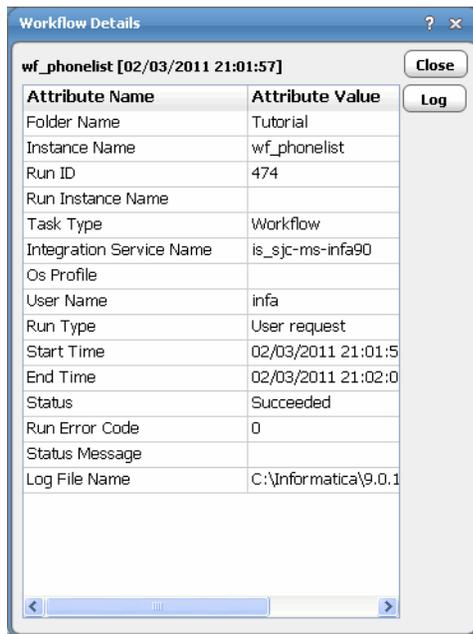
- Task Summary

- **PID** - The Parent ID for the task, either the Workflow Run ID or the Worklet Run ID, for worklet tasks.
- **ID** - The Run ID for worklet tasks (e.g. Worklet Run ID).
- **Name** - The name of the task.

The indenting of the tasks allows for a graphical view of the parent child hierarchy for the tasks relative the their parent, which will be either the workflow itself or a worklet of the workflow.

- **Start Time** - The time the task began.
- **Completion Time** - The time the task completed.

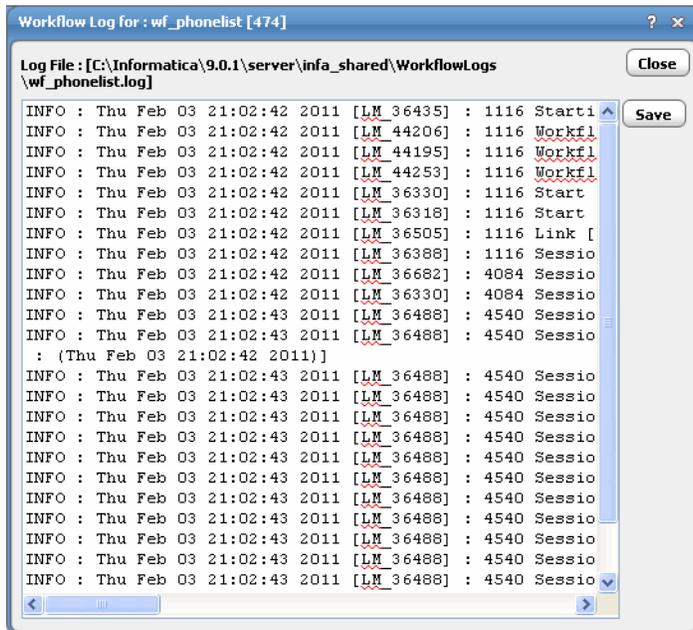
- **Status** - The status of the task.
- **Task Menu Options** - There are several menu options that may be initiated for a selected workflow task.
 - **Task Details** - Select to display the **Task Details** dialog. This is the same as the **Task Details** button and is available once the task is completed. If the task is still running, this option is disabled.
 - **Restart From** - Select to restart the workflow from the selected task. This is the same as the **Restart From** button and is available once the workflow is completed. If the workflow is still running, this option is disabled.
 - **Stop Task** - Select to stop the selected workflow task running on the Integration Service. When you stop a task in the workflow, the Integration Service stops processing the task and all other tasks in its path. This option is available while the task is running. Once the task is no longer running, the option is disabled.
 - **Abort Task** - Select to abort the selected, running workflow task. This option is available while the task is running. Once the task is no longer running, the option is disabled.
- **Buttons**
 - **Refresh** - Click to refresh the workflow status by re-requesting the workflow and task status from PowerCenter.
 - **Restart From** - Click to restart the workflow from a selected task.
 - **Details** - Click to view the workflow details by selecting the **Details** button or by double-clicking anywhere in the **Workflow Run** pane.



The **Workflow Details** dialog displays run details for the workflow. You can double click on any attribute to display an informational dialog containing the attributes full value.

For example, you can double-click **Log File Name** to display the full long file name in an **Information** dialog.

Click the **Log** button to display the **Workflow Log** dialog.



By default, logs are retrieved given a request to the PowerCenter, but a direct read can be configured given job or connection options. Logs only apply to workflows and sessions and the **Log** button is only available for workflow and sessions.

Click the **Save** button to save a copy of the logs to disk.

- **Task Details** – Given a selected task from the task details pane, you can view its details by clicking **Task Details** or by double-clicking a selected task.

The **Task Details Dialog** displays run details for the task. You can double-click on any attribute to display an informational dialog containing the attribute's full value. For example, you can double-click **Status Message** to display the full status message in an **Information** dialog.

For session tasks, click the **Log** button to display the **Session Log** dialog. Given a selected session task, you can also view the session log of completed sessions.

Note: The Log button only applies to workflow and session tasks.

By default, logs are retrieved given a request to PowerCenter, but a direct read may be configured given job or connection options (see Step 3). Logs only apply to workflows and sessions so the **Log** button is only available for these.

Click the **Save** button to save a copy of the logs to disk.

9. When you have completed viewing the job activity details, click **OK** to close the dialog.

Controlling Informatica Tasks and Workflows

Scheduler provides the following job control capabilities:

- [Recovering a Workflow, page 39](#)
- [Recovering a Task, page 39](#)
- [Restarting a Workflow from a Specific Task, page 40](#)

Recovering a Workflow

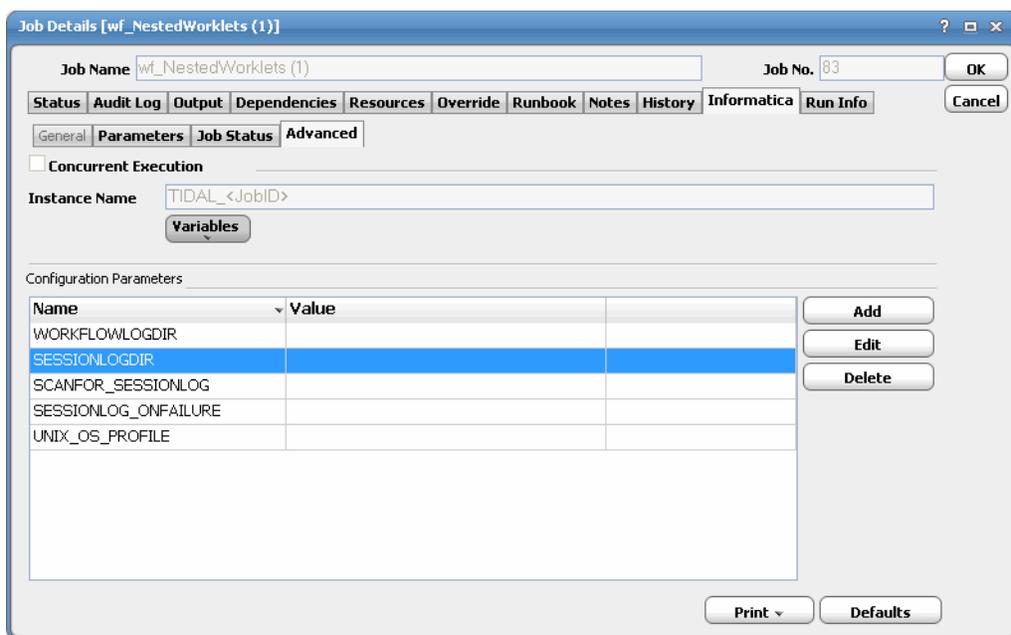
You can recover an aborted, stopped, suspended, or terminated workflow if you enable the workflow for recovery. During the course of running a workflow, if the workflow is subsequently aborted, stopped, suspended or terminated, you may wish to recover this workflow from Scheduler.

To recover a workflow:

1. From the **Job Activity** pane, edit the Informatica job corresponding to workflow you wish to recover.

- a. From the **Informatica** tab, **Advanced** tab, include the following Configuration Parameter:

RECOVER=Y



- b. Click **OK** to save this run instance.

2. Select **Job Control>Rerun** option from the context menu to rerun the workflow in recovery mode.

Any recoverable session/task is recovered.

Recovering a Task

After enabling the workflow, you can recover a failed task from the enabled workflow by completing one of the following PowerCenter supported recovery methods:

- Choose **Recover Task** from the context menu
- Edit the Informatica job from the **Job Activity** pane

To recover a task from the context menu:

1. Double-click the workflow in the Workflow Monitor to expand it and display the task.
2. Right-click the session and choose **Recover Task** from the context menu. The Integration Service recovers the failed session according to the recovery strategy. This starts a new workflow and recovers only the failed task.

To recover a task by editing the job from the Job Activity pane:

1. Double-click the workflow in the Workflow Monitor to expand it and display the task.
2. From the **Job Activity** pane, edit the Informatica job corresponding to the workflow you wish to recover.
3. From the **Advanced** tab of the **Informatica** tab, include the following configuration parameter:

```
RECOVER=Y  
STARTTASK=taskPath
```

where `taskPath` is the full qualified task path

4. Click **OK** to save this run instance.
5. Select **Job Control>Rerun** from the context menu to rerun the workflow in recovery mode.

Restarting a Workflow from a Specific Task

On occasion, you may need to restart an Informatica workflow without starting from the beginning. From the Informatica **Job Detail** dialog within the Web client, you can restart an Informatica workflow starting from any task in the job. Before rerunning, you can modify certain parameters on the instance.

To restart a workflow from a specific task:

1. From the **Job Activity** console, double-click the job you need to rerun to open the **Job Detail** dialog.
2. Select the task to restart from.
3. Click the **Restart From** button.

The job will automatically rerun from the selected task.

Controlling Adapter and Agent Jobs

Scheduler provides the following job control capabilities for either the process currently running or the job as a whole:

- **Holding a Job**—Hold a job waiting to run.
- **Aborting a Job**—Abort an active job.
- **Rerunning a Job**—Rerun a job that completed.
- **Making One Time Changes to an Adapter or Agent Job Instance**—Make last minute changes to a job.
- **Deleting a Job Instance before It Has Run**—Delete a job instance before it has run.

Holding a Job

Adapter/agent jobs are held in the same way as any other Scheduler jobs.

Adapter/agent jobs can only be held before they are launched. Once a job reaches the Adapter/Agent system, it cannot be held or suspended.

To hold a job:

1. From the **Job Activity** pane, right-click on the job.
2. Select **Job Control>Hold/Stop**.

Aborting a Job

Adapter/agent jobs are aborted in the same way as any other Scheduler jobs.

To abort a job:

1. From the **Job Activity** pane, right-click on the job.
2. Select **Job Control>Cancel/Abort**.

Rerunning a Job

On occasion, you may need to rerun an Adapter/Agent job. You can override parameter values first, if necessary, from the Adapter/Agent tab.

To rerun a job:

1. From the **Job Activity** pane, right-click the Adapter/Agent job you need to rerun.
2. Select **Job Control>Rerun** option from the context menu.

Making One Time Changes to an Adapter or Agent Job Instance

Prior to a run or rerun, you can edit data on the specific **Adapter/Agent** tab. To ensure that there is an opportunity to edit the job prior to its run, you can set the **Require operator release** option on the **Options** tab in the Adapter **Job Definition** dialog. Use this function to make changes to an Adapter job after it enters Waiting on Operator status as described in the following procedure.

To make last minute changes:

1. From the **Job Activity** pane, double-click the Adapter/Agent job to display the **Job Details** dialog.
2. Click the Adapter tab.
3. Make the desired changes to the job and click **OK** to close the **Job Details** dialog.
4. If this job is Waiting on Operator, perform one of the following tasks:
 - To release the job, select **Job Control->Release**.
 - To rerun the job with changes, select **Job Control->Rerun**.

Deleting a Job Instance before It Has Run

Adapter/Agent job instances are deleted in the same way as any other Scheduler job.

Deleting a job from the **Job Activity** pane removes the job from the Scheduler job activity only. The original definition is left in tact.

To delete a job instance:

1. From the **Job Activity** pane, right-click the Adapter/Agent job to be deleted.
2. Select **Remove Job(s) From Schedule**.



4

Troubleshooting the Informatica Adapter

Overview

This chapter describes how to resolve issues with the Informatica Adapter which entails:

- [Troubleshooting, page 43](#)
- [Testing Using Command Line Mode, page 44](#)
- [Monitoring with JConsole, page 45](#)

Troubleshooting

This section describes how to enable debugging and some of the most common issues.

Enabling Debugging

Enable high debugging in the Informatica Adapter logs located at the log directory under the Adapter's GUID directory (7640B420-5530-11DE-8812-7B8656D89593}). This can be enabled by:

- Setting the "Host Environment Log" to **High Debug** in the Master Connection Definition from the **Connections** console.

Warning: This will trigger high debug for all Adapter logs, not just informatica. You may need to disable, then re-enable the Informatica connection for this value to take affect.

- Alternatively, set the "ServiceLog" value of the *service.props* to **FINEST**. This only affects the Informatica logs.

Once high debug logging is enabled, review the logs for errors.

Informatica Service Fails to Start

If the Informatica Service fails to start, the diagnostic log with high debug enabled may point out problems with your configuration.

If the service fails to start due to:

- *java.lang.UnsatisfiedLinkError*

This is usually due to an unsatisfied dependency. Verify that your environment's library path is properly set up. For instructions, see [To install and configure the Informatica Libraries:](#), page 9.

Note: For Windows installations, the Informatica Libraries requires Microsoft Visual C++ runtime components. Refer to the same section on how to obtain these from Microsoft.

- *java.lang.UnsatisfiedLinkError....Cannot load IA 32-bit.dll on a AMD 64-bit platform* or similar message.

This means the platform for the Informatica libraries for your distribution are not compatible with the Java installed in your environment. For 64-bit Informatica Libraries, you must also run with 64-bit version of the JVM. For 32-bit, you need a 32-bit JVM.

- -117 (INFA_CANNOT_CREATE_LOCALE)

This means the adapter failed to initialize. Verify the Locale Path for the Informatica Library as specified by the `TDLINFA_LOCALE.service.props` value of the Informatica Adapter. See [Configuring service.props, page 47](#) for further details.

Informatica Connection Is Red

If the Informatica connection is red, it has failed to establish a connection to the Integration Service. Perform the following steps for diagnosis:

The diagnostic log with high debug enabled may point out problems with your configuration.

- -109 (INFA_CANNOT_ESTABLISH_CONNECTION)

Verify correct values for **Repository Service** tab of the Connection definition:

- Check the PowerCenter user used to connect, verifying both user name, password and security domain as appropriate.
- Verify proper configuration of the **INFA_DOMAINS_FILE** environment variable and that it points to a valid *domains.infa* file. Also, verify the contents of the file are valid for the Integration Service defined in the Connection definition and proper network connectivity to the PowerCenter servers specified.

- If the Informatica Connection fails to establish a database connection to the Repository database:

- Verify the correct JDBC drivers have been configured given the CLASSPATH value in *service.props*.
- Verify correct values for the **Repository Database** tab of the Connection definition.
- Verify the Database User used to connect, verifying both user name and password.

Also, ensure this user has read access to the following PowerCenter database views: REP_SUBJECT, REP_WORKFLOWS, REP_TASK_INST, REP_WORKFLOW_DEP, REP_TASK_INST_RUN. Without this, there will be failures interrogating the system for Folders and Workflows.

- If you are using a demo license which permits you define an MSSql or Oracle Adapter connection, you can attempt to create one of these, while utilizing the same parameters as those used in Informatica. This will help isolate the problem. If one of these connections is successful then the likely problem is the JDBC driver in use. Copy the *.jar* file from the **lib** directory of the corresponding Adapter GUID directory to the GUID **lib** directory of Informatica. For example: `\{7640B420-5530- 11DE- 8812-7B8656D89593}\lib`

Testing Using Command Line Mode

You can test proper configuration of the Informatica Adapter from the command line, which is useful for debugging purposes. Note, this is not a fully integrated test and can only be used to validate:

- Proper environment configuration
- Ability to connect to PowerCenter and run workflows, given some valid PowerCenter user.

Note: A database user is configured in order to provide picklists for Folder and Workflows for ease of job configuration. This will not test for a valid PowerCenter Repository Database User.

To test Informatica configuration from the command line:

1. The **infalib** archive contains directories **locale**, **lib** and a test script (*.bat* or *.sh*). These should have been extracted to: C:\Program Files\TIDAL\master\services\infa\, for example. In the same directory, copy *jniLM.jar* located at:

```
master\services\{7640B420?5530?11DE?8812?7B8656D89593}\lib
```

2. In this same directory, copy *infaservice.pkg* located at:

```
master\services
```

You should have the following similar structure:

```
infa \
  locale
    lib
    test.bat
    jniLM.jar
    infaservice.pkg
```

3. Modify and save the contents of the test script for your specific environment. The script contains usage information which explains the parameters in use.

Parameters **-l**, **-d**, **-i**, **-r**, **-u**, **-p**, **-f**, **-w** in the following sample *test.bat* script for Windows would need to be changed for testing of your specific environment.

```
java -Xms256m -Xmx512m -cp infaservice.pkg; jniLM.jar
com.tidalsoft.service.informatica.logic.jniLM.jniLoadManager -l "locale" -d Domain_Infa85 -i
IS_Infa85 -r PowerCenter85 -u tidal -p tidal -f Tutorial_Tidal -w wf_PhoneList
```

4. Run the test script.

Monitoring with JConsole

The Java Monitoring and Management Console (JConsole) is a JMX technology-compliant graphical management tool bundled with JDK 5 (and later). JConsole connects to a running JVM and gathers information from the JVM MBeans in the connected JMX agent.

You can use JConsole to monitor the Informatica Adapter's resource usage.

To monitor the Informatica Adapter using JConsole:

1. Add the following in the Informatica *service.props*, prior to starting the adapter:

JmxOn=Y

2. Start the jconsole executable located in the **JDK_HOME/bin**, where **JDK_HOME** is the directory where the JDK is installed.
3. Create a new connection with the following JMX URL:

```
service:jmx:rmi://localhost/jndi/rmi://localhost/Informatica
```




5

Configuring service.props

About Configuring service.props

The **service.props** file is used to configure adapter behavior. **service.props** is located in the \config directory located under the Adapter's GUID directory, You can create both the directory and file if it does not yet exist. Properties that can be specified in service.props control things like logging and connection configuration. Many of the properties are specific to certain adapters; others are common across all adapters.

service.props Properties

The table below lists many of the parameters that can be specified in service.props. Some properties apply to all adapters (shaded in the table) and some properties are adapter-specific as indicated by the **Applicable Adapter(s)** column. The properties are listed in alphabetical order.

Property	Applicable Adapter(s)	Default	What It Controls
BYPASS_SEC_VALIDATION	Oracle Apps	N	If set to Y, the secondary user validation is bypassed. If not, secondary user validation is performed.
CLASSPATH	All	<none>	(Optional) - The path to the JDBC driver. If the default CLASSPATH used when the Adapter process is started does not include an appropriate JDBC driver jar required to connect to the PowerCenter Repository Database, you will need to specify this <i>service.props</i> configuration
CONN_SYNC	Informatica, Oracle Apps, SAP	N	Setting this flag to Y allows synchronous connections without overloading the RDOOnly Thread. If set to N, the adapter might stop trying to reconnect after an outage or downtime.
DISCONN_ON_LOSTCONN	Informatica	N	Setting this flag to Y avoids an unnecessary logout call to the Informatica server when the connection is lost. This logout call usually hangs.
EnableDynamicPollingInterval	All	N	Use to avoid frequent polling on long-running jobs. When set to Y in service.props of a particular adapter, these properties are enabled: MinDynamicPollInterval—Minimum value should be 5 seconds. MaxDynamicPollIntervalInMin—Maximum value should be 5 minutes. PercentOfEstDuration—Default value is 5.

Property	Applicable Adapter(s)	Default	What It Controls
HADOOP_JAVA_HOME	Sqoop	<none>	If the Java version used in the Hadoop environment is lower than Java 8, then install the same lower JDK version in the in the Master and include the path of the JDK in this property.
IGNORE_CODES	Informatica	<none>	This parameter can be set in service.props, job configuration and connection configuration parameters. The order of precedence is service.props (applicable for all jobs running in all connections), job level (only for that particular job), and connection (applicable for all jobs in the connection). This parameter is used to specify Informatica-specific error codes, separated by commas (,), that you want to ignore while running a job.
IGNORESUBREQ	Oracle Apps	N	Y or N. Setting this flag to Y stops huge job xml file transfers back and forth between the adapter and the AdapterHost during polls when a single request set has multiple sub-requests of more than 100. The default value is N or empty.
kerbkdc	MapReduce	<none>	If the Hadoop cluster is Kerberos secured, use this value to specify the KDC Server. For example, kerbkdc=172.25.6.112
kerbrealm	MapReduce	<none>	If the Hadoop cluster is Kerberos secured, use this value to specify the Kerberos Realm. For example, kerbrealm=TIDALSOFT.LOCAL
Keystore	BusinessObjects, BusinessObjects BI, BusinessObjects DS, Cognos, JD Edwards, Oracle Applications, UCS Manager, VMware, Web Service	<none>	Specify Keystore=c:\\<adapter_certificate_directory>\\<your_trusted_keystore>.keystore when importing certificates into a Java keystore.
LAUNCH_DELAY (in milliseconds)	Informatica	<none>	This parameter can be set in service.props, job configuration and connection configuration parameters. The order of precedence is service.props (applicable for all jobs running in all connections), job level (only for that particular job), and connection (applicable for all jobs in the connection). If a non-zero value is set for this parameter, then the jobs are delayed for the specified number of milliseconds before being submitted to Informatica.

Property	Applicable Adapter(s)	Default	What It Controls
LoginConfig	BusinessObjects BI Platform, BusinessObjects Data Services	<none>	Specifies the location of the login configuration if using WinAD or LDAP authentication. For example: LoginConfig=c:\\windows\\bscLogin.conf where "c:\\windows\\bscLogin.conf" is the location of the login configuration information. Note the use of \\ if this is a Windows location.
MaxLogFiles	Informatica, JDBC, PeopleSoft	50	(Optional) - Number of logs to retain.
OUTPUT_ASYNC_LOGOUT	Informatica	N	Setting this flag to Y avoids jobs getting stuck in Gathering Output status.
OUTPUT_SYNC	All	Y	Enables concurrent output gathering on a connection. To enable this feature, set the value to N.
POLL_SYNC	All	Y	Enables concurrent polling on connections of the same type. This is helpful when there is a heavily load on one connection of an adapter. The heavily loaded connection will not affect the other adapter connection. To enable this feature, set the value to N.
QUERY_TIMEOUT	Oracle Apps	N	Y or N. If set to Y, the timeout value defined using the parameter QUERY_TIMEOUT_VALUE is applied to the SQL queries. Default value is N or empty.
QUERY_TIMEOUT_VALUE	Oracle Apps	unset	The time period in seconds that SQL queries wait before timeout. If 0 or not set, there is no timeout.
READPCHAINLOG	SAP	Y	Used to control the log gathering in SAP Process Chain jobs. This property depends on the Summary Only check box of the job definition Options tab.
SCANFOR_SESSIONSTATS	Informatica	Y	Y or N - Set this parameter to N to turn off the default behavior of Informatica jobs collecting the session statistics during the job run.
SCANFOR_SESSIONSTATS_AFTER_WF_ENDS	Informatica	N	Y or N - Set this parameter to Y to turn off the gathering of session statistics during each poll for the status of Informatica jobs.
TDLINFA_LOCALE	Informatica	<none>	Points to the Load Manager Library locale directory. See "Configuring the Informatica Adapter" in the <i>Informatica Adapter Guide</i> for how to set this for Windows and Unix environments.
TDLINFA_REQUESTTIMEOUT	Informatica	<none>	(Optional) - The number of seconds before an API request times out. The default is 120 seconds, if not specified.
TDLJDBC_LIBPATH	JDBC	<none>	(Windows only, optional) An alternate path to the JDBC library files. The library file path should have been configured given system environment variables. This option is available in case you wish to use an alternate set of libraries and may be helpful for trouble-shooting purposes.

service.props Properties

Property	Applicable Adapter(s)	Default	What It Controls
TDLJDBC_LOCALE	JDBC	<none>	The path to the JDBC locale files.
TRANSACTION_LOG_BATCH_SIZE	MS SQL	5000	Set this parameter if more than 5000 lines need to be read from the transaction table.
version_pre898	JD Edwards	N	If running on a JD Edwards server version that is less than 8.9.8, set version_pre898=Y.