



# Cisco Workload Automation Web Service Adapter Guide

Version 6.3

**First Published:** August, 2015

**Last Updated:** September 7, 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](http://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 6

### **Introducing the Web Service Adapter 7**

- Overview 7
- Prerequisites 7

### **Configuring the Web Service Adapter 9**

- Overview 9
- Licensing an Adapter 9
- Securing the Web Services Adapter 10
  - Defining Runtime Users 10
  - Authorizing Schedulers to Work with Web Service Jobs 11
    - Defining a Security Policy 11
    - Defining a Web Service CWA User 12
- Configuring the HTTPS Protocol 14
  - Obtain Security Certificates 14
  - Export Security Certificates 15
  - Import Target Server Certificates Into a Java Keystore 16
- Defining a Web Service Connection 16
  - Adding a SOAP Web Service Connection 17
  - Adding a REST Web Service Connection 19
    - Without Authentication 20
    - With HTTP Authentication 20
    - With OAuth 1.0 Authentication 21
    - With OAuth 2.0 Authentication 26
- Verifying Web Service Connection Status 28

### **Working with Web Service Jobs 29**

- Overview 29
- Adding a SOAP Web Service Job 29
- Adding a REST Web Service Job 33
  - GET Method Example 33
  - POST Method Example 38
- Defining a Web Service Action 42
- Monitoring Job Activity 44
  - Output Tab 44
  - WebService Tab 46
    - For a SOAP Web Service Job 47
    - For a REST Web Service Job 49
  - Run Info Tab 50
    - For a SOAP Web Service Job 51
    - For a REST Web Service Job 51
- Controlling Adapter and Agent Jobs 52
  - Holding a Job 52
  - Aborting a Job 53

---

Rerunning a Job 53

Making One Time Changes to an Adapter or Agent Job Instance 53

Deleting a Job Instance before It Has Run 53

## **Configuring service.props 55**

About Configuring service.props 55

service.props Properties 55



# Preface

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

## Audience

This guide is for administrators who install and configure the Web Service 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 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 Web Service Adapter Guide*.

Version Number	Issue Date	Reason for Change
6.1.0	October 2012	New Cisco version.
6.2.1	June 2014	Available in online Help only.
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 Web Service Adapter documentation into one document.
6.3	August 2016	Rebranded “Cisco Tidal Enterprise Scheduler (TES)” to “Cisco Workload Automation (CWA)”.  Miscellaneous edits for the 6.3 release.



# 1

## Introducing the Web Service Adapter

### Overview

The Cisco Workload Automation (CWA) Web Service Adapter is an API-level integration solution. This solution hides implementation details in screens that connect to Web Service providers and define Web Service tasks as part of CWA job definitions. Up to version 6.0.2, the Web Service Adapter only supports SOAP (Simple Object Access Protocol) Web Services. Versions 6.0.3 and above provides support for REST (Representational State Transfer) Web Services.

As a platform independent solution, the adapter can run on any platform where the CWA master runs.

REST Web Service is a stateless client-server architecture in which clients access and manipulate Web resources through HTTP protocol. It does not introduce additional specification (as oppose to SOAP/WSDL) on top of the existing HTTP methods, definitions, and entities. This means an HTTP client can interact with a REST Web Service provider without having to incorporate any supporting stack.

### Prerequisites

- The WSDL file that defines the Web Service or a URL to the WSDL.
- The REST Web Service.
- The Web Service must support SOAP 1.1 or 1.2.
- If HTTP Authentication is required by the Web Service's Web server, the username and password.

Refer to your *Cisco Workload Automation Compatibility Guide* for a full list of software and hardware prerequisites.

## Prerequisites





# 2

## Configuring the Web Service Adapter

### Overview

While the Web Service adapter software is already installed as part of a normal installation of CWA, you must perform the following steps to license and configure the adapter before you can run Web Service jobs:

- [Licensing an Adapter, page 9](#) – License the connection(s) to the Web Service instance. You cannot define a Web Service connection until you have applied the Web Service license from Cisco.
- [Securing the Web Services Adapter, page 10](#) – Define a Web Service Authentication user to authorize a connection to be established to the Web Service's Web server and permit requests to be made on behalf of the authenticated account.
- [Configuring the HTTPS Protocol, page 14](#) – Configure the HTTPS protocol if used in your environment.
- [Defining a Web Service Connection, page 16](#) – Define a Web Service connection so the master can communicate with the Web Service.
- [Verifying Web Service Connection Status, page 28](#) – Verify the Web Service connection is healthy.

See [Configuring service.props, page 55](#) for information about general and adapter-specific properties that can be set to control things like logging and connection properties.

### 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 Web Services Adapter

There are two types of users associated with the Web Service adapter, Runtime Users and Schedulers. You maintain definitions for both types of users from the **Users** pane.

### ■ Runtime Users

Runtime users in the context of Web Service jobs represent those users and passwords required for HTTP Authentication. Not all Web Service operations require authentication, but for those that do, runtime user(s) will need to be defined. For REST Web Service that requires OAuth authentication, runtime users do not need to be defined.

### ■ Schedulers

Schedulers are those users who will define and/or manage Web Service jobs. There are three aspects of a user profile that grant and/or limit access to scheduling jobs that invoke Web Services:

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

## Defining Runtime Users

### To define a runtime 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–

Right-click a user in the **Runtime Users** pane and select **Edit Runtime User** from the shortcut menu (*Edit mode*).

The **User Definition** dialog displays (Figure 3).

3. If this is a new user definition, enter the new user name in the **User/Group 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 Web Service jobs, click **Add** on the **Passwords** tab.
7. Select **WebService** 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 Web Service will be available for use with Web Service 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.

The screenshot shows the 'User Definition' dialog box. It has three tabs: 'Passwords', 'Kerberos', and 'Description'. The 'Passwords' tab is selected. In this tab, there are input fields for 'User Name' (containing 'WS User'), 'Full Name' (containing 'Web Service User'), and 'Domain' (containing 'CISCO'). Below these are two empty input fields for 'Windows/FTP' password. A table is present with columns 'Adapter', 'Password Type', and 'Password'. One row is filled with 'WebService Password', 'Password', and '\*\*\*'. To the right of the table are buttons for 'Add', 'Edit', and 'Delete'. At the top right of the dialog are 'OK' and 'Cancel' buttons.

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

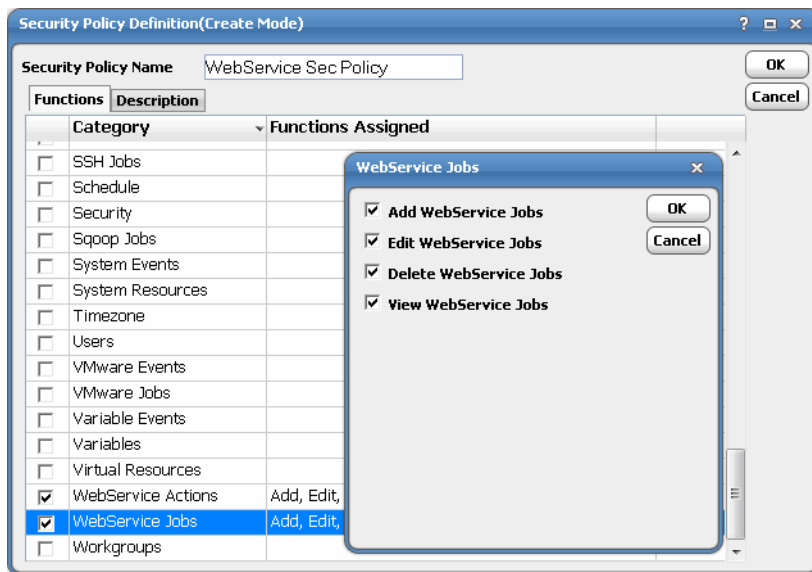
## Authorizing Schedulers to Work with Web Service Jobs

### Defining a Security Policy

**To define a Security Policy that authorizes access to Web Service jobs:**

1. In the **Navigator** pane, select **Security Policies** to display the **Security Policies** pane.

2. Select a security policy for the Web Service job privileges and double-click on it to display its **Security Policy Definition** dialog.



3. Scroll down the list of function categories and double-click on the **WebService Jobs** category to display the available functions.
4. Double-click the category row to select the desired job privileges then click **OK**.

A check mark displays next to the **WebService Jobs** function category indicating that one or more functions are selected within the category.

If needed, different security policies with varying authorized functions can be created to provide different levels of access for a variety of users.

5. Click **OK** to save the security policy.

## Defining a Web Service CWA User

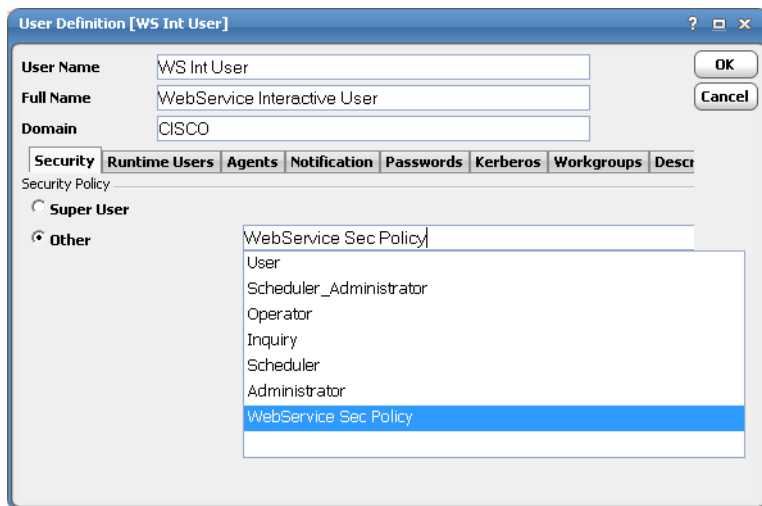
### To define a CWA user to work with Web Service jobs:

1. From the **Navigators** 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*).

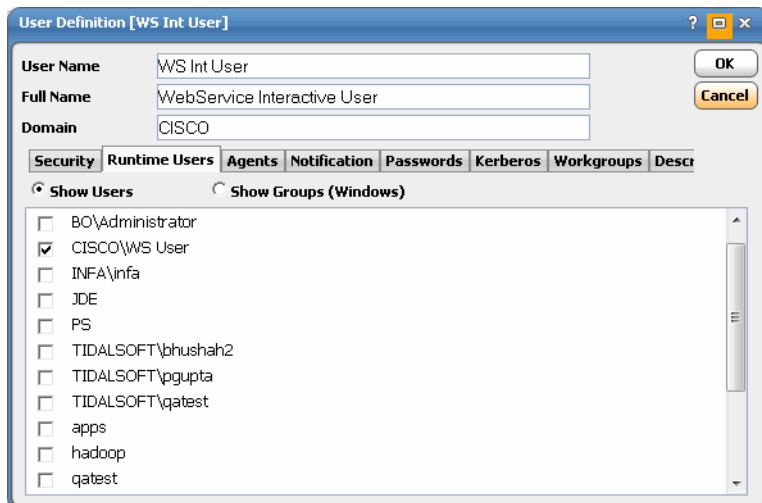
-or-

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.

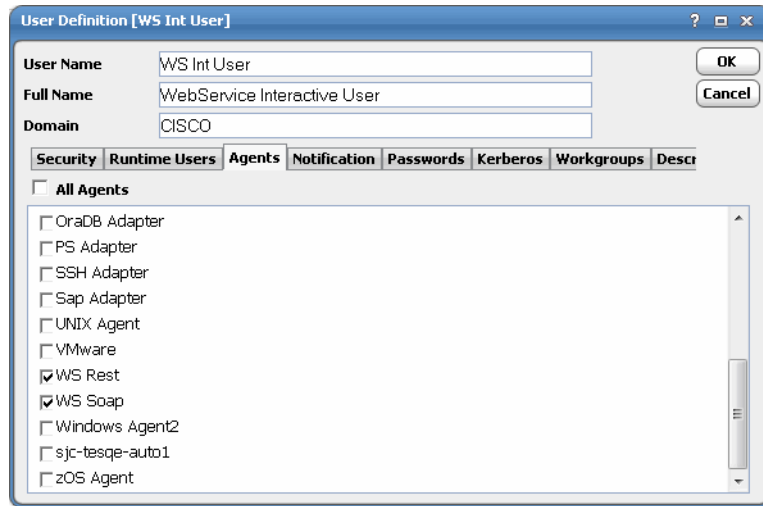


3. If this is a new user definition, enter the new user name in the **User/Group Name** field.
4. For documentation, enter the **Full Name** or description associated with this user.
5. In the **Domain** field, select a Windows domain associated with the user account required for authentication, if necessary.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



8. Select the Web service users that this scheduling user may use for Web service authentication in Web service jobs.

9. Click the **Agents** tab.



10. Select the check boxes for the Web service connections that this scheduling user can access when scheduling jobs.

11. Click **OK** to save the user definition.

## Configuring the HTTPS Protocol

It is recommended that Web Service Web servers be configured to use SSL via the HTTPS protocol for Data Services/Data Integrator. If your environment is configured to use HTTP, you can skip this section.

For complete instructions on configuring Web Service servers to use the HTTP or HTTPS protocol, refer to the Web Service documentation that ships with the product.

## Obtain Security Certificates

From a Windows desktop, you can obtain a security certificates for each target Web Service server using the Microsoft Internet Explorer Certificate Cache.

**Note:** Although other procedures are available for obtaining the required certificates, the procedure below can be performed from your Windows desktop.

**Note:** You need to carry out the following instructions only if your server certificate is generated in-house (that is, self-signed) or if your server certificate is signed by a Certification Authority that is not trusted by the version of Java you are using. Alternatively, if your adapter connection fails by reporting the error “unable to find valid certification path to requested target” you need to carry out the following instructions.

### To obtain target Web Service server security certificates

1. Open the Internet Explorer browser and navigate to the following dispatch URL (replacing servername and port as it applies to your environment).

```
https://<adminHost:adminPort>/admin/servlet/webservices
```

where `adminHost` is where the Data Integrator Administrator is installed and `adminPort` is the port the Data Integrator Administrator is listening on.

A **Security Alert** message displays.

2. Click **View Certificate** to open the **Certificate** dialog.

3. Click **Install Certificate**.
4. On the **Certificate Import Wizard Welcome** panel, click **Next**.
5. On the **Certificate Store** panel, use the default option **Automatically select the certificate store based on the type of certificate** and click **Next**.
6. On the **Completing Certificate Import Wizard** panel, click **Finish**.
7. If a **Security Warning** message displays informing you that you are about to install a certificate from a certification authority, click **Yes** to continue with the certificate installation.  
  
A message stating *The import was successful* displays.
8. Click **OK** to close the message and return to the **Certificate** dialog.
9. Click **OK** on the **Certificate** dialog. You can close your browser now.
10. Repeat the process for each Web Service server that you want to connect to with the Web Service Adapter.

## Export Security Certificates

After you have obtained the security certificates for the target servers, you must export them from the Internet Explorer cache to a local directory.

### To export the cached certificates to a local directory

1. On the local computer, create the following directory for the certificates:

```
C:\WebServer-Certs
```

2. In Internet Explorer, select **Tools>Internet Options**.
3. On the **Internet Options** dialog, select the **Content** tab.
4. In the Certificates area, click **Certificates**.
5. On the **Certificates** dialog, select the **Trusted Root Certification Authorities** tab to display the list of trusted certificates. This list should contain the certificates for the target servers that were obtained in the previous procedure (see [Obtain Security Certificates, page 14](#)).
6. Scroll through the list of certificates to find the certificates.
7. Perform the following procedure for each target server certificate:
  - a. Select the certificate and click **Export** to launch the Certificate Export Wizard.
  - b. On the **Welcome** panel, click **Next**.
  - c. On the **Export File Format** panel, use the default option DER encoded binary X.509 (.CER) and click **Next**.
  - d. On the **File To Export** panel, enter the complete path to the *WebServer-Certs* directory and a unique name for the certificate:  
  

```
C:\WebServer-Certs\servername.cer
```
  - e. Click **Next**.
  - f. On the **Completing the Certificate Export Wizard** panel, click **Finish** to complete the export.

A message stating *The export was successful* displays.

- g. Click **OK** to close the message box.
8. After all target server certificates have been exported, click **Close** to exit the **Certificates** dialog.
9. Click **OK** to close the **Internet Options** dialog.

## Import Target Server Certificates Into a Java Keystore

**Note:** These instructions assume that a JRE or JDK is in your system PATH.

### To import certificates into a Java keystore

1. Open a Windows **Command Prompt** window.
2. Change to the directory where the certificates are stored by entering the following commands:

```
c:
cd \WebServer-Certs
```

3. Use the Java keytool utility to import a certificate. The following syntax is used:

```
keytool -import -file <certificate-filename> -alias <server-name> -keystore WebServer.keystore
```

For example:

```
C:\WebServer-Certs>keytool -import -file sdkpubs01.crt -alias sdkpubs01 -keystore
WebServer.keystore
```

4. When prompted to create a password for the keystore, enter a password at the prompt. The keystore utility displays the certificate information.
5. At the **Trust this certificate? [no]** prompt, type **yes** and press **Enter**. The certificate is imported into the **WebServer.keystore** keystore and the following message displays:

```
Certificate was added to keystore
```

6. Repeat this procedure for each target server.
7. Navigate to the following folder where the Web Service Adapter is installed and create a new directory named **config**:

```
<install dir>\master\services\{2C290052-71BA-47BC-85BB-D65E06459001}\config
```

8. Create a text file named *service.props* if it doesn't already exist.
9. Open the *service.props* text file and add the following line:

```
Keystore=c:\\WebServer-Certs\\WebServer.keystore
```

(Note the use of escaped backslashes for Windows directories).

See [Configuring service.props, page 55](#) for information about general and adapter-specific properties that can be set to control things like logging and connection properties.

**Note:** This feature is effective only if the Master is running and you have configured HTTPS protocol by specifying the "Keystore" property in Adapter's service.props.

## Defining a Web Service Connection

You must create a connection to the Web Service host before CWA can run your Web Service jobs. These connections also must be licensed before CWA can use them.



A Web Service connection represents a connection to a single REST or SOAP Web server and one service defined in its WSDL. A single installation allows for multiple connections to be established to various REST and SOAP Web Service providers.

A connection is created through the **Connection Definition** dialog.

## Adding a SOAP Web Service Connection

**To add a SOAP Web Service connection:**

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button or right-click and select **Add Connection>WebService Adapter** from the context menu to display the **Web Service Connection Definition** dialog.

3. Enter a name for the new connection in the **Name** field.
4. Click the **General** tab.
5. In the **Job Limit** field, select the maximum number of concurrent active processes that CWA should submit to the Web Service host on this server at one time.
6. In the **Default Runtime User** list, select the name of a default user for Web Service jobs.

The runtime user is used for HTTP authentication.

Only users that have been defined with Web Service passwords display in this list. The user selected is automatically supplied as the runtime user in CWA Web Service job definitions.

**Note:** If you intend to connect Web Service adapter to another CWA 6.x platform, select a user account from the "Interactive User" group. Only and interactive user is allowed to connect to a CWA platform.

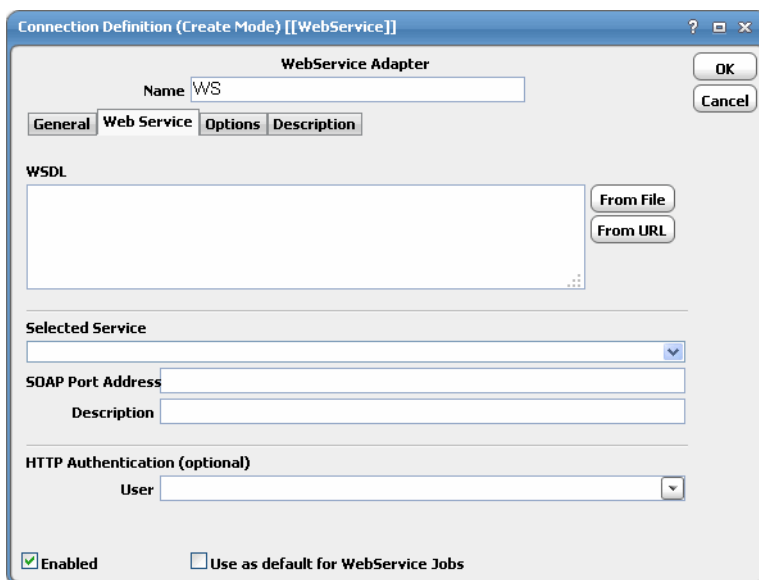
7. Click the **Web Service Connection** tab.

The **Web Service Type** dialog displays.



8. Select the **SOAP Web Service** option, and then click **OK**.

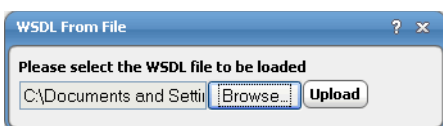
The **Web Service** tab displays as follows:



9. Click either **From File** or **From URL** to load the WSDL file.

- **From File** - Load the WSDL file from an existing file via the **WSDL From File** dialog.

Due to internet browser security constraint, WSDL upload from file will be successful only if the WSDL is self contained and does not import a schema or other WSDL's. To supply WSDL that imports other files, you must download it through the **From URL** option.



- **From URL** - Load the WSDL file from a WSDL URL

## Defining a Web Service Connection

**Note:** You can also import a schema from within a WSDL. To import a schema, load the WSDL. If the WSDL imports any schemas, the adapter will follow the reference paths to load them. If the schema imports additional schemas, the adapter will load every one.

If HTTP Authentication is required to access the WSDL URL on the Web server, specify the HTTP Authentication user for the connection first, then load the WSDL from the URL.

10. Select a service from the **Selected Service** list.

The Web service connection is defined by selecting a service defined in a WSDL file.

11. After selecting a file or entering a WSDL URL, click **OK**.

The **Web Service** tab displays as follows:

Connection Definition (Edit Mode) [WS Soap[WebService]]

WebService Adapter

Name: WS Soap

General Web Service Options Outages Description

SOAP Web Service

WARNING: This Web Service connection is bound to 1 job(s).  
Changing its settings may affect connectivity of the dependent services!

WSDL

<?xml version="1.0" encoding="UTF-8"  
standalone="yes"?> <!-- Generated by JAX-WS RI  
at http://jax-ws.dev.java.net. RI's version is  
JAX-WS RI 2.1.6 in JDK 6. --> <definitions  
targetNamespace="http://www.tidalsoftware.com/tes  
webservice" name="TESWebService"  
xmlns="http://schemas.xmlsoap.org/wsdl/"

From File From URL

Selected Service

TESWebService / TESWebServicePort12 (SOAP 12)

SOAP Port Address: http://sjc-tesqe-auto1.tidalsoft.local:8080/api/tes-6.0/webService/T

Description: TESWebService provides Web Service API to Tidal Enterprise S

HTTP Authentication

User: TIDALSOFT\qatest

☒ Enabled ☒ Use as default for Webservice Jobs

12. Optionally, in the **HTTP Authentication** section, select an authorized runtime user from the list for use with Web service jobs.

13. Click **OK**.

See your *Cisco Workload Automation User Guide* for instructions on using the **Options**, **Outages**, and **Description** tabs. These tabs are not specific to this adapter.

## Adding a REST Web Service Connection

A REST Web Service is a stateless client-server architecture in which clients access and manipulate Web resources through HTTP protocol. It does not introduce additional specification (as oppose to SOAP/WSDL) on top of the existing HTTP methods, definitions, and entities. This means an HTTP client can interact with a REST Web Service provider without having to incorporate any supporting stack. You can create a REST connection without authentication, with HTTP authentication, or with OAuth authentication.

## Without Authentication

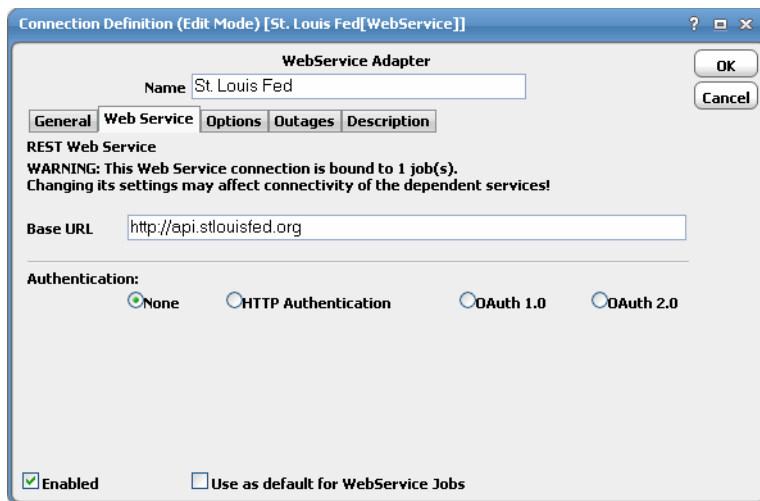
### To add a REST connection without authentication:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button or right-click and select **Add Connection>WebService Adapter** from the context menu to display the **Web Service Connection Definition** dialog.
3. In the **Name** field, enter a name for the new connection.
4. Click the **General** tab.
5. In the **Job Limit** field, select the maximum number of concurrent active processes that CWA should submit to the REST Web Service host on this server at one time.
6. Click the **Web Service Connection** tab.

The **Web Service Type** dialog displays.

7. Select **REST Web Service**, and then click **OK**.

The **Web Service** tab displays as follows:



8. In the **Base URL** field, enter the URL for the REST service.
9. From the **Authentication** section, select **None**.
10. Click **OK**.

## With HTTP Authentication

### To add a REST connection with HTTP authentication:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button or right-click and select **Add Connection>WebService Adapter** from the context menu to display the **Web Service Connection Definition** dialog.
3. Enter a name for the new connection in the **Name** field.
4. Click the **General** tab.

5. In the **Job Limit** field, select the maximum number of concurrent active processes that CWA should submit to the REST Web Service host on this server at one time.

6. Click the **Web Service Connection** tab.

The **Web Service Type** dialog displays.

7. Select **REST Web Service**, and then click **OK**.

The **Web Service** tab displays as follows:

Connection Definition (Create Mode) [[WebService]]

WebService Adapter

Name: WS w HTTP Auth

General | **Web Service** | Options | Description

REST Web Service

Base URL: http://sjc-tesqa-bsiu3:8080/api/tes-6.0

Authentication:

☐ None ☒ HTTP Authentication ☐ OAuth 1.0 ☐ OAuth 2.0

User: TIDALSOFT\qatest

☒ Enabled ☐ Use as default for WebService Jobs

8. In the **Base URL** field, enter the URL for the REST service.
9. From the **Authentication** section, select **HTTP Authentication**.
10. From the **User** list, select the runtime user for Web Service's that represents those users and passwords required for HTTP Authentication
11. Click **OK** to save the connection.

## With OAuth 1.0 Authentication

### To add a REST connection with OAuth 1.0 authentication:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button or right-click and select **Add Connection>WebService Adapter** from the context menu to display the **Web Service Connection Definition** dialog.
3. Enter a name for the new connection in the **Name** field.
4. Click the **General** tab.
5. In the **Job Limit** field, select the maximum number of concurrent active processes that CWA should submit to the REST Web Service host on this server at one time.
6. Click the **Web Service Connection** tab.

## Defining a Web Service Connection

The **Web Service Type** dialog displays.

7. Select **REST Web Service**, and then click **OK**.

The **Web Service** tab displays as follows:

Connection Definition (Create Mode) [[WebService]]

**WebService Adapter**

Name: WS w HTTP Auth

General | **Web Service** | Options | Description

REST Web Service

Base URL: http://sjc-tesqa-bsiu3:8080/api/tes-6.0

Authentication:

☐ None ☐ HTTP Authentication ☒ OAuth 1.0 ☐ OAuth 2.0

Consumer Key:

Consumer Secret:

Access Token:

Token Secret:

Send OAuth Parameters by

☒ Authorization Header ☐ Form Parameter ☐ Query Parameter

Additional OAuth Parameters...

Click the 'Authorize' button if you need to authorize your application to acquire or renew Access Token and Token Secret.

Authorize...

☒ Enabled ☐ Use as default for WebService Jobs

8. In the **Base URL** field, enter the URL for the REST service.

9. From the **Authentication** section, select **OAuth 1.0**.

10. Enter the following values that are required to authenticate with the service provider. You should have obtained the Consumer Key and Secret after registering your Web Service application with the provider.

- Consumer Key
- Consumer Secret
- Access Token

**Note:** If you need to acquire or renew the Access Token, see [Running the OAuth 1.0 Authentication Wizard, page 23](#).

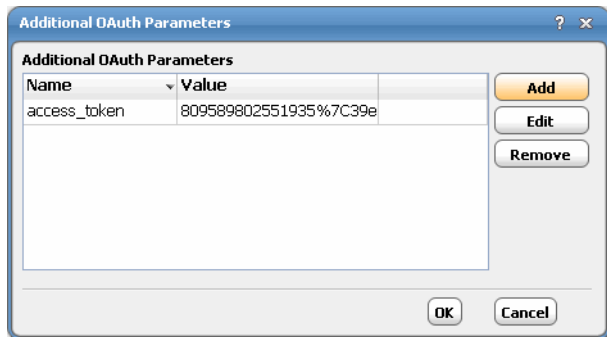
- Token Secret

11. In the **Send OAuth Parameters by** section, select one of the following options:

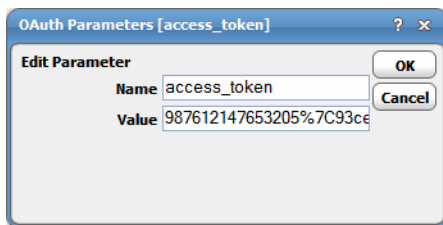
- Authorization Header
- Form Parameter
- Query Parameter

12. Optionally, click **Additional OAuth Parameters...** to add or edit additional OAuth parameters.

The **Additional OAuth Parameters** dialog displays.



- a. Click **Add** or **Edit** to display the **OAuth Parameters** dialog.



- b. In the **Name** field, enter the parameter name.
- c. In the **Value** field, enter the parameter value.
- d. Click **OK** to return to the **Additional OAuth Parameters** dialog.
- e. Click **OK** to return to the **Web Service Connection** tab.
13. Click **OK** to save the connection.

### Running the OAuth 1.0 Authentication Wizard

To use OAuth 1.0 authentication, you are required to to authenticate with the service provider with four values, Consumer Key, Consume Secret, Access Token, and Token Secret.

**Note:** The OAuth 1.0 Authentication Wizard will only work with Web Service providers that support the Out of Band (OOB) callback mechanism.

**To acquire or renew these values:**

1. On the **Web Service** tab, click **Authorize** to launch the wizard.

**OAuth 1.0 Authorization**

This wizard will guide you through the OAuth 1.0 authorization flow with the Web Service provider.

Enter the following information you obtained from the Web Service provider. Then click the 'Next' button.

Required for 'Request Token' Request

Consumer Key	ecwy3uk4ow7bu
Consumer Secret	yxwAxp4xyiUb8n
Permission Scope	
Request Token URL	https://api.linkedin.com/uas/oauth/requestToken

Next > Cancel

2. On the **Authorization** panel, enter the following values that are required to authenticate with the service provider. You should have obtained the Consumer Key and Secret after registering your Web Service application with the provider.

- Consumer Key
- Consumer Secret
- Permission Scope
- Request Token URL

3. Click **Next** to connect to the Web Service provider and view the response.

**OAuth 1.0 Authorization**

The Web Service provider has granted Request Token and Token Secret as following.

Request Token	GkcwcEa0xSZAEMUURZWMjWmSeTxUGqdNjgxGLD8dc
Token Secret	EDibxOnQ4b1VNNi6cbCKOmJrI3hkEKoztsYVa7M1U

View Response...

**Do The Following**

In the text box below, enter authorization URL and the value of Request Token from above, in the format required by the Web Service provider. A typical format is as following  
[https://www.provider.com/oauth/authorize?oauth\\_token=<request token>](https://www.provider.com/oauth/authorize?oauth_token=<request token>)  
 Then click the Authorize button to punch out to the Web Service provider web site. Follow the instructions by the Web Service provider to authorize your application and obtain the Verifier code.  
 When done, come back to this wizard and click the 'Next' button to continue.

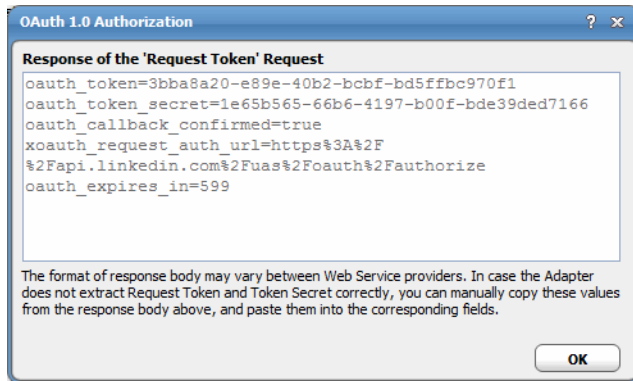
Authorize URL: [https://api.twitter.com/oauth/authorize?oauth\\_token=GkcwcEa0xSZAEMUUR](https://api.twitter.com/oauth/authorize?oauth_token=GkcwcEa0xSZAEMUUR)

Authorize...

< Back Next > Cancel



- Click **View Response** to display the **View Response** panel.



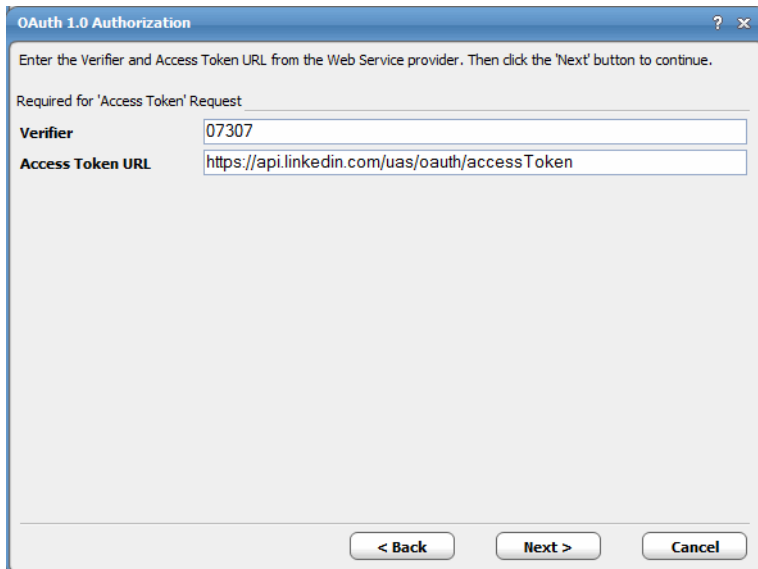
- Click **OK** to return to the **Authorization** panel.
- In the **Authorization URL** field, enter the authorization URL and Request Token value displayed in the **Request Token** field above in the format required by the Web Service provider.

For example, a typical format is as follows:

[https://www.provider.com/oauth/authorize?oauth\\_token=<request token>](https://www.provider.com/oauth/authorize?oauth_token=<request token>)

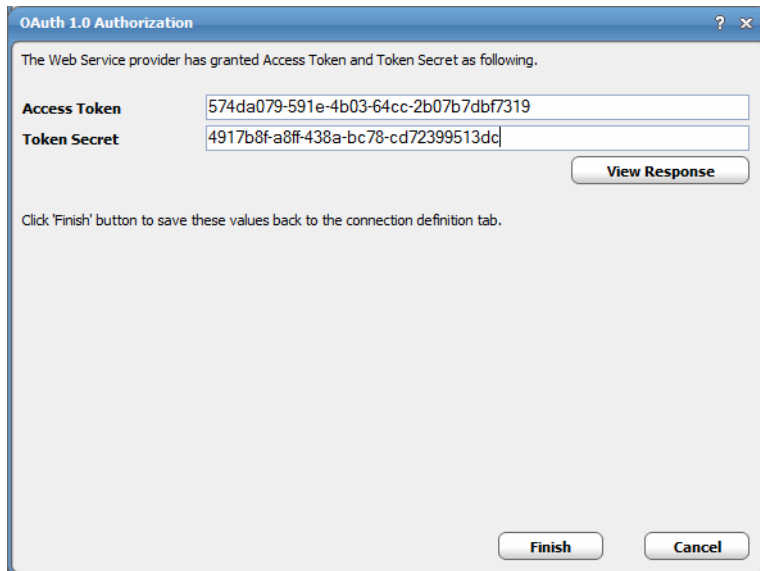
- Click **Authorize** to display the Web Service provider Web site and follow the instructions provided to authorize your application and obtain the Verifier code.
- When complete, return to this wizard, then click **Next** to continue with your OAuth authorization.

The **Verifier and Access Token URL** panel displays.



- In the **Verifier** field, enter the verifier you obtained in Step 7 above.
- In the **Access Token URL** field, enter the URL you obtained from the Web Service provider.
- Click **Next**.

The **Access and Token Granted** panel displays.



12. (Optional) Click **View Response** to display the **View Response** panel.
13. Click **Finish** to save the values and return to the **Web Service Connection** tab.

## With OAuth 2.0 Authentication

### To add a REST connection with OAuth 2.0 authentication:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button or right-click and select **Add Connection>WebService Adapter** from the context menu to display the **Web Service Connection Definition** dialog.
3. Enter a name for the new connection in the **Name** field.
4. Click the **General** tab.
5. In the **Job Limit** field, select the maximum number of concurrent active processes that CWA should submit to the REST Web Service host on this server at one time.
6. Click the **Web Service Connection** tab.

The **Web Service Type** dialog displays.
7. Select **REST Web Service**, and then click **OK**.

The **Web Service** tab displays as follows:

Connection Definition (Create Mode) [[WebService]]

WebService Adapter

Name: Facebook Graph

General Web Service Options Description

REST Web Service

Base URL: https://graph.facebook.com/

Authentication:

☐ None ☐ HTTP Authentication ☐ OAuth 1.0 ☒ OAuth 2.0

OAuth 2.0 Authentication:

☐ Basic ☒ OAuth Parameters

Send OAuth Parameters by

☐ Authorization Header ☐ Form Parameter ☒ Query Parameter

OAuth Parameters

Name	Value
access_token	987612147653205%7C93ce5394a31668523c99b1c9.

Add Edit Remove

☒ Enabled ☐ Use as default for WebService Jobs

8. In the **Base URL** field, enter the URL for the REST service.

9. In the **Authentication** section, select **OAuth 2.0**.

10. In the **OAuth 2.0 Authentication** section:

a. Select one of the following options:

- **Basic** - If selecting this option, enter the Client ID and Client Secret.
- **OAuth Parameters** - If selecting this option, select one of the following options from the **Send OAuth Parameters by** section:
  - Authorization Header
  - Form Parameter
  - Query Parameter

b. To add a parameter, click **Add** to display the **OAuth Parameters** dialog.

OAuth Parameters

Add Parameter

Name:

Value:

OK Cancel

c. Enter the parameter name and its value, then click **OK**.

11. Click **OK**.

## Verifying Web Service Connection Status

If the CWA master cannot connect or loses its connection to a Web Service instance, you will see a red status light next to your Web Service connection in the **Connections** pane. You can still define jobs from the CWA Web client regardless of the connection status.



# 3

## Working with Web Service Jobs

### Overview

You can start creating and scheduling SOAP and REST Web service jobs once you have:

- added the runtime users required for HTTP authentication to run jobs.
- defined your Web service connection(s).

You can create a Web service job using the context menu within the **Jobs** pane. You can also edit, copy and delete an existing Web service job. If you add a Web service job to a CWA job group, items common between the job group and the Web service job are inheritable. However, unless the parent group has a Web Service adapter assigned to it, you must clear the **Inherited** option and choose an appropriate Web Service connection on the **Run** tab.

Selecting the option to **Add a Web Service Job** from the CWA **Jobs** pane displays the **Web Service Job Definition** dialog.

A Web Service job is an invocation of a Web Service operation defined for the connection. When the jobs runs, a SOAP or REST request with the operation name and its arguments is sent to the Web Service. The output, if any, is a SOAP or REST response.

### Adding a SOAP Web Service Job

A CWA job is a set of instructions about how, when and where to perform an automated task. For a Web Service job, all scheduling criteria are available. The only difference between a Web Service job and a standard operating system job is that you specify a Web Service request instead of a command, program or script. In the job rule definition, as with other jobs, you can specify a short name for the job (job alias), where to run the job (agent), the days and the times to run the job, the dependencies needing to be satisfied before it can run and other runtime criteria.

A job or job group definition can be added to the production schedule either manually on demand or automatically through a calendar. Each entry of the job into the production schedule is called a job instance. A job instance is an occurrence of the job definition at a specific time. Job instance history can be reviewed for auditing purposes. Some properties of jobs are described below.

#### To add a SOAP Web Service job:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click and select **Add>Web Service Job** from the context menus to display the **Web Service Job Definition** dialog.
3. In the **Job Name** field, enter a name up to 50 characters in length for your job.

The Job Name is an identifier for CWA only. All of the other job definition information, such as Job Class, Owner and Parent Group, is also the same as a standard CWA job and is used in the same way.

## Adding a SOAP Web Service Job

4. On the **Run** tab, select the SOAP Web Service connection from the **Agent/Adapter Name** list.

If you are including your Web Service job into a group, note that unless the parent group selected has an Web Service host connection assigned, you must clear the **Inherited** option on the **Run** tab before you can select an Web Service connection.

5. Click the **Web Service Job** tab.

The screenshot shows the 'WebService Job Definition [WS 1]' dialog box. At the top, there are fields for 'WebService Job Name' (WS 1), 'Job Class' (empty), 'Parent Group' (empty), and 'Owner' (Schedulers). Below these are tabs for 'WebService', 'Schedule', 'Run', 'Dependencies', 'Resources', 'Job Events', 'Options', 'Run Book', 'Notes', 'History', and 'Images'. The 'Run' tab is selected. Under the 'Run' tab, there are sub-tabs for 'Operation', 'Soap Message', 'Parameters', 'Output Format', and 'Options'. The 'Operation' sub-tab is selected, showing a dropdown menu with 'GetQuote' selected. Below the dropdown is a 'symbol' field containing 'CSCO'. To the right of the symbol field are buttons for '...', 'string', and 'Remove'. At the bottom left, there is a checked 'Enabled' checkbox. At the bottom right, it says 'Last Modified: 07/23/2010 13:59:21'.

6. On the **Operation** tab of the **Web Service Job** tab, select the operation from the **Operation** list.

This list is populated from the Web Service definition (i.e., the WSDL specified in the Web Service adapter connection).

The **Operation** drop-down contains a list of all the operations defined for the service.

7. When you select an operation, the **Parameters** section changes to prompt you for the arguments of that operation.

Fill in either the hard-coded values for each parameter or type a parameter name preceded by a colon (for example, `:stocksymbol1`). The colon indicates that this will be a parameter replaced at runtime.

## Adding a SOAP Web Service Job

8. Click the **SOAP Message** tab to see the SOAP request that will be sent when the job runs. You can also edit the SOAP message directly by selecting the **Override** option.

The screenshot shows the 'WebService Job Definition [WS 1]' dialog box. The 'WebService Job Name' is 'WS 1'. The 'Job Class' and 'Parent Group' are empty. The 'Owner' is 'Schedulers'. The 'Override' checkbox is checked. The 'SOAP Message' tab is selected, showing the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope
xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:soap="http://schemas.xmlsoap.org/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wssoap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:xsd="http://www.w3.org
/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body soapenv:encodingStyle="">
<GetQuote xmlns="http://www.webserviceX.NET/"><symbol>CSCO</symbol></GetQuote>
</soapenv:Body>
</soapenv:Envelope>
```

The 'Enabled' checkbox is checked. The 'Last Modified' date is '07/23/2010 13:59:21'.

9. Click the **Parameters** tab if you have specified parameters on the **Operation** tab preceded by a colon.

These parameters will be listed on the **Parameters** tab.

The screenshot shows the 'WebService Job Definition [WebService 1]' dialog box. The 'WebService Job Name' is 'WebService 1'. The 'Job Class' and 'Parent Group' are empty. The 'Owner' is 'Schedulers'. The 'Parameters' tab is selected, showing a table with the following data:

Name	Value
STOCKSYMBOL	<STOCKSYMBOL.8>

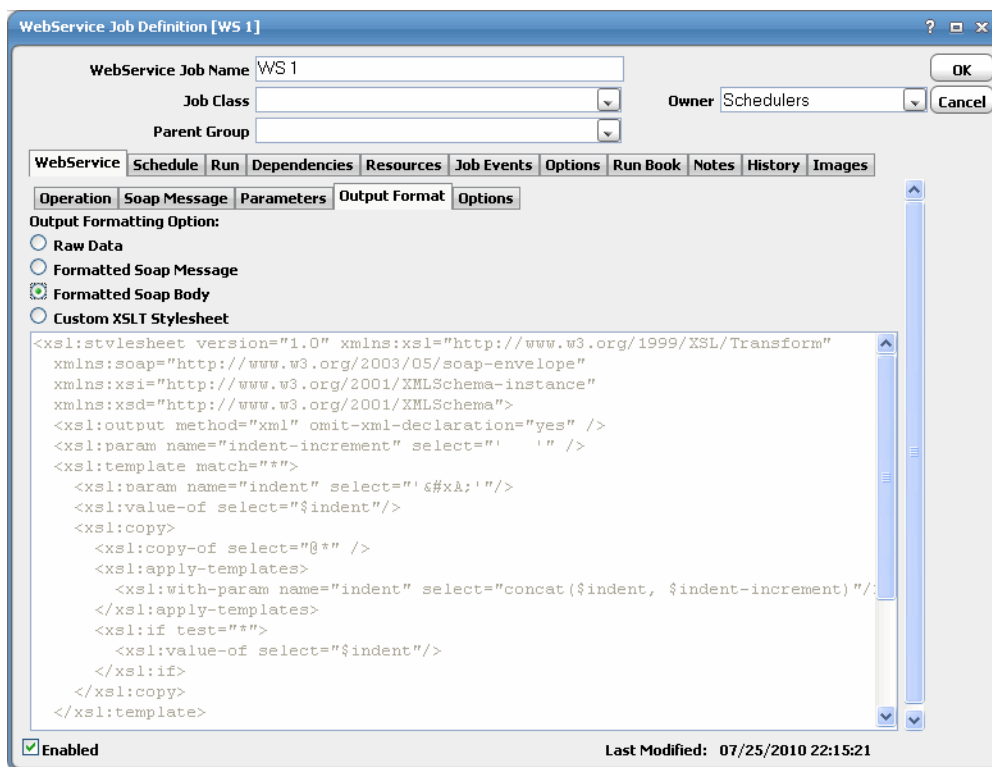
The 'Enabled' checkbox is checked. The 'Last Modified' date is '07/23/2010 13:59:21'.

10. For each parameter listed, click **Edit** to display the **Variable Definition** dialog.



11. Provide a hard-coded value or click **Variables** to insert a dynamically replaced variable value, then click **OK**.

12. Click the **Output Format** tab to configure the output format.



You can apply an XSLT stylesheet or a predefined format to the SOAP response to transform it before it is returned in the job output.

This tab includes the following options:

- **Raw Data** – Click this option to see the raw data as output.
- **Formatted Soap Message** – Click this option to see a formatted soap message as output.
- **Formatted Soap Body** – Click this option to see a formatted soap body as output.
- **Custom XSLT Stylesheet** – Click this option to use a custom XSLT stylesheet applied to the output.

XSLT is a language for transforming XML documents into other XML documents, or can be used to extract tagged elements. For more information, go to [www.w3.org/TR/xslt](http://www.w3.org/TR/xslt).

- **Load from File** – Click this button to apply a predefined format to the SOAP response to transform it before it is returned in the job output.



## Adding a REST Web Service Job

13. Select the output formatting from the options listed in the **Output Formatting Option** section.

If selecting **Custom XSLT Stylesheet**:

- a. Click **Load from File** to display the **Select XSLT File** dialog.
- b. Locate the XSLT file and click **Open**.

The file text displays in the **Custom XSLT Stylesheet** field.

14. Click the **Options** tab to set the amount of time you want the job to run before timing out (in seconds).

15. Click **OK** to add the job.

## Adding a REST Web Service Job

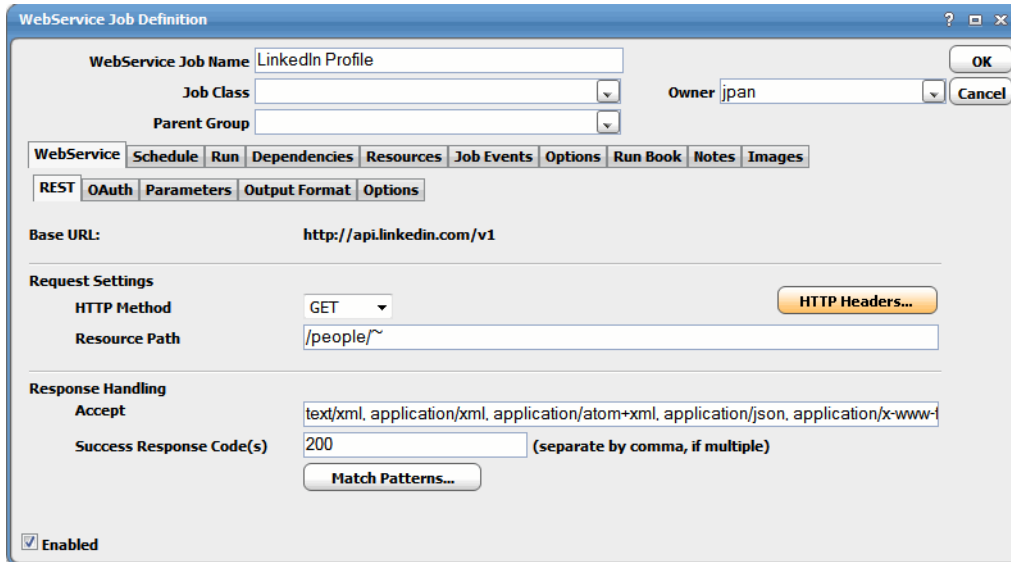
### GET Method Example

**To add a REST Web Service job using the GET method:**

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click and select **Add>Web Service Job** from the context menus to display the **Web Service Job Definition** dialog.
3. In the **Job Name** field, enter a name up to 50 characters in length for your job.
4. On the **Run** tab, select an agent/adaptor name from the **Agent/Adapter Name** list.

## Adding a REST Web Service Job

5. Click the **Web Service Job** tab.



The **WebService Job Definition** dialog box is shown. The **WebService Job Name** is "LinkedIn Profile". The **Job Class** is empty. The **Owner** is "jpan". The **Parent Group** is empty. The **REST** tab is selected. The **Base URL** is "http://api.linkedin.com/v1". The **Request Settings** section shows **HTTP Method** as "GET" and **Resource Path** as "/people/~". The **Response Handling** section shows **Accept** as "text/xml, application/xml, application/atom+xml, application/json, application/x-www-t" and **Success Response Code(s)** as "200". The **Match Patterns...** button is visible. The **Enabled** checkbox is checked.

6. On the **REST** tab's **Request Setting** section, select the **GET (Read)** request method from the **HTTP Method** list.

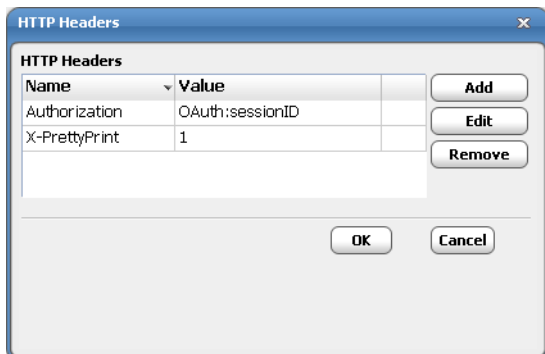
Other standard methods are:

- **POST** (Create)
- **PUT** (Update)
- **DELETE** (Delete)

7. (Optional) In the **Resource Path** field, enter the path (under the base URL) to the Web resource this request will apply (including query string, if applicable).

If not specified, the base URL alone will be used to carry out the request.

8. Click **HTTP Headers** to specify HTTP headers to be sent along with the request. The **HTTP Headers** dialog displays.

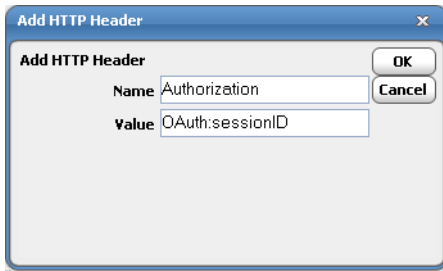


The **HTTP Headers** dialog box is shown. It contains a table with the following headers and values:

Name	Value
Authorization	OAuth:sessionID
X-PrettyPrint	1

Buttons for **Add**, **Edit**, and **Remove** are present. At the bottom are **OK** and **Cancel** buttons.

Click **Add** to display the **Add HTTP Header** dialog, enter the parameters, then click **OK**.

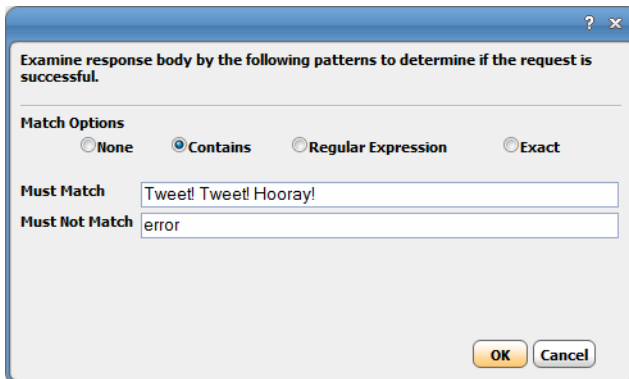


CWA parameters are supported in the value field of each HTTP header entry.

9. (Optional) In the **Accept** field of the **Response Handling** section, specify the MIME type expected of the response.
10. (Optional) In the **Success Response Code(s)** field, specify the HTTP response code(s) expected in the response.

If multiple numbers are entered, the Web Service adapter will consider success of job run as long as one of the numbers is received. Use a comma to separate the numbers. For example, **200,201**.

11. (Optional) Click **Match Patterns** to set the parameters for examining the response body to determine if the request is successful.



12. In the **Match Options** section, select one of the following options:

- **None** – Disables pattern matching option, even though pattern text remain specified.
- **Contains** – The response body must contain the text specified in the **Must Match** field (if not empty) and must not contain the text in the **Must Not Match** field (if not empty).
- **Regular Expression** – The response body must match the regular expression specified in the **Must Match** field (if not empty) and must not match the regular expression in the **Must Not Match** field (if not empty).
- **Exact** – The response body must be exactly the same as the text specified in the **Must Match** field (if not empty) and must not be exactly the same as the text in the **Must Not Match** field (if not empty).

13. Click **OK**.

14. Click the **OAuth** tab to add, edit, delete, or override parameters inherited from a connection.

The dialog box is titled "WebService Job Definition [Facebook Attending]". It has a "WebService Job Name" field with the value "Facebook Attending", a "Job Class" dropdown, a "Parent Group" dropdown, and an "Owner" field with the value "jpan". There are "OK" and "Cancel" buttons. Below these are tabs for "WebService", "Schedule", "Run", "Dependencies", "Resources", "Job Events", "Options", "Run Book", "Notes", "History", and "Images". The "OAuth" tab is selected. Under the "OAuth" tab, there is a "REST" button and a "Parameters" button. Below these is an "Override" checkbox. The "Send OAuth Parameters by" section has three radio buttons: "Authorization Header", "Form Parameter", and "Query Parameter". Below this is a table for "OAuth Parameters" with columns "Name" and "Value". The table contains one row with "access\_token" and "242312155803205%7C93ce5394a316". To the right of the table are "Add", "Edit", and "Remove" buttons. At the bottom, there is an "Enabled" checkbox and a "Last Modified" timestamp: "09/14/2011 17:29:02".

15. (Optional) Check the **Override** checkbox if you want to override the current OAuth parameters.

- a. In the **Send OAuth Parameters** by section, select one of the following options:
  - Authorization Header
  - Form Parameter
  - **Query Parameter**
- b. Select the parameter, then click **Edit** to display the **OAuth Parameter** dialog.

The dialog box is titled "OAuth Parameters [access\_token]". It has an "Edit Parameter" section with a "Name" field containing "access\_token" and a "Value" field containing "2423987087%793cd5394a3". There are "OK" and "Cancel" buttons.

- c. Modify the name and/or value, then click **OK**.

16. Click the **Parameters** tab if you have specified parameter (i.e., prefixed by a colon) in the **REST** tab.

WebService Job Definition [Twitter Status]

WebService Job Name: Twitter Status

Job Class: [dropdown]

Parent Group: [dropdown]

Owner: qatest

Buttons: OK, Cancel

Tabs: WebService, Schedule, Run, Dependencies, Resources, Job Events, Options, Run Book, Notes, History, Images

Sub-tabs: REST, OAuth, Parameters, Output Format, Options

Name	Value
status	<JobName> (<JobID>) stated by

Edit button

☒ Enabled

Last Modified : 10/17/2011 15:50:11

17. To edit a parameter, highlight the parameter, then click **Edit** to display the **Parameter Definition** dialog.

Parameter Definition [runtime\_status]

Parameter Define

Parameter Name: runtime\_status

Parameter Value: Action '<ActionName>' trigger

Buttons: OK, Cancel

Variables button

18. Edit the values, and then click **OK**.
19. Click the **Output Format** tab to configure the output format.
20. Click the **Options** tab to set the amount of time you want the job to run before timing out (in seconds).
21. Click **OK** to add the job.

## POST Method Example

To add a REST Web Service job using the POST method:

1. Click the **Web Service Job** tab.

**WebService Job Definition**

WebService Job Name: Facebook Attending

Job Class: [dropdown]

Owner: Schedulers

Parent Group: [dropdown]

Base URL: http://sjc-tesqe-auto1.tidalsoft.local:8080/api/tes-6.0/

**Request Settings**

HTTP Method: POST

Resource Path: 236734218439208/attending

Content-Type: [dropdown]

Request Body: [text area]

**Response Handling**

Accept: text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded

Success Response Code(s): 200

Match Patterns...

☒ Enabled

2. On the **REST** tab's **Request Setting** section, select the **POST** request method.
3. Click **HTTP Headers** to specify HTTP headers to be sent along with the request. The **HTTP Headers** dialog displays.
4. Click **Add** to display the **Add HTTP Header** dialog, enter the parameters, then click **OK**.  
CWA parameters are supported in the value field of each HTTP header entry.
5. (Optional) In the **Resource Path** field, enter the path (under the base URL) to the Web resource this request will apply (including query string, if applicable).  
If not specified, the base URL alone will be used to carry out the request.
6. In the **Content Type** field, enter the content type manually or click **Select** to specify the content type.

This describes the request body to be sent to the Web Service provider.

For example:

Select ▾

- application/x-www-form-urlencoded
- application/json
- application/xml
- text/xml

## Adding a REST Web Service Job

If a value other than **application/x-www-form-urlencoded** is selected for the content type, the text that makes up the request body to be sent to the Web Service provider displays in the **Request Body** field.

**WebService Job Definition**

WebService Job Name: TES sacmd agent -C LIST

Job Class: [dropdown]

Parent Group: [dropdown]

Owner: jpan

Base URL: http://localhost:8080/api/production/post

**Request Settings**

HTTP Method: POST

Resource Path: [empty]

Content-Type: text/xml

Request Body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<atom:entry xmlns:atom="http://purl.org/atom/ns#">
  <atom:id>1</atom:id>
  <atom:title>HTTP</atom:title>
  <CmdLine.tescmd>
    <cmd>agent -C LIST</cmd>
  </CmdLine.tescmd>
</atom:entry>
```

**Response Handling**

Accept: text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded

Success Response Code(s): 200 (separate by comma, if multiple)

☒ Enabled

If specifying the content by selecting **application/x-www-form-urlencoded**, the adapter displays the **Form Parameters** field allowing you to specify URL encoded form parameters (i.e., name-value pairs).

The screenshot shows the 'WebService Job Definition' dialog box. The 'WebService Job Name' is 'Twitter Tweet'. The 'Job Class' and 'Parent Group' are set to default values. The 'Owner' is 'jpan'. The 'Base URL' is 'https://api.twitter.com/1'. The 'Request Settings' section shows 'HTTP Method' as 'POST', 'Resource Path' as '/statuses/update.json', and 'Content-Type' as 'application/x-www-form-urlencoded'. A dropdown menu for 'Content-Type' is open, showing options: 'application/x-www-form-urlencoded', 'application/json', 'application/xml', and 'text/xml'. The 'Form Parameters' table is empty. The 'Response Handling' section shows 'Accept' as 'text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded' and 'Success Response Code(s)' as '200'. The 'Enabled' checkbox is checked.

WebService Job Definition

WebService Job Name: Twitter Tweet

Job Class: [dropdown]

Parent Group: [dropdown]

Owner: jpan

Base URL: https://api.twitter.com/1

Request Settings

HTTP Method: POST

Resource Path: /statuses/update.json

Content-Type: application/x-www-form-urlencoded

Form Parameters

Name	Value
------	-------

Response Handling

Accept: text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded

Success Response Code(s): 200 (separate by comma, if multiple)

Match Patterns...

Enabled



## Adding a REST Web Service Job

You can add, edit, and remove parameters by selecting the form parameter listed, then clicking the appropriate button.

**Webservice Job Definition**

Webservice Job Name:  OK

Job Class:  Owner:  Cancel

Parent Group:

**Webservice** | Schedule | Run | Dependencies | Resources | Job Events | Options | Run Book | Notes | Images

REST | OAuth | Parameters | Output Format | Options

Base URL:

**Request Settings**

HTTP Method:  HTTP Headers...

Resource Path:

Content-Type:  Select v

**Form Parameters**

Name	Value
status	Tweet! Tweet! Hooray!

Add Edit Remove

**Response Handling**

Accept:

Success Response Code(s):  (separate by comma, if multiple)

Match Patterns...

☒ Enabled

7. In the **Accept** field of the **Response Handling** section, specify the MIME type for the response.
8. (Optional) In the **Success Response Code(s)** field, specify the HTTP response code(s) expected in the response.  
If multiple numbers are entered, the Web Service adapter will consider success of job run as long as one of the numbers is received. Use a comma to separate the numbers. For example, **200,201**.
9. (Optional) Click **Match Patterns** to set the parameters for examining the response body to determine if the request is successful.

The **Match Patterns** dialog displays.

10. Click the **OAuth** tab to add, edit, delete, or override parameters inherited from a connection.
11. Click the **Parameters** tab if you have specified OAuth parameters.

These parameters will be listed on the **Parameters** tab.

12. Click **Add** to display the **Variable Definition** dialog.
13. Enter the parameter values, and then click **OK**.


To edit an existing parameter, select the parameter and click **Edit** to display the **Variable Definition** dialog.

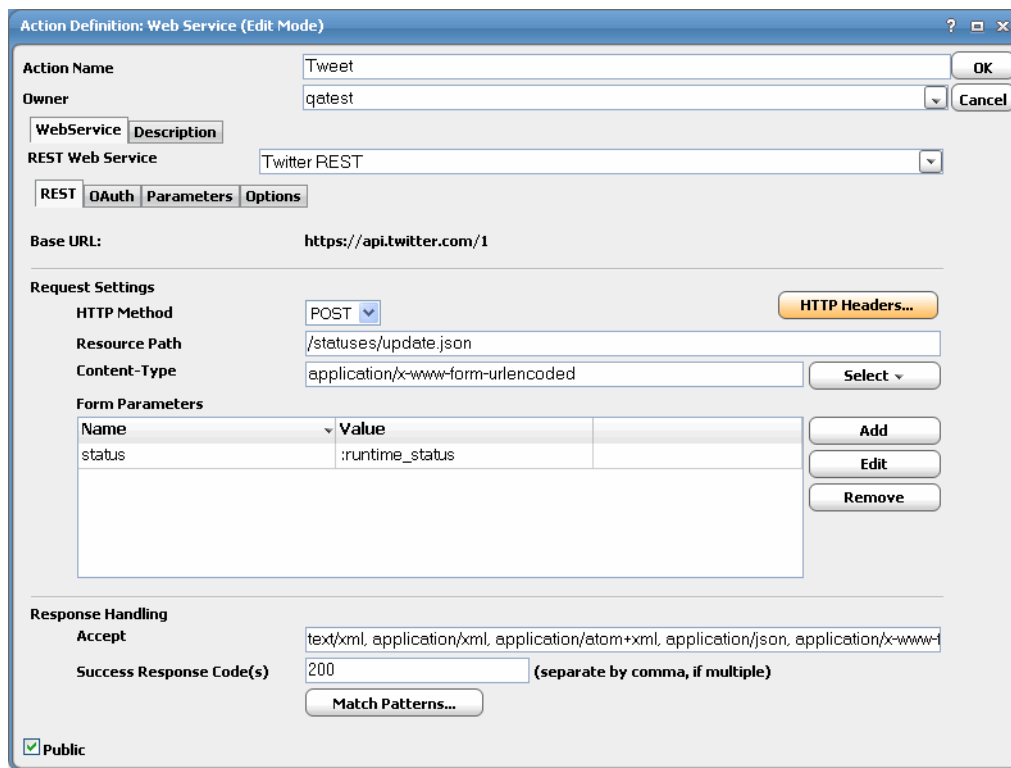
14. Click the **Output Format** tab to configure the output format.
15. Click the **Options** tab to set the amount of time you want the job to run before timing out (in seconds).
16. Click **OK** to add the job.

## Defining a Web Service Action

The Web Service adapter allows you to trigger events as an CWA Action type. This action can then be associated with any CWA event, including job events such as "Job Completed Normally" or file, email, variable events, etc (refer to CWA documentation on how to associate actions with scheduling events). When the action triggers a custom event in Web Services, any pending scheduled task waiting on the event will kick off.

### To define an action:

1. In the **Navigator** pane, select **Definitions>Actions>Web Service Actions** to display the **Web Service Actions** pane.
2. Right-click **Web Service** and select **Add WebService Action** from the context menus.
3. On the CWA toolbar, click the **Add**  button to display the **Action Definition** dialog.



**Action Definition: Web Service (Edit Mode)**

**Action Name:** Tweet OK

**Owner:** qatest Cancel

**WebService** **Description**

**REST Web Service:** Twitter REST

**REST** **OAuth** **Parameters** **Options**

**Base URL:** https://api.twitter.com/1

**Request Settings**

**HTTP Method:** POST HTTP Headers...

**Resource Path:** /statuses/update.json

**Content-Type:** application/x-www-form-urlencoded Select ▾

**Form Parameters**

Name	Value
status	:runtime_status

Add Edit Remove

**Response Handling**

**Accept:** text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded

**Success Response Code(s):** 200 (separate by comma, if multiple) Match Patterns...

☒ **Public**

4. In the **Action Name** field, enter the name of the new action.
5. Select the owner of the action from the **Owner** list.
6. From the **REST Web Service** drop-down list, select the previously defined REST connection you want to associate the action with.

The **Base URL** field contains the URL for the selected REST Web Service and is read-only.

7. From the **HTTP Method** list, select the request method. In the example above, **POST** is selected.
8. Click **HTTP Headers** to specify HTTP headers to be sent along with the request. The **HTTP Headers** dialog displays.
9. Click **Add** to display the **Add HTTP Header** dialog, enter the parameters, then click **OK**.

CWA parameters are supported in the value field of each HTTP header entry.

10. (Optional) In the **Resource Path** field, enter the path (under the base URL) to the Web resource this request will apply (including query string, if applicable).

If not specified, the base URL alone will be used to carry out the request.

11. In the **Content Type** field, enter the content type manually or click **Select** to specify the content type.

This describes the request body to be sent to the Web Service provider.

If text is selected, the text that makes up the request body to be sent to the Web Service provider displays in the **Request Body** field.

If specifying the content by selecting **application/x-www-form-urlencoded**, the adapter displays the **Form Parameters** field allowing you to specify URL encoded form parameters (i.e., name-value pairs).

12. You can add, edit, and remove parameters by selecting the form parameter listed, then clicking the appropriate button.
13. In the **Accept** field of the **Response Handling** section, specify the MIME type for the response.
14. (Optional) In the **Success Response Code(s)** field, specify the HTTP response code(s) expected in the response.  
  
If multiple numbers are entered, the Web Service adapter will consider success of job run as long as one of the numbers is received. Use a comma to separate the numbers. For example, **200,201**.
15. (Optional) Click **Match Patterns** to set the parameters for examining the response body to determine if the request is successful.
16. Click the **OAuth** tab to add, edit, delete, or override parameters inherited from a connection.
17. Click the **Parameters** tab if you have specified OAuth parameters. These parameters will be listed on the **Parameters** tab.
18. Click **Add** to display the **Variable Definition** dialog.
19. Enter the parameter values, and then click **OK**.
20. To edit an existing parameter, select the parameter and click **Edit** to display the **Variable Definition** dialog.
21. Click the **Output Format** tab to configure the output format.
22. Click the **Options** tab to set the amount of time you want the job to run before timing out (in seconds).
23. Click **OK** to add the action.

The **Description** field is a read-only field that displays the description corresponding to the selected custom event.

## Monitoring Job Activity

The **Job Details** dialog displays by double-clicking on a job instance record in the **Job Activity** pane or by right-clicking and selecting the **Details** option from the context menu. The **Job Detail** dialog provides information on the job after it has completed or as it is still running. The tabs of this dialog specific to the Web Service adapter are the **Output**, **Web Service**, and **Run Info** tabs.

Job Details [WS 1 (1)]

Job Name: WS 1 (1) Job No.: 665 OK

Current Status

Status: Completed Normally Reruns: 46

Est. Start Time: 7:30 AM Disable Carryover: ☐

Act. Start Time: 7:30 AM

Est. Duration: 0 min 6 s

Act. Duration: 0 min 7 s

Job Owner: qatest

Scheduled By: Calendar

Exit Code: 0

External ID: 2

Print Defaults

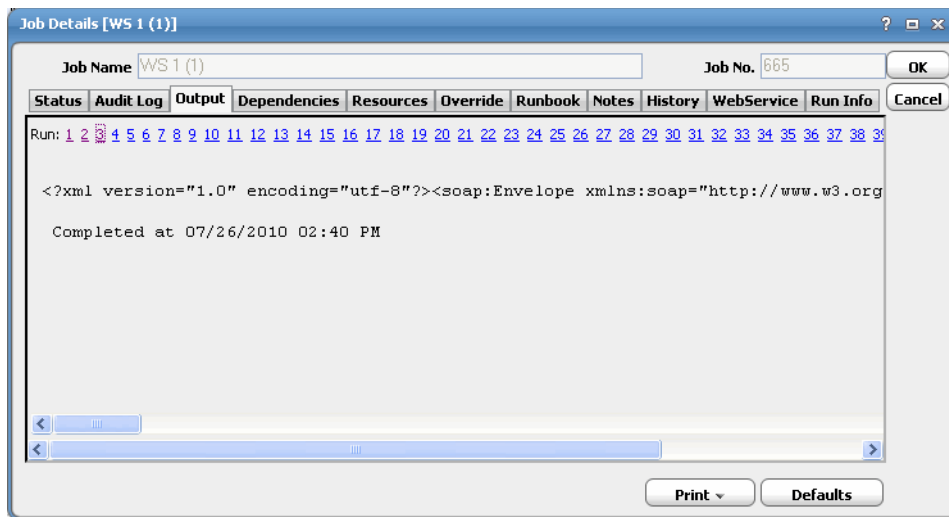
## Output Tab

The **Output** tab of the **Job Detail** dialog, if job is configured to save output, displays the SOAP/REST Response from the Web Service. CWA can be configured to save, append, or discard job output by default from the **Defaults** tab of the **System Configuration** dialog. Regardless of the system default, any individual job instance can be configured from its job definition to override the system default. If you have the **Append** option configured, each time a job is rerun that run's output is separated by a block of number signs (#).

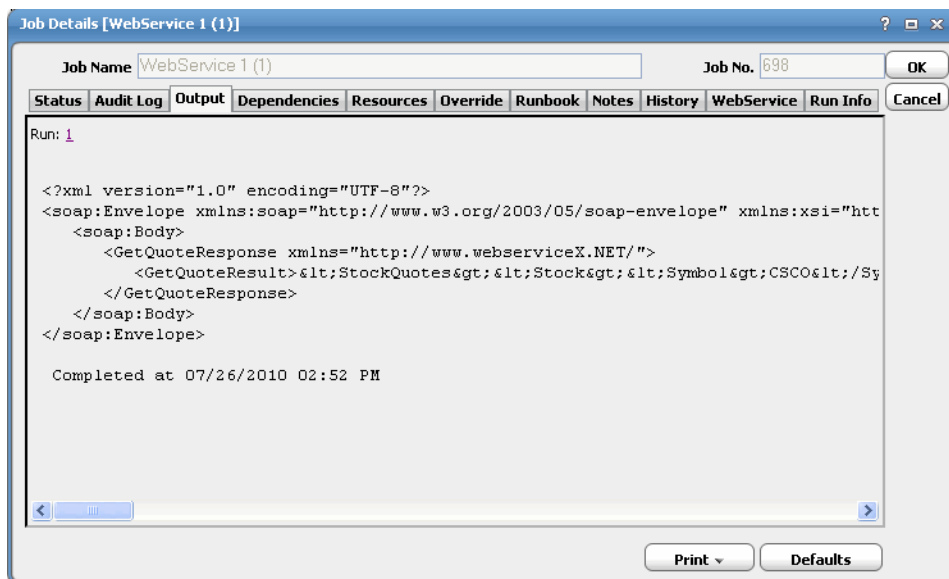
**Note:** CWA's default is to discard job output. If you want to be able to view job output, you must select the Save Output option on the Options tab in the Job Definition dialog or change the system default on the Defaults tab in the System Configuration dialog.

## Monitoring Job Activity

This example shows the output for a selection of **Raw Data** as the output format:



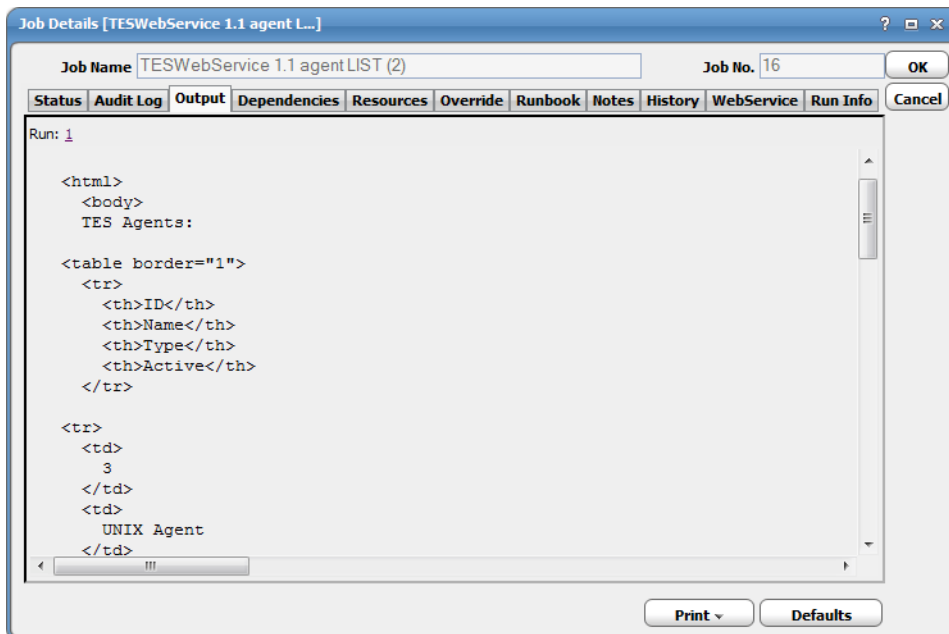
This example shows the output for a selection of **Formatted Soap Message** as the output format:



This example shows the output for a selection of **Formatted Soap Body** as the output format:



This example shows the **Output** tab when an XSLT is applied to extract just the <Result> value from the SOAP response:



## WebService Tab

The **Webservice** tab of the **Job Detail** dialog contains the request with the variables used when this job was submitted. This tab allows you to override the parameter values for a job rerun. In addition to overriding individual parameter values, you can also directly modify the SOAP message.

## For a SOAP Web Service Job

The **WebService** tab of a SOAP Web Service job displays as follows:

Job Details [WebService 2 (1)]

Job Name: WebService 2 (1) Job No. 718 OK

Status Audit Log Output Dependencies Resources Override Runbook Notes History WebService Run Info Cancel

Operation Soap Message Parameters Output Format Options

Operation: GetQuote

symbol :stocksymbol string Remove

Print Defaults

This tab contains the following sub-tabs:

- **Operation** - Contains the SOAP Web Service operation defined for the connection.
- **Soap Message** - Contains the SOAP request that was sent when the job ran. You can edit the SOAP message directly by selecting the **Override** option.

Job Details [TES SOAP 1.2 listrule (1)]

Job Name: TES SOAP 1.2 listrule (1) Job No. 2335 OK

Status Audit Log Output Dependencies Resources Override Runbook Notes History WebService Run Info Cancel

Operation Soap Message Parameters Output Format Options

☒ Override

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope" xmlns:soap="http://schemas.xmlsoap.org/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wssoap12="http://schemas.xmlsoap.org/
wsdl/soap12/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance">
<soapenv:Body soapenv:encodingStyle="">
<listrule xmlns="http://www.tidalsoftware.com/teswebservice"/>
</soapenv:Body>
</soapenv:Envelope>
```

Print Defaults

## Monitoring Job Activity

- **Parameters** – Contains the parameters associated with the job. You can click **Edit** to display the **Variable Definition** dialog and manually enter a new parameter value or click **Variables** to select a system-defined variable.

Job Details [Twitter Status (1)]

Job Name: Twitter Status (1) Job No. 2325

Buttons: OK, Cancel

Tabs: Status, Audit Log, Output, Dependencies, Resources, Override, Runbook, Notes, History, Webservice, Run Info

Sub-tabs: REST, OAuth, Parameters, Output Format, Options

Name	Value
status	<JobName> (<JobID>) stated by <JobRunUser>

Edit button

Print Defaults

- **Output Format** – Contains the output format for the response. You can select a different output format after the job has run.

Job Details [TES SOAP 1.2 listrule (1)]

Job Name: TES SOAP 1.2 listrule (1) Job No. 2335

Buttons: OK, Cancel

Tabs: Status, Audit Log, Output, Dependencies, Resources, Override, Runbook, Notes, History, Webservice, Run Info

Sub-tabs: Operation, Soap Message, Parameters, Output Format, Options

Output Formatting Option

☐ Raw Data

☒ Formatted Soap Message

☐ Formatted Soap Body

☐ Custom XSLT Stylesheet

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsl:output method="xml" omit-xml-declaration="no" />
  <xsl:param name="indent-increment" select="' ' ' ' />
  <xsl:template match="*">
    <xsl:param name="indent" select="'&#xA;'" />
    <xsl:value-of select="$indent" />
    <xsl:copy>
      <xsl:copy-of select="*" />
      <xsl:apply-templates>
        <xsl:with-param name="indent" select="concat($indent, $indent-increment)" />
      </xsl:apply-templates>
      <xsl:if test="*">
        <xsl:value-of select="$indent" />
      </xsl:if>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="comment()|processing-instruction()">
    <xsl:copy />
  </xsl:template>
</xsl:stylesheet>
```

Print Defaults



## Monitoring Job Activity

- **Options** – Contains the amount of time set for the job to run before timing out (in seconds). You can modify this amount on this tab.

The screenshot shows the 'Job Details [Twitter Status (1)]' dialog box with the 'Options' tab selected. The 'Job Name' is 'Twitter Status (1)' and the 'Job No.' is '2325'. The 'Timeout (in seconds)' is set to '75'. The 'Options' tab is highlighted in the sub-tab bar. At the bottom, there are 'Print' and 'Defaults' buttons.

## For a REST Web Service Job

The **WebService** tab of a REST Web Service job displays as follows:

The screenshot shows the 'Job Details [Twitter Status (1)]' dialog box with the 'WebService' tab selected. The 'Base URL' is 'https://api.twitter.com/1'. The 'Request Settings' section includes 'HTTP Method' (POST), 'Resource Path' (statuses/update.json), and 'Content-Type' (application/x-www-form-urlencoded). The 'Form Parameters' table has one row: 'status' with value ':status'. The 'Response Handling' section includes 'Accept' (text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded) and 'Success Response Code(s)' (200). At the bottom, there are 'Print' and 'Defaults' buttons.

This tab contains the following sub-tabs:

- **REST** – Contains the REST Web Service settings defined for the connection.

- **OAuth** – Contains the OAuth parameters inherited from a connection. You can override, add, edit or delete additional parameters on this tab.

Job Details [Twitter Status (1)]

Job Name: Twitter Status (1) Job No.: 2325

Tabs: Status | Audit Log | Output | Dependencies | Resources | Override | Runbook | Notes | History | Webservice | Run Info

Sub-tabs: REST | OAuth | Parameters | Output Format | Options

☒ Override

Send OAuth Parameters by:

☐ Authorization Header ☐ Form Parameter ☒ Query Parameter

Name	Value
Example Parameter	3333

Buttons: Add, Edit, Remove

Buttons: Print, Defaults

- **Parameters** – Contains the parameters associated with the job. You can click **Edit** to display the **Variable Definition** dialog and manually enter a new parameter value or click **Variables** to select a system-defined variable
- **Output Format** – Contains the output format for the response. You can select a different output format after the job has run.
- **Options** – Contains the amount of time set for the job to run before timing out (in seconds). You can modify this amount on this tab.

### Overriding Parameters

#### To override the parameter value listed:

1. On the **Web Service** tab, click the **Parameters** tab.
2. Highlight the parameter, then click **Edit** to display the **Variable Definition** dialog and manually enter a new parameter value. You can also click **Variables** to select a system-defined variable.
3. Click **OK**.
4. On the **Job Activity** pane, right-click on this job and select **Job Control>Rerun** from the context menu.

### Run Info Tab

The **Run Info** tab of the **Job Detail** dialog contains the request that was submitted to the Web Service. Each tab reflects the last run of this Web Service job instance. This tab is read-only.

**Note:** This may or may not be the same thing you see on the Webservice tab depending on whether you have made any changes to this job instance since the last run.

## For a SOAP Web Service Job

The **Run Info** tab of a SOAP Web Service job displays as follows:

Job Details [WS (1)]

Job Name: WS (1) Job No.: 706

OK Cancel

Status Audit Log Output Dependencies Resources Override Runbook Notes History Webservice Run Info

Operation Soap Message Parameters Output Format Options

Operation: GetDelayedChart

Symbol	a	string
AdditionalSymbols	b	string
StartTime	c	string
EndTime	d	string
Style	AreaPercentage	string
Width	e	int
Height	f	int

Print Defaults

This tab contains the following sub-tabs:

- **Operation** - Contains the SOAP Web Service operation defined for the connection.
- **Soap Message** - Contains the SOAP request that was sent when the job ran.
- **Parameters** - Contains the parameters associated with the job.
- **Output Format** - Contains the output format for the response.
- **Options** - Contains the amount of time set for the job to run before timing out (in seconds).

## For a REST Web Service Job

The **Run Info** tab of the **Job Detail** dialog contains the request that was submitted to the Web Service. Each tab reflects the last run of this Web Service job instance. This tab is read-only.

**Note:** This may or may not be the same thing you see on the WebService tab depending on whether you have made any changes to this job instance since the last run.

Job Details [Twitter Status (1)]

Job Name: Twitter Status (1) Job No. 2325

Tabs: Status | Audit Log | Output | Dependencies | Resources | Override | Runbook | Notes | History | WebService | Run Info

Sub-tabs: REST | OAuth | Output Format | Options

Base URL: https://api.twitter.com/1

**Request Settings**

HTTP Method: POST

Resource Path: statuses/update.json

Content-Type: application/x-www-form-urlencoded

**Form Parameters**

Name	Value
status	Twitter Status (2325) stated by qatest at 14:27:39, 11/3/2011

**Response Handling**

Accept: text/xml, application/xml, application/atom+xml, application/json, application/x-www-form-urlencoded

Success Response Code(s): 200 (separate by comma, if multiple)

Buttons: HTTP Headers..., Select, Add, Edit, Remove, Match Patterns, Print, Defaults

This tab contains the following sub-tabs:

- **REST** - Contains the REST Web Service settings defined for the connection.
- **OAuth** - Contains the OAuth parameters inherited from a connection.
- **Parameters** - Contains the parameters associated with the job.
- **Output Format** - Contains the output format for the response.
- **Options** - Contains the amount of time set for the job to run before timing out (in seconds).

## 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**.



# 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.

## service.props Properties

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, <b>kerbkdc=172.25.6.112</b>
kerbrealm	MapReduce	<none>	If the Hadoop cluster is Kerberos secured, use this value to specify the Kerberos Realm.  For example, <b>kerbrealm=TIDALSOFT.LOCAL</b>
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.