opentext™

OpenText™ Identity Manager
Driver for Workday Implementation Guide

January 2025

Legal Notice

Copyright 2009 - 2024 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

For additional information, such as certification-related notices and trademarks, see http://www.microfocus.com/about/legal/ (http://www.microfocus.com/about/legal/).

1 Understanding the Workday Driver 7

Driver Concepts . . . 7
- Data Management . . . 7
- How the Driver Works . . . 8

Support for Standard Driver Features . . . 9
- Local Platforms . . . 9
- Remote Platforms . . . 10
- Supported Operations . . . 10

2 Installing and Configuring the Workday Driver 13

Workday Driver Packages . . . 13
Checklist for Installing and Configuring the Driver . . . 14
Creating Integration System Group(s) and User(s) in Workday Application . . . 15
- Creating Integration System Security Group(s) . . . 15
- Create Integration System . . . 17
- Creating Integration System User(s) . . . 17

Generating Client ID and Token Endpoint for OAuth 2.0 Authentication Method . . . 18
- Creating x509 Public Key Certificate . . . 18
- Generating Client ID and Token Endpoint . . . 19

Installing the Driver Files . . . 19
- Installing the Driver Files as a Root User . . . 20
- Installing the Driver Files as a Non-Root User . . . 20
- Configuring Remote Loader for the Driver Instance . . . 22

Creating the Driver Object . . . 22
- Importing the Current Driver Packages . . . 22
- Installing and Configuring the Driver Object . . . 23

Configuring Phone Contact Information in Subscriber Channel . . . 38
- Fetching Country Phone Code ID and Phone Type ID from Workday . . . 38
- Adding the Phone Contact Information in Subscriber Channel . . . 39

Configuring Entitlements . . . 40
- Fetching Entitlements from Workday Portal . . . 40
- Assigning Permissions to a User . . . 41

Configuring the Driver to Include Additional Transaction Type References . . . 41
Deploying the Driver Object . . . 42
Starting the Driver . . . 43
Activating the Driver . . . 43
Advanced Synchronization of Worker Photos from the Workday Portal . . . 43

3 Customizing the Workday Driver 51

Customizing the Publisher Channel for Additional Attributes . . . 51
- Modifying an Existing Attribute . . . 51
- Adding a New Attribute . . . 52

Customizing the Subscriber Channel for Additional Attributes . . . 53
Customizing Entitlements . . . 55
- Prerequisites . . . 56
- Supported Entitlements . . . 56

4 Lifecycle Business Process for the Workday Driver 59

Business Process Use Cases . . . 59

Details of Business Process Package and Associated Files 60
Upgrading the Workday Driver to Use Business Process Lifecycle 60
Configuring Business Process 61

5 Attributes Mapping & Filtering 63

User Attributes 63
Job Profile Attributes 82
Job Family Attributes 83
Relation Attributes 83
Photo Attributes 89
Location Attributes 89
Organization Attributes 91
Delta Object Attributes 92
Future Object Attributes 93
Attribute Filter Settings 93
Users Attributes Filter Settings 93
Relations Attributes Filter Settings 94
Job Family Attributes Filter Settings 95
Location Attributes Filter Settings 96
Photo Attributes Filter Settings 96
Job Profile Attributes Filter Settings 97
Organization Attributes Filter Settings 97
Delta Object Attributes Filter Settings 98

6 Upgrading an Existing Driver 99

7 Securing Communication 101

8 Managing the Driver 103

Configuring Dynamic Attributes Using the Workday Driver 103
Synchronizing Custom IDs Using the Workday Driver 104
Configuring Derived Attributes for Worker 105
Configuring Custom Fields Using the Workday Driver 106
Adding New Attribute to the Filter 107
Migrating Individual Object from Workday: 107

9 Troubleshooting the Driver 109

Driver Cache File and Driver State Settings 109
Troubleshooting Driver Processes 110
Workday Driver Fails to Start After Changing the Cache Directory 110
Workday Driver Displays an Error Message While Shutting Down 110
Workday Driver Displays an Error Message When Workday Is Under Maintenance 111
Setting Paging Size Parameter Value 111
Error Message for Incorrectly Configuring OAuth 2.0 112
Error Displayed for Assigning Roles and User-Based Security Groups 112
Issue in Migrating Delta and Future Object for Business Process 114
FutureObjectCreation and CleanDeltaAndFutureObject Job Might Fail to Execute Successfully 114
Issue in Performing CPRS Computation for User-Based Security Group 114

Query-Ex fails to Fetch the Next Set of Pages Due to Expiry of the Token 114

10 Frequently Asked Questions 115

How to confirm successful establishment of JWT connection? 115
Why are Delta Objects and Full State Objects created with the suffix “<Termination Date + 1>” in the termination business process? 116
Why does same day termination also create Delta and Full state objects? 116
Why does the rescind transaction not update the Future Full State Object? 116
How are photos synchronized from Azure AD to Workday through the OpenText Identity Manager? . . . 117

11 Known Issues in Workday Driver 119

Modifying the Device Types Attributes are not Supported Independently 119
Unable to Restrict Entitlement Resource Assignments 119

A Driver Properties 121

Driver Configuration 121
- Driver Module 121
- Driver Object Password 122
- Authentication 122
- Startup Option 122
- Driver Parameters 123

Global Configuration Values 123

B Trace Level 125

C IDV Schema Extension 127

D IDV Proposed Structure 129

E Understanding the Driver State Storage 131

1 Understanding the Workday Driver

The Workday driver for Identity Manager can seamlessly provision users from the Workday cloud application, keeping the user identity information consistent across the Identity Vault (IDV). This driver is also capable of updating some of the connected system attributes such as User ID, contact information etc., back to the Workday application.

This document provides the detailed use cases that are required to be solved by the Workday driver to establish a seamless connectivity between Identity Manager and Workday.

IMPORTANT: The Workday driver shim is supported on Identity Manager 4.7 SP4 and above. The driver supports Workday version v32.0 and above.

This section provides the following information on the Workday driver:

- "Driver Concepts" on page 7
- "Support for Standard Driver Features" on page 9

Driver Concepts

This section provides the following information:

- "Data Management" on page 7
- "How the Driver Works" on page 8

Data Management

The Workday driver communicates with Workday using the Workday SOAP API; especially the Human_Resource API. Some of the important Human resource APIs used in the driver, are mentioned below:

- Get_Workers
- Get_Worker_Photos
- Get_Locations
- Get_Organizations
- Get_Job_Profiles
- Get_Job_Families
- Update_Workday_Account
- Get_Assign_User-Based_Security_Group
- Get_Assign_User-Based_Security_Groups
- Put_Assign_User-Based_Security_Groups
- Assign_Roles

- Change_Other_IDs
- Get_Server_Timestamp
- Put_Worker_Photo

SOAP

SOAP (Simple Object Access Protocol) is an XML-based protocol for exchanging messages. It defines the message exchange but not the message content. The driver supports SOAP 1.1.

SOAP documents are organized into three elements:

- **Envelope:** The root XML node.
- **Header:** Provides context knowledge such as a transaction ID and security information.
- **Body:** The method-specific information.

SOAP follows the HTTP request/response message model, which provides SOAP request parameters in an HTTP request and SOAP response parameters in an HTTP response.

XML

XML (Extensible Markup Language) is a generic subset of Standard Generalized Markup Language (SGML) that allows for exchange of structured data on the Internet.

HTTP

HTTP is a protocol used to request and transmit data over the Internet or other computer network. The protocol works well in an Internet infrastructure and with firewalls.

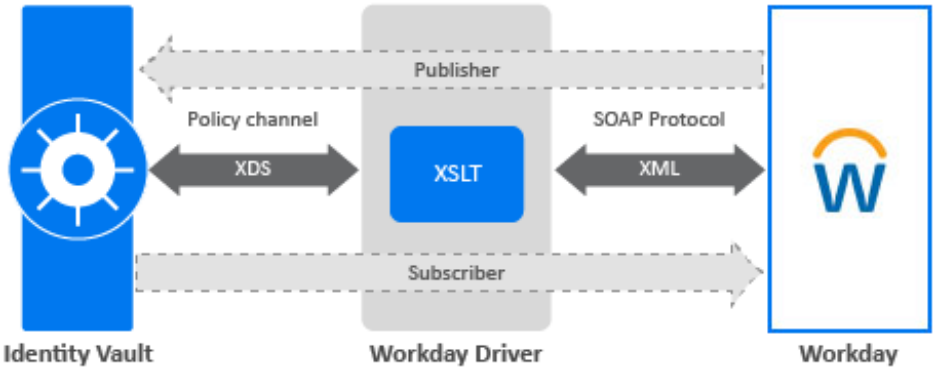
HTTP is a stateless request/response system because the connection is usually maintained only for the immediate request. The client establishes a TCP connection with the server and sends it a request command. The server then sends back its response.

NOTE: Workday communication always happens over HTTPS.

How the Driver Works

The following diagram illustrates the data flow from Workday service to Identity Manager and *vice-versa*:

Figure 1-1 *Workday Driver Data Flow*



The Workday driver polls data from Workday as SOAP request. Workday processes the request, and returns a SOAP response to the driver shim. The driver processes the response, converting it into appropriate XDS, a specialized form of XML, using XSLT style sheets. The driver stores the response data in a cache file. The Publisher channel periodically polls for additions and modifications to the objects in Workday. On a successful retrieval of the changes, it stores the polling time for use in the next polling cycle. Driver uses the cache file to submit the optimized XDS modify document to the Identity Manager Engine.

The Identity Manager engine uses XDS to represent events in the Identity Vault. On the Subscriber channel, the Identity Manager engine passes the XDS event to the driver policy channel, which consists of basic policies, DirXML Script, and XSLT style sheets. The driver shim receives the XDS document from the policy channel. The driver shim uses XSLT style sheets to convert the XDS document to appropriate SOAP request.

The Publisher channel uses two different ways of polling the Workday server for fetching the changes:

- **Recurring Polling:** Recurring polling happens at a fixed interval. This type of polling happens for Worker, Relation and Job Profile object classes. Recurring polling queries the Workday server for changes from the last successful recurring polling cycle.
- **Scheduled Polling:** Scheduled polling happens once in a day at a fixed time. This type of polling happens for Terminated Workers, Location, Worker Photo, Job Family and Organization object classes.

Support for Standard Driver Features

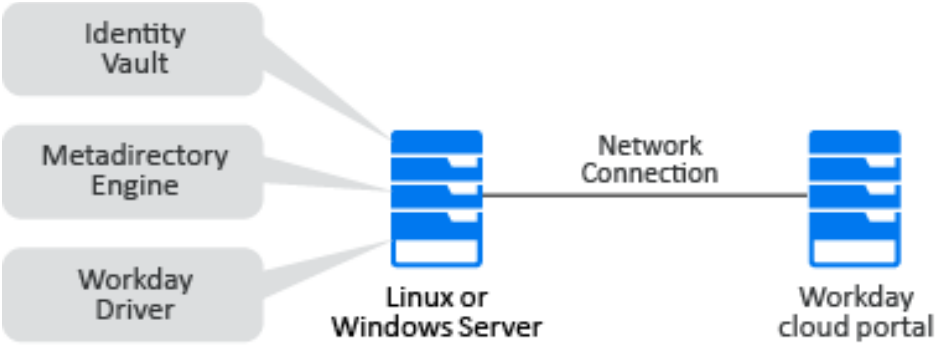
The following sections provide information about how the Workday driver supports these standard driver features:

Local Platforms

A local installation is an installation of the driver on the Metadirectory server. You can install the Workday driver on the operating systems supported for the Metadirectory server.

For information about the operating systems supported for the Metadirectory server, see Considerations for Installing Drivers with the OpenText Identity Manager Engine (https://www.netiq.com/documentation/identity-manager-4.10/idm_overview_planning/data/identity-

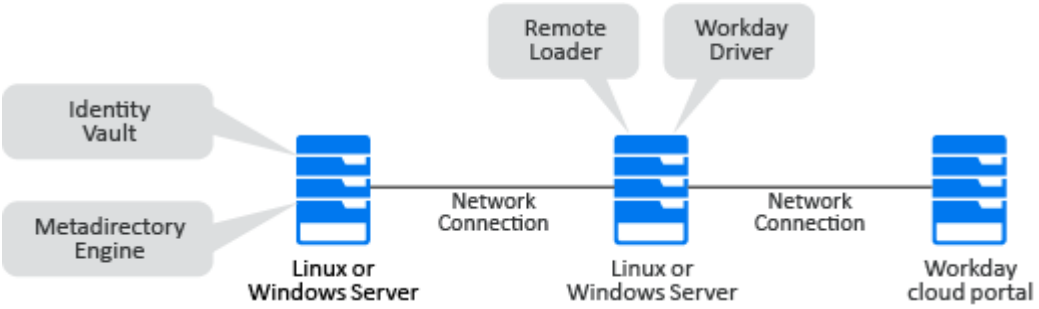
manager-server-guidelines.html#guidelines-for-installing-drivers-with-identity-manager-engine) in the OpenText™ Identity Manager CE 24.4 (v4.10) Overview and Planning Guide (https://www.netiq.com/documentation/identity-manager-4.10/idm_overview_planning/data/identity-manager-overview-and-planning.html).



Remote Platforms

The Workday driver can use the Remote Loader service to run on a server other than the Metadirectory server. You can install the Workday driver on the operating systems supported for the Remote Loader.

For information about the supported operating systems, see NetIQ Identity Manager Install and Upgrade Guide for Linux.



Supported Operations

The Workday driver supports the following operations on the Subscriber channel:

- Update user ID.

  When a user is renamed in IDV, the driver updates the user ID back in the Workday.
- Update user attributes such as email id, phone number, Custom IDs in OpenText Identity Vault.

  When a user is updated in IDV, the updated user information can be synchronized with Workday.

  NOTE: The delete operation for email ID and primary phone number (work and home) is not supported in the OpenText Identity Vault.

- Add and Modify operations of photo object in IDV.
- Managing permissions through user-based security groups and organizational roles.

NOTE: Workday driver does not support the following:

- Creating and deleting users in Workday
- Password synchronization

The following operations are supported on the Publisher channel:

- Users Object:
  - Add users (Employee and Contingent Worker Both)
  - Modify users

    When a user is updated in Workday, the updated user information is synchronized with the IDV.
  - Delete users

    Users cannot be deleted in Workday, hence there will be no delete event from workday.
  - Position or Relation object: Each job /position is treated as relation irrespective of primary position or non-primary position. The primary position information is also stored as part of the user object.
  - Location Object: Driver synchronizes locations and creates a separate object in IDV.
  - Job Family Object: Driver synchronizes job family and creates a separate object in IDV.
  - Job Profile Object: Driver synchronizes job profile and creates a separate object in IDV.
  - Organization Object: Driver synchronizes organizations and creates a separate object in IDV. Organization is configurable and can have different values such as Cost Center, Company, and HR Company etc.
  - Worker Photo Object: Driver synchronizes worker’s photo from Workday.

2 Installing and Configuring the Workday Driver

This section contains information about installing and configuring the Workday driver.

The installation and configuration process for the driver varies depending on whether you want to install the driver with the OpenText Identity Manager server or with the Remote Loader. After installing the driver, you must manually activate it. For more information, see "Activating the Driver" on page 43.

The following sections provide instructions to create the driver:

- "Workday Driver Packages" on page 13
- "Checklist for Installing and Configuring the Driver" on page 14
- "Creating Integration System Group(s) and User(s) in Workday Application" on page 15
- "Generating Client ID and Token Endpoint for OAuth 2.0 Authentication Method" on page 18
- "Installing the Driver Files" on page 19
- "Creating the Driver Object" on page 22
- "Configuring Phone Contact Information in Subscriber Channel" on page 38
- "Configuring Entitlements" on page 40
- "Configuring the Driver to Include Additional Transaction Type References" on page 41
- "Deploying the Driver Object" on page 42
- "Starting the Driver" on page 43
- "Activating the Driver" on page 43
- "Advanced Synchronization of Worker Photos from the Workday Portal" on page 43

Workday Driver Packages

The following packages are available for this driver:

Name	Package Name	Description
Workday Base Configuration	NETQWDBASE	This package contains the base configuration. It is a mandatory package.
Workday Default Configuration	NETQWDDCFG	These packages contain the default configuration information for the Workday workers (users). Always leave this package selected.
Workday Job Family Synchronization Configuration	NETQWDJBFSYN	This package is used for synchronizing job family.

Name	Package Name	Description
Workday Job Profile Synchronization Configuration	NETQWDJOBSYN	This package is used for synchronizing job profile.
Workday Location Synchronization Configuration	NETQWDLOCSYN	This package is used for synchronizing locations.
Workday Organization Synchronization Configuration	NETQWDORGSYN	This package is used for synchronizing Organization.
Workday Photo Synchronization Configuration	NETQWDPHTSYN	This package is used for synchronizing worker's photo.
Workday Position Relation Configuration	NETQWDPRLN	This package is used for synchronizing relation objects.
Workday Retry Events Configuration	NETQWDRETRY	This package is used for configuring retry events.
Workday Business Process Configuration	NETIQWDBP	This package is used for managing the worker's lifecycle within an organization.
Workday Entitlement Configuration	NETIQWDENT	This package is used for synchronizing entitlements.

Checklist for Installing and Configuring the Driver

Use the following checklist to ensure that you complete all of the tasks required to set up and use the Workday driver: For Organization Object including Cost Center, Company and HR Company etc

- Create Integration System Security Group(s) and Integration System User(s) in the Workday application. For more information, see "Creating Integration System Group(s) and User(s) in Workday Application" on page 15.
- Generate Access Token URL and Client ID in Workday application. For more information see, "Generating Client ID and Token Endpoint" on page 19
- Ensure that your environment meets the requirements for installing and configuring the Workday driver. For more information, see Chapter 1, "Understanding the Workday Driver," on page 7.
- Install the driver files on the OpenText Identity Manager engine or on a Remote Loader server. For more information, see "Installing the Driver Files" on page 19.
- Ensure that you have the appropriate packages installed and imported for the driver. For more information, see "Importing the Current Driver Packages" on page 22.
- Deploy the driver to the OpenText Identity Vault. For more information, see "Deploying the Driver Object" on page 42.
- Start the driver. For more information, see "Starting the Driver" on page 43.

Creating Integration System Group(s) and User(s) in Workday Application

Before you can install and configure the Workday driver, you must create Integration System Security Group(s) containing Integration System User(s) in the Workday application. Ensure that the Integration System User has all the required permissions before installing the driver. Perform the following steps to create Integration System Groups and Users:

- "Creating Integration System Security Group(s)" on page 15
- "Create Integration System" on page 17
- "Creating Integration System User(s)" on page 17

Creating Integration System Security Group(s)

1. Access the **Create Security Group** task and create an Integration System Security Group.
2. To grant the security group access to the domains required by your integration, perform the following steps for each domain:
   - Access the **View Domain** report and find the domain.
   - Select **Domain** > **Edit Security Policy Permissions**.
   - Add the security group that you created in Step 1 to the Integration Permissions and select **Get** and **Put**.
   - Access the **Activate Pending Security Policy Changes** task and activate the changes that you made in previous step.

Once the Integration System Security Group has been created, you must provide the following rights to the group:

- "Domain Rights for the Security Group:" on page 15
- "Domain Rights for Assignable Role-Based Security Group Entitlement" on page 16
- "Domain Rights for User-Based Security Group Entitlement" on page 17
- "Business Process Types Rights for the Security Group" on page 17

Domain Rights for the Security Group:

You can provide the following domain read rights to the security group based on your requirements:

Domain Rights	Description
Job Information	Provide this right if you want to synchronize the job information for the group users. Select only **Get** option for this right.
Manage: Location	Provide this right if you want to synchronize the location data for the group users. Select only **Get** option for this right.

Domain Rights	Description
Manage: Organization Integration	Provide this right if you want to synchronize the Organization data. Select only **Get** option for this right.
Person Data: Date of Birth	Provide this right if you want to synchronize the worker's date of birth. Select only **Get** option for this right.
Person Data: Gender	Provide this right if you want to synchronize the worker's gender information. Select only **Get** option for this right.
Person Data: ID Information	Provide this right if you want to synchronize the worker's other IDs. Select both **Get** and **Put** options for this right.
Person Data: Personal Photo	Provide this right if you want to synchronize the worker's photo from both channels. Select **Get**, **Put**, **View** and **Modify** options for this right.
Workday Accounts	This right should be provided to the security group by default. Select both **Get** and **Put** options for this right.
Worker Data: All Positions	Provide this right if you want to synchronize all position data help by the worker. Select both **Get** and **Put** options for this right. Select only **Get** option for this right.
Worker Data: Current Job Profile Information	Provide this right if you want to synchronize the worker's current job profile information. Select only **Get** option for this right.
Worker Data: Organization Information	Provide this right if you want to synchronize the worker's Organization information. Select only **Get** option for this right.
Worker Data: Public Worker Reports	Provide this right if you want to synchronize the public worker reports. Select only **Get** option for this right.

Domain Rights for Assignable Role-Based Security Group Entitlement

You can provide the following rights to the assignable role entitlement based on your requirements:

Domain Rights	Description
Manage: Organization Roles	Provide this right if you want to synchronize the Organization roles. Select **Put** option for this right.
Org Designs: Assign Roles	Provide this right if you want to assign roles for the organization. Select **Put** option for this right.

Domain Rights for User-Based Security Group Entitlement

You can provide the following domain rights to the user-based security group entitlement based on your requirements:

Domain Rights	Description
User-Based Security Group Administration	The administration rights provided to the user-based security group. Select both **Get** and **Put** options for this right.

Business Process Types Rights for the Security Group

You can provide the initiate access to the following business process types rights based on your requirements:

Business Process Types Rights	Description
Contact Change	Provide this right if you want to synchronize the contact change information of the worker.
Edit Other IDs	Provide this right if you want to synchronize the edit other IDs information of the worker.

Create Integration System

1. Access **Create Integration System**.
2. Input **System Name**.
3. Input (optional) **Comment** and **Integration Tags** under **System ID**.
4. Provide "Cloud Integration Template" as value to **New Using Template**.
5. Click **OK**.

Creating Integration System User(s)

1. Access the **Create Integration System User** task and configure a system user account for the integration. Set the **Session Timeout Minutes** to its default value of 0 to prevent session expiration. An expired session can cause the integration to time out before it successfully completes.
2. Select **Security Profile** > **Assign Integration System Security Groups**.
3. In the **Integration System Security Group to Assign** screen, select the security group that you created in "Creating Integration System Security Group(s)" on page 15.
4. Create the Integration System following the steps listed in "Create Integration System" on page 17.
5. Access **View Integration System** Report and select the **Integration System** that has been created.
6. Select **Workday Account** > **Edit Account for Integration System** and select the Workday account that you created in Step 1 on page 17.

7 (Optional) Under the **Global Preferences** option, select a preferred locale and display language for the integration system user.

8 Access the **Maintain Password Rules** task and add the integration system user to the **System Users exempt from password expiration** field.

Generating Client ID and Token Endpoint for OAuth 2.0 Authentication Method

Before creating the driver object, you must generate the workday client ID and the end point token URL. It is also recommended to create a public key certificate prior to creating the client ID and token endpoint. The following sections explain the procedure to create a Public Key Certificate, and the Client ID and Token Endpoint.

Creating x509 Public Key Certificate

The following steps explain the procedure to create a public key certificate:

1 Run the following `keytool` command on your platform:

```
keytool -genkey -keyalg RSA -alias Workday -keystore JWTkeystore.ks -
storetype PKCS12 -storepass <Workday123!> -validity 360 -keysize 2048
```

NOTE: ◆The `keytool` command is common for both Linux and Windows platforms.

- You must specify the `storetype` as `PKCS12` as shown in the above `keytool` command.

2 Enter the following details when prompted:

- First Name
- Last Name
- Organizational Unit
- Organization
- City
- Province
- Country Code

3 Enter `Yes` to confirm. The keystore file is created, for example: `JWTkeystore.pkcs12`.

4 Extract the public key and the certificate by entering the following command:

```
keytool -export -alias Workday -keystore JWTkeystore.ks -rfc -file
publickey.cert
```

5 You will be prompted to enter the password. Enter the password to create the certificate, for example `publickey.cert`.

NOTE: The password does not display on the screen when entered. Ensure to remember the entered password or save it in a convenient location.

6 Run the following command, to display the certificate.

```
cat publickey.cert
```

7 Copy the certificate from the `BEGIN CERTIFICATE` line till `END CERTIFICATE` line, and save it in a convenient location.

Generating Client ID and Token Endpoint

Perform the following steps to generate the Client ID and Token Endpoint:

1 Login to Workday application.

2 Search and click on **Register API Client**.

   2a In the **Client Name** field, enter the name of the client.

   2b In the **Client Grant Type** field, select **Jwt Bearer Grant**.

   2c In the **x509 Certificate** field, select **Create x509 Public Key**.

   2d In the Create x509 Public Key page, enter your name and paste the certificate generated as shown in "Creating x509 Public Key Certificate" on page 18.

   2e Select **Access Token Type** as **Bearer**.

   2f Enter the **Redirection URL**. This could be your organization's home page URL.

   2g Select the following **Scope** values from Workday REST API:

   - Contact Information
   - Integration
   - Jobs & Positions
   - Organizations & Roles
   - Personal Data
   - System
   - Tenant Non-configurable
   - Include Workday Owned Scope

3 Click **OK** to generate the **Client ID** and the **Token Endpoint**.

4 Save the **Client ID** and the **Token Endpoint** in a convenient location, as these values will be required to configure OAuth 2.0 authentication method for the Workday driver.

Installing the Driver Files

The Workday driver is a Java-based driver and can be run on the OpenText Identity Manager engine or on a Remote Loader server. It works only with native and Java Remote Loader.

- "Installing the Driver Files as a Root User" on page 20
- "Installing the Driver Files as a Non-Root User" on page 20
- "Configuring Remote Loader for the Driver Instance" on page 22

Installing the Driver Files as a Root User

You must install the driver files manually. The Workday driver files are installed on the OpenText Identity Manager server at the same time as the OpenText Identity Manger engine. The installation program extends the OpenText Identity Vault's schema and installs the driver shim and the driver configuration file, but does not create the driver in the OpenText Identity Vault or upgrade an existing driver's configuration. For information about creating the driver, see "Creating the Driver Object" on page 22.

If you want to run the driver with Remote loader, verify that the `WorkdayShim.jar` file is present in the following directories for the Remote Loader:

- **Linux:** `/opt/novell/eDirectory/lib/dirxml/classes`
- **Windows:** `C:\novell\remoteloader\lib`

Else, copy the file from the OpenText Identity Vault server to the appropriate directory based on the operating system.

- **Linux**: Install the new `novell-DXMLWday.rpm` in your driver installation directory by running the following command in a terminal window:
  - If you are installing the binary, run the command: **`rpm -Ivh (binaries-path)/novell-DXMLWday.rpm`**
- **Windows**: Copy `WorkdayShim.jar` in the `{IDM-Install-Folder\nds\lib}` directory of the OpenText Identity Manager Engine. If you want to run the driver with Remote loader on Windows, verify that the `WorkdayShim.jar` file is present in the `{remoteloader\lib}` directory.

To install the dependent `jar` files, perform the following actions on your platform as applicable:

- **Linux**:
  - Copy the jar file `WorkdayUtils.jar`, from *<IDM_WorkDayDriver_1300>*/common folder and place it in `/opt/novell/eDirectory/lib/dirxml/classes/` directory in the OpenText Identity Manager engine.
  - Copy the `WorkdayUtils.jar`, `imgscalr-lib-4.2.jar`, `jcce-1.3.0.jar`, and `jcce-oauth-1.3.0.jar` from *<IDM_WorkDayDriver_1300>*/common folder and place it in the driver installation folder. For example, `/opt/novell/eDirectory/lib/dirxml/classes/` (for local or remote installation).
- **Windows**:
  - Copy the jar files WorkdayUtils.jar from *<IDM_WorkDayDriver_1300>*/common folder and place it in OpenText Identity Manager engine path \NetIQ\IdentityManager\NDS\lib.
  - Copy the imgscalr-lib-4.2.jar, jcce-1.3.0.jar and jcce-oauth-1.3.0.jar from *<IDM_WorkDayDriver_1300>*/common folder and place it in the driver installation folder. For example, `\NetIQ\IdentityManager\NDS` (local installation) or `\Novell\RemoteLoader\64bit` (remote installation).

Installing the Driver Files as a Non-Root User

1. Verify that the **`/rpm`** directory exists and contains the **`_db.*`** file.

2. The **_db.*** file is created during a non-root installation of the OpenText Identity Manager engine. The absence of this file indicates that the OpenText Identity Manager is not installed properly. In such a case, reinstall the OpenText Identity Manager to correctly place the file in the mentioned directory.
3. To set the root directory to the location of non-root in OpenText Identity Manager, enter the following command in the command prompt:

   ```
   ROOTDIR=<non-root eDirectory location>
   ```

   This will set the environmental variables to the directory where OpenText Identity Manager is installed as a non-root user.
4. For example, to install the Workday driver rpm, use this command:

   ```
   rpm --dbpath $ROOTDIR/rpm -Ivh --relocate=/usr=$ROOTDIR/opt/novell/
   eDirectory --relocate=/etc=$ROOTDIR/etc --relocate=/opt/novell/
   eDirectory=$ROOTDIR/opt/novell/eDirectory --relocate=/opt/novell/
   dirxml=$ROOTDIR/opt/novell/dirxml --relocate=/var=$ROOTDIR/var --
   badreloc --nodeps --replacefiles /home/user/novell-DXMLWday.rpm
   ```

   NOTE: In the above command **/opt/novell/eDirectory** is the location where non-root OpenText Identity Manager is installed, and **/home/user/** is the home directory of the non-root user.
5. To install the dependent `jar` files, perform the following actions based on your platform:
   - **Linux**:
     - Copy the jar file `WorkdayUtils.jar`, from <*IDM_WorkDayDriver_1300*>/`common` folder and place it in `/opt/novell/eDirectory/lib/dirxml/classes/` directory in the OpenText Identity Manager engine.
     - Copy the `WorkdayUtils.jar`, `imgscalr-lib-4.2.jar`, `jcce-1.3.0.jar`, and `jcce-oauth-1.3.0.jar` from <*IDM_WorkDayDriver_1300*>/`common` folder and place it in the driver installation folder. For example, `/opt/novell/eDirectory/lib/dirxml/classes/` (for local or remote installation).
   - **Windows**:
     - Copy the jar files `WorkdayUtils.jar` from from <*IDM_WorkDayDriver_1300*>/`common` folder and place it in OpenText Identity Manager engine path `\NetIQ\IdentityManager\NDS\lib`.
     - Copy the `imgscalr-lib-4.2.jar`, `jcce-1.3.0.jar` and `jcce-oauth-1.3.0.jar` from <*IDM_WorkDayDriver_1300*>/`common` folder and place it in the driver installation folder. For example, `\NetIQ\IdentityManager\NDS` (local installation) or `\Novell\RemoteLoader\64bit` (remote installation).

You must configure the driver to use an SSL connection between the Remote Loader and the OpenText Identity Manager server. For more information, see Creating a Secure Connection to the OpenText Identity Manager Engine (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b18xta1v.html#t461ov2hxlp7) in the OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html).

For more information on configuring the Remote Loader, see "Configuring Remote Loader for the Driver Instance" on page 22.

Configuring Remote Loader for the Driver Instance

For more information about specifying values for these parameters, see Configuring the Remote Loader and Drivers (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b18xta1v.html) in the OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html).

Below is an example of a configuration file with some sample values.

```
-description "Workday Driver"
-commandport 8000
-connection "port=8090"
-trace 3
-tracefile "/opt/netiq/workdaydriver.log" (or
"C:\novell\remoteloader\64bit\WorkdayDriver-Trace.log" on Windows)
-tracefilemax 100M
-class "com.netiq.dirxml.driver.workday.WDDriverShim"
```

Once the driver instance is running, you can use the command line on Linux or Remote Loader Console on Windows to instruct the Remote Loader to perform a function. For example, turn the trace on or off on Windows or stop the driver instance.

Creating the Driver Object

To create a Workday driver object you must install the driver packages and then modify the configuration to suit your environment. After you create and configure the driver object, you need to deploy it to the OpenText Identity Vault and start the driver.

- "Importing the Current Driver Packages" on page 22
- "Installing and Configuring the Driver Object" on page 23

Importing the Current Driver Packages

The driver packages contain the items required to create a driver, such as policies, entitlements, filters, and Schema Mapping policies. These packages are only available in Designer and can be updated after they are initially installed.

To verify that you have the most recent version of the driver packages in the Package Catalog:

1. Open Designer.
2. In the Outline view, right-click the Package Catalog.
3. Click **Import Package**.
4. Browse to the location where you downloaded the Workday packages.
5. Select the required packages for Workday. Alternately, you can click **Select All** to import all of the packages displayed. For the available Workday packages, see "Workday Driver Packages" on page 13.

6 Click **OK** to import the selected packages, then click **OK** in the successfully imported packages message.

7 After the current packages are imported, continue with "Installing and Configuring the Driver Object" on page 23.

Installing and Configuring the Driver Object

1 In **Designer**, open your project.

2 From the **Palette**, drag-and-drop the Workday driver to the desired driver set in the **Modeler**.

   The Workday driver is under the **Enterprise** category in the Palette.

3 Select the required packages that you want to install for the Workday driver, then click **Next**. For more information, see "Workday Driver Packages" on page 13.

4 Click **OK** on the **Package Dependencies** screen.

5 Fill the **Driver Name** field, then click **Next**.

6 In the **Driver Parameters** page, the following sections and fields are displayed. Enter the values in the corresponding fields as shown below and click **Next**.

   - **Driver Parameters**: The **Driver Options** field defaults to **show** and the following sections are displayed:
     - **Authentication Parameters**: Select the authentication type required as shown in the following Authentication Parameters table. The available options are:
       - **Basic**: This option requires the user ID and password to authenticate and establish connection with Workday.
       - **OAuth 2.0**: This an open authentication protocol. This authentication method enables a third-party application to access and share data from a HTTP service across connected applications (for example, Workday).

Table 2-1 *Authentication Parameters*

Field/Description	Sample Field Value
If you select **Basic**, the following fields appear: - **Workday Login ID**: Specify the user ID that is used to login to Workday. Ensure that the user has required permission to fetch and modify data from Workday. Perform the following steps to find the tenant name: 1. Log in to the Workday sandbox. 2. On the first page, find the tenant details appearing in the upper left corner of the page. - **Workday Login Password**: Specify the password that is used to login to Workday.	- **Workday Login ID**: `<IntegrationUsername@tenantName>` - **Workday Login Password**: `<workday password>`

Field/Description	Sample Field Value
If you select **OAuth2.0**, the following fields appear: ◆ **Access token URL**: Specify the URL of the server used for requesting token access. For more information to generate the URL, see "Generating Client ID and Token Endpoint" on page 19. ◆ **Client ID**: Specify the Workday Client ID for OAuth 2.0 authentication. For more information to generate the Client ID, see "Generating Client ID and Token Endpoint" on page 19. ◆ **User Name**: Specify the Workday Integration User Name for OAuth 2.0. ◆ **Keystore File**: Specify the path of the keystore file. For more information, see "Creating x509 Public Key Certificate" on page 18. ◆ **Alias**: Specify the Alias name for secret (private) key in the keystore file. This is the alias name that was provided when creating the keystore certificate. For more information, see "Creating x509 Public Key Certificate" on page 18. ◆ **Keystore Password**: Enter the password of the Keystore. This is the password that was provided when creating the keystore certificate. For more information, see "Creating x509 Public Key Certificate" on page 18. **NOTE:** Ensure the file password and the private key passwords are same when creating the keystore.	◆ **Access token URL**: *<https://wd2-impl-services1.workday.com/ccx/oauth2/tenantname/token>* ◆ **Client ID**: *<AjkwZTU5GUtOWIopb00MWrnmkkzODYtZjU0MjNkNDIfgykl>* ◆ **User Name**: *<name of the user>* ◆ **Keystore File**: *<C:\keystore>* ◆ **Alias**: *<keystore alias name>* ◆ **Keystore Password**: *<password for the keystore file>*

◆ **Connection Parameters**: Specify the Connection Parameters as explained in the following table:

Table 2-2 Connection Parameters

Field/Description	Sample Field Value
Workday Login URL: Specify the login URL of Workday. Perform the following steps to find the Host Name: 1. Log in to the Workday sandbox or production instance. 2. Search for **Public Web Services** in the search box. 3. Select **Public Web Services** from the search results. 4. Scroll down to **Human Resources (Public)** and click on the ellipsis (3 dots). Then click the **Web Services** list and select **View WSDL**. 5. Scroll down to the end to find the `soapbind:address` location value assigned to the location element in the WSDL file. 6. Copy and save the Workday `soapbind:address` location part of the value. For example, `location="https://wd5-impl-services9.workday.com/....."`	*<https://hostName/ccx/service/tenantName/Human_Resources/v32.0>*
Workday API Version	v32.0 onwards **NOTE:** This value should not be changed.
Proxy Authentication: Proxy authentication enables you enhance the security of your environment by placing a middle ware layer between the Workday driver and the OpenText Identity Vault.	If you decide to use a proxy authentication, then specify the following fields: ◆ **Proxy hostname and port**: Specify the host address and the host port. Choose an unused port number on your server. ◆ User name: Enter the proxy user ID. ◆ **User Password**: Set the proxy user password.

- **General Parameters**: Specify the General Parameters as shown in the following table:

Table 2-3 General Parameters

Field/Description	Sample Field Value
Cache Files Directory: Specify the absolute path of the directory where the driver has to create cache files which is required for its operations. As cache files can be large in size, it is recommended to provide the directory with no size limit.	`/var/opt/novell/eDirectory/data/dib` or `C:/NetlQ/IDM/NDS/DIBFiles/`

- **SSL Parameters**: Specify the SSL Parameters values as explained in the following table:

Table 2-4 SSL Parameters

Field/Description	Sample Field Value
Workday Keystore File: Specify the path and the name of the keystore file that contains the trusted certificates for the remote server to provide server authentication.	For example: ◆ In Windows: `<C:\security\keystore>` ◆ In Linux: `</var/opt/novell/eDirectory/workday/idmworkday.jks>` To obtain a Workday tenant root CA certificate and import it to the Java keystore. see Chapter 7, "Securing Communication," on page 101.
Workday Keystore Password	Specify the password for the keystore file.

- **Advanced Parameters**: Select **Show** to configure the Advanced Parameters.
  - **Paging Parameters**: Specify the Paging Parameters as explained in the following table:

Table 2-5 Paging Parameters

Field/Description	Sample Field Value
Page size: Specify the number of pages to be processed at a time by the driver. This value is used for both the recurring and scheduled polling cycle and should be set as per the Java heap available to the driver.	Specify the page size in this field. By default, the paging size is set to 50. For more information on setting paging size value see, "Setting Paging Size Parameter Value" on page 111.
Query-Ex timeout in seconds: Specify the timeout in seconds for Query Ex.	*<120>*

  - **Transformation Parameters**: Specify the Transformation Parameters as shown in the following table:

Table 2-6 Transformation Parameters

Field/Description	Sample Field Values
Transformation Parameters: Set the following parameters on the Publisher channel and select the required values for each of them from the list of default values or you can create custom values by clicking on the ✚ icon. The default Parameter Names configured are: ◆ Parameter Name: **`wdOrgList`** Custom values for this parameter can also be added by clicking on the ✚ icon. You can collect such Organization values from the Workday implementation team. ◆ Parameter Name: **`wdIntFieldAttrList`** ◆ Parameter Name: **`wdAutoCompleteBP`** ◆ Parameter Name: **`wdCustomIdList`** ◆ Parameter Name: **`wdFetchPrivatePhoneInformation`** ◆ Parameter Name: **`wdTransactionLimit`**	The descriptions for the default **Parameter Name** and their associated **Parameter Value** are shown below: ◆ **`wdOrgList`**: The values here indicate the types of Organizational data that will be returned from a call of the `Get_Worker` SOAP API. **Parameter Values**: ◆ `Supervisory` ◆ `Cost_Center` ◆ `Company` ◆ `Cost_Center_Hierarchy` ◆ `Matrix` ◆ `Region` ◆ **`wdIntFieldAttrList`**: Add the names of any custom, override fields created in Workday for Workers and that should be synchronized to the OpenText Identity Vault. This option can be used to synchronize data to the OpenText Identity Vault that is not currently synchronized by the shim. ◆ **`wdAutoCompleteBP`**: Set this to **true** to mark the business process for the modification as complete and immediately commit the update to the Worker. ◆ **`wdCustomIdList`**: Add the names of any Other/Custom IDs that the driver should synchronize to the OpenText Identity Vault and/or which should have data from the OpenText Identity Vault written back to Workday using the `Change_Other_IDs` API. ◆ **`wdFetchPrivatePhoneInformation`**: When set to **true**, the driver will retrieve phone information if that number has been marked `private` in Workday. By default this is set to **false** which means meaning only the phone numbers marked `public` will be published. ◆ **`wdTransactionLimit`**: This parameter indicates maximum number of transactions that the `wd-TransactionLog` attribute can hold. The following values are available to select: ◆ 3 ◆ 5 ◆ 10 ◆ 15 ◆ 30 **NOTE:** The `wd-TransactionLog` attribute is a multi-valued attribute, and is capable of storing multiple transactions.

7 Fill in the Subscriber Options as shown below and click **Next**.

IMPORTANT: ◆To place the imported custom stylesheet in the required folder, you must have the required modify folder permissions.

- When you are importing the custom stylesheets and placing it in the required folder, you must ensure that the folder modify rights must be available only for the system administrator or the user running the OpenText Identity Vault process.

- **Subscriber Options**
  - **Worker Settings**
    - **Import Worker Stylesheet**: Specify the absolute path to import the custom stylesheet for a worker. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be considered.

      For example:

      - **Linux**: `/home/NOVLSPML2-its-XDSInputTransform_worker.xml`
      - **Windows**: `C:\NOVLSPML2-its-XDSInputTransform_worker.xml`
  - **Worker Photo Settings**
    - **Import Worker Photo Stylesheet**: Specify the absolute path to import the custom stylesheet for worker's photo. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be considered.

      For example:

      - **Linux**: `/home/NOVLSPML2-its-XDSInputTransform_workerphoto.xml`
      - **Windows**: `C:\ NOVLSPML2-its-XDSInputTransform_workerphoto.xml`

8 Fill in the Publisher Options as shown below and click **Next**.

IMPORTANT: When you are importing the custom stylesheets, you must ensure that it can be modified only by the system administrator or the user running the OpenText Identity Vault process.

- **Publisher Options**:
  - **Enable Publisher Channel**: **Yes**
  - **Heartbeat Interval**: This option is used to configure the driver shim to send a periodic status message on the Publisher channel, when there is no Publisher traffic for a specified number of minutes. By default, this is set to 10 minutes.
  - **Recurring API**: Specify the values in the fields as shown below:
    - **Polling interval in minutes**: Number of minutes between the polling cycles. This is applicable for both worker and job profile. By default, the polling interval is set to 10 minutes.
    - **Migrate on Startup**: These parameters are valid only at driver startup. Driver will continue with regular polling after the task at startup is completed. If driver shuts down unexpectedly during migration, just restart the driver and migration will continue from where it stopped.

If you select the value as:

- True: Migration is initiated for the Recurring APIs (i.e Worker and Job Profile) that are enabled in the configuration, on driver startup. After migration is completed, the driver continues with regular polling cycle.
- False: Regular polling for Recurring APIs that are enabled in the configuration is initiated. This parameter can force the migration by clearing cache files.

IMPORTANT: You must ensure not restart the driver with this parameter set to true for a completed migration, unless re-migration is required.

- **Enable Delta and Future Object Creation on initial Migration**: Set the vale as true to enable the Delta and Future Object Creation upon the initial migration of objects from Workday.

IMPORTANT: To enable this feature, it is mandatory to set the **Migrate on Startup** value as True.

  - **Number of days of Prior Transactions to be considered for Future Object creation**: The number of past days for which the delta object should be created as a part of migration.
- **Effective Poll Time**: Set the time interval for the effective polling. This polling is applicable for `Worker` and `Job Profile`. The effective polling can be performed once in a day at a selected time slot.

- **Worker**:
  - **Poll Worker**: Select Enable to poll worker objects from Workday.
    - **Worker Poll Settings**: Select Show to view worker poll settings and specify the values as shown in the following table:
    - **Response Group Element List**: Specify the elements to be included as part of the response received from Workday. `Worker_Response_Group` section in Workday `Get_Workers` API displays the list of all the elements that can be included. This list has the following default values:
      - `Include_Reference`
      - `Include_Additional_Jobs`
      - `Include_User_Account`
      - `Include_Personal_Information`
      - `Include_Employment_Information`
      - `Include_Organizations`
      - `Exclude_Company_Hierarchies`
      - `Exclude_Matrix_Organizations`
      - `Include_Transaction_Log_Data`
      - `Include_Management_Chain_Data`
    - **Field and Parameter Criteria Provider Reference List**: The integration system IDs will be placed in the `Field_And_Parameter_Criteria_Data` section of the GET SOAP API request.

- **Transaction Log Criteria Reference List:** Enter the transaction type references to be checked for changes during normal polling. For more information on Additional transaction type references, see "Configuring the Driver to Include Additional Transaction Type References" on page 41. These are primarily `Business_Process_Type` transaction type references. To see changes for additional business processes, add the `Business_Process_Type` to this list. A few of the `Business_Process_Type` references configured by default are shown below:
  - `Hire Employee`
  - `Add Additional Job`
  - `Add Retiree Status`
  - `Assign Matrix Organization`
  - `Assign Workers`
  - `Change Job`
  - `Change Business Title`
  - `Change Legal Name`
  - `Change Organization Assignments for Worker`
  - `Change Organization Assignments for Workers by Organization`
  - `Change Preferred Name`
  - `Change Personal Information`
  - `Contact Information Event`
  - `Demote Employee`
  - `Promote Employee`
  - `Edit ID Information`
  - `Change Personal Information`
  - `Edit Other IDs for Worker`
  - `Contact Information Event`
  - `Terminate Employee`
  - `Contract Contingent Worker`
  - `End Contingent Worker Contract`
  - `Transfer Employee`
  - `Transfer Contingent Worker`
  - `Request Leave of Absence`
  - `Request Return from Leave of Absence`
  - `Change Photo`
  - `Home Contact Information Event`
  - `Work Contact Information Event`

- **Import Worker Stylesheet**: Specify the absolute path to import custom stylesheet for worker. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be merged with the default one.

  For example:

  - **Linux**: `/home/NOVLSPML2-its-XDSInputTransform_worker.xml`
  - **Windows**: `C:\NOVLSPML2-its-XDSInputTransform_worker.xml`

- **Job Profile**:
  - **Poll JobProfile**: Select **Enable** to poll JobProfile objects from Workday.
    - **Job Profile Poll Settings**: Select **Show** to view JobProfile poll settings.
      - **Response Group Elements List:** Specify the elements to be included as part of the response received from Workday. The `Job_Profile_Response_Group` section in `Get_Job_Profiles` API displays the list of elements that can be included.

        This list has the following default values:

        - Include_Reference
        - Include_Job_Profile_Compensation_Data
        - Include_Job_Profile_Basic_Data
      - **Field and Parameter Criteria Provider Reference List**: Specify the integration system IDs for the override fields created in Workday. The integration system IDs will appear in the `Field_And_Parameter_Criteria_Data` section of the GET request criteria.
      - **Import Job Profile Stylesheet**: Specify the absolute path to import custom stylesheet for job profile. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be merged with the default one.

        For example:

        - **Linux**: `/home/NOVLSPML2-its-XDSInputTransform_jobprofile.xml`
        - **Windows**: `C:\NOVLSPML2-its-XDSInputTransform_jobprofile.xml`

- **Scheduled API**:
  - **Scheduler**: Specify the time of the day to poll objects from Workday. This is applicable to objects such as, Terminated Workers, Location, Organization job Family, and Worker Photo.

- **Terminated Workers**:
  - **Poll Terminated Workers**: Select **Enable** to poll JobProfile objects from Workday.
    - **Terminated Workers Poll Settings**: Select **Show** to view JobProfile poll settings.

- **Termination Business types to poll**: Specify the list of worker business termination types to poll. For example, Terminate Employee, End Contingent Worker Contract.
- **Number of days to fetch**: Specify the number of days to be queried for terminated workers.

- **Location**:
  - **Poll Location**: Select **Enable** to poll Location objects from Workday.
    - **Location Poll Settings**: Select **Show** to view Location poll settings.
    - **Location Types to Poll**: Specify the list of location types to poll. The `Location_Type_Reference` section in Workday `Get_Locations` API displays the list of elements that can be included.
    - **Location Usage Types to Poll**: Specify the list of location usage types to poll. The `Location_Usage_Reference` section in Workday `Get_Locations` API displays the list of elements that can be included.
    - **Response Group Elements List**: Specify the elements to be included as part of the response received from Workday. The `Location_Response_Group` section in Workday `Get_Locations` API displays the list of elements that can be included.
    - **Field and Parameter Criteria Provider Reference List**: Specify the integration system IDs for the override fields created in Workday. The integration system IDs will appear in the `Field_And_Parameter_Criteria_Data` section of the GET request criteria.
    - **Import Location Stylesheet**: Specify the absolute path to import custom stylesheet for location. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be merged with the default one.

      For example:
      - **Linux**: `/home/NOVLSPML2-its-XDSInputTransformJocation.xml`
      - **Windows**: `C:\ NOVLSPML2-its-XDSInputTransform_location.xml`

- **Worker Photo**:
  - **Poll Worker Photo**: Select **Enable** to poll worker's photo from Workday.
    - **Worker Photo Poll Settings**: Select **Show** to view worker's photo poll settings.
    - **Resize Worker Photo**: Select **Yes** to resize the worker's photo.
      - **Maximum Photo Height**: It is recommended to specify the maximum photo height as 200.
      - **Maximum Photo Width**: It is recommended to specify the maximum photo width as 140.

- **Worker Photo Attribute Settings**:
  - **Compress Worker's Photo**: Resize worker's photo to the specified height and width and stores it in OpenText Identity Vault's `photo` attribute. The OpenText Identity Vault's photo attribute will store only those photos which are less than 64KB in size.
  - **Compress Worker's Photo and Retain the Original Photo File**: Resize the worker's photo to the specified dimensions and stores it in OpenText Identity Vault's `photo` attribute. Also, stores a copy of the original photo in the `wd-originalPhoto` photo stream attribute.
- **Photo Update Report URL:** Specify the URL for Workday Photo Update Report.
- **Photo Update Calculated Field**: Specify the Calculated Field created on Workday. This field records the DateTime value for last photo update event for the workers.
- **Photo Report Namespace**: Specify the namespace of the custom report under Web Services Options.
- **Import Worker Photo Stylesheet**: Specify the absolute path to import the custom stylesheet for Worker Photo. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be merged with the default one.

  For example:
  - **Linux**: `/home/NOVLSPML2-its-XDSInputTransform_workerphoto.xml`
  - **Windows**: `C:\ NOVLSPML2-its-XDSInputTransform workerphoto.xml`

- **Job Family**:
  - **Poll JobFamily**: Select **Enable** to poll job family objects from Workday.
    - **Job Family Poll Settings**: Select **Show** to view Job Family poll settings.
    - **Response Group Elements List**: Specify the elements to be included as part of the response received from workday. **Job_Family_Response_Group** section in Workday **Get_Job_Families** API displays the list of elements that can be included.
    - **Field and Parameter Criteria Provider Reference List**: Specify the integration system IDs for the override fields created in Workday. The integration system IDs appear in the `Field_And_Parameter_Criteria_Data` section of the GET request criteria.
    - **Import Job Family Stylesheet**: Specify the absolute path to import custom stylesheet for job family. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be merged with the default one.

For example:

- **Linux**: `/home/NOVLSPML2-its-XDSInputTransformjobfamily.xml`
- **Windows**: `C:\NOVLSPML2-its-XDSInputTransformjobfamily.xml`

- **Organization**:
  - **Poll Organization**: Select **Enable** to poll organization objects from Workday.
    - **Organization Poll Settings**: Select **Show** to view organization poll settings.
    - **Response Group Elements List:** Specify the elements to be included as part of the response received from workday. **Organization_Response_Group** section in Workday **Get_Organizations** API displays the list of elements that can be included.
    - **Field and Parameter Criteria Provider Reference List**: Specify the integration system IDs for the override fields created in Workday. The integration system IDs appear in the `Field_And_Parameter_Criteria_Data` section of the GET request criteria.
    - **Import Organization Stylesheet**: Specify the absolute path to import custom stylesheet for organization. Leave this field empty to operate with default stylesheet. If stylesheet is configured, the data in the custom stylesheet will be merged with the default one.

      For example:

      - **Linux**: `/home/NOVLSPML2-its-XDSInputTransform_organization.xml`
      - **Windows**: `C:\NOVLSPML2-its-XDSInputTransform_organization.xml`

- **Advanced Entitlement Settings**:
  - **Advanced Entitlement Settings**: Select the value as **Show** to configure the entitlement settings.
    - **Staffing Web Service URL**: Enter the URL of the staffing web service specific to your tenant. The procedure to obtain the staffing web service URL is similar to obtaining the Workday login URL (as shown in the Connection Parameters table), except that you must perform a search for **Staffing (Public)** instead of **HR Services (Public)**.
    - **Organization Types ID Values List**: The list of values of **Organization_Type_ID**s for the organizations that you want to collect the role entitlements for. If you specify any values here, only then the filter is applied. Else, by default IDV considers all the Organization Type ID values defined in Workday.

      For example, some the standard Organization Type ID values are:

      - `Supervisory`
      - `Cost_Center`
      - `Company`
      - `Cost_Center_Hierarchy`

- `Matrix`
- `Region`, etc.

NOTE: OpenText Identity Console does not support full migration of objects from Workday Identity Vault if you select the **Migrate into IDVault** option. To migrate all objects, you must set the values in **Migration on Startup** or **Scheduled Polling** fields accordingly in the **Driver Parameters** in Designer.

---

9 Fill in the following fields for the Remote Loader information, then click **Next**:

Connect To Remote Loader: Select **Yes** or **No** to determine if the driver will use the Remote Loader. For more information, see Configuring the Remote Loader and Drivers (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b18xta1v.html) in the *NetIQ Identity Manager Install and Upgrade Guide for Linux* in the OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html).

If you select **No**, skip to Step 8. If you select **Yes**, use the following information to complete the configuration of the Remote Loader:

Host Name: Specify the IP address or DNS name of the server where the Remote Loader is installed and running.

Port: Specify the port number for this driver. Each driver connects to the Remote Loader on a separate port. The default value is 8090.

KMO: Specify the Key Name (for example, `kmo=remotecert`) of the Key Material Object (KMO) containing the keys and certificate to be used for SSL.

If you used spaces in the certificate name, you need to enclose the KMO object nickname in single quotation marks.

Remote Loader Password: Specify a password to control access to the Remote Loader. It must be the same password that is specified as the Remote Loader password on the Remote Loader.

Driver Password: Specify a password for the driver to authenticate to the OpenText Identity Manager server. It must be the same password that is specified as the Driver Object Password on the Remote Loader.

10 On the Install Workday Default Configuration page, fill in the following fields for **User - Publisher** settings and **Advanced Settings**, then click **Next**:

Merge Employee and Contingent workers: Select **true** to merge Employee and Contingent workers in OpenText Identity vault. Selecting the value as **true** will maintain a single user object in OpenText Identity vault for converting an employee to contingent worker, or vice-versa.

User Placement Container: This global configuration value is used to get a reference to the container IDV where new users will be created. It may be different than user search container. The DN is expected in slash-format such as `data\workday\users`. If you want to configure a custom container, specify custom as the value and define a custom placement policy.

User Search Container in: This global configuration value is used to get a reference to the container where Workday users are stored. This value is used for all workers (both Employee and Contingent workers). If you want separate containers for different type of users, then use custom option in `matching/creation/placementpolicy`. The DN provided here should in slash-format, such as `data\workday\users`.

Default Password Policy: Specify the path for the default password policy to be used while creating random password for new Users, for example `Security\Password Policies\Sample Password Policy`. A sample policy `NETQWDDCFG-pub-cp-user-default-password` is added in the default configuration package. This policy adds a random password to new users created in OpenText Identity Vault. The password generated confirms to the password policy configured in the GCV (`default.password.policy`). This policy has to be added to the creation policy set to get executed.

User Advanced Settings: If this option is set to **show**, fill the following fields:

- **Creation Settings**: If this option is set to **show**, fill the following fields:
  - **Mandatory Attributes for User creation:** Lists all the required attributes for creating a new user object. Each of the specified attributes must be included in the Publisher filter. By default, the following attribute are available to select. You can add any custom attribute by clicking on the icon:
    - `Given Name`
    - `Surname`
    - `WorkforceID`
    - `wd-WorkerIDType`
    - `wd-UserName`
  - **User CN Attribute:** Specify the name of an attribute to be used for creating CN of user. If custom policy is defined to create CN, mention the value as **Custom**.
- **Match Settings**: If this option is set to **show**, fill the following fields:
  - **Matching Attributes for User:** Select a method for matching the users in IDV. Select **Custom**, if you want to add a customized policy. The system provide following options to match the user under publisher matching policy:
    - `Match WorkforceID`
    - `Match CN`
    - `Match all above in order`
    - `Match Custom-Define yourself`
- **Contact Settings**: If this option is set to **show**, fill the following fields
  - **Default Country ISO Code:** Specify the default country ISO Code for modifying the contact data.
  - **Work Contact Usage Type:** Specify the value of communication usage type as configured in Workday for work contact information.
  - **Home Contact Usage Type:** Specify the value of communication usage type as configured in Workday for home contact information.
  - **Work Phone Contact Details:** Specify the mapping attributes for different types of contact devices used at work.
    - **Device Type:** Specify the ID of the contact type as configured in Workday. For example, Landline, Mobile etc.
    - **Attribute Name:** Specify the name of an attribute, which holds the contact values for the given device type. If there is only a single value for th given device type, you can specify a single-value or a multi-value type. In case, there are multiple values for the given device type, you must only specify multi-value attribute.

- **Home Phone Contact Details:** Specify the mapping attributes for different types of contact devices used at home.
  - **Device Type:** Specify the ID of the contact type as configured in Workday. For example, Landline, Mobile etc.
  - **Attribute Name:** Specify the name of an attribute, which holds the contact values for the given device type. If there is only a single value for th given device type, you can specify a single-value or a multi-value type. In case, there are multiple values for the given device type, you must only specify multi-value attribute.
- **Country Phone Codes:** Select the country phone code of the user. You can add a country phone code that are available in Workday. For the procedure to fetch a country phone code from Workday, see "Fetching Country Phone Code ID and Phone Type ID from Workday" on page 38.

11 On the subsequent installation pages, fill the following fields as shown in the respective ***<Object>* - Publisher Settings** pages, then click **Next**:

***<Object>* Placement Container:** This global configuration value is used to get a reference to the container in IDV where new objects will be created. It may be different than object search container. The DN is expected in slash-format such as `Data\Workday\relation`. If you want to configure a custom container, specify custom as the value and define a custom placement policy. The following containers are created by default:

- Relation Objects: `data\workday\relation`
- Job Family Objects: `data\workday\jobfamily`
- Location Objects: `data\workday\location`
- Photo Objects: `data\workday\photo`
- Job Profile Objects: `data\workday\jobprofile`
- Organization Objects: `data\workday\organization`

***<Object>* Search Container:** This global configuration value is used to search the container in IDV. The DN is expected in slash-format such as `Data\Workday\relation`.

NOTE: The field values for the search containers are the same as placement containers.

Advanced Settings: Defaults to **Hide**, set as **Show** to configure advanced settings for the object. The associated fields to configure the advanced settings appear.

- **Advanced Settings**
  - ***<Object>* Advanced Settings**: Defaults to **Hide**, set as **Show** to configure advanced settings.
    - **Creation Settings**
      - **Mandatory Attributes for *<Object>* creation**: Specify the list of required attributes for creating a new Photo object. Each of the attributes specified must be included in the Publisher filter.

        For example, `wd-RelationID`.
      - ***<Object>* CN Attribute**: Specify the attribute name used for CN creation of Photo.

        For example, `wd-RelationID`.

- **Match Settings**
  - **Matching Attribute for *<Object>***: Specify the OpenText Identity Vault attribute name mapped to the application attribute. This attribute must be part of Mandatory Attributes list in creation settings.

    For example, `wd-relationID`

12 In the Retry Attributes page, specify a comma separated list of attribute modifications to retry when the user is activated in the **Attributes to modify when the user is activated.**

   For example, `wd-HomePrimaryPhone`, `wd-WorkPrimaryPhone`, `Internet EMail Address`, `homeEmailAddress`

13 Review the summary of tasks that will be completed to create the driver, then click **Finish**.

14 After you have installed the driver, you can change the configuration for your environment. or more information, see "Creating the Driver Object" on page 22.

Configuring Phone Contact Information in Subscriber Channel

This section explains how to configure the phone contact information of a Workday user in the Subscriber channel. As a prerequisite, you must first fetch the Country Phone Code and Phone Type ID values from Workday.

The following procedures explain how to fetch these IDs and configure the phone contact information:

- "Fetching Country Phone Code ID and Phone Type ID from Workday" on page 38
- "Adding the Phone Contact Information in Subscriber Channel" on page 39

Fetching Country Phone Code ID and Phone Type ID from Workday

Prior to configuring the phone contact information, you must fetch the Country Phone Code and the Phone Type ID from Workday. The procedures to fetch this information from Workday is explained in the following sections:

- "Fetching Country Phone Code ID from Workday" on page 38
- "Fetching Phone Type ID from Workday" on page 39

Fetching Country Phone Code ID from Workday

The following procedure explains how to fetch a country phone code value from Workday.

1 Login to Workday.

2 Perform a search for **Country Phone Code**.

3 In the search results, click **Country Phone Code** link.

4 Navigate to [...] > Integration ID > View ID.

5 Copy the value displayed in the ID column. For example, ALB_355 is for the country Albania. This ID value will be added to the Country Phone Codes in Designer.

Fetching Phone Type ID from Workday

The following procedure explains how to fetch a phone type ID from Workday.

1 Login to Workday.

2 Perform a search for Devices.

3 In the search results, select Phone Device Types.

4 Scroll over the corresponding Phone Type, and navigate to [...] > Integration ID > View ID.

5 Copy the ID value. For example, in Workday, the ID value for telephone device type is 1063.5, and for mobile device type is 1063.1.

Adding the Phone Contact Information in Subscriber Channel

To add the Home and Work phone contact information in the Subscriber channel, perform the below tasks:

1 In Designer, double click the connector line, and navigate to GCVs > User Config.

2 In the Advanced Settings section, set User Advanced Settings to True.

3 Scroll to the Country Phone Code field and add the Country Code ID value as fetched in "Fetching Country Phone Code ID from Workday" on page 38 by clicking the icon. If the required value is already present, you can just select the required ID value from the list.

4 Click Apply > OK, and deploy the driver.

In OpenText Identity Console:

1 Click the User Management tile.

   1a Select the required user.

   1b On the Modify Users page, select the Others tab.

   1c Ensure the following attributes are present in the Valued Attributes list.

   - `wd-HomePrimaryPhone`
   - `wd-HomePrimaryDeviceType`
   - `wd-WorkPrimaryPhone`
   - `wd-WorkPrimaryDeviceType`

   If these attributes are not present, add them from the unvalued attribute list by clicking the + sign.

2 Edit these attributes and add their corresponding values, as shown below:

   - `wd-HomePrimaryPhone`: Add the new phone number with the country code. For example, <*+297 1122334456*>.
   - `wd-HomePrimaryDeviceType`: Add the home device type ID as fetched from Workday. For more information, see "Fetching Phone Type ID from Workday" on page 39.

- `wd-WorkPrimaryPhone`: Add the new phone number with the country code. For example, *<+297 8877665554>*.
- `wd-WorkPrimaryDeviceType`: Add the work device type ID as fetched from Workday. For more information, see "Fetching Phone Type ID from Workday" on page 39.

IMPORTANT: You must ensure to add the ID values to the above mentioned attributes based on the device types respectively.

For example:

- For a mobile device type:
  - `wd-HomePrimaryPhone` as *<+297 1122334456>*
  - `wd-HomePrimaryDeviceType` ID as **1063.1**
- For a telephone device type:
  - `wd-WorkPrimaryPhone` as *<+297 8877665554>*
  - `wd-WorkPrimaryDeviceType` ID **1063.5**

---

3 Save.

Configuring Entitlements

In Designer, after creating the driver object, double click the connector line and navigate to **GCVs > Entitlement** tab. Specify the following parameters to configure entitlements:

- **Enable User-Based Security Group Entitlement**: Select the value as **true** to enable the entitlements feature for User-Based Security Group.
- **Enable Assignable Role Entitlement**: Select the value as **true** to enable the entitlements feature for Assignable Roles.

NOTE: After configuring the Workday driver for entitlements, you can deploy the driver. Post deployment, perform the procedures as explained in the following sections.

Fetching Entitlements from Workday Portal

You must ensure to perform a "CodeMap refresh" to fetch all the user-based security groups and assignable roles from Workday. The code map refresh, helps to facilitate the grant or revoke permissions through OpenText Identity Manager. For the procedure to refresh the code map, see Entitlement Query Settings (https://www.netiq.com/documentation/identity-manager-4.10/help_idm_identityapps_dash_admin/data/entitlement-query-settings-configuration.html) section in the Help for the Identity Applications Dashboard (https://www.netiq.com/documentation/identity-manager-4.10/help_idm_identityapps_dash_admin/data/front.html).

Assigning Permissions to a User

You can assign the required permissions to a new user, or change an already assigned permission from an existing user to a new user. In OpenText Identity Manager, this changing of the assignment is a two step process as follows:

1. First, you must revoke the currently assigned resource permission, and submit the request.
2. Then, grant the permission to another user and submit the request again.

For more information to perform the above steps, see About the Resource Editor (https://www.netiq.com/documentation/identity-manager-4.10/identity_apps_design/data/resource-editor.html) section in the OpenText™ Identity Manager - Administrator's Guide to Designing the Identity Applications (https://www.netiq.com/documentation/identity-manager-4.10/identity_apps_admin/data/bookinfo.html).

Configuring the Driver to Include Additional Transaction Type References

Additional transaction type references can be configured in the Publisher driver parameter **Transaction Log Criteria Reference List for Normal Polling** (**wdWorkerTransactLogCritList**). By default, if a simple value is added to this parameter, this becomes a `Business_Process_Type` transaction type reference.

In this case, the transaction reference types for each business process types must be obtained from the Workday application. To obtain the business process type values from Workday, login to the Workday application > Go to **Transaction Types available for Subscription** report > click on the required event > click on **Integration IDs** > collect the value for the **Business Process Types** ID as displayed. For example, if the Business Process Type is **Termination** in the Workday application, the corresponding `wdWorkerTransactLogCritList` driver configuration transaction type value will be **Terminate Employee**.

If a transaction type reference other than `Business_Process_Type` is needed, the format of the value entered in **Transaction Log Criteria Reference List for Normal Polling** should have the name of the transaction type reference, a colon and then the value for that transaction type reference. For example,

```
Event_Lite_Type_ID:Custom Object Non-Effective-Dated Change
```

or

```
WID:08e574a19f6b400fac5dbe71ea854f1d
```

As per the above examples, any values without a colon will default to a `Business_Process_Type`. Values with a colon will be parsed with the value of the transaction type reference on the left of the colon and the value for that transaction type reference on the right of the colon.

Apart from `Business_Process_Types`, Workday 32.0 provides the following transaction type references:

- `WID`

- `Event_Lite_Type_ID`
- `Rorganization_Activity_Type`

Deploying the Driver Object

After the driver object is created in Designer, it must be deployed into the OpenText Identity Vault.

1. In Designer, open your project.
2. In the Modeler, right-click the driver icon or the driver line, then select **Live > Deploy**.
3. If you are authenticated to the OpenText Identity Vault, skip to Step 5; otherwise, specify the following information:
   - **Host:** Specify the IP address or DNS name of the server hosting the OpenText Identity Vault.
   - **Username:** Specify the DN of the user object used to authenticate to the OpenText Identity Vault.
   - **Password:** Specify the user's password.
4. Click **OK**.
5. Read through the deployment summary, then click **Deploy**.
6. Read the successful message, then click **OK**.
7. Click **Define Security Equivalence** to assign rights to the driver and all required jobs.

   The driver and all required jobs require the rights to objects within the OpenText Identity Vault. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user.

   7a Click **Add**, then browse to and select the object with the correct rights.

   7b Click **OK** twice.

   For more information about defining a Security Equivalent User in objects for drivers in the OpenText Identity Vault, see Establishing a Security Equivalent User (https://www.netiq.com/documentation/identity-manager-4.10/security/data/establishing-security-equivalent-user-in-identity-manager.html) in the OpenText™ Identity Manager Security Guide (https://www.netiq.com/documentation/identity-manager-4.10/security/data/identity-manager-security-guide.html).
8. Click *Exclude Administrative Roles* to exclude users that should not be synchronized.

   You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization.

   8a Click **Add**, then browse to and select the user object you want to exclude.

   8b Click **OK**.

   8c Repeat Step 8a and Step 8b for each object you want to exclude.

   8d Click **OK**.
9. Click **OK**.

Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver and cause events to occur. OpenText Identity Manager is an event-driven system, so after the driver is started, it won't do anything until an event occurs.

To start the driver:

1. In Designer, open your project.
2. In the Modeler, right-click the driver icon or the driver line, then select **Live > Start Driver**.

   The driver cannot initialize completely unless it successfully connects to the Remote Loader and loads the Workday driver shim.

For information about management tasks for the driver, see Chapter 8, "Managing the Driver," on page 103.

Activating the Driver

To activate the Workday driver, activate the Metadirectory engine, then activate the driver by using the separate Workday activation key. If you created the driver in a driver set that has not been activated, you must activate the Metadirectory engine and the driver within 90 days. Otherwise, the driver stops working.

If driver activation has expired, the following error message is displayed in the ndstrace window:

```
DirXML Log Event -------------------
Driver: \METADIRECTORY\system\DriverSet\WorkdayDriver
Channel: Subscriber
Status: Error
Message: Code(-9075) Shutting down because DirXML engine evaluation period
has expired. Activation is required for further use.
```

To use the driver, you must reactivate it.

For more information on activation, see Activating OpenText Identity Manager (https://www.netiq.com/documentation/identity-manager-4.10/idm_overview_planning/data/activating-identity-manager.html) in the OpenText™ Identity Manager CE 24.4 (v4.10) Overview and Planning Guide (https://www.netiq.com/documentation/identity-manager-4.10/idm_overview_planning/data/identity-manager-overview-and-planning.html).

Advanced Synchronization of Worker Photos from the Workday Portal

The photo synchronization between Workday and the driver happens once in every 24-hours. If any Organization has a large number of worker base, the driver takes longer than usual to fetch all the worker's photo. Sometimes in such cases, the process might fail due to timeout.

IMPORTANT: It is highly recommended to use the photo change business process to fetch worker photos from Workday. The following procedure to synchronize worker photos from the Workday is not recommended as this functionality might be deprecated in the near future.

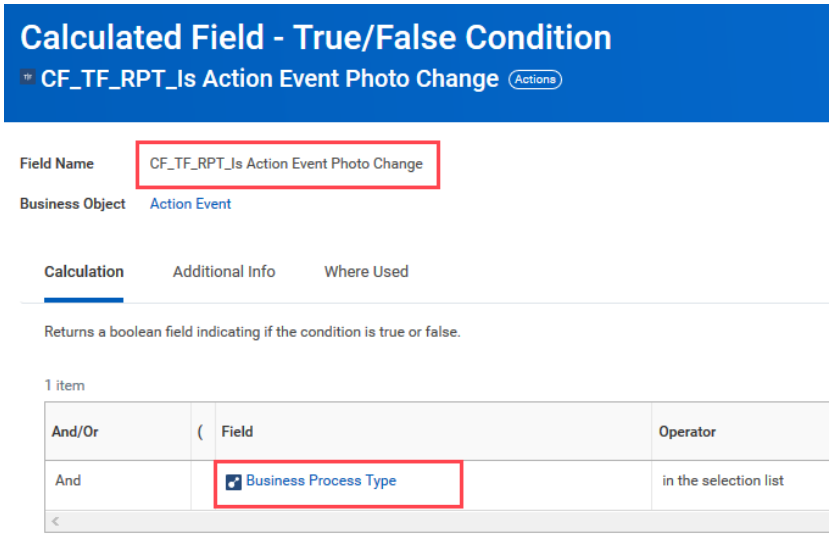
The Workday system stores the date/time value of each change in worker's photo. These date/time values can be used via calculated fields and custom reports (created using the calculated fields) to improve driver's performance while performing photo synchronization. The Workday driver uses this custom report to compare the photo related details based on the last date/time value of the worker's photo since the previous polling.

NOTE: The following configuration is not mandatory and only recommended to improve the driver's performance.

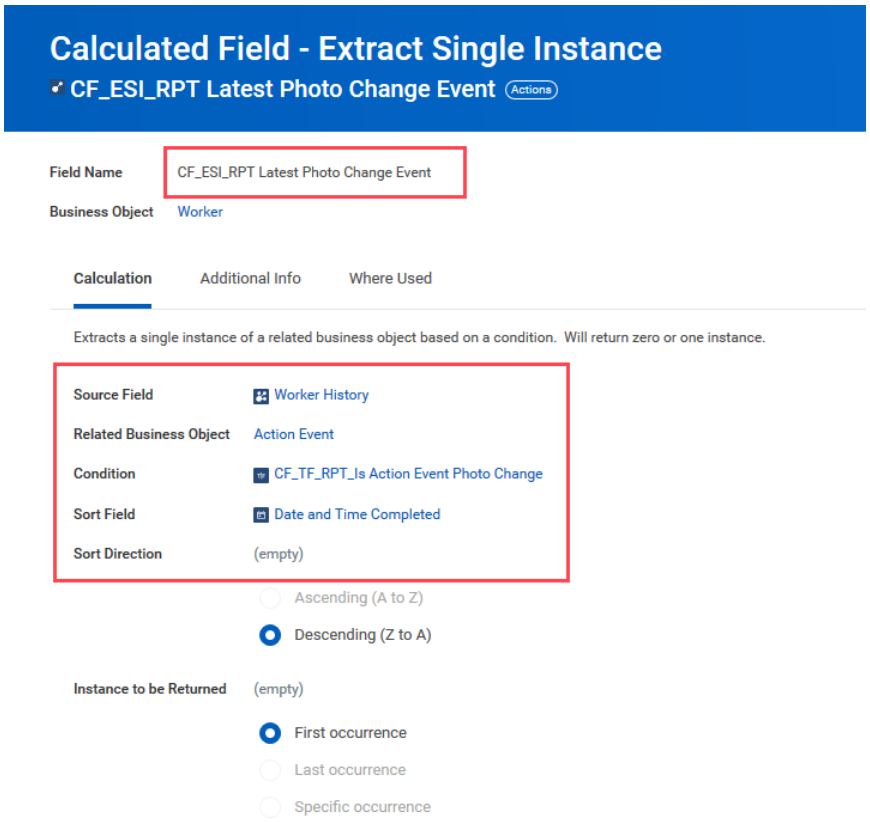
Perform the following steps to create calculated fields and custom report in the Workday portal:

1 Create the following calculated fields using the **Create Calculated Field** option in the Workday portal:

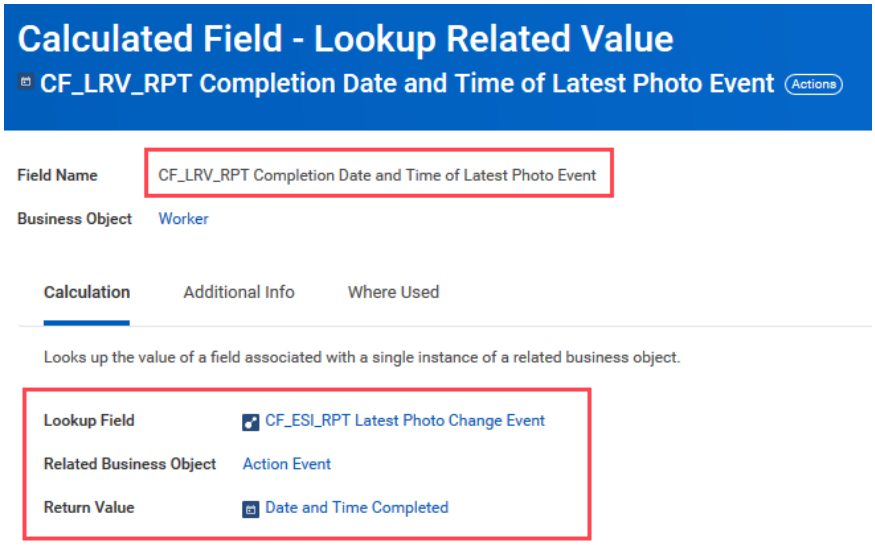
   1a Create a calculated field. For example, we've created a field named **CF_TF_RPT_Is Action Event Photo Change**.



   1b Create another calculated field using the field created in the previous step as a **Condition**. For example, we've created the second field with the name **CF_ESI_RPT Latest Photo Change Event**.



1c Create a third calculated field using the field created in previous step as a **Lookup Field**. For example, we have created the field named **CF_LRV_RPT Completion Date and Time of Latest Photo Event**.



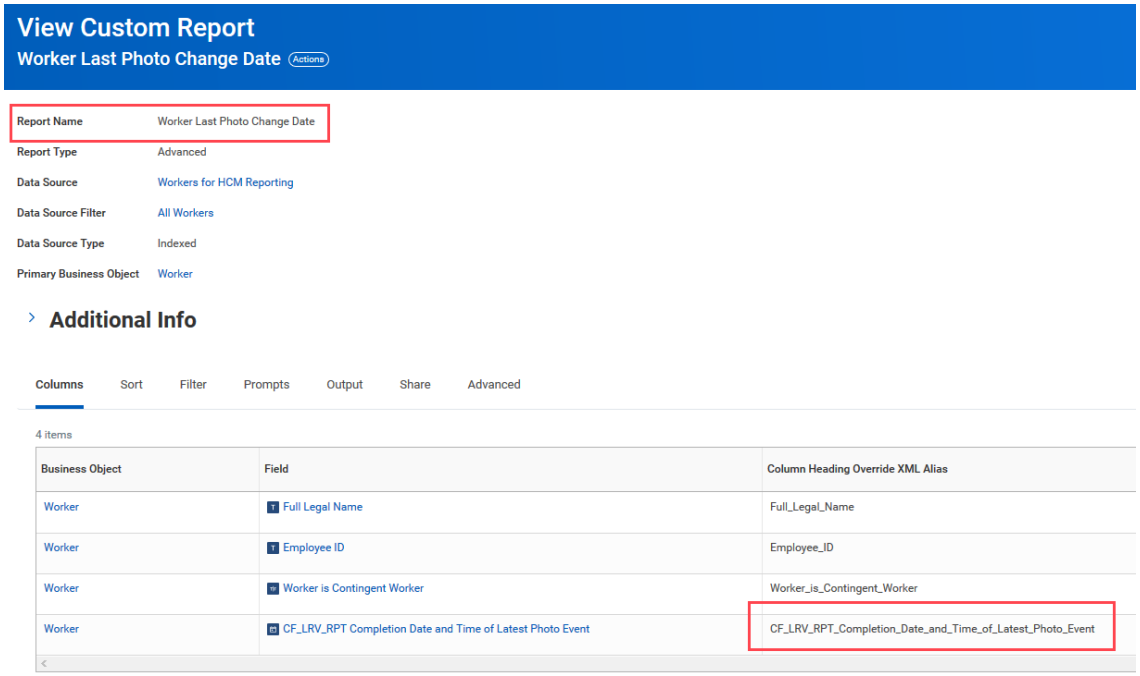
2 Create a custom report using the **Create Custom Report** option in Workday portal. You must ensure that the integration user has access to this report or the report is created as an integration user.

Create a custom report with the following mandatory fields:

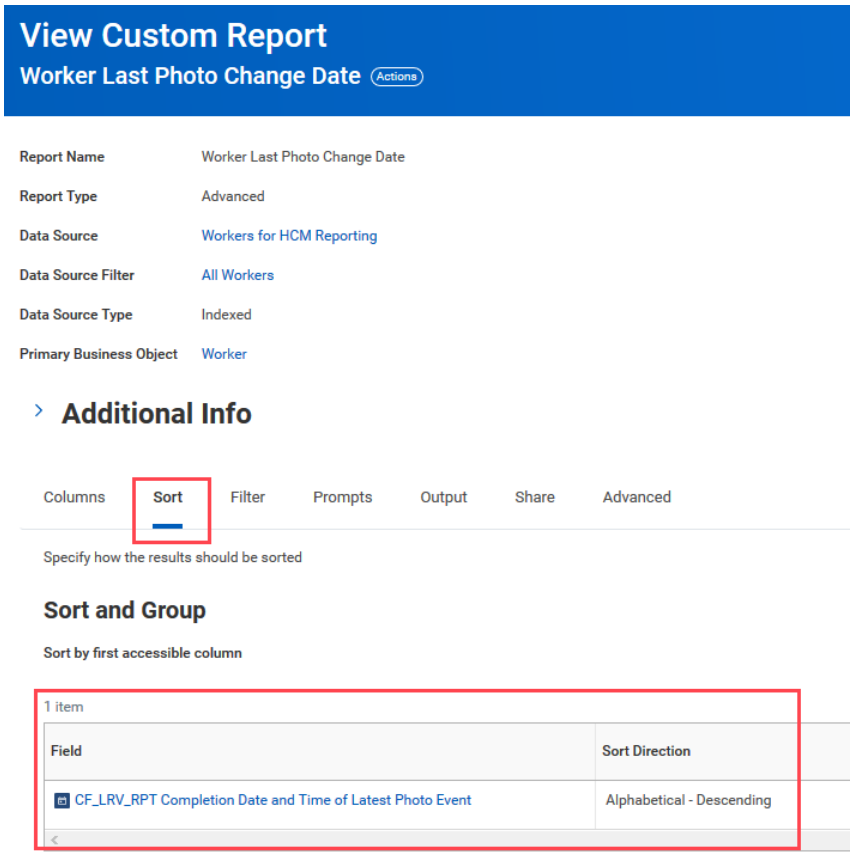
- Employee ID
- Worker is Contingent Worker
- The calculated field created (**CF_LRV_RPT Completion Date and Time of Latest Photo Event**) in Step 1c.

Note down the calculated field name appearing under **Column Heading Override XML Alias** as shown in the below image and specify the same in the **Photo Update Calculated Field** in Designer while configuring the driver.

NOTE: You can add additional fields as per your requirement.

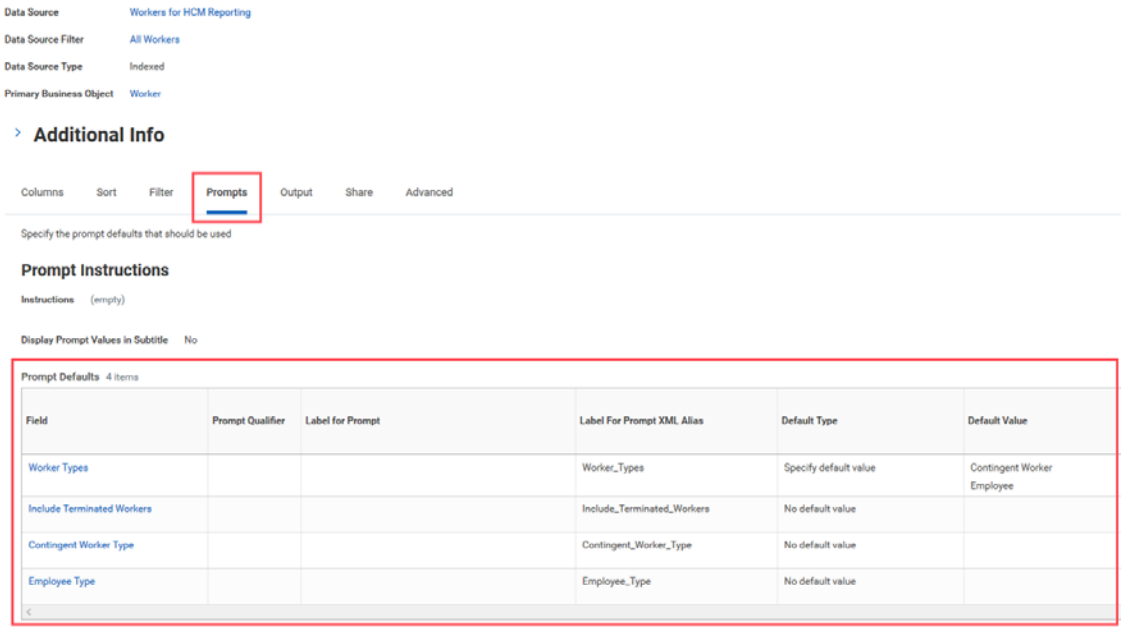


3 Click on the **Sort** tab to enable **Sorting Field** for the newly created custom report based on alphabetical descending order. This helps to display the latest changes at the top of the report.



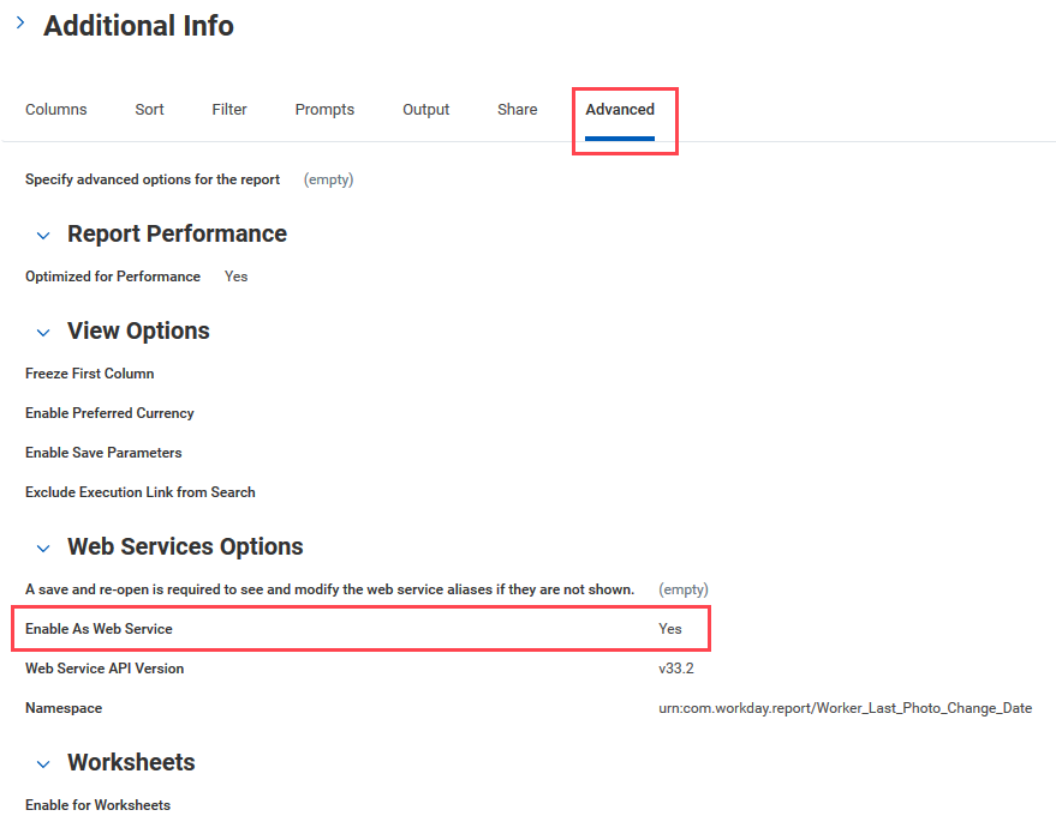
4 Create prompts for the report under the **Prompts** tab.

Provide default values for **Worker Types** as **Contingent Worker** and **Employee**. This will ensure that both types of Workers are included in the report.

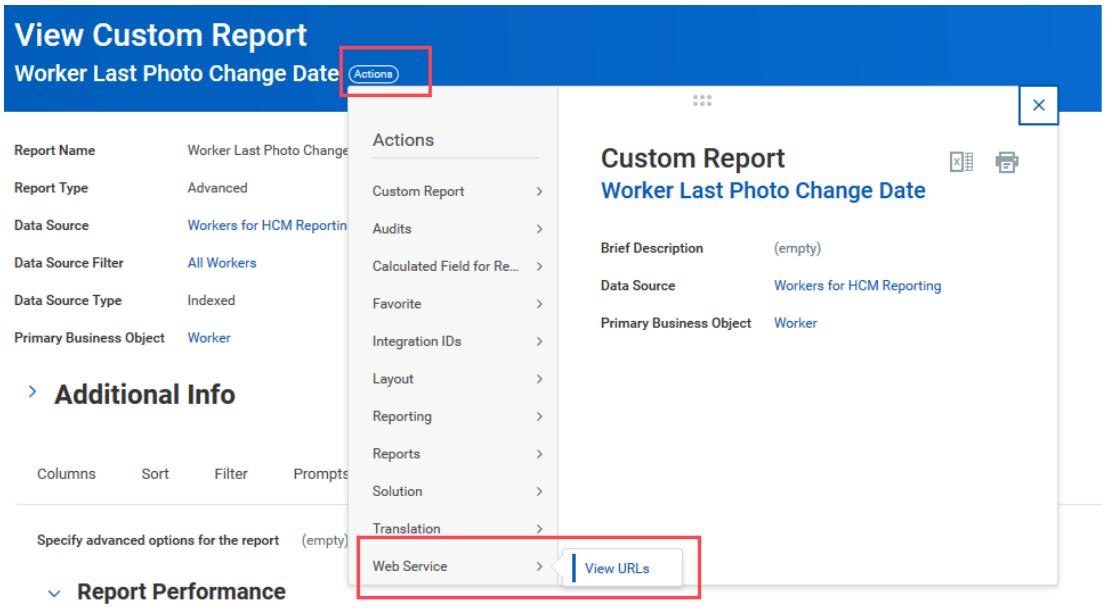


5 Enable the report as Web Service under the **Advanced** tab.

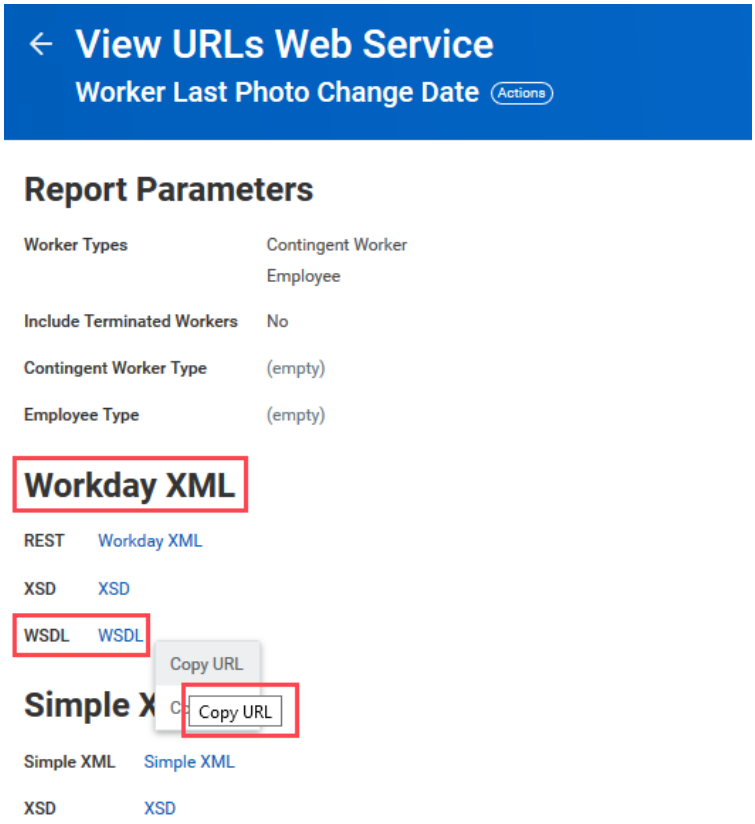
NOTE: Note down the Namespace value and specify the same in the Photo Report Namespace field in Designer while configuring the driver.



6 Click on Actions > Web Service > View URLs to get the Web Service URL of the report.



7 Copy the URL for **Workday XML** > **WSDL** option and save it. You need to specify this URL in the **Photo Update Report URL** field in Designer while configuring the driver.



3 Customizing the Workday Driver

NOTE: The resulting XML is the data interpreted between OpenText Identity Manager and Workday environment. For the most efficient processing, the external stylesheet must return the attributes only that have a different value or any new attributes. However, it is advised to retain the code that generates association as is, to ensure that the results of the external stylesheet are correctly merged with the results from the internal stylesheet.

Apart from the default attributes available in OpenText Identity Manager, you can customize the attributes by modifying the default stylesheet of Workday Driver. As Workday supports multiple attributes, and the business need may not use all the attributes that are supported by Workday, the stylesheet customization provides the flexibility to manage only the attributes that are required for your business environment.

The following sections explain how the attributes can be customized:

- "Customizing the Publisher Channel for Additional Attributes" on page 51
- "Customizing the Subscriber Channel for Additional Attributes" on page 53
- "Customizing Entitlements" on page 55

Customizing the Publisher Channel for Additional Attributes

You can customize the stylesheet to:

- "Modifying an Existing Attribute" on page 51
- "Adding a New Attribute" on page 52

Modifying an Existing Attribute

You can modify an existing attribute value in the custom stylesheet. For example, consider a scenario where you want to update your domestic phone number in place of your international phone number. In this scenario, you can just modify the phone attribute in the custom stylesheet as shown below:

- Stylesheet for phone attribute (in bold) is shown below:

```
<attr>
<xsl:attribute name="attr-name">
<xsl:value-of select='"wd-WorkPrimaryPhone"'/>
</xsl:attribute>
<value>
<xsl:value-of select="$primary_work_phone_node/
@wd:International_Formatted_Phone"/>
</value>
</attr>
```

- The modified custom stylesheet:

```
<attr>
<xsl:attribute name="attr-name">
<xsl:value-of select='"wd-WorkPrimaryPhone"'/>
</xsl:attribute>
<value>
<xsl:value-of select="$primary_work_phone_node/
@wd:National_Formatted_Phone"/>
</value>
</attr>
```

You can customize the Workday objects and their attributes using custom stylesheet over the Publisher and Subscriber channel. The custom XSLT paths must be specified for the required objects as per your environment to exchange information between Workday and OpenText Identity Vault. For more information on customization, see Step 8 on page 28.

IMPORTANT: These above mentioned attribute values (in the custom stylesheet) are sample values and must be used for reference purposes only. If you want to add a new custom attribute, or modify an existing attribute for the respective workday supported objects (files), there is no significant action or insertion required though the Workday driver configuration. The Workday driver shim, by default, would handle the existing objects that are supported by Workday and also their respective attribute transformations.

Adding a New Attribute

To add a new attribute, the perform the following steps:

1. Fetch the SOAP response from Workday.
2. Identify the XPATH of the required attribute from the fetched SOAP response.
3. Build the XPATH for the attribute in the required format. This path will be added in the XSLT stylesheet.
4. Add the XPATH of the new attribute to the XSLT stylesheet. For example, if you are adding an attribute for Marital Status, add the following to the XSLT stylesheet.

```
<!--  **** Worker : Marital Status************* -->
<attr>
<xsl:attribute name="attr-name">
<xsl:value-of select='"wd-MaritalStatusID"'/>
</xsl:attribute>
<value>
<xsl:value-of select="wd:Worker_Data/wd:Personal_Data/
wd:Marital_Status_Reference/wd:ID[@wd:type='Marital_Status_ID']"/>
</value>
</attr>
```

NOTE: In the above example, the bold text is the XPATH that is built for Marital Status attribute, referring to the fetched schema file from Workday.

---

5. Extend the schema file for the new attribute. The attribute added in the XSLT stylesheet should be added in the schema extension (.sch) file also, so that OpenText Identity Manager supports that attribute.

Ensure to add the attribute name to the Object Class, as shown below:

```
"wd-MaritalStatusID" ATTRIBUTE ::=
{
 Operation ADD,
 SyntaxID SYN_CI_STRING,
 Flags {DS_STRING_ATTR},
 ASN1ObjID {2 16 840 1 113719 1 14 4 1 2746}
}
```

6 In OpenText Identity Console, click the **IDM Administration** tile.

7 On the Driver Dashboard, click the Workday Driver icon.

8 Click the **Data Transformation and Synchronization** tab.

9 Click **Filter** in the **Publisher Channel**.

10 Add the attribute name, for example `wd-MaritalStatusID`. To add:

- **10a** Select **User** as class and click **Add Attribute**.
- **10b** Click **Show All Attribute** to display the list of attributes applicable for the class.
- **10c** Add `wd-MaritalStatusID`, and save.
- **10d** Select the attribute from the class list and set the **Publisher** option to **Synchronize** to match the attribute name with the schema extension name.

11 Add the XSLT stylesheet path as prepared in Step 4 on page 52. To add the path:

- **11a** Click the **Configuration** tab.
- **11b** Expand the Driver Parameters section and click the **Subscriber Settings** tab.
- **11c** Add the path in the **Import Worker Stylesheet** field.

12 Save.

13 Restart the driver for the changes to take effect.

To add the XSLT stylesheet path, in Designer:

1 Double click the connector line, and navigate to **Driver Configuration > Driver Parameters.**

2 Select Publisher Options > Worker > Worker Poll Settings > Import Worker Stylesheet.

3 Restart the driver for the changes to take effect.

Customizing the Subscriber Channel for Additional Attributes

The procedure involves the following steps for subscriber channel:

1 Fetch the XDS event from the Workday log.

2 Identify the XPATH of the required attribute from the fetched XDS event.

3 Build the XPATH for the attribute in the required format, to add to the custom stylesheet.

4 Add the XPATH of the new attribute to the custom stylesheet. You can copy the custom stylesheet, edit and upload it as per your requirement.

NOTE:

- Ensure that you provide a complete updated stylesheet file when configuring the driver.
- This feature is supported only for `worker` and `photo` objects.

For example, you have to add BirthDate template and the Caller template (to call the BirthDate template) in the custom stylesheet for the BirthDate attribute.

The examples for BirthDate and BirthDate Caller templates are shown below:

Sample BirthDate template:

```
<xsl:template name="Date_of_Birth">
<xsl:param name="Birthdate"/>
<xsl:param name="workerIdValue"/>
<xsl:param name="workerIdType"/>
<!-- Prepare SOAP envelope for DOB -->
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:bsvc="urn:com.workday/bsvc">
<soapenv:Header />
<soapenv:Body>
<xsl:message>version is: <xsl:value-of select="$wdVersion"/></
xsl:message>
<bsvc:Change_Personal_Information_Request bsvc:version="{$wdVersion}">
<bsvc:Change_Personal_Information_Data>
<bsvc:Worker_Reference bsvc:Descriptor="?">
<bsvc:ID bsvc:type="{$workerIdType}">
<xsl:value-of select="$workerIdValue"/></bsvc:ID>
</bsvc:Worker_Reference>
<bsvc:Personal_Information_Data>
<bsvc:Date_of_Birth><xsl:value-of select="$Birthdate"/></
bsvc:Date_of_Birth>
</bsvc:Personal_Information_Data>
</bsvc:Change_Personal_Information_Data>
</bsvc:Change_Personal_Information_Request>
</soapenv:Body>
</soapenv:Envelope>
</xsl:template>
```

Sample BirthDate Caller template:

```
<!--Personal Info attributes -->
 <xsl:variable name="lv-Birthdate" select="modify-attr[(@attr-name='wd-
BirthDate')]/add-value/value/text()"/>
<xsl:call-template name="Date_of_Birth">
<xsl:with-param name="Birthdate" select="$lv-Birthdate"/>
<xsl:with-param name="workerIdType" select="$lv-wd-WorkerIDType"/>
<xsl:with-param name="workerIdValue" select="$lv-wd-EMPLID"/>
</xsl:call-template>
```

NOTE: In the above example, the bold text is the XPATH that is built for BirthDate attribute by referring to the fetched schema file from Workday.

5 In OpenText Identity Console, click the **IDM Administration** tile.

6 On the Driver Dashboard, click the Workday Driver icon.

7 Click the **Data Transformation and Synchronization** tab.

8 Click **Filter** in the **Subscriber Channel**.

9 Add the attribute `wd-BirthDate`. To add:

   9a Select **User** as class and click **Add Attribute**.

   9b Click **Show All Attribute** to display the list of attributes applicable for the class.

   9c Add `wd-BirthDate`, and save.

10 Select the attribute from the class list and set the **Subscriber** option to **Synchronize** to match the attribute name with the schema extension name.

11 Restart the driver for the changes to take effect.

In Designer:

1 double click the connector line and navigate to **Driver Configuration > Driver Parameters**.

2 Select Subscriber Options > Worker Settings > Import Worker Stylesheet.

3 Add the XSLT stylesheet path as prepared in Step 4 on page 53, and deploy the driver.

4 Restart the driver for the changes to take effect.

NOTE: When supplied stylesheets are used as a template to create external stylesheets, it is critical that the results of processing the response document through the stylesheet will return the right association value in the resulting XML. For the most efficient processing, the external stylesheet must return the attributes only that have a different value or any new attributes. However, it is advised to retain the code that generates association as is, to ensure that the results of the external stylesheet are correctly merged with the results from the internal stylesheet.

Customizing Entitlements

The Workday Driver supports entitlements to manage assigned permissions and revocations. This feature is available as an optional package. You can use the entitlements feature to manage the user memberships in User-Based Security Groups, and for assigning or revoking organizational roles to workers. From the Workday driver perspective, a User-Based Security Group grants the rights to the users in Workday. Similarly, the Organizational Role entitlement grants permissions to workers to perform certain tasks based on roles in the organization.

The driver must be enabled for configuring the supported entitlements, and managing the permissions of a user account in OpenText Identity Manager. The entitlements can be granted or revoked only by the entitlement permission administrators.

- "Prerequisites" on page 56
- "Supported Entitlements" on page 56

Prerequisites

To configure entitlements, you must have the following drivers, and a user with the applicable role as follows:

- Role and Resource Driver
- User Application Driver
- Administrative privileged user

Supported Entitlements

The Workday Driver supports the following entitlement types:

- "User-Based Security Group" on page 56
- "Assignable Role" on page 57

User-Based Security Group

The User-Based Security Group entitlement is a set of access permissions provided for a user belonging to a user-based security group in Workday. This entitlement grants or revokes the membership to a User-Based Security group for a user. If the User-Based Security Group entitlement is granted, the user is added to the user-based security group, and if revoked, the user is removed.

When configuring the driver, in the **Entitlement Configuration** page, select **Enable User-Based Security Group Entitlement** as **true**.

You can synchronize the user memberships of the User-Based Security Groups from Workday to OpenText Identity Manager through this entitlement using the Controlled Permission Reconciliation Services (CPRS) feature. For more information, see Using Controlled Permission Reconciliation Services (https://www.netiq.com/documentation/identity-manager-4.10/identity_apps_admin/data/netiq-identity-manager-cprs-controlled-permission-reconciliation-services.html) in the OpenText™ Identity Manager - Administrator's Guide to the Identity Applications (https://www.netiq.com/documentation/identity-manager-4.10/identity_apps_admin/data/bookinfo.html).

NOTE: Prior to using the CPRS feature, you must migrate all users from Workday at least once post upgrading to Workday driver 1.3. During the migration process, a cache mapping for `user-name` and `worker-id` is created.

For more information, see the **Migrate on Startup** field settings in Step 8 on page 28.

Assignable Role

Every Workday organization defines certain roles, which can be assigned directly to workers in Workday. The role assignments that are defined in one Workday organization is independent from the assignment defined in another Workday organization. As a result of this, the Workday assignable role entitlement for a role contains the name of the organization in its entitlement association.

For example, consider the Organizations as Org1 and Org2 and the defined roles as Role1 and Role2. Also assume the following associations in Workday organization:

- Org1 defines Role1 and Role2, and
- Org2 defines only Role1

In such a case, the OpenText Identity Manager would fetch three entitlement associations from both the Workday organizations as follows:

- Role1#Org1
- Role2#Org1
- Role1#Org2

Now, if Role1#Org1 is assigned or revoked, the assignment changes in Workday will be applicable only in Org1 for Role1, and the Org2 assignment for Role1 remains unchanged.

You must ensure to grant sufficient privileges to the Workday integration user in order to execute the **Assign Roles Web Service** task in Workday. For more information, see "Prerequisites to be Performed in Workday for Assignable Roles" on page 57.

You can synchronize the worker assignments for organizational roles from Workday to IDM through this entitlement, using the Controlled Permission Reconciliation Services (CPRS) feature. For more information, see Using Controlled Permission Reconciliation Services (https://www.netiq.com/documentation/identity-manager-4.10/identity_apps_admin/data/netiq-identity-manager-cprs-controlled-permission-reconciliation-services.html) in the OpenText™ Identity Manager - Administrator's Guide to the Identity Applications (https://www.netiq.com/documentation/identity-manager-4.10/identity_apps_admin/data/bookinfo.html).

NOTE: Workday supports inherited role assignment from a supervisory organization to its sub organization. But, the Resource catalog in OpenText Identity Manager stores the information of assigned resources only, and that information is specific to organizations. Hence, by publishing the CPRS computation, you can store all the inherited role assignments.

For the procedure to configure entitlements in Designer, see "Configuring Entitlements" on page 40.

Prerequisites to be Performed in Workday for Assignable Roles

To enable role provisioning from Workday, refer to the following steps:

1. Login to the Workday tenant.
2. **Start Proxy** session with a super user.
3. Create a **User-Based Security Group**, and assign the Workday integration user to this group.
4. Search for the **User-Based Security Group** name as created in Step 3, and click the link from the search results.

5 Click > **Security Group** > **Maintain Domain Permissions for Security Group**.

6 Navigate to the **Integration Permission** section > **Domain Security Policies permitting Put access** field, and search and select **Org Designs: Assign Roles** > **OK** > **Done.**

7 Search for **Activate Pending Security Policy Changes** and click the link from the search results.

8 Add a comment, and click **OK**.

9 In the **Activate Pending Security Policy Changes** page, check the **Confirm** checkbox and click **OK**.

10 Search for **View Security for Securable Item**, and select the link from the search results.

11 Provide a name in the **Assign Roles** field, and click **OK**.

12 In the **Tasks** list, click **View Security** button corresponding to the **Assign Roles Web Service** task.

13 In the **Business Process Security** table, navigate to Initiating Action for **Business Process** > **Assign Roles** > **Business Process Policy** > **Edit**.

14 Scroll to find the **Assign Roles Web Service** task.

15 In the corresponding **Security Groups** field, search and select the name of the **User-Based Security Group** as created in Step 3, and click **OK** > **Done**.

16 To activate these changes, repeat Step 7 to Step 9.

4 Lifecycle Business Process for the Workday Driver

Workday HCM (Human Capital Management) system has multiple lifecycle business processes available for hiring, termination, move worker position, and attribute management for an employee or contingent worker. The lifecycle business process feature is available with Workday 1.3 and above.

This chapter comprises the following sections:

- "Business Process Use Cases" on page 59
- "Details of Business Process Package and Associated Files" on page 60
- "Upgrading the Workday Driver to Use Business Process Lifecycle" on page 60
- "Configuring Business Process" on page 61

Business Process Use Cases

As a part of lifecycle business process, the following use cases are supported for a worker:

- **Hire Business Process**:
  - **Hiring an employee for future date**: This use case helps to manage the worker details for workers hired for a future date.

    For example, if a worker is being hired for a future date, the hiring company will have the worker's generic information updated for the record, which is partial. To have the complete information (which includes employee ID, official email ID, etc) updated, you must specify the number of days prior to keep the worker record ready.

    Having this information ready and available helps to prevent any on-boarding issues. Hence, the advance number of days is specified (in the field **Future Object Creation Days in Advance**) with respect to the hiring date, so that a future state object of the worker is created to execute a seamless on-boarding process.

    For more information, see "Configuring Business Process" on page 61.
  - **Hire Contingent worker for future date**: For a Contingent worker to be hired on a future date, you can acquire and have all the required information updated. All the future state objects (such as worker details and position details) are created in advance to the effective hiring date.
- **Rehire Business Process**:
  - **Rehire employee for future date**: You can set the number of days in advance to acquire and update all the information for hiring an ex-employee. OpenText Identity Manager creates all the future state objects as per the number of days set in advance with respect to the effective hiring date of the ex-employee.

- **Rehire Contingent worker for future date**: Similar to the rehire employee, you can set the number of days in advance for the future state objects to be created for a contingent worker.
- **Convert Contingent worker to employee**: You can set the number of days in advance to convert a contingent worker to a regular employee. OpenText Identity Manager acquires and updates all the required information and creates the future state objects in advance to the conversion effective date.

- **Terminate Business Process**:
  - **Terminate Employee**: You can set the number of days in advance to deactivate the worker's details, due to termination of an employee.
  - **Contract closure for Contingent worker**: You can set the number of days in advance to deactivate the worker's details, due to expiry of the contract period.
  - **Employee Retirement**: When the employee qualifies for retirement, you can set the number of days in advance to deactivate the employee's details.
- **Attribute Management of Contact Information**: You can manage the attributes of the workers contact information such as, home address etc.

Details of Business Process Package and Associated Files

You must ensure to select the required package and install the required business process files to configure the business process feature.

For more information about packages and associated files, see:

- "Importing the Current Driver Packages" on page 22, for the package details.
- "Installing the Driver Files" on page 19, for driver files.

Upgrading the Workday Driver to Use Business Process Lifecycle

The Business Process package is available for driver versions 1.3 and above. As this package is optional, it is not readily available to configure when you upgrade your existing driver. You must manually add the package to the Installed Packages list in Designer's Package Management page (**Double click the connector line > Packages > ✚ > select the Workday Business Process package > OK > Apply**) to configure the Business Process functionality.

Configuring Business Process

As a part of Workday Business Process, you can specify the container details where the information for delta and future objects must be created in the OpenText Identity Vault.

After creating the driver object in designer, double click the connector line and navigate to **GCVs > Business Process**. The Business Process Configuration page appears. Specify the values as shown below:

- **Enable Delta and Future Object Creation**: Select the value to **true**. The associated fields appear as shown below:
  - **Delta Object Placement Container**: The location of the container in OpenText Identity Vault where the delta objects will be created.

    For example, `data\workday\deltaObjects` is the location in OpenText Identity Vault, and the name of the container is `deltaObjects`.
  - **Future Object Placement Container**: The location of the container in OpenText Identity Vault where the future objects will be created.

    For example, `data\workday\futureObjects` is the location in OpenText Identity Vault, and the name of the container is `futureObjects`.
  - **Future Object Creation Days in Advance**: The number of days in advance the future object data must be created in OpenText Identity Vault.

    For example:
    - 3: Indicates that the future object will be created 3 days in advance to the effective hiring date.
    - 0: Does not create a future object in advance.
    - -1: Indicates that the future object is to be created after the effective hiring date.
  - **Delta and Future Object Retention Period in Days**: The number of days the Delta and Future Object must be retained in OpenText Identity Vault. If you specify -1, the delta and future state objects will not be cleared.

5 Attributes Mapping & Filtering

This chapter provides information about mapping of various attributes available for the Workday driver:

- "User Attributes" on page 63
- "Job Profile Attributes" on page 82
- "Job Family Attributes" on page 83
- "Relation Attributes" on page 83
- "Photo Attributes" on page 89
- "Location Attributes" on page 89
- "Organization Attributes" on page 91
- "Delta Object Attributes" on page 92
- "Future Object Attributes" on page 93
- "Attribute Filter Settings" on page 93

User Attributes

The below table lists OpenText Identity Vault user attributes. These attributes can be retrieved using the `Get_Workers` API.

Table 5-1 *Mapping of User Attributes*

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-LegalFirstName	Given name	wd:Worker_Data// wd:Legal_Name_Data//wd:First_Name	First name for a worker.
wd-LegalMiddleName	Initials	wd:Worker_Data// wd:Legal_Name_Data// wd:Middle_Name	Middle name for a worker.
wd-LegalLastName	Surname	wd:Worker_Data// wd:Legal_Name_Data//wd:Last_Name	Last name for a worker.
wd-LegalPersonalTitle	personalTitle	wd:Worker_Data// wd:Legal_Name_Data// wd:Title_Reference/@wd:Descriptor	Description of the reference to the title of a worker.
wd-LegalGenerationQualifier	Generational Qualifier	wd:Worker_Data// wd:Legal_Name_Data// wd:Social_Suffix_Reference/ @wd:Descriptor	Description of social suffix for a worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
workforceID	workforceID	wd:Worker_Data/wd:Worker_ID	The ID for the worker or the contingent worker.
wd-WorkerType	wd-WorkerType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Worker_Type_Reference/ wd:ID[@wd:type='Employee_Type_ID']	Unique identifier of the worker type for the position.
wd-AddressType	wd-AddressType	wd:Worker_Data/wd:Personal_Data// wd:Address_Data/wd:Usage_Data/ wd:Type_Data/wd:Type_Reference/ wd:ID[@wd:type='Communication_Usage_Type_ID']	Reference ID for the communication usage type.
wd-BirthDate	wd-BirthDate	wd:Worker_Data//wd:Birth_Date	Birth date of the worker.
wd-Country	co	wd:Worker_Data/wd:Personal_Data// wd:Address_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-3_Code']	Three character code for the country of the primary work address of the worker.
wd-StreetAddress	SA	wd:Worker_Data/wd:Personal_Data// wd:Address_Data/ wd:Address_Line_Data	The address line of the primary work address of the worker.
wd-City	Physical Delivery Office Name	wd:Worker_Data/wd:Personal_Data// wd:Address_Data/wd:Municipality	City part of the primary work address of the worker.
wd-State	S	wd:Worker_Data/wd:Personal_Data// wd:Address_Data/ wd:Country_Region_Descriptor	The region part of the primary work address of the worker. For example, state/ province information.
wd-PostalCode	Postal Code	wd:Worker_Data/wd:Personal_Data// wd:Address_Data/wd:Postal_Code	The postal code part of the primary work address of the worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-BusinessAddressLine1	wd-BusinessAddressLine1	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_1']	Address line 1 of the primary address of the worker.
wd-BusinessCity	wd-BusinessCity	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Municipality	City part of the primary address of the worker.
wd-BusinessCountry	wd-BusinessCountry	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-3_Code']	Three character code of the country of the primary address of the worker.
wd-BusinessPostalcode	wd-BusinessPostalcode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Postal_Code	Postal code part of the primary address of the worker.
wd-ManagementLevel	wd-ManagementLevel	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Profile_Summary_Data/ wd:Management_Level_Reference/ wd:ID[@wd:type='Management_Level_ID']	Unique identifier for the management level of the position of the worker.
wd-ManagerWorkforceID	managerWorkforceID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Organizations_Data// wd:Organization_Support_Role/ wd:Organization_Role_Data/ wd:Worker_Reference/ wd:ID[@wd:type='Employee_ID' or @wd:type='Contingent_Worker_ID']	Workforce ID of the manager of the worker.
wd-TransactionLog	wd-TransactionLog	wd:Worker_Data/ wd:Transaction_Log_Entry_Data/ wd:Transaction_Log_Entry/ wd:Transaction_Log_Data	Transaction log entry data of the worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-isManager	isManager	wd:Worker_Data//wd:Role_Data// wd:Organization_Role_Reference/ wd:ID[@wd:type='Organization_Role_ID ' and text()='Manager']	Boolean attribute indicating if the worker is a manager.
wd-BusinessState	wd-BusinessState	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Country_Region_Descriptor	Region part of the primary address of the worker. For example, state/ province information.
wd-BusinessTitle	Title	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Business_Title	Business title for the position of the primary job of the worker.
wd-C-Active	wd-C-Active	wd:Worker_Data// wd:Worker_Status_Data/wd:Active	Boolean attribute identifying whether the contingent worker is active.
wd-C-ActivePositionIDs	wd-C-ActivePositionIDs	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data[@wd:Primary_Job ='1']	Text attribute identifying position ID of a contingent worker.
wd-C-Hiredate	wd-C-Hiredate	wd:Worker_Data// wd:Worker_Status_Data/wd:Hire_Date	The most recent hire date for the contingent worker.
wd-C-Home_Countries	wd-C-Home_Countries	wd:Worker_Data// wd:Home_Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	Three character code of the home country for contingent worker' primary jobs.
wd-C-Host_Countries	wd-C-Host_Countries	wd:Worker_Data// wd:Host_Countries_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	Three character code of the host country for contingent worker's international assignment jobs.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-C-International_Assignment	wd-C-International_Assignment	wd:Worker_Data// wd:Has_International_Assignment	Boolean attribute indicating that the contingent worker has international assignments.
wd-C-OriginalHiredate	wd-C-OriginalHiredate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Original_Hire_Date	Earliest hire date of the contingent worker.
wd-C-Rehire	wd-C-Rehire	wd:Worker_Data// wd:Worker_Status_Data/wd:Rehire	Boolean attribute indicating whether the contingent worker is a rehire based on the most recent hire event.
wd-C-Retired	wd-C-Retired	wd:Worker_Data// wd:Worker_Status_Data/wd:Retired	Boolean attribute identifying whether the contingent worker is currently retired.
wd-C-Terminated	wd-C-Terminated	wd:Worker_Data// wd:Worker_Status_Data/wd:Terminated	Boolean attribute identifying whether the contingent worker is currently terminated.
wd-C-TerminationDate	wd-C-TerminationDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Termination_Date	Most recent termination date of the contingent worker.
wd-C-HireRescinded	wd-C-HireRescinded	wd:Worker_Data// wd:Worker_Status_Data/ wd:Hire_Rescinded	Boolean attribute indicating whether the contingent worker's hire was rescinded.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-E-Active	wd-E-Active	wd:Worker_Data// wd:Worker_Status_Data/wd:Active	Boolean attribute identifying whether the worker is active.
wd-E-ActivePositionIDs	wd-E-ActivePositionIDs	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/wd:Position_ID	Text attribute identifying position ID of an worker.
wd-E-Hiredate	wd-E-Hiredate	wd:Worker_Data// wd:Worker_Status_Data/wd:Hire_Date	The most recent hire date for the worker.
wd-E-Home_Countries	wd-E-Home_Countries	wd:Worker_Data// wd:Home_Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-3_Code']	Three character code of the home country for worker's primary jobs.
wd-E-Leave	wd-E-Leave	wd:Worker_Data// wd:Worker_Status_Data/ wd:Leave_Status_Data/wd:On_Leave	Boolean attribute identifying whether the worker is currently on leave.
wd-E-LeaveDate	wd-E-LeaveDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Leave_Status_Data/ wd:Leave_Start_Date	Start date of the current leave event of the worker.
wd-E-LeaveEndDate	wd-E-LeaveEndDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Leave_Status_Data/ wd:Estimated_Leave_End_Date	Estimated end date of the current leave event of the worker.
wd-E-LeaveType	wd-E-LeaveType	wd:Worker_Data// wd:Worker_Status_Data/ wd:Leave_Status_Data/ wd:Leave_of_Absence_Type_Reference/ wd:ID[@wd:type='Leave_of_Absence_Type_ID']	Unique identifier for the leace type of the current leave event of the worker.
wd-E-HireReason	wd-E-HireReason	wd:Worker_Data// wd:Worker_Status_Data/ wd:Hire_Reason_Reference/ wd:ID[@wd:type='Event_Classification_Subcategory_ID']	Unique identifier of reason for hire from the most recent hire event of the worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-E-TerminationReason	wd-E-TerminationReason	wd:Worker_Data// wd:Worker_Status_Data/ wd:Primary_Termination_Reason_Reference/ wd:ID[@wd:type='Event_Classification_Subcategory_ID']	Unique identifier of the primary reason for the worker's most recent termination.
wd-E-Host_Countries	wd-E-Host_Countries	wd:Worker_Data// wd:Host_Countries_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-3_Code']	Three character code of the host country for worker's international assignment jobs.
wd-E-International_Assignment	wd-E-International_Assignment	wd:Worker_Data// wd:Has_International_Assignment	Boolean attribute indicating that the worker has international assignments.
wd-E-OriginalHiredate	wd-E-OriginalHiredate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Original_Hire_Date	Earliest hire date of the worker.
wd-E-Rehire	wd-E-Rehire	wd:Worker_Data// wd:Worker_Status_Data/wd:Rehire	Boolean attribute indicating whether the worker is a rehire based on the most recent hire event.
wd-E-Retired	wd-E-Retired	wd:Worker_Data// wd:Worker_Status_Data/wd:Retired	Boolean attribute identifying whether the worker is currently retired.
wd-E-Terminated	wd-E-Terminated	wd:Worker_Data// wd:Worker_Status_Data/wd:Terminated	Boolean attribute identifying whether the worker is currently terminated.
wd-E-TerminationDate	wd-E-TerminationDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Termination_Date	Most recent termination date of the worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-HomeCity	homeCity	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Municipality	City part of the primary home address of the worker.
wd-HomeCountry	wd-HomeCountry	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Country_Reference/wd:ID[@wd:type='ISO_3166-1_Alpha-3_Code']	Three character country code of the primary home address of the worker.
wd-HomePostalCode	homeZipCode	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Postal_Code	The postal code part of the primary home address of the worker.
wd-HomeState	homeState	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Country_Region_Descriptor	The region part of the primary home address of the worker. For example, state/province information.
wd-HomeStreetAddress	wd-HomeStreetAddress	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Address_Line_Data	The address line of the primary home address of the worker.
wd-JobClassificationType	wd-JobClassificationType	wd:Worker_Data/wd:Employment_Data/wd:Worker_Job_Data/wd:Position_Data//wd:Job_Classification_Reference/@wd:Descriptor	Description for the job classification of the position.
wd-JobGroupReference	wd-JobGroupReference	wd:Worker_Data/wd:Employment_Data/wd:Worker_Job_Data/wd:Position_Data//wd:Job_Group_Reference/@wd:Descriptor	Description for the job group of the position.
wd-JobProfileID	jobCode	wd:Worker_Data/wd:Employment_Data/wd:Worker_Job_Data/wd:Position_Data//wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']	Unique identifier for the job profile of the position.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-JobProfileName	wd-JobProfileName	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Job_Profile_Name	Th name of the job profile of the position.
wd-LocationID	wd-LocationID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Business_Site_Summary_Data/ wd:Location_Reference/ wd:ID[@wd:type='Location_ID']	Unique identifier for the location data of the user.
wd-LocationName	L	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Business_Site_Summary_Data/ wd:Name	Name of the location of the worker.
wd-NationalID	wd-NationalID	wd:Worker_Data// wd:National_ID_Data/wd:ID	National identifier ID.
wd-PayRateType	wd-PayRateType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Pay_Rate_Type_Reference/ wd:ID[@wd:type='Pay_Rate_Type_ID']	Unique identifier to the pay rate type for the position.
wd-PositionEffectiveDate	wd-PositionEffectiveDate	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/@wd:Effective_Date	Effective date for position of the primary job of the worker.
wd-PositionEndDate	wd-PositionEndDate	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:End_Date	The effective date of end employment business process.
wd-PositionID	wd-PositionID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Position_ID	Text attribute identifying position ID of the primary job the worker.
wd-PositionStartDate	wd-PositionStartDate	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Start_Date	Date the worker first started working in the primary job position.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-PositionTimeType	wd-PositionTimeType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Position_Time_Type_Reference/ wd:ID[@wd:type='Position_Time_Type_ID']	Unique identifier to the position time type.
wd-PositionTitle	wd-PositionTitle	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Position_Title	Text attribute identifying position title of the primary job the worker.
wd-PreferredFirstName	wd-PreferredFirstName	wd:Worker_Data// wd:Preferred_Name_Data// wd:First_Name	First name of the preferred name of the worker.
wd-PreferredGenerationQualifier	wd-PreferredGenerationQualifier	wd:Worker_Data// wd:Preferred_Name_Data// wd:Social_Suffix_Reference/ @wd:Descriptor	Description of social suffix of the preferred name for a worker.
wd-PreferredLastName	wd-PreferredLastName	wd:Worker_Data// wd:Preferred_Name_Data// wd:Last_Name	Last name of the preferred name of the worker.
wd-PreferredMiddleName	wd-PreferredMiddleName	wd:Worker_Data// wd:Preferred_Name_Data// wd:Middle_Name	Middle name of the preferred name of the worker.
wd-PreferredPersonalTitle	wd-PreferredPersonalTitle	wd:Worker_Data// wd:Preferred_Name_Data// wd:Title_Reference/@wd:Descriptor	Description of the reference to the title of the preferred name of a worker.
wd-WID	wd-WID	wd:Worker_Reference/ wd:ID[@wd:type='WID']	Unique identifier of the worker or contingent worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-JobFamily	wd-JobFamily	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Profile_Summary_Data/ wd:Job_Family_Reference/ wd:ID[@wd:type='Job_Family_ID'] ||| wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Profile_Summary_Data/ wd:Job_Family_Reference/ @wd:Descriptor	Unique identifier and description for the job family of the worker.
wd-WorkerIDType	wd-WorkerIDType	wd:Worker_Data/wd:Worker_ID/ wd:Worker_Reference[wd:ID[@wd:type ='Employee_ID']]	Employee ID / Contingent Worker ID.
wd-BusinessAddressLine2	wd-BusinessAddressLine2	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_2']	Address line 2 of the primary address of the worker.
wd-BusinessAddressLine3	wd-BusinessAddressLine3	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_3']	Address line 3 of the primary address of the worker.
wd-InternetEMailAddress	InternetEMailAddress	wd:Worker_Data/wd:Personal_Data// wd:Email_Address_Data/ wd:Email_Address	Work email address of the worker.
wd-HomeInternetEMailAddress	homeEmailAddress	wd:Worker_Data/wd:Personal_Data// wd:Email_Address_Data/ wd:Email_Address	Home email address of the worker.
wd-UserName	wd-UserName	wd:Worker_Data//wd:User_Name	User name that is used to sign onto the Workday application.
wd-orgManagerWorkforceID	wd-orgManagerWorkforceID	wd:Worker_Data/ wd:Organization_Data// wd:Organization_Support_Role/ wd:Organization_Role_Data/ wd:Worker_Reference/ wd:ID[@wd:type='Employee_ID']	Workforce ID of the worker's manager in the Organization.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-JobClassificationSummaryData	wd-JobClassificationSummaryData	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Classification_Summary_Data/ wd:Job_Group_Reference/ wd:ID[@wd:type='Job_Classification_Group_ID'] ||| wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Classification_Summary_Data/ wd:Job_Classification_Reference/ @wd:Descriptor ||| wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Classification_Summary_Data/ wd:Job_Classification_Reference/ wd:ID[@wd:type='Job_Classification_Reference_ID']	Contains job classification from the job profile for the position and additional job classification specified for the position.
wd-Gender	wd-Gender	wd:Worker_Data/wd:Personal_Data/ wd:Gender_Reference/ wd:ID[@wd:type='Gender_Code']	Unique identifier for the gender of the worker.
wd-StandardHours	wd-StandardHours	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Scheduled_Weekly_Hours	Scheduled weekly hours for position of the primary job of the worker.
wd-FullTimeEquivalent	wd-FullTimeEquivalent	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Full_Time_Equivalent_Percentage	Full time equivalent percentage for position of the primary job of the worker.
wd-ShortCountryCode	wd-ShortCountryCode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	Two character code for the country of the primary address of the worker.
wd-C-ContractorRefID	wd-C-ContractorRefID	wd:Worker_Data// wd:Worker_Contract_Data/ wd:Supplier_Reference/ wd:ID[@wd:type='Supplier_Reference_ID']	Unique identifier of the supplier for the contingent worker.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-ShortStateCode	wd-ShortStateCode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data//wd:Address_Data/ wd:Country_Region_Reference/ wd:ID[@wd:type='ISO_3166-2_Code']	Two character code for region of the primary address of the worker.
wd-ReportingHierarchy	wd-ReportingHierarchy	wd:Worker_Data/ wd:Management_Chain_Data/ wd:Worker_Supervisory_Management_Chain_Data/ wd:Management_Chain_Data/ wd:Manager_Reference/wd:ID	Sequence of manager references of the management chain data.
wd-DepartmentHierarchy	wd-DepartmentHierarchy	wd:Worker_Data/ wd:Management_Chain_Data/ wd:Worker_Supervisory_Management_Chain_Data/ wd:Management_Chain_Data/ wd:Organization_Reference/wd:ID	Sequence of Organization IDs of the management chain data.
wd-E-HireRescinded	wd-E-HireRescinded	wd:Worker_Data// wd:Worker_Status_Data/ wd:Hire_Rescinded	Boolean attribute indicating whether the worker's hire was rescinded.
wd-E-ContinuousServiceDate	wd-E-ContinuousServiceDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Continuous_Service_Date	Continuous service date for the worker.
wd-C-ContinuousServiceDate	wd-C-ContinuousServiceDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Continuous_Service_Date	Continuous service date for the contingent worker.
wd-HostCompanyRefID	wd-HostCompanyRefID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data[wd:Position_Data[wd:International_Assignment_Data]]/ wd:Position_Organizations_Data/ wd:Position_Organization_Data/ wd:Organization_Data[wd:Organization_Type_Reference/wd:ID='COMPANY']/ wd:Organization_Reference_ID/	Unique identifier for the host Organization of the worker's international assignment job.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-HostCompanyCode	wd-HostCompanyCode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data[wd:Position_Data[wd:International_Assignment_Data]]/ wd:Position_Organizations_Data/ wd:Position_Organization_Data/ wd:Organization_Data[wd:Organization_ Type_Reference/wd:ID='COMPANY']/ wd:Organization_Code	Organization code for the host Organization of the worker's international assignment job.
wd-HostLocation	wd-HostLocation	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data[wd:Position_Data[wd:International_Assignment_Data]]/ wd:Position_Data/ wd:Business_Site_Summary_Data/ wd:Name	Name of the location for the host Organization of the worker's international assignment job.
wd-HostCompanyName	wd-HostCompanyName	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data[wd:Position_Data[wd:International_Assignment_Data]]/ wd:Position_Organizations_Data/ wd:Position_Organization_Data/ wd:Organization_Data[wd:Organization_ Type_Reference/wd:ID='COMPANY']/ wd:Organization_Name	Organization name for the host Organization of the worker's international assignment job.
wd-Home_Country_ISO_2	wd-Home_Country_ISO_2	wd:Worker_Data// wd:Home_Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	Two character code for the worker's home country.
wd-Host_Country_ISO_2	wd-Host_Country_ISO_2	wd:Worker_Data// wd:Host_Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	Two character code for the worker's host country for international assignments.
wd-VisaID	wd-VisaID	wd:Worker_Data//wd:Visa_ID_Data/ wd:ID	Visa identifier
wd-GovernmentID	wd-GovernmentID	wd:Worker_Data// wd:Government_ID_Data/wd:ID	Government ID
wd-PassportID	wd-PassportID	wd:Worker_Data// wd:Passport_ID_Data/wd:ID	Passport ID
wd-LicenseID	wd-LicenseID	wd:Worker_Data//wd:License_ID_Data/ wd:ID	License identifier

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-DisabilityID	wd-DisabilityID	wd:Worker_Data/wd:Personal_Data/ wd:Disability_Status_Data/ wd:Disability_Reference/ wd:ID[@wd:type='Disability_ID']	Person's disability identifier.
wd-ArmedForcesID	wd-ArmedForcesID	wd:Worker_Data/wd:Personal_Data/ wd:Military_Service_Data/ wd:Status_Reference/ wd:ID[@wd:type=' Armed_Forces_Status_ID']	Person's military status identifier.
wd-WorkPrimaryDeviceType	wd-WorkPrimaryDeviceType	wd:Worker_Data/wd:Personal_Data// wd:Phone_Data/ wd:Phone_Device_Type_Reference/ wd:ID[@wd:type='Phone_Device_Type_ID']	Unique identifier for primary work phone device type.
wd-HomePrimaryDeviceType	wd-HomePrimaryDeviceType	wd:Worker_Data/wd:Personal_Data// wd:Phone_Data/ wd:Phone_Device_Type_Reference/ wd:ID[@wd:type='Phone_Device_Type_ID']	Unique identifier for primary home phone device type.
wd-WorkPrimaryPhone	wd-WorkPrimaryPhone	wd:Worker_Data/wd:Personal_Data// wd:Phone_Data/ @wd:International_Formatted_Phone	Formatted international phone number with extension of the primary work phone of the worker.
wd-HomePrimaryPhone	wd-HomePrimaryPhone	wd:Worker_Data/wd:Personal_Data// wd:Phone_Data/ @wd:International_Formatted_Phone	Formatted international phone number with extension of the primary home phone of the worker.
wd-C-ContractEndDate	wd-C-ContractEndDate	wd:Worker_Data// wd:Worker_Contract_Data/ wd:Contract_End_Date	That date the contingent worker's contract ends.
wd-Merge	wd-Merge		This attribute when set on worker object, triggers the synchronization of the workers with IDV.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-Merge-effectiveDate	wd-Merge-effectiveDate		If the wd-Merge-effectiveDate is set on the worker object class prior to setting wd-Merge to true, the query for the sync event will use the value for wd-Merge-effectiveDate as the effective date provided it is in the future.
wd-E-TerminationLastDayOfWork	wd-E-TerminationLastDayOfWork	wd:Worker_Data// wd:Worker_Status_Data/ wd:Termination_Last_Day_of_Work	Last day worked for the worker's termination event.
wd-C-TerminationLastDayOfWork	wd-C-TerminationLastDayOfWork	wd:Worker_Data// wd:Worker_Status_Data/ wd:Termination_Last_Day_of_Work	Last day worked for the contingent worker's termination event.
wd-C-SupplierID	wd-C-SupplierID	wd:Worker_Data// wd:Worker_Contract_Data/ wd:Supplier_Reference/ wd:ID[@wd:type='Supplier_ID']	Unique identifier of the supplier for the contingent worker.
wd-DeathDate	wd-DeathDate	wd:Worker_Data//wd:Date_of_Death	The worker's death date.
wd-GovernmentIDData	wd-GovernmentIDData	wd:Worker_Data// wd:Government_ID_Data/wd:ID ||| wd:Worker_Data// wd:Government_ID_Data/ wd:ID_Type_Reference/ wd:ID[@wd:type='Government_ID_Type_ID'] ||| wd:Worker_Data// wd:Government_ID_Data/ wd:Issued_Date ||| wd:Worker_Data// wd:Government_ID_Data/ wd:Expiration_Date	The worker's government identification data.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-PassportData	wd-PassportData	wd:Worker_Data// wd:Passport_ID_Data/wd:ID ||| wd:Worker_Data// wd:Passport_ID_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code'] ||| wd:Worker_Data// wd:Passport_ID_Data/ wd:Expiration_Date	The worker's passport data.
wd-LicenseIDData	wd-LicenseIDData	wd:Worker_Data//wd:License_ID_Data/ wd:ID ||| wd:Worker_Data// wd:License_ID_Data/ wd:ID_Type_Reference/ wd:ID[@wd:type='License_ID_Type_ID']	The worker's license identification data.
wd-EthnicityID	wd-EthnicityID	wd:Worker_Data/wd:Personal_Data/ wd:Ethnicity_Reference/ wd:ID[@wd:type='Ethnicity_ID']	The worker's ethnicity identifier.
wd-HispanicOrLatino	wd-HispanicOrLatino	wd:Worker_Data/wd:Personal_Data/ wd:Hispanic_or_Latino	The worker's ethnicity identifier, to identify if the worker is Hispanic or Latin.
wd-CitizenshipStatusCode	wd-CitizenshipStatusCode	wd:Worker_Data/wd:Personal_Data/ wd:Citizenship_Status_Reference/ wd:ID[@wd:type='Citizenship_Status_Code']	The worker's unique identification code for the citizenship status.
wd-CityOfBirth	wd-CityOfBirth	wd:Worker_Data/wd:Personal_Data/ wd:City_of_Birth	The worker's city of birth.
wd-CountryOfBirth	wd-CountryOfBirth	wd:Worker_Data/wd:Personal_Data/ wd:Country_of_Birth_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	The worker's country of birth.
wd-RegionOfBirth	wd-RegionOfBirth	wd:Worker_Data/wd:Personal_Data/ wd:Region_of_Birth_Reference/ wd:ID[@wd:type='ISO_3166-2_Code']	The worker's region of birth.
wd-MaritalStatusID	wd-MaritalStatusID	wd:Worker_Data/wd:Personal_Data/ wd:Marital_Status_Reference/ wd:ID[@wd:type='Marital_Status_ID']	The worker's marital status identifier.
wd-PreferredLanguage	wd-PreferredLanguage	wd:Worker_Data/ wd:User_Account_Data/ wd:User_Language__Reference/ wd:ID[@wd:type='User_Language_ID']	The worker's preferred language for communication.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-HomeAddressLine1	wd-HomeAddressLine1	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_1']	The worker's home address line 1.
wd-HomeAddressLine2	wd-HomeAddressLine2	wd:Worker_Data/wd:Personal_Data//wd:Address_Data/wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_2']	The worker's home address line 2.
wd-CertificationData	wd-CertificationData	wd:Worker_Data/wd:Qualification_Data/wd:Certification/wd:Certification_Data/wd:Certification_Name ||| wd:Worker_Data/wd:Qualification_Data/wd:Certification/wd:Certification_Data/wd:Issuer ||| wd:Worker_Data/wd:Qualification_Data/wd:Certification/wd:Certification_Data/wd:Certification_Number	The worker's certifications data.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-EmergencyContactData	wd-EmergencyContactData	wd:Worker_Data//wd:Related_Person/wd:Emergency_Contact/wd:Emergency_Contact_Data/@wd:Primary ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Name_Data/wd:Preferred_Name_Data/wd:Name_Detail_Data/@wd:Formatted_Name ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Address_Data/wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_1'] ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Address_Data/wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_2'] ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Address_Data/wd:Municipality ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Address_Data/wd:Country_Region_Reference/wd:ID[@wd:type='ISO_3166-2_Code'] ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Address_Data/wd:Postal_Code ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Address_Data/wd:Country_Reference/wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code'] ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Phone_Data/@wd:International_Formatted_Phone ||| wd:Worker_Data//wd:Related_Person/wd:Personal_Data/wd:Contact_Data/wd:Phone_Data/@wd:International_Formatted_Phone	The worker's emergency contact data.
wd-E-TerminationRegrettable	wd-E-TerminationRegrettable	wd:Worker_Data//wd:Worker_Status_Data/wd:Regrettable_Termination	Indicates if the worker's termination was regrettable.
wd-E-TerminationInvoluntry	wd-E-TerminationInvoluntry	wd:Worker_Data//wd:Worker_Status_Data/wd:Termination_Involuntary	Indicates if the worker's termination was involuntary.

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-E-SeniorityDate	wd-E-SeniorityDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Seniority_Date	Indicates the date when the worker attained seniority in the Organization.
wd-C-TerminationRegrettable	wd-C-TerminationRegrettable	wd:Worker_Data// wd:Worker_Status_Data/ wd:Regrettable_Termination	Indicates if the contingent worker's termination was regrettable.
wd-C-TerminationInvoluntry	wd-C-TerminationInvoluntry	wd:Worker_Data// wd:Worker_Status_Data/ wd:Termination_Involuntary	Indicates if the contingent worker's termination was involuntary.
wd-C-SeniorityDate	wd-C-SeniorityDate	wd:Worker_Data// wd:Worker_Status_Data/ wd:Seniority_Date	Indicates the date when the contingent worker attained seniority in the Organization.

Job Profile Attributes

The below table lists OpenText Identity Vault Job Profile attributes. These attributes can be retrieved using the `Get_Job_Profiles` API.

Table 5-2 *Mapping of Job Profile Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-CompensationGradeID	wd-CompensationGradeID	wd:Job_Profile_Data/ wd:Job_Profile_Compensation_Data/ wd:Compensation_Grade_Reference/ wd:ID[@wd:type='Compensation_Grade_ID']	Unique identifier for the compensation grade
wd-CompensationGradeProfileID	wd-CompensationGradeProfileID	wd:Job_Profile_Data/ wd:Job_Profile_Compensation_Data/ wd:Compensation_Grade_Profile_Reference/ wd:ID[@wd:type='Compensation_Grade_Profile_ID']	Unique identifier for the compensation grade profile
wd-Inactive	wd-Inactive	wd:Job_Profile_Basic_Data/ wd:Inactive	Boolean attribute indicating if job profile is inactive

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-JobCode	wd-JobCode	wd:Job_Profile_Data/ wd:Job_Code	Job code for the job profile
wd-JobFamilyID	wd-JobFamilyID	wd:Job_Profile_Basic_Data/ wd:Job_Family_Data/ wd:Job_Family_Reference/ wd:ID[@wd:type='Job_Family_ID']	Unique identifier for the job family of the job profile.
wd-JobProfileID	wd-JobProfileID	wd:Job_Profile_Data/ wd:Job_Code	Job code for the job profile.
wd-JobTitle	wd-JobTitle	wd:Job_Profile_Basic_Data/ wd:Job_Title	Name of the job profile.
wd-WID	wd-WID	wd:Job_Profile_Reference/ wd:ID[@wd:type='WID']	Unique identifier for the job profile.
wd-UnionRequired	wd-UnionRequired	wd:Unions_Data/ wd:Requirement_Option_Reference/ wd:ID[@wd:type='Requirement_ID']	Indicates if the worker should be assigned to a union.

Job Family Attributes

The below table lists OpenText Identity Vault Job Family attributes. These attributes can be retrieved using the `Get_Job_Families` API

Table 5-3 *Mapping of Job Family Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-Inactive	wd-Inactive	wd:Job_Family_Data/wd:Inactive	Boolean attribute indicating if job family is inactive
wd-JobFamilyID	wd-JobFamilyID	wd:Job_Family_Data/wd:ID	Unique identifier for the job family
wd-JobFamilyName	wd-JobFamilyName	wd:Job_Family_Data/wd:Name	Text attribute identifying job family name
wd-WID	wd-WID	wd:Job_Family_Reference/ wd:ID[@wd:type='WID']	Unique identifier for the job family

Relation Attributes

The below table lists OpenText Identity Vault Relation attributes. These attributes can be retrieved using the `Get_Workers` API.

Table 5-4 Mapping of Relation Attributes

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-ManagerWorkforceID	managerWorkforceID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Organizations_Data// wd:Organization_Support_Role/ wd:Organization_Role_Data/ wd:Worker_Reference/ wd:ID[@wd:type='Employee_ID' or @wd:type='Contingent_Worker_ID']	Workforce ID of the manager of the position
wd-BusinessTitle	Title	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Business_Title	Business title for the position
wd-BusinessAddressLine1	wd-BusinessAddressLine1	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Address_Line_Data	Address line 1 of the position
wd-BusinessCity	wd-BusinessCity	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Municipality	City part of address of the position
wd-BusinessCountry	wd-BusinessCountry	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Country_Reference/ @wd:Descriptor	Description for the country of the position
wd-BusinessPostalcode	wd-BusinessPostalcode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Postal_Code	Postal code part of the address of the position

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-BusinessState	wd-BusinessState	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Country_Region_Reference/@wd:Descriptor	Region part of the address for the position. For example, state/ province information
wd-JobClassificationType	wd-JobClassificationType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Job_Classification_Reference/@wd:Descriptor	Display information for the job classification of the position
wd-JobGroupReference	wd-JobGroupReference	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Job_Group_Reference/ @wd:Descriptor	Display information for the job group of the position
wd-JobProfileName	wd-JobProfileName	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Job_Profile_Name	The name of the job profile
wd-JobProfileID	jobCode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Job_Profile_Reference/ wd:ID[@wd:type='Job_Profile_ID']	Unique identifier for the job profile
wd-LocationID	wd-LocationID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Location_Reference/ wd:ID[@wd:type='Location_ID']	Unique identifier for the location of the position
wd-LocationName	L	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Business_Site_Summary_Data/wd:Name	Name of the location for the position

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-PayRateType	wd-PayRateType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Pay_Rate_Type_Reference/@wd:Descriptor	Text attribute identifying the pay rate type for the position
wd-PositionEffectiveDate	wd-PositionEffectiveDate	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ @wd:Effective_Date	Effective date for position
wd-PositionEndDate	wd-PositionEndDate	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:End_Date	The effective date of the end employment business process
wd-PositionID	wd-PositionID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Position_ID	Text attribute identifying the position
wd-PositionstartDate	wd-PositionstartDate	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Start_Date	Date the worker first started working in this position
wd-PositionTimeType	wd-PositionTimeType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Position_Time_Type_Reference/@wd:Descriptor	Text attribute identifying the position time type
wd-PositionTitle	wd-PositionTitle	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Position_Title	Text attribute identifying the position title
wd-RelationID	wd-RelationID	wd:Worker_Data/ wd:Worker_ID-C/ E:wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Position_ID	Unique identifier for the relation based on the worker ID, worker type and position ID

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-WorkerIDType	wd-WorkerIDType	wd:Worker_Data/ wd:Worker_ID/ wd:Worker_Reference[wd:ID[@wd:type='Employee_ID']]	Employee ID / Contingent Worker ID
wd-WorkerType	wd-WorkerType	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Worker_Type_Reference/ @wd:Descriptor	The worker type for the position
workforceID	workforceID	wd:Worker_Data/ wd:Worker_ID	The ID for the worker or the contingent worker
wd-WID	wd-WID	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Position_Reference/ wd:ID[@wd:type='WID']	Unique identifier for the position of the worker
wd-Primary	wd-Primary	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ @wd:Primary_Job	Boolean attribute indicates whether the position is the primary position for a worker
wd-ShortCountryCode	wd-ShortCountryCode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	To character code representing country of the position

Attribute Name	NDS-Name	Workday SOAP Response Path	Description
wd-JobClassificationSummaryData	wd-JobClassificationSummaryData	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Classification_Summary_Data/ wd:Job_Group_Reference/ wd:ID[@wd:type='Job_Classification_Group_ID'] ||| wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Classification_Summary_Data/ wd:Job_Classification_Reference/@wd:Descriptor ||| wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Job_Classification_Summary_Data/ wd:Job_Classification_Reference/ wd:ID[@wd:type='Job_Classification_Reference_ID']	Contains job classification from the job profile for the position and additional job classification specified for the position
wd-StandardHours	wd-StandardHours	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Scheduled_Weekly_Hours	Scheduled weekly hours for the position
wd-FullTimeEquivalent	wd-FullTimeEquivalent	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data/ wd:Full_Time_Equivalent_Percentage	Full time equivalent percentage for position
wd-ShortStateCode	wd-ShortStateCode	wd:Worker_Data/ wd:Employment_Data/ wd:Worker_Job_Data/ wd:Position_Data// wd:Address_Data/ wd:Country_Region_Reference/ wd:ID[@wd:type='ISO_3166-2_Code']	Two character code representing region part of the address

Photo Attributes

The below table lists OpenText Identity Vault Photo attributes. These attributes can be retrieved using the `Get_Worker_Photos` API.

Table 5-5 *Mapping of Photo Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-photo	wd-photo	wd:Worker_Photo_Data/ wd:File	Photo file content in `base64Binary` format
wd-photoFilename	wd-photoFilename	wd:Worker_Photo_Data/ wd:Filename	File name of the photo
wd-photoID	wd-photoID	For worker photo: wd:Worker_Photo_Reference/ wd:ID[@wd:type='Employee_ID'] -E For contingent worker's photo: wd:Worker_Photo_Reference/ wd:ID[@wd:type='Contingent_Worker_ID'] -C	For worker, the SOAP response path is appended with `-E` option and for contingent workers, the SOAP response path is appended with `-C` option.
wd-originalPhoto	wd-originalPhoto	wd:Worker_Photo_Data/ wd:File	This attribute holds the data and specifications of the worker's original photo.

Location Attributes

The below table lists OpenText Identity Vault Location attributes. These attributes can be retrieved using the `Get_Locations` API.

Table 5-6 *Mapping of Location Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-AddressType	wd-AddressType	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Usage_Data/ wd:Type_Data/ wd:Type_Reference/ wd:ID[@wd:type='Communication_Usage_Type_ID']	Communication usage type ID for the address information of the location
wd-BusinessAddressLine1	wd-BusinessAddressLine1	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_1']	Address line 1 of the address of the location

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-BusinessCity	wd-BusinessCity	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Municipality	City part of the address of the location
wd-BusinessCountry	wd-BusinessCountry	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Country_Reference/ @wd:Descriptor	Text description of the country for the address of the location
wd-BusinessPostalcode	wd-BusinessPostalcode	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Postal_Code	The postal code part of the address of the location
wd-BusinessState	wd-BusinessState	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Country_Region_Descriptor	The region part of the address of the location. For example, street name, street number, suite number, apartment etc.
wd-Inactive	wd-Inactive	wd:Location_Data/ wd:Inactive	Boolean attribute indicates if the location is inactive
wd-LocationID	wd-LocationID	wd:Location_Data/ wd:Location_ID	The unique location ID
wd-LocationName	wd-LocationName	wd:Location_Data/ wd:Location_Name	Name of the location
wd-LocationUsageID	wd-LocationUsageID	wd:Location_Data/ wd:Location_Usage_Reference/ wd:ID[@wd:type='Location_Usage_ID']	Unique identifier for the usage of the location. For example, business site, workspace etc.
wd-BusinessAddressLine2	wd-BusinessAddressLine2	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_2']	Address line 2 of the address of the location
wd-BusinessAddressLine3	wd-BusinessAddressLine3	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Address_Line_Data[@wd:Type='ADDRESS_LINE_3']	Address line 3 of the address of the location

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-StateCode	wd-StateCode	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Country_Region_Reference/ wd:ID[@wd:type='ISO_3166-2_Code']	Unique identifier for the region part of the address of the location. For example, state/ province information
wd-ShortCountryCode	wd-ShortCountryCode	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-2_Code']	Two digit code representing the country
wd-LongCountryCode	wd-LongCountryCode	wd:Location_Data/ wd:Contact_Data/ wd:Address_Data/ wd:Country_Reference/ wd:ID[@wd:type='ISO_3166-1_Alpha-3_Code']	Three digit code representing the country
wd-WID	wd-WID	wd:Location_Reference/ wd:ID[@wd:type='WID']	Unique identifier for the location

Organization Attributes

The below table lists OpenText Identity Vault Organization attributes. These attributes can be retrieved using the `Get_Organizations` API.

Table 5-7 *Mapping of Organization Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-Inactive	wd-Inactive	wd:Organization_Data/ wd:Inactive	Inactive indicator for the Organization
wd-OrganizationCode	wd-OrganizationCode	wd:Organization_Data/ wd:Organization_Code	Text attribute identifying Organization code
wd-OrganizationID	wd-OrganizationID	wd:Organization_Data/ wd:Reference_ID	Integration reference ID used for integration purposes
wd-OrganizationName	wd-OrganizationName	wd:Organization_Data/ wd:Name	Name of the Organization that appears on page and reports
wd-OrganizationSubType	wd-OrganizationSubType	wd:Organization_Data/ wd:Organization_Subtype_Reference/ wd:ID[@wd:type='Organization_Subtype_ID']	Unique identifier for the sub-type of the Organization

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-OrganizationType	wd-OrganizationType	wd:Organization_Data/ wd:Organization_Type_Reference/ wd:ID[@wd:type='Organization_Type_ID']	Unique identifier for the type of the Organization
wd-SuperiorOrganizationID	wd-SuperiorOrganizationID	wd:Organization_Data/ wd:Hierarchy_Data/ wd:Superior_Organization_Reference/ wd:ID[@wd:type='Organization_Reference_ID']	Unique identifier for the immediately superior Organization to the Organization
wd-ManagerEID	wd-ManagerEID	wd:Organization_Data/ wd:Manager_Reference/ wd:ID[@wd:type='Employee_ID']	Unique identifier for the manager of teh Organization
wd-OrganizationLevel	wd-OrganizationLevel	wd:Organization_Data/ wd:Supervisory_Data/ wd:Staffing_Restrictions_Data / wd:Job_Profile_Restriction_Summary_Data/ wd:Management_Level_Reference/@Descriptor	Display information used to describe management level
wd-WID	wd-WID	wd:Organization_Reference/ wd:ID[@wd:type='WID']	Unique identifier to reference an Organization

Delta Object Attributes

The below table lists OpenText Identity Vault Delta Object attributes.

Table 5-8 *Mapping of Delta Object Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-WDAssociation	wd-WDAssociation	NA	The associated value of the delta object
wd-IDVAssociation	wd-IDVAssociation	NA	The DN of the original object
wd-EffectiveDate	wd-EffectiveDate	NA	The event effective date
wd-creationDate	wd-creationDate	NA	The event creation date
wd-ChangelogValue	wd-ChangelogValue	NA	The differential information (in Base64 encrypted format) of the worker's state, between creation and effective date.

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-WorkerLogEntries	wd-WorkerLogEntries	NA	The transaction events of the worker
wd-ProcessingStatus	wd-ProcessingStatus	NA	The current status of the delta object. The values are as follows: ◆ 1: indicates a processed delta object ◆ 0: indicates an unprocessed delta object

Future Object Attributes

The below table lists OpenText Identity Vault Future Object attribute.

Table 5-9 *Mapping of Delta Object Attributes*

Attribute Name	nds-name	Workday SOAP Response Path	Description
wd-DeltaObject	wd-DeltaObject	NA	The destination details of delta object

Attribute Filter Settings

This section provides information about the OpenText Identity Vault Filter settings in the driver. The Workday will be the authoritative source for all the above attributes except user ID and Email address which are generated in the OpenText Identity Manager.

- "Users Attributes Filter Settings" on page 93
- "Relations Attributes Filter Settings" on page 94
- "Job Family Attributes Filter Settings" on page 95
- "Location Attributes Filter Settings" on page 96
- "Photo Attributes Filter Settings" on page 96
- "Job Profile Attributes Filter Settings" on page 97
- "Organization Attributes Filter Settings" on page 97
- "Delta Object Attributes Filter Settings" on page 98

Users Attributes Filter Settings

The following table provides information about the Users Attributes Filter settings:

Table 5-10 Users Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
Given Name	app	sync	reset
managerWorkforceID	app	sync	reset
Surname	app	sync	reset
wd-WorkerIDType	app	sync	reset
workforceID	app	sync	reset
wd-WID	app	sync	reset
wd-UserName	app	sync	reset
wd-E-Active	app	sync	reset
wd-C-Active	app	sync	reset
wd-HomePrimaryPhone	app	sync	sync
wd-HomePrimaryDeviceType	app	sync	sync
wd-WorkPrimaryPhone	app	sync	sync
wd-WorkPrimaryDeviceType	app	sync	sync
Internet Email Address	app	sync	sync
homeEmailAddress	app	sync	sync
DirXML-EntitlementRef	OpenText Identity Vault	Ignore	Notify

NOTE: Only the following attributes can be updated on Subscriber channel:

- User ID
- Phone Numbers
- E-Mail Address
- Custom IDs

Modifying any other attribute on the Subscriber channel is not supported. For more information on the procedure to update phone numbers, see "Configuring Phone Contact Information in Subscriber Channel" on page 38.

Relations Attributes Filter Settings

The following table provides information about the Relations Attributes Filter settings:

Table 5-11 Relations Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
L	app	sync	reset
managerWorkforceID	app	sync	reset
Title	app	sync	reset
wd-BusinessAddressLine1	app	sync	reset
wd-BusinessCity	app	sync	reset
wd-BusinessCountry	app	sync	reset
wd-BusinessPostalcode	app	sync	reset
wd-BusinessState	app	sync	reset
wd-FullTimeEquivalent	app	sync	reset
wd-JobClassificationType	app	sync	reset
wd-JobFamilyID	app	sync	reset
wd-JobGroupReference	app	sync	reset
wd-JobProfileName	app	sync	reset
wd-LocationID	app	sync	reset
wd-PayRateType	app	sync	reset
wd-PositionEffectiveDate	app	sync	reset
wd-PositionEndDate	app	sync	reset
wd-PositionID	app	sync	reset
wd-PositionstartDate	app	sync	reset
wd-PositionTimeType	app	sync	reset
wd-PositionTitle	app	sync	reset
wd-Primary	app	sync	reset
wd-RelationID	app	sync	reset
wd-WorkerIDType	app	sync	reset
wd-WorkerType	app	sync	reset
workforceID	app	sync	reset
wd-WID	app	sync	reset

Job Family Attributes Filter Settings

The following table provides information about the Job Family Attributes Filter settings:

Table 5-12 Job Family Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
wd-Inactive	app	sync	reset
wd-JobFamilyID	app	sync	reset
wd-JobFamilyName	app	sync	reset
wd-WID	app	sync	reset

Location Attributes Filter Settings

The following table provides information about the Location Attributes Filter settings:

Table 5-13 Location Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
wd-AddressType	app	sync	reset
wd-BusinessAddressLine1	app	sync	reset
wd-BusinessCity	app	sync	reset
wd-BusinessCountry	app	sync	reset
wd-BusinessPostalcode	app	sync	reset
wd-BusinessState	app	sync	reset
wd-Inactive	app	sync	reset
wd-LocationID	app	sync	reset
wd-LocationName	app	sync	reset
wd-LocationUsageID	app	sync	reset
wd-BusinessAddressLine2	app	sync	reset
wd-BusinessAddressLine3	app	sync	reset
wd-StateCode	app	sync	reset
wd-ShortCountryCode	app	sync	reset
wd-LongCountryCode	app	sync	reset
wd-WID	app	sync	reset

Photo Attributes Filter Settings

The following table provides information about the Photo Attributes Filter settings:

Table 5-14 Photo Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
photo	app	sync	sync
wd-photoFilename	app	sync	sync
wd-photoID	app	sync	sync
wd-originalPhoto	app	sync	Ignore

Job Profile Attributes Filter Settings

The following table provides information about the Job Profile Attributes Filter settings:

Table 5-15 Job Profile Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
wd-CompensationGradeID	app	sync	reset
wd-CompensationGradeProfileID	app	sync	reset
wd-Inactive	app	sync	reset
wd-JobCode	app	sync	reset
wd-JobFamilyID	app	sync	reset
wd-JobTitle	app	sync	reset
wd-WID	app	sync	reset

Organization Attributes Filter Settings

The following table provides information about the Organization Attributes Filter settings:

Table 5-16 Organization Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
wd-Inactive	app	sync	reset
wd-OrganizationCode	app	sync	reset
wd-OrganizationID	app	sync	reset
wd-OrganizationName	app	sync	reset
wd-OrganizationSubType	app	sync	reset
wd-OrganizationType	app	sync	reset
wd-SuperiorOrganizationID	app	sync	reset
wd-ManagerEID	app	sync	reset
wd-OrganizationLevel	app	sync	reset
wd-WID	app	sync	reset

Delta Object Attributes Filter Settings

The following table provides information about the Organization Attributes Filter settings:

Table 5-17 Delta Object Attributes Filter Settings

Attribute Name	Merge Authority	Publisher	Subscriber
CN	Default	Ignore	Notify
wd-ProcessedEntries	Default	Ignore	Notify
wd-ProcessingStatus	Default	Ignore	Notify

6 Upgrading an Existing Driver

For upgrading the driver packages, see the respective readme. Before upgrading your driver packages, ensure to perform the following actions:

- Backup the driver configuration by using the designer driver export.
- Backup the driver customizations such as driver configuration, GCVs, policies etc. from designer for post package upgrade and review.
- Follow package upgrade instructions and review GCVs / driver configuration settings in designer prior to reconcile to the OpenText Identity Vault.

7 Securing Communication

If the remote Web service you are accessing allows HTTPS connections, you can configure the driver to take advantage of this increased security.

IMPORTANT: Only certificates from Java keystore are accepted. So, make sure that the keystore of the certificates is a Java keystore and the Workday tenant SSL certificate is available in the Java keystore.

The communication between the OpenText Identity vault and Workday is established through the Subscriber and Publisher channels. To establish a secure communication between them, you need a trust store containing a certificate issued by the certificate authority that signed the server's certificate.

It is recommended to import the chain of CA certificates from issuer's trusted URL. Workday instance uses certificates from a trusted CA. For example, if Workday is using the certificates from Digicert (trusted CA), you must verify the authenticity of the certificates from the Digicert's website before downloading and importing them into the Java Keystore.

For testing purposes, you can download the certificates from the browser directly but this is not recommended. In case you want to view and download the certificates from browser, perform the following steps:

1. Enter the Workday login URL in your browser (for example Chrome), and click **Enter**.

   NOTE: Different browsers have different procedures to view the certificates. Steps 1 to 6 considers Chrome browser as an example.

   ---

2. Click the lock icon near the browser, and select **Certificate (Valid)**. The certificate is displayed.
3. Click **Certification Path**. The Certification Path displays the hierarchical structure of the structure of all the certificates.
4. Select the root certificate (the top most parent certificate), and click **View Certificate**. The root certificate is displayed.
5. To save the certificate to your system, click **Details > Copy to File > Next > Next**.
6. Enter a filename for the certificate and save it to a location as required.
7. Import the saved Workday tenant root CA certificate into your trust store or create a new trust store by entering the following command at the command prompt:

   ```
   keytool -import -file name_of_cert_file -trustcacerts -noprompt -
   keystore filename -storepass password
   ```

   For example:

   ```
   keytool -import -file tree_ca_root.b64 -trustcacerts -noprompt -
   keystore dirxml.keystore -storepass novell
   ```

8 Configure the driver to use the trust store you created in Step 1:

   8a In OpenText Identity Console, click the **IDM Administration** tile.

   8b On the Driver Dashboard, click the driver's icon.

   8c Select the **Configuration** tab, then expand the **Driver Parameters** section.

   8d On the **Driver Settings** tab, in **SSL Parameters** > **Workday Keystore file** setting, specify the path to the trust store you created in Step 1.

9 Save.

8 Managing the Driver

As you work with the Workday driver, there are several management tasks you might need to perform, including the following:

- Starting, stopping, and restarting the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver's health status
- Backing up the driver
- Inspecting the driver's cache files
- Viewing the driver's statistics
- Using the DirXML Command Line utility to perform management tasks through scripts Securing the driver and its information

  Because these tasks, as well as several others, are common to all OpenText Identity Manager drivers, they are included in the *OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html)*.

This chapter comprises the following sections:

- "Configuring Dynamic Attributes Using the Workday Driver" on page 103
- "Migrating Individual Object from Workday:" on page 107

Configuring Dynamic Attributes Using the Workday Driver

This section provides information on how to use various dynamic attributes provided by the Workday driver.

- "Synchronizing Custom IDs Using the Workday Driver" on page 104
- "Configuring Derived Attributes for Worker" on page 105
- "Configuring Custom Fields Using the Workday Driver" on page 106
- "Adding New Attribute to the Filter" on page 107

Synchronizing Custom IDs Using the Workday Driver

The driver can be configured to synchronize **Other IDs** displayed on the **IDs** tab of the **Personal** section in Workday. The driver and API refer to these other IDs as Custom IDs. Using the `Change_Other_IDs_Request` API, the driver can update custom ids in Workday. Perform the following steps to configure Custom IDs for the Workday driver:

1. Collect `Custom_ID_Type_ID` of the Custom IDs that need to be synchronized with OpenText Identity Manager. This information can be collected from the Workday implementation team. The IDs are case-sensitive.
2. In Designer, go to **Workday Driver Properties** > **Driver Configurations** > **Driver Parameters** > **Driver Options** > **Transformation Parameters** and specify the following:
   - **Parameter Name:** `wdCustomIdList`
   - **Parameter Value:** Multi valued list of all the IDs that are required to synchronize data between Workday and OpenText Identity Manager. For example, `CUSTOM_ID_TYPE-6-10` is a parameter value of one Custom ID in Workday driver.
3. In Designer, add a driver schema mapping for `{IDV attribute name}` to `{WD Custom Attribute}_Custom_ID` on the Workday driver. For example, `CUSTOM_ID_TYPE-6-10` to `CUSTOM_ID_TYPE-6-10_Custom_ID`.
4. If multiple Custom IDs are being updated from OpenText Identity Manager on Subscriber channel, then for each custom ID, you must create `wd-user` auxiliary class attribute in Designer or OpenText Identity Console to hold the Workday Reference ID of the Custom ID. This stores an internal Workday reference which is needed in order to update the value of the Custom ID in Workday. These auxiliary class attributes should be created as single valued and can be case ignore string. For example, `AUX_ID_WID`.
5. In schema mapping policy, add a mapping of `{WD Custom Attribute}_Custom_ID_Shared_Reference` with the attribute created in the previous step. For example, `CUSTOM_ID_TYPE-6-10_Custom_ID_Shared_Reference` to `AUX_ID_WID`.
6. Once a custom ID is specified in the `wdCustomIdList`, the driver shim will also return an attribute of the format `{WD Custom Attribute}_Description`. For example, `CUSTOM_ID_TYPE-6-10_Description`.
7. In driver filter, add an entry for the OpenText Identity Manager attribute that is mapped to `{WD Custom Attribute}_Custom_ID` so that it flows to Workday on the subscriber and/or from Workday on the publisher channels. For example, `AUX_ID`.
8. If one or more custom IDs are updated from OpenText Identity Manager, then create a driver filter entry for each newly created attribute holding the custom ID attributes so they flow back to OpenText Identity Manager on the publisher channel. For example, `AUX_ID`, `AUX_ID_WID`, `AUX_ID_Description`.
9. If one or more custom IDs are updated from OpenText Identity Manager, then create an output transformation policy to append the Shared Reference of the custom ID that is changing to the driver-operation-data. Ensure that value which is appended to the driver-operation-data is prefaced with `WD-` even though the attribute name in the data mapping does not have `WD-`.
   - Append the XML element `WD-{WD Custom Attribute}_Custom_ID_Shared_Reference` to driver-operation-data. For example, `WD-CUSTOM_ID_TYPE-6-10_Custom_ID_Shared_Reference`.
   - Append the text from the IDV source attribute. For example, `AUX_ID_WID`.

- Create a rule for each Custom ID attribute that you are sending to Workday. The policy should be added after `NETQWDDCFG-otp-user` output transformation policy. If the attribute holding the custom ID is not in the IDV, nothing will be appended to the driver-operation-data. A sample policy rule has been shown below:

```
<rule>
   <description>Add driver-operation-data element for modify AUX_ID
events</description>
    <conditions>
        <and>
            <if-op-attr name="AUX_ID_Custom_ID" op="changing"/>

            <if-operation mode="nocase" op="equal">modify</if-
operation>

            <if-class-name mode="nocase" op="equal">User</if-class-
name>
        </and>
    </conditions>
    <actions>
        <do-if>
            <arg-conditions>
                <and>
                    <if-attr name="AUX_ID_WID" op="available"/>
                </and>
            </arg-conditions>
            <arg-actions>
               <do-append-xml-element expression="driver-operation-
data" name="WD-CUSTOM_ID_TYPE-6-10_Custom_ID_Shared_Reference"/>
                <do-append-xml-text expression="driver-operation-
data/WD-CUSTOM_ID_TYPE-6-10_Custom_ID_Shared_Reference">
                    <arg-string>
                        <token-attr name="AUX_ID_WID"/>
                    </arg-string>
                </do-append-xml-text>
            </arg-actions>
            <arg-actions/>
        </do-if>
    </actions>
</rule>
```

Configuring Derived Attributes for Worker

When an organization is added to the `wdOrgList` transformation parameter of the Workday driver, the shim will automatically derive the following 3 attributes on worker object for each organization type/organization sub-type combination:

- `_RefID` is used for data in the `Organization_Reference_ID` element
- `_Name` is used for data in the `Organization_Name` element
- `_Code` is used for data in the `Organization_Code` element
- `|`*`RoleID`*`_Employee_ID`
- `|`*`RoleID`*`_Contingent_Worker_ID`

The above derived attributes should be suffixed to the `Organization-type | Organization-sub-type`. These attributes are available to be added to the schema mapping policy and to the filter. For example:

```
Supervisory|Department_RefID
Supervisory|Department_Name
Supervisory|Department_Code
```

In the above example, `Supervisory` is the Organization type and `Department` is the Organization sub-type. The values for Organization type and Organization sub-type are derived from the following SOAP API response path of `Get_Workers` API:

```
wd:Worker_Data/wd:Organization_Data//
wd:Organization_Data[wd:Organization_Type_Reference/
wd:ID[@wd:type='Organization_Type_ID'] | wd:Worker_Data/
wd:Organization_Data//wd:Organization_Data[wd:Organization_Type_Reference/
wd:ID[@wd:type='Organization_Type_ID']/wd:Organization_Subtype_Reference/
wd:ID[@wd:type='Organization_Subtype_ID']
```

`RoleID` is derived from the following SOAP API response path of `Get_Workers` API:

```
wd:Worker_Data/wd:Organization_Data//
wd:Organization_Data[wd:Organization_Type_Reference/
wd:iD[@wd:type='Organization_Type_ID']/wd:Organization_Support_Role_Data/
wd:Organization_Support_Role/wd:Organization_Role_Reference/
wd:iD[@wd:type='Organization_Role_ID']
```

Configuring Custom Fields Using the Workday Driver

Worker has a functionality where a Workday implementer can create override fields that have data calculated from the data in one or more other data fields. This option can be used to get data that the current driver shim does not return. For instance, one customer wanted to synchronize the pay grade which is a part of the compensation data returned if `Include_Compensation` were included in the response group data. However, since `Include_Compensation` includes all the salary information, the shim was purposely designed to not return that data. To get the pay grade for this customer, the Workday implementers set up an override field for just the pay grade.

In order to use Workday override calculated field information from the Worker data in the driver, you must take the following steps:

1. Obtain the Integration System ID from the Workday team. for more information, see "Obtaining Integration System ID from Workday" on page 107.
2. Place the Integration System ID in the Publisher driver parameters in the **Field and Parameter Criteria Provider Reference List** of the corresponding class.
3. Obtain the override field name.
4. Once you have obtained the name of the override fields, the field names must be placed in the `wdIntFieldAttrList` transformation parameters.
5. The override fields are now available for data mapping or use in policies by appending _CF to the override field name. For example, the override field returned by the shim will be `mailStop_CF` for `mailstop` override field. This will be the application attribute name used in the schema mapping or input and output transformation policies.

Obtaining Integration System ID from Workday

Perform the following steps to obtain the integration system ID from Workday:

1. Search for the **View Integration Field Override Service** in Workday.
2. Select your **Integration Field Override Service** and click **OK**.
3. View your **Integration Field** and click on the value of **Integration System**. For example, **WorkDay_Driver**.
4. Collect the value from **System ID** field. Sample value for my tenant System ID is **WorkDay_Driver/WorkDay_Driver/StartHere**.
5. Place the Integration System ID in the Publisher driver parameters in the **Field and Parameter Criteria Provider Reference List** of the corresponding class

Adding New Attribute to the Filter

If you want the additional attributes (newly added to filter) to be populated on workers who are already migrated to the OpenText Identity Vault, You can perform one of the following steps:

- Migrate the workers again through the driver which in turn clears the cache.
- No changes required. The attributes will flow automatically as part of regular polling by the driver, provided that there is a change on that worker in Workday.

Another sync option is to set the `wd-Merge` attribute to **True** on the workers to sync. If the `wd-Merge-effectiveDate` is set before setting the `wd-Merge` to **True**, the sync will be according to the `wd-Merge-effectiveDate` if that is in the future. Otherwise, the current date will be used as the effective date for the sync. However, for this option to work correctly, all workers that have the `wd-Merge` attribute set to true, must already have the correct value for the `wd-WorkerIDType` (Employee_ID or Contingent_Worker_ID).

If the worker has not been modified in Workday, the driver will never get that user data during polling and the new attributes will not be populated for that worker in OpenText Identity Vault. However, there will be delay in populating additional attributes until that worker gets modified in Workday.

Migrating Individual Object from Workday:

You may migrate individual user object either employee or Contingent worker with relation (Position). You may also migrate other object such as job profile, job family, location, organization including cost center, company, HR Company etc. To migrate these objects, use OpenText Identity Console's **Migrate into IDVault** option. This process will update the IDV object with latest information.

- **Migrating users with relation:**

1. In OpenText Identity Console, click the IDM Administration tile. "**Migrate into IDVault**" option to migrate individual employee or contingent worker.
2. On the Driver Dashboard, locate the driver and click the driver icon.
3. Click the **Data Transformation and Synchronization** tab.
4. Click the **Migrate into IDVault** tab.

5 Click **Edit migration criteria**.

6 Select user as class and select **Show all attributes from all classes**.

7 Select **workforce ID** and **wd-workerIDType** and provide the required data.

wd-workerIDType can be `Employee_ID` or `Contingent_Worker_ID`. Workforce ID is the unique ID in Workday.

8 Select **OK**.

- **Migrating job profile:**

1 In OpenText Identity Console navigate to the **Migrate into IDVault** tab to migrate job profile.

2 Select **Edit migration criteria**.

3 Select **wd-Jobprofile** as class and select **Show all attributes from all classes**.

NOTE: You have to first add wd-Jobprofile in the filter list.

---

4 Select **wd-JobProfileID** and provide the ID of the object to migrate.

- **Migrating job family:**

1 In OpenText Identity Console navigate to the **Migrate into IDVault** tab to migrate job family.

2 Select **Edit migration criteria**.

3 Select **wd-Jobfamily** as class and select **Show all attributes from all classes**.

4 Select **wd-JobFamilyID** and provide the ID of the object to migrate.

- **Migrating Location:**

1 In OpenText Identity Console navigate to the **Migrate into IDVault** tab to migrate location.

2 Select **Edit migration criteria**.

3 Select **wd-Location** as class and select **Show all attributes from all classes**.

4 Select **wd-LocationID** and provide the ID of the object to migrate.

- **Migrating Organization – Cost center, company, HR company:**

1 In OpenText Identity Console navigate to the **Migrate into IDVault** tab to migrate organization.

2 Select **Edit migration criteria**.

3 Select **wd-organization** as class and select **Show all attributes from all classes**.

4 Select **wd-OrganizationID** and provide the ID of the object to migrate.

- **Migrating user's photo:**

1 In OpenText Identity Console navigate to the **Migrate into IDVault** tab to migrate photo.

2 Select **Edit migration criteria**.

3 Select **wd-Photo** as class and select **Show all attributes from all classes**.

4 Select **wd-photoID** and provide the ID of the object to migrate.

9 Troubleshooting the Driver

You can log OpenText Identity Manager Events by setting the log level 5 at the driver and also at remote loader. Using this log in combination provides you with tracking control at a very granular level.

This section contains the following information on error messages:

- "Driver Cache File and Driver State Settings" on page 109
- "Troubleshooting Driver Processes" on page 110
- "Workday Driver Fails to Start After Changing the Cache Directory" on page 110
- "Workday Driver Displays an Error Message While Shutting Down" on page 110
- "Workday Driver Displays an Error Message When Workday Is Under Maintenance" on page 111
- "Setting Paging Size Parameter Value" on page 111
- "Error Message for Incorrectly Configuring OAuth 2.0" on page 112
- "Error Displayed for Assigning Roles and User-Based Security Groups" on page 112
- "Issue in Migrating Delta and Future Object for Business Process" on page 114
- "FutureObjectCreation and CleanDeltaAndFutureObject Job Might Fail to Execute Successfully" on page 114
- "Issue in Performing CPRS Computation for User-Based Security Group" on page 114
- "Query-Ex fails to Fetch the Next Set of Pages Due to Expiry of the Token" on page 114

Driver Cache File and Driver State Settings

As part of cache management, the shim develops sets of cache files. It compares the attribute value after conversion from SOAP response and passes only changed value to the driver channel. The driver will develop these cache files either on the server where OpenText Identity Manager engine is running (in Native setup) or at the remote loader server where the driver shim is running. You may delete these files in the case of corruption (If any). After deletion, the driver will redevelop these files but the driver will lose all caches and has to build up cache again from the beginning.

The Driver state setting file keeps the information about the last poll timestamp and also the pages fetched in the initial migration. A sample state file is shown below:

```
state__u5fTREEu5fSLES_system_driverset1_workdayu20v25.xml
```

Troubleshooting Driver Processes

Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DSTrace. You should only use it during testing and troubleshooting the driver. Running DSTrace while the drivers are in production increases the utilization on the OpenText Identity Manager server and can cause events to process very slowly. For more information, see Viewing Identity Manager Processes in the NetIQ Identity Manager Driver Administration Guide.

Workday Driver Fails to Start After Changing the Cache Directory

Workday driver fails to start after changing the cache directory in the driver configuration parameters or cache files are removed from the existing location.

You can start the driver after some time of changing the cache directory.

Workday Driver Displays an Error Message While Shutting Down

Workday driver displays the following error message if a shutdown command is executed during the polling time or during the migration process when the cache update is in progress.

```
<output>
                <status
level="error">com.netiq.dirxml.driver.workday.exception.PollException:
Error
encountered while polling JobProfile
        at
com.netiq.dirxml.driver.workday.WDPublisherHandler.pollJobProfile(WDPublis
herHandler.java:533)
        at
com.netiq.dirxml.driver.workday.WDPublisherHandler.doPoll(WDPublisherHandl
er.java:256)
        at
com.netiq.dirxml.driver.workday.WDPublisherShim.doPoll(WDPublisherShim.jav
a:140)
        at
com.netiq.dirxml.driver.workday.WDPublisherShim.start(WDPublisherShim.java
:116)
        at com.novell.nds.dirxml.remote.loader.Driver.run(Driver.java:899)
        at java.lang.Thread.run(Thread.java:748)
```

```
Caused by: com.netiq.dirxml.driver.workday.exception.CacheException: Cache
shutdown in progress...
        at
com.netiq.dirxml.driver.workday.cache.MapDbCache.checkShutdown(MapDbCache.
java:250)
        at
com.netiq.dirxml.driver.workday.cache.CacheManager.openJobProfileCacheFile
s(CacheManager.java:735)
        at
com.netiq.dirxml.driver.workday.WDPublisherHandler.pollJobProfile(WDPublis
herHandler.java:505)
        ... 5 more
</status>
        </output>
```

There is no workaround at this moment but you can ignore the error message and wait for the driver to shut down.

Workday Driver Displays an Error Message When Workday Is Under Maintenance

Workday driver displays the following error message when the driver requests for time stamp from the Workday server and if the server is under maintenance.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/"><SOAP-ENV:Body><SOAP-ENV:Fault xmlns:SOAP-ENV="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:wd="urn:com.workday/
bsvc"><faultcode>SOAP-ENV:Client.validationError</
faultcode><faultstring>Invalid request</faultstring></SOAP-ENV:Fault></
SOAP-ENV:Body></SOAP-ENV:Envelope>
```

This is assumed that the Workday server is under maintenance when you receive this error while requesting for time stamp.

Setting Paging Size Parameter Value

If Workday data is fetched using SOAP APIs such as Get_Workers, Get_JobProfiles etc., the data is fetched in chunks. The driver fetches data from Workday in pages to limit the number of records to be processed at a time. The Workday driver fetches one page at a time, processes and stores it in the cache, updates the driver storage, and then requests for the next page. The Paging Size value which is already set is directly proportional to the Java heap that is available to the driver, as it uses the heap for local storage of the data records returned by Workday until it is processed and stored in the driver cache as part of every poll request.

The Paging Size value is used for both the recurring and scheduled polling cycle and should be set as per the Java heap available to the driver.

If a smaller value is set as the page size, multiple pages with smaller chunks of data are processed as part of poll and the frequency of the driver processing and storing the data to the driver cache would increase.

You must also check and set the remote loader configuration for Java heap size and number of instances of remote loader running, as these parameter values would be limiting the actual virtual memory available for the Java heap.

Error Message for Incorrectly Configuring OAuth 2.0

If any parameters are missed while configuring the Workday driver with OAuth 2.0 authentication method, the connection terminates and displays an error.

For example, if the **Client ID** parameter value is missing the following error message appears:

```
DirXML Log Event -------------------
     Driver:   \SLES12SP3_MAHESH_1302_TREE\system\driverset1\Workday
     Channel:  Subscriber
     Status:   Fatal
     Message:  Code(-9005) The driver returned a "fatal" status indicating
that the driver should be shut down. Detail from driver: Error occured in
subscriber init. Reason :
com.netiq.dirxml.driver.workday.exception.StatusException: Error: Client
Id cannot be empty.
        at
com.netiq.dirxml.driver.workday.soap.conn.OAuthConnection.handleEmptyOAuth
Param(OAuthConnection.java:120)
        at
com.netiq.dirxml.driver.workday.soap.conn.OAuthConnection.&lt;init>(OAuthC
onnection.java:54)
        at
com.netiq.dirxml.driver.workday.soap.conn.AuthConnFactory.getAuthConnInsta
nce(AuthConnFactory.java:63)
        at
com.netiq.dirxml.driver.workday.WDSubscriberShim.init(WDSubscriberShim.jav
a:120)
        at
com.novell.nds.dirxml.engine.Subscriber.init(Subscriber.java:274)
        at
com.novell.nds.dirxml.engine.Subscriber.&lt;init>(Subscriber.java:136)
        at com.novell.nds.dirxml.engine.Driver.startShim(Driver.java:1722)
        at com.novell.nds.dirxml.engine.Driver.initialize(Driver.java:329)
        at com.novell.nds.dirxml.engine.Driver.&lt;init>(Driver.java:295)
        at
com.novell.nds.dirxml.engine.DriverEntry.run(DriverEntry.java:626)
        at java.lang.Thread.run(Thread.java:748)
```

Workaround: You must stop the driver, specify the required parameter and configure the driver as explained in "Creating the Driver Object" on page 22.

Error Displayed for Assigning Roles and User-Based Security Groups

While configuring entitlements for Assignable Roles and User-Based Security Groups, the following error could appear:

```
DirXML Log Event --------------------
     Driver:   \SLES12SP3_1302_TREE\system\driverset1\Workday
     Channel:  Subscriber
     Object:   \SLES12SP3_1302_TREE\data\workday\users\abaker001
     Status:   Error
     Message:  Subscriber execute exception :
com.netiq.dirxml.driver.workday.exception.StatusException: <?xml
version="1.0" encoding="utf-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://
schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Body><SOAP-ENV:Fault
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wd="urn:com.workday/bsvc"><faultcode>SOAP-
ENV:Client.authenticationError</faultcode><faultstring>invalid username or
password</faultstring></SOAP-ENV:Fault></SOAP-ENV:Body></SOAP-
ENV:Envelope>
        at
com.netiq.dirxml.driver.workday.soap.conn.AbsAuthConnection.sendRequest(Ab
sAuthConnection.java:365)
        at
com.netiq.dirxml.driver.workday.entitlements.EntitlementEventProcessor.pro
cess(EntitlementEventProcessor.java:72)
        at
com.netiq.dirxml.driver.workday.WDSubscriberShim.processEntitlementModific
ation(WDSubscriberShim.java:252)
        at
com.netiq.dirxml.driver.workday.WDSubscriberShim.process(WDSubscriberShim.
java:218)
        at
com.netiq.dirxml.driver.workday.WDSubscriberShim.execute(WDSubscriberShim.
java:183)
        at
com.novell.nds.dirxml.engine.Subscriber.execute(Subscriber.java:473)
        at
com.novell.nds.dirxml.engine.Subscriber.execute(Subscriber.java:304)
        at
com.novell.nds.dirxml.engine.Subscriber$ModifyProcessor.process(Subscriber
.java:1760)
        at
com.novell.nds.dirxml.engine.Subscriber.processEvent(Subscriber.java:1197)
        at
com.novell.nds.dirxml.engine.Subscriber.processEvents(Subscriber.java:1010
)
        at
com.novell.nds.dirxml.engine.Driver.submitTransaction(Driver.java:901)
        at
com.novell.nds.dirxml.engine.DriverEntry.submitTransaction(DriverEntry.jav
a:1174)
        at
com.novell.nds.dirxml.engine.DriverEntry.processCachedTransaction(DriverEn
try.java:1058)
        at
com.novell.nds.dirxml.engine.DriverEntry.eventLoop(DriverEntry.java:866)
        at
com.novell.nds.dirxml.engine.DriverEntry.run(DriverEntry.java:640)
        at java.lang.Thread.run(Thread.java:748)
```

Issue: This error indicates that the user does not have the required permissions to assign the resource.

Workaround: Provide the require permission to the user for performing the action.

Issue in Migrating Delta and Future Object for Business Process

Issue: The delta and future objects will not be migrated when the **Enable Delta and Future Object Creation** is set to **True**.

Workaround: For a successful migration, it is mandatory to set the **Migrate on Startup** option to **True**.

FutureObjectCreation and CleanDeltaAndFutureObject Job Might Fail to Execute Successfully

ssue: Occurs when a user deploys the driver with the missing **Security Equals** attribute value for FutureObjectCreation and CleanDeltaAndFutureObject jobs. The operation fails due to inadequate permissions.

Workaround: Refer to Step 7, in the section "Deploying the Driver Object" on page 42.

Issue in Performing CPRS Computation for User-Based Security Group

Workaround: Prior to using the CPRS feature, you must migrate all users from Workday to the OpenText Identity Vault atleast once post upgrading to Workday driver 1.3.

During the migration process, a cache mapping for `user-name` and `worker-id` is created. This mapping is used for displaying the assignments in CPRS. The mapping information is available in `user-name-worker-id-mapping.txt` file under cache folder.

Query-Ex fails to Fetch the Next Set of Pages Due to Expiry of the Token

Workaround: The recommended Query-Ex timeout option must be set to either 120 or more. If you specify a value less than 120, the token for Query-Ex might expire.

10 Frequently Asked Questions

This section contains information on frequently asked questions while setting up the Workday driver:

- "How to confirm successful establishment of JWT connection?" on page 115
- "Why are Delta Objects and Full State Objects created with the suffix "<Termination Date + 1>" in the termination business process?" on page 116
- "Why does same day termination also create Delta and Full state objects?" on page 116
- "Why does the rescind transaction not update the Future Full State Object?" on page 116
- "How are photos synchronized from Azure AD to Workday through the OpenText Identity Manager?" on page 117

How to confirm successful establishment of JWT connection?

The successful JWT connection with all required parameters can now be confirmed from the log in level 3, as shown below:

```
Workday Ent PT:Workday Ent: Creating new OAuth Connection...
Workday Ent PT:Workday Ent: Extracting private key from the keystore
file...
Workday Ent PT:Workday Ent: JWT Bearer Token initialized with
OAuth2 Client Id: ZmE1NjNhOGYtZWY1Mi00Yzg2LTlmZmUtOGJjOTAyNjhjYjEy
Workday Integration Username: idmuser2
Public-Private-Key Keystore File: /home/JWTkeystore.jks
Workday Ent PT:Workday Ent: Sending Request to Workday...
Workday Ent PT:
<soapenv:Envelope xmlns:bsvc="urn:com.workday/bsvc" xmlns:soapenv="http://
schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <bsvc:Server_Timestamp_Get bsvc:version="v32.0"/>
  </soapenv:Body>
</soapenv:Envelope>
Workday Ent PT:Workday Ent: Generating signed JWT...
Workday Ent PT:Workday Ent: Requesting new access token...
Workday Ent PT:Workday Ent: Fetching Page 1 of 1 Pages.
Workday Ent PT:Workday Ent: Received Response from Workday with response
code: 200
```

Why are Delta Objects and Full State Objects created with the suffix “<Termination Date + 1>” in the termination business process?

This occurs because, the termination is in effect from the end of termination date. Hence, the termination information, such as Active status, Termination Date, Termination Reason, Regrettable or non-Regrettable, Voluntary or non-Voluntary are suffixed.

Why does same day termination also create Delta and Full state objects?

As the termination business process always queries for the next day as the effective date, the business process treats it as a future event.

Why does the rescind transaction not update the Future Full State Object?

Rescind process is an event that is triggered by the HR administrator to terminate a business process that was initiated for an expected worker to join the organization. The rescind event is triggered if the expected worker defers to join the organization. As the HR administrator would have started the upstream business process, a rescind event is triggered to terminate it.

To create a future state object, the business process and workflows are initiated based on the created future state event.

For example, assume a business process is initiated for hiring, and the full state object is created “x“ days prior to the effective date of hiring. In some cases, there may be changes to update the worker object, post the creation of full state object. These changes when incorporated to the worker object, they do not reflect in the full state object. Hence, there will be a difference between the worker object and the full state object.

So, depending on when a rescind event is triggered, the Workday driver ensures to update the changes to the associated object.

Consider the two scenarios for triggering the rescind event:

- **If the rescind event is triggered before creation of the full state object**:

  In this scenario, the full state object will include the rescind information. This is because, the rescind event was already triggered prior to the full state object creation. As a result of this, the future state object will not be created.

- **If the rescind event is triggered after creation of the full state object**:

  In this scenario, the full state object will not include the rescind information. As the rescind event was triggered post creation of the full state object, this event updates only the current state object, and not the full state object.

How are photos synchronized from Azure AD to Workday through the OpenText Identity Manager?

Azure AD supports photo synchronization in OpenText Identity Manager User container. However, Workday supports photo synchronization through a different container called `wd-Photo`.

Follow the procedure given below to complete the synchronization:

1. In OpenText Identity Console, click the **IDM Administration** tile.
2. On the Driver Dashboard, click the Workday Driver icon.
3. Click the **Data Transformation and Synchronization** tab.
4. Click the **Filter** in the **Subscriber channel**, add the Azure attribute `jpegPhoto`. To add:
   - **4a** Select **User** as class and click **Add Attribute**.
   - **4b** Click **Show All Attribute** to display the list of attributes applicable for the class.
   - **4c** Add `jpegPhoto`, and save.
   - **4d** Select the attribute from the class list and set the following:
     - Publisher: **Ignore**
     - Subscriber: **Notify**
5. For the Workday attribute `wd-Photo` class, set the Subscriber channel to Synchronize.
6. Place the following policy in the subscriber etp after `NETQWDDCFG-sub-etp-SupportedOperations-users`:

```
<?xml version="1.0" encoding="UTF-8"?><policy>
    <rule>
        <description>subPhotoSync</description>
        <comment xml:space="preserve">SubPhotoSynz</comment>
        <conditions>
            <and>
                <if-class-name mode="nocase" op="equal">User</if-class-
name>
                <if-op-attr name="jpegPhoto" op="changing"/>
            </and>
        </conditions>
        <actions>
            <do-set-local-variable name="workerID" scope="policy">
                <arg-string>
                    <token-attr name="workforceID"/>
                </arg-string>
            </do-set-local-variable>
            <do-set-local-variable name="eType" scope="policy">
                <arg-string>
                    <token-attr name="wd-WorkerIDType"/>
                </arg-string>
            </do-set-local-variable>
            <do-set-local-variable name="photoVal" scope="policy">
                <arg-string>
                    <token-op-attr name="jpegPhoto"/>
                </arg-string>
            </do-set-local-variable>
```

```
            <do-add-dest-attr-value class-name="wd-Photo" name="photo"
when="after">
                <arg-association>
                    <token-local-variable name="workerID"/>
                    <token-text xml:space="preserve">-</token-text>
                    <token-local-variable name="eType"/>
                </arg-association>
                <arg-value type="octet">
                    <token-local-variable name="photoVal"/>
                </arg-value>
            </do-add-dest-attr-value>
        </actions>
    </rule>
</policy>
```

11 Known Issues in Workday Driver

This sections explains the known issues that are associated with the Workday driver.

- "Modifying the Device Types Attributes are not Supported Independently" on page 119
- "Unable to Restrict Entitlement Resource Assignments" on page 119

Modifying the Device Types Attributes are not Supported Independently

Issue: Modifying the device types attributes are not supported independently in the Subscriber Channel.

Workaround: You can modify them along with their respective phone number attributes, as shown below:

- The `wd-HomePrimaryDeviceType` attribute can be modified along with `wd-HomePrimaryPhone` attribute.
- Similarly, the `wd-WorkPrimaryDeviceType` can be modified along with `wd-WorkPrimaryPhone`.

For more information to configure the phone contact attributes, see "Configuring Phone Contact Information in Subscriber Channel" on page 38

Unable to Restrict Entitlement Resource Assignments

Issue: For an organizational role in Workday, the single and multi assignments can be restricted. However, the same cannot be restricted by executing the Workday SOAP API PUT operation. Hence, assignments made by entitlement resource from the Resource Catalog cannot be restricted.

Workaround: There is no workaround at the moment.

A Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the Workday driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so forth) are common to all drivers. Refer to Driver Properties (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b94pq23.html) in the OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html) for information about the common properties.

The information is presented from Identity Console's perspective. If a field is different in Designer, it is marked with a Designer icon.

Refer to the following sections for more information:

- "Driver Configuration" on page 121
- "Global Configuration Values" on page 123

Driver Configuration

In Identity Console:

1. Click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.

In Designer:

1. Open a project in the Modeler.
2. Right-click the driver icon or line, then select click **Properties > Driver Configuration.**

The following sections describe driver configuration in details:

Driver Module

The driver module changes the driver from running locally to running remotely or the reverse.

Java: Used to specify the name of the Java class that is instantiated for the shim component of the driver. This class can be located in the `classes` directory as a class file, or in the `lib` directory as a `.jar` file. If this option is selected, the driver is running locally.

The Java class name is: `com.netiq.dirxml.driver.workday.WorkdayShim`

Native: Used when driver is connecting directly from the server where Identity Manager engine is installed.

Connect to Remote Loader: Used when the driver is connecting remotely to the connected system. Designer includes two sub options:

- **Driver Object Password:** Specifies a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim.
- **Remote Loader Client Configuration for Documentation:** Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.

Driver Object Password

Driver Object Password: Use this option to set a password for the driver object. If you are using the Remote Loader, you must enter a password on this page or the remote driver does not run. This password is used by the Remote Loader to authenticate itself to the remote driver shim.

Authentication

The authentication section stores the information required to authenticate to the connected system.

Authentication ID: This option is not used with the Workday driver.

Authentication Context: This option is not used with the Workday driver.

Remote Loader Connection Parameters: Used only if the driver is connecting to the application through the Remote Loader. The parameter to enter is `hostnamexxx.xxx.xxx.xxx portxxxx kmocertificatename`, when the host name is the IP address of the application server running the Remote Loader server and the port is the port the Remote Loader is listening on. The default port for the Remote Loader is 8090.

The `kmo` entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine.

Example: `hostname10.0.0.1 port8090 kmoIDMCertificate`

Cache limit (KB): Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. Click **Unlimited** to set the file size to unlimited in Designer.

Application Password: This option is not used with the Workday driver.

Remote Loader Password: Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.

Startup Option

The Startup Option section allows you to set the driver state when the Identity Manager server is started.

Auto start: The driver starts every time the Identity Manager server is started.

Manual: The driver does not start when the Identity Manager server is started. The driver must be started through Designer or Identity Console.

Disabled: The driver has a cache file that stores all of the events. When the driver is set to Disabled, this file is deleted and no new events are stored in the file until the driver state is changed to Manual or Auto Start.

Do not automatically synchronize the driver: This option only applies if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

Driver Parameters

The Driver Parameters section lets you configure the driver-specific parameters. When you change driver parameters, you tune driver behavior to align with your network environment. The parameters are divided into the following categories:

- Driver Options
- Subscriber Options
- Publisher Options

To configure the above options, see Step 6 in the section "Creating the Driver Object" on page 22.

Global Configuration Values

Global Configuration Values (GCVs) can be used by the driver to control its functionality. GCVs are defined in the driver or in the driver set. Driver set GCVs can be used by all drivers residing in the driver set.

The Workday driver includes predefined GCVs. You can also add your own GCVs as required for the additional policy implementation in the driver. For information on most of the GCVs, see "Creating the Driver Object" on page 22.

However, the following GCV's should be configured after installing the driver:

- **User Config**: This GCV helps to configure settings for a user in the Publisher channel. For more information, see Step 10 in the section "Creating the Driver Object" on page 22.
- **Business Process**: As a part of Workday Business Process, you can specify the container details where the information for delta and future objects will be created in the Identity Vault. For more information, see "Configuring Business Process" on page 61.
- **Retry Event Config:** You can specify certain attributes in this field for which events get logged on the Workday portal and the driver might retry to configure them after activating the user. Mention the required list of attributes separated by comma in this field. The following attributes will be listed by default:
  - wd-HomePrimaryPhone
  - wd-WorkPrimaryPhone
  - Internet EMail Address
  - homeEmailAddress

NOTE: ◆If any secondary phone attributes are configured as part of the driver, you can add them to this retry event config list.

- Ensure to extend the Schema to include `DirXML-WDSyncAttr` attribute in IDV for configuring the retry events. For more information, see Appendix C, "IDV Schema Extension," on page 127.

- To get a reference to the container in IDV where new objects will be created, you must specify the following GCVs:
  - **Relation Config**
  - **JobFamily Config**
  - **Location Config**
  - **Photo Config**
  - **Job Profile Config**
  - **Organizations Config**

  To configure these GCVs, see Step 11 in the section "Creating the Driver Object" on page 22.
- **Entitlement**: You can now enable entitlements for User-Based Security Groups and Organizational Roles. To enable entitlements, double click the connector line and navigate to **GCVs > Entitlement** tab. For more information, see "Customizing Entitlements" on page 55.

For more information on GCVs, see When and How to Use Global Configuration Values (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bh15mpk.html) in OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html).

B Trace Level

The driver supports the following trace levels:

Table B-1 *Supported Trace Levels*

Level	Description
0	Provides error/warning/exception
1	Previous level and success messages.
2	Previous level and input/output trace
3	Previous level and policy processing/SOAP calls
4 & 5	More debug messages

For information about setting driver trace levels, see Viewing OpenText Identity Manager Processes (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b1rc1vm.html) in the OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html).

C IDV Schema Extension

As part of solution, we need to extend the schema of eDirectory so that the Workday driver can create these additional objects in the IDV. The proposed schema extension in `ldif` and `sch` formats are available for download. The following classes are added in IDV.

- Auxiliary Class for User: wd-User

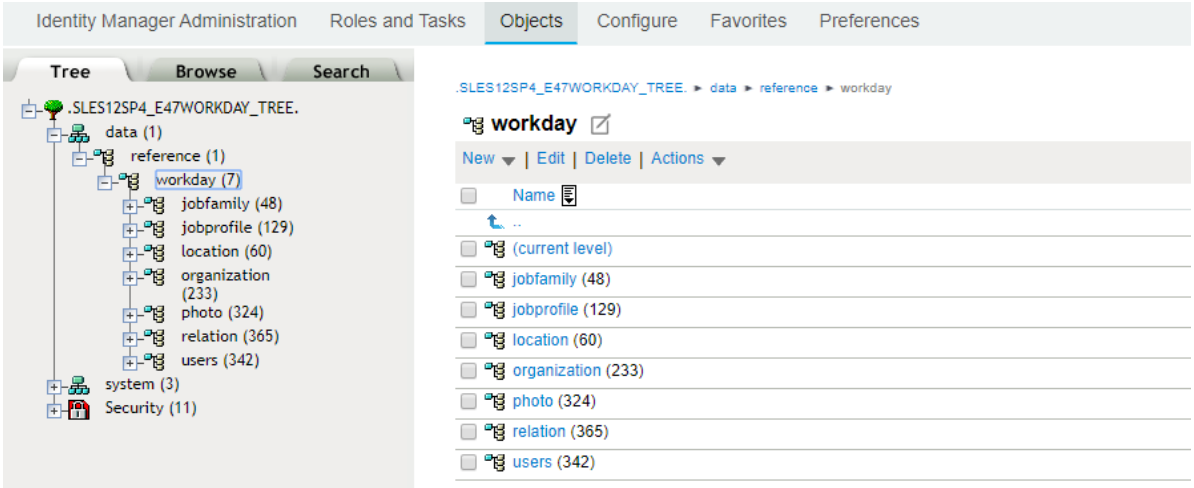
  The schema file should only be extended with the `ndssch` utility. Using any other method for importing the schema file fails for certain attributes. A sample syntax for importing the schema file has been provided below:

  ```
  /opt/novell/eDirectory/bin/ndssch -h idv.server.DNS.name -F /tmp/wd-schema.log admin.services.data /tmp/wd-schema.sch
  ```

- Effective Class for Relation Objects: wd-Relation
- Effective Class for Job Family Objects: wd-Jobfamily
- Effective Class for Job Profile Objects: wd-Jobprofile
- Effective Class for Location Objects: wd-Location
- Effective Class for Organization Objects: wd-Organization
- Effective Class for Photo Objects: wd-Photo
- Effective Class for Retry Settings: DirXML-WDSyncAttr

D IDV Proposed Structure

As part of solution, it is proposed to have a different container for different objects in IDV. The proposed structure is as follow:



E Understanding the Driver State Storage

The Workday driver periodically updates its current status in XML format in the `DirXML-DriverStorage` attribute which is stored in OpenText Identity Vault. Driver uses this XML document to resume its operation after a restart. Each element in the XML document is presented in the following format:

```
<Object Class-Poll Type-Status Type>Value</Object Class-Poll Type-Status Type>
```

For example,

```
<worker-regular-poll-start-timestamp>03-12-2019 14:37:20</worker-regular-poll-start-timestamp>
```

In the above example, **Worker** is the object class, **Regular Poll** is the poll type and **Start Timestamp** is the status type. The information related to the following object classes are stored in the XML document:

- Worker
- Terminated Worker
- Job Profile
- Job Family
- Location
- Organization
- Photo

The following Poll Types are stored in the XML document:

- Migration
- Migration_Restart
- Regular_Poll

The following table explains the various status types that are stored in the XML document:

Table E-1 *Explanation of Status Types*

Status Type	Description
start-timestamp	Start timestamp of the specified latest poll type
end-timestamp	End timestamp of the specified latest poll type
restart-timestamp	Restart timestamp of the specified latest poll type

Status Type	Description
last-page-poll-start-timestamp	Start timestamp of the last page that is fetched during the specified latest poll type
last-page-poll-end-timestamp	End timestamp of the last page that is fetched during the specified latest poll type
last-poll-timestamp	The last timestamp for which the polling was requested to the Workday server
last-effective-poll-timestamp	The last timestamp for which the effective polling was requested to the Workday server
migration-completed	Timestamp of the migration completion
last-request-doc	The last request that was sent to the Workday server. This request is stored in `Base64` format.
page-count	The number of pages fetched during the specified latest poll type
total-page	The total number of pages in the Workday response during the specified latest poll type
entry-time	The entry timestamp for which the polling was requested to the Workday server

NOTE: ◆The `last-request-doc` status type will not be stored in the conventional format. It stores the last request that was made to the Workday server irrespective of object class or poll type.

◆ The effective polling happens once in a day and happens for worker and job profile.