opentext™

OpenText™ Identity Manager

Driver Implementation Guide for Sentinel

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.

Contents

About This Guide **5**

1 Introduction **7**

Components for Identity Tracking 7
DirXML-Accounts Attribute 7
Driver for Sentinel 8
Sentinel REST API Interface 8
How the Driver Works 8
Data Transfer Between Systems 11

2 What's New? **13**

What's New in Version 4.0.1? 13
What's New in Version 4.0.0.0? 13
What's New in Version 2.0.0? 13
Upgrade Procedure 13
. . . . . Upgrading the Installed Packages 13
Applying the Driver Patch 14

3 Checklist for Enabling Identity Tracking **17**

4 Planning the Driver Shim Installation **19**

System Requirements 19
Planning the Installation 19
Installing the Driver Shim on the Identity Manager System 20
Installing the Driver Shim on a Remote System 21

5 Installing the Driver Shim **23**

Installing on Windows 23
Installing on Linux 24

6 Creating and Configuring the Driver **25**

Understanding the Configuration 25
Creating the Driver 26
Deploying the Driver 27
Starting the Driver 28
Verifying the Functionality 28

7 Activating the Driver **31**

8 Understanding the Schema Mapping **33**

9 Migrating Data **37**

Disabling the Subscriber Channel 37
Migrating the Data from Identity Vault to Sentinel 37
Migrating Data Without Manager References 37
Migrating Data with Manager References 37

About This Guide

This guide introduces a Sentinel or OpenText Identity Manager Administrator to the process of integrating identity information stored in OpenText Identity Manager with Sentinel's event collection.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Other Information in the Library

For more information about the library for OpenText Identity Manager, see the following resources:

- OpenText™ Identity Manager CE 24.4 (v4.10) documentation website (https://www.netiq.com/documentation/identity-manager-4.10/)
- OpenText™ Identity Manager CE 24.4 (v4.10) driver documentation website (https://www.netiq.com/documentation/identity-manager-4.10-drivers/)

Additional Documentation

- *Sentinel Documentation*

Contacting Information

- Technical Support
- Self Support
- Patch download site
- Identity Manager Support Forum
- Sentinel Support Forum
- Identity Manager TIDs
- Sentinel TIDs

For specific product issues, contact Micro Focus Support at https://www.microfocus.com/support-and-services/.

Additional technical information or advice is available from several sources:

- Product documentation, Knowledge Base articles, and videos: https://www.microfocus.com/support-and-services/
- The Micro Focus Community pages: https://www.microfocus.com/communities/

1 Introduction

Users in an IT environment have accounts with multiple applications and sometimes have multiple account identifiers with a single application. For example, if a user has accounts with both Active Directory and an LDAP directory, the user can log in to either of the application or both applications.

Sentinel tracks events related to user activities in applications but, without additional data, Sentinel cannot correlate account activity in different applications with the single user who initiated the actions.

The Driver for Sentinel provides the additional data required to correlate actions in disparate applications with the initiating use. The driver integrates Sentinel with OpenText Identity Manager to track the user identity associated with each user account and which events those identities have performed. This allows you to rapidly solve a variety of complex business problems. For example, the account tracking solution helps you monitor rogue administration and define what action is taken if this occurs.

- "Components for Identity Tracking" on page 7
- "How the Driver Works" on page 8
- "Data Transfer Between Systems" on page 11

Components for Identity Tracking

This section provides information about the components required to integrate Sentinel with Identity Manager.

DirXML-Accounts Attribute

The DirXML-Accounts attribute on an Identity Vault User object tracks information about accounts that a user has in different applications. Identity Manager drivers that manage the account information for a user in an application, create and maintain the DirXML-Accounts attribute values. For example, the Active Directory driver maintains the DirXML-Accounts values for the account identifiers that a user has in Active Directory.

The Driver for Sentinel uses the DirXML-Accounts values to create and manage account records in Sentinel. If the application has multiple ways of identifying a single application account, there might be multiple account records in Sentinel for a single account. For example, Active Directory has five different identifiers for the same account. The Active Directory driver provides information about four of these identifiers in the DirXML-Accounts attribute. The driver synthesizes the fifth value from one of the identifiers provided by the Active Directory driver.

Table 1-1 shows that the DirXML-Accounts attribute stores the different identifiers for John's account. Active Directory has four different account identifiers for the same account and the LDAP directory has one.

Table 1-1 Contents of the DirXML-Accounts Attribute

Driver/Application	Account Identifier Type	Account Identifier Sample Data
Active Directory	sAMAccountName	jsmith
Active Directory	userPrincipalName	jsmith@company.com
Active Directory	LDAPDN	cn=John Smith,cn=users,dc=company,dc=com
Active Directory	association	5d377f84f3ab534babbf12edd6540d77
LDAP	LDAPDN	cn=jsmith,cn=users,dc=company,dc=com

This allows for correlation between all of the account identities in the systems managed by Identity Manager. You can also validate business policies with this information.

Driver for Sentinel

The driver is an Identity Manager driver that sends the account identifier and the account status from the Identity Vault to the Sentinel REST API interface. The account identifier data is used to track the accounts, the status of the identities, and the account access information.

The driver implements data sharing policies with Sentinel. You can control the actions by using Identity Console to define filters and policies.

Sentinel REST API Interface

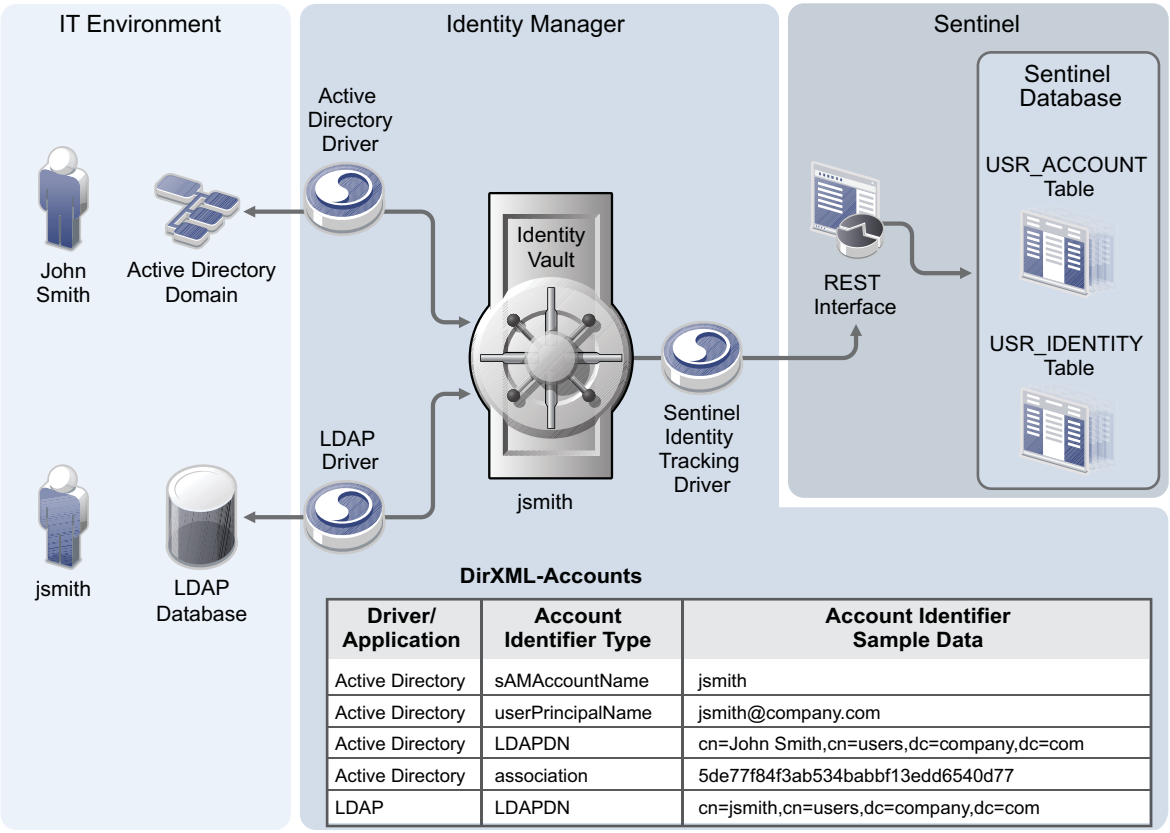
The Sentinel REST API interface integrates the data from the driver to Sentinel. The interface performs functions, such as remote protocol connections and data mapping.

How the Driver Works

Sentinel receives information from various Identity Collectors and then stores the data in the database. If the same user has multiple identifiers for a single account in an application, Sentinel treats each identifier as a unique account.

The driver enables you to track all account identifiers for each user and to track the status of those accounts, so you have a complete picture of user activities. Figure 1-1 illustrates how the driver works to capture this information.

Figure 1-1 *Synchronizing Account Data*



Driver/ Application	Account Identifier Type	Account Identifier Sample Data
Active Directory	sAMAccountName	jsmith
Active Directory	userPrincipalName	jsmith@company.com
Active Directory	LDAPDN	cn=John Smith,cn=users,dc=company,dc=com
Active Directory	association	5de77f84f3ab534babbf13edd6540d77
LDAP	LDAPDN	cn=jsmith,cn=users,dc=company,dc=com

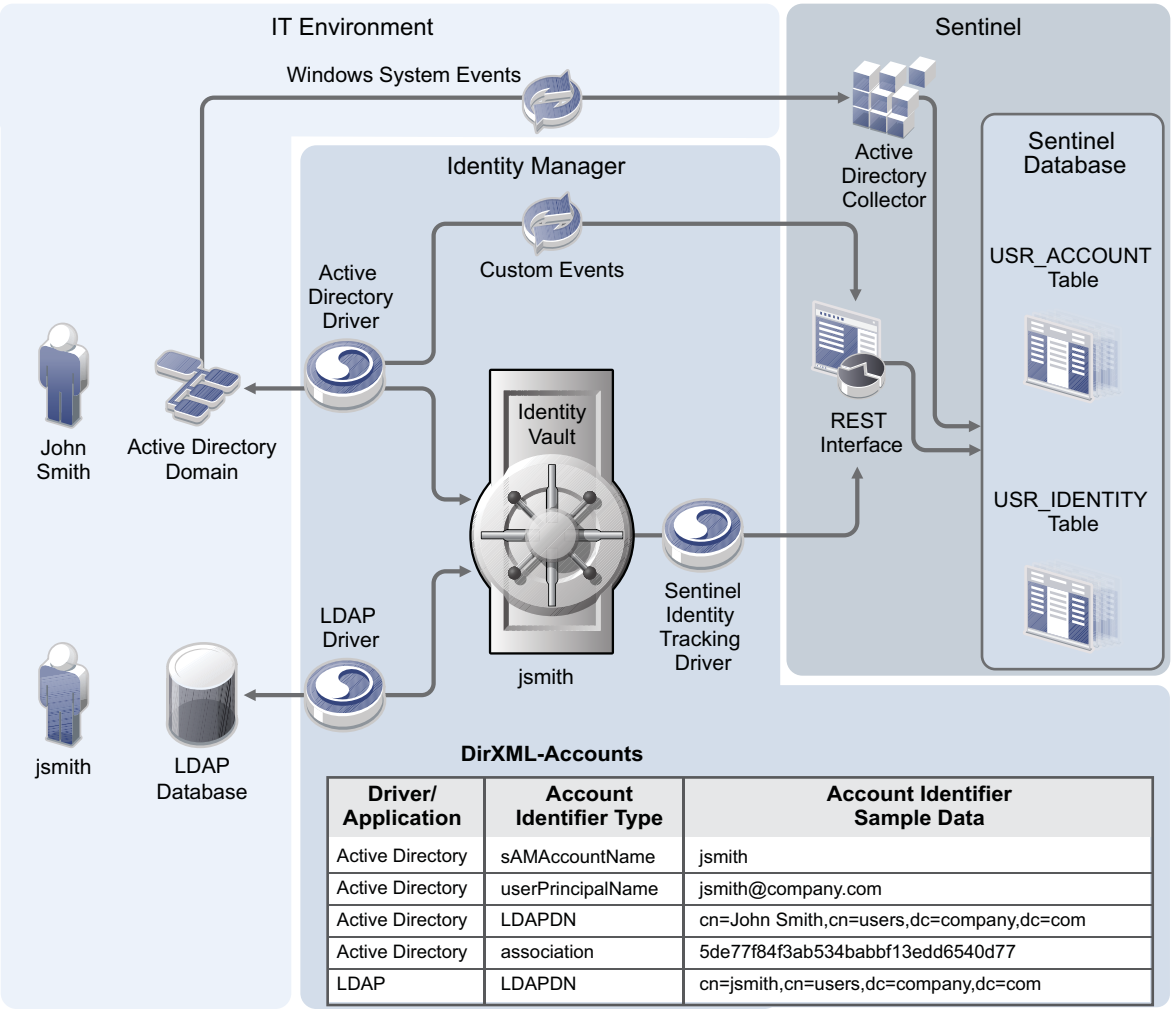
1. The Active Directory driver creates an account for John Smith in Active Directory and synchronizes the information to the Identity Vault.
2. The Identity Vault, which contains the DirXML-Accounts attribute, creates an account for John Smith. The DirXML-Accounts attribute stores the different account identifiers from Active Directory.
3. The driver detects that the DirXML-Accounts attribute is added and sends this information to the Sentinel REST Interface.
4. The LDAP driver detects the new account created in the OpenText Identity Vault, then synchronizes this information to the LDAP database.
5. The LDAP driver creates a new account for John Smith in the LDAP database as follows:

   `cn=jsmith,cn=users,dc=company,dc=com`

6. The LDAP driver synchronizes the new account information back to the Identity Vault. The Identity Vault adds a new entry to the DirXML-Accounts attribute.
7. The Sentinel driver detects the change to the DirXML-Accounts attribute, then sends this information to the Sentinel REST Interface.
8. The Sentinel REST Interface stores the account data in the USR_IDENTITY table in the Sentinel database.
9. The Sentinel correlation engine uses the information in the USR_IDENTITY table to generate reports of account activity per identity across all the systems provisioned by Identity Manager.

The second half of this solution allows other Sentinel Collectors to use the account information to track whether your organization enforces business policies. Figure 1-2 shows how Sentinel uses the custom events and the events from other Collectors to provide a complete record of John Smith's accounts.

Figure 1-2 *Synchronizing Events*



Driver/ Application	Account Identifier Type	Account Identifier Sample Data
Active Directory	sAMAccountName	jsmith
Active Directory	userPrincipalName	jsmith@company.com
Active Directory	LDAPDN	cn=John Smith,cn=users,dc=company,dc=com
Active Directory	association	5de77f84f3ab534babbf13edd6540d77
LDAP	LDAPDN	cn=jsmith,cn=users,dc=company,dc=com

1. The Active Directory driver creates an account for John Smith in the OpenText Identity Vault.
2. The Sentinel driver detects this new account and sends the account information to the Sentinel REST Interface, which stores the information in the USR_IDENTITY table.
3. John Smith logs in to Active Directory, and that information is sent to Sentinel through the Active Directory Collector.
4. The Active Directory Collector receives the login event directly from Windows without going through the Identity Vault. Sentinel records this information in the USR_ACCOUNT table indicating that `cn=John Smith,cn=users,dc=company,dc=com` logged in at a specific time.
5. If John Smith's CN in Active Directory is renamed to John D. Smith, the Active Directory driver synchronizes the information to the Identity Vault.

6. The DirXML-Accounts attribute is updated with the new information, and the Sentinel driver detects this change.
7. The Sentinel driver synchronizes the new account information to the Sentinel REST interface.
8. The Sentinel REST interface reads the new account information and writes it to the USR_IDENTITY table.
9. When John Smith logs in again to Active Directory, the Active Directory Collector records the login information.
10. Sentinel performs a lookup on the USR_IDENTITY table and detects that John Smith and John D. Smith are the same user account. Sentinel can keep a complete record of user actions.
11. The driver policies define and add custom audit events to each Identity Manager driver. The policies add a layer of intelligence to Identity Manager and Sentinel by defining the business logic. These policies are part of each driver that ships with Identity Manager.
12. You can generate useful reports about user accounts from Sentinel.

The Sentinel Identity Tracking driver provides the infrastructure to allow Sentinel to track each user account. This awareness allows you to enforce business policies.

Data Transfer Between Systems

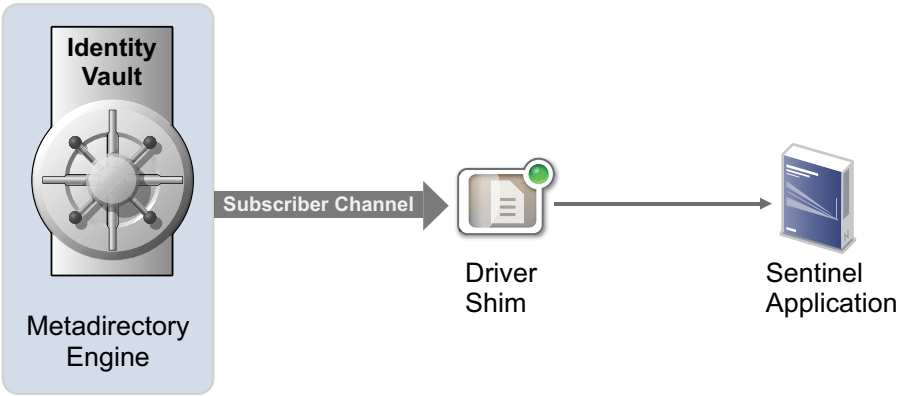
There are two data transfer channels between the OpenText Identity Vault and the connected application:

- **Publisher Channel:** Transfers data and events from the connected application to the OpenText Identity Vault. The Sentinel Identity Tracking driver does not support this channel.
- **Subscriber Channel:** Transfers data and events from the OpenText Identity Vault to the connected application. The Sentinel Identity Tracking driver supports only data transfers from the Identity Vault to Sentinel. Communication is one-way only.

  The Subscriber channel does the following:

  - Watches for additions and modifications to the Identity Vault objects.
  - Makes changes to Sentinel's internal representation of Identities and Accounts that reflect those changes.

Figure 1-3 *Data Transfer Between Systems*



2 What's New?

What's New in Version 4.0.1?

This version of the driver does not provide any new features.

What's New in Version 4.0.0.0?

This version provides the following key features and functions:

- The driver now supports Sentinel 7.1 and later in both FIPS and non-FIPS mode.

  NOTE: If you upgrade your Sentinel system to version 7.1 or later, you must download the `Sentinel client.jar` files and Sentinel driver shim from the download web site.

- MSSP (Managed Security Service Provider) administrators can now set up identity tracking for MSSP tenants by using the name of the tenant as used in the Sentinel Collector configuration instead of the database record ID of the tenant.

What's New in Version 2.0.0?

This version provides the following key features and functions:

- The Driver for Sentinel, formerly known as the Sentinel Driver, removes the dependency on the Collector for Identity Manager and Sentinel integration.
- This driver now facilitates simplified configuration. The previous version of the driver used a JMS bus and sent information to a Sentinel Collector, which then sent the information to the Sentinel database. This driver uses the native Sentinel 7.0.1 REST APIs to perform the integration, which sends the account information directly to the Sentinel database.

Upgrade Procedure

The driver upgrade process involves upgrading the installed driver packages and updating the driver files.

This section provides general instructions for updating a driver. For information about updating the driver to a specific version, search for that driver patch in the OpenText Patch Finder Download Page and follow the instructions from the Readme file accompanying the driver patch release.

Upgrading the Installed Packages

1. Download the latest available packages.

To configure Designer to automatically read the package updates when a new version of a package is available, click **Windows > Preferences > NetIQ > Package Manager > Online Updates** in Designer. However, if you need to add a custom package to the Package Catalog, you can import the package `.jar` file. For more information about creating custom packages, see Upgrading the Installed Packages (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/t4d53bn8cowh.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)*.

2 Upgrade the installed packages.

   2a Open the project containing the driver.

   2b Right-click the driver for which you want to upgrade an installed package, then click **Driver > Properties**.

   2c Click **Packages**.

   If there is a newer version of a package, there is check mark displayed in the Upgrades column.

   2d Click **Select Operation** for the package that indicates there is an upgrade available.

   2e From the drop-down list, click **Upgrade**.

   2f Select the version that you want to upgrade to, then click **OK**.

   NOTE: Designer lists all versions available for upgrade.

   2g Click **Apply**.

   2h (Conditional) Fill in the fields with appropriate information to upgrade the package, then click **Next**.

   Depending on which package you selected to upgrade, you must fill in the required information to upgrade the package.

   2i Read the summary of the packages that will be installed, then click **Finish**.

   2j Review the upgraded package, then click **OK** to close the Package Management page.

   For detailed information, see the Upgrading the Installed Packages (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/t4d53bn8cowh.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)*.

Applying the Driver Patch

The driver patch updates the driver files. You can install the patch as a `root` or `non-root` user.

Prerequisites

Before installing the patch, complete the following steps:

1 Take a back-up of the current driver configuration.

2 (Conditional) If the driver is running with the OpenText Identity Manager engine, stop the Identity Vault and the driver instance.

3 (Conditional) If the driver is running with a Remote Loader instance, stop the Remote Loader instance and the driver instance.

4 In a browser, navigate to the OpenText Patch Finder Download Page.

5 Under **Patches**, click **Search Patches**.

6 Specify **Identity Manager *nn* Sentinel Driver nn** in the search box.

7 Download and unzip the contents of the patch file to a temporary location on your server.

For example, `IDM402_SENTINEL_4000.tar.gz`.

Applying the Patch as a Root User

In a root installation, the driver patch installs the driver files RPMs in the default locations on Linux. On Windows, you need to manually copy the files to the default locations.

1 Update the driver files:

- **Linux:** Log in to your server as `root` and run the following command in a command prompt:

  `rpm -Uvh <Driver Patch File Temporary Location>/linux/netiq-DXMLsentinel-REST-version.noarch.rpm`

  For example, `rpm -Uvh <IDM402_SENTINEL_4000.tar.gz>/linux/netiq-DXMLsentinel-REST-4.0.0.0-7.noarch.rpm`

- **Windows:** Navigate to the `<Extracted Driver Patch File Temporary Location>\nt` folder and copy the following files to `<IdentityManager installation>\NDS\lib` or `<IdentityManager installation>\RemoteLoader\<architecture>\lib` folder.
  - `SentinelRESTShim.jar`
  - `sentinel-client-base.jar`
  - `sentinel-client-base-java.jar`
  - `sentinel-client-beans.jar`

2 (Conditional) If the driver is running locally, start the OpenText Identity Vault and the driver instance.

3 (Conditional) If the driver is running with a Remote Loader instance, start the Remote Loader instance and the driver instance.

Applying the Patch as a Non-Root User

1 Verify that `<non-root eDirectory location>/rpm` directory exists and contains the file, `_db.000`.

The `_db.000` file is created during a non-root installation of the OpenText Identity Manager engine. Absence of this file might indicate that OpenText Identity Manager is not properly installed. Reinstall OpenText Identity Manager to correctly place the file in the directory.

2 To set the `root` directory to non-`root` OpenText eDirectory location, enter the following command in the command prompt:

`ROOTDIR=<non-root eDirectory location>`

This will set the environmental variables to the directory where OpenText eDirectory is installed as a non-`root` user.

3 Download the patch and untar or unzip the downloaded file.

4 To install the driver files, enter the following command:

```
rpm --dbpath $ROOTDIR/rpm -Uvh --relocate=/usr=$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 <rpm-location>
```

For example, to install the Sentinel driver 4.0.0.0-7 RPM, use this command:

```
rpm --dbpath $ROOTDIR/rpm -Uvh --relocate=/usr=$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/netiq-DXMLsentinel-REST-
4.0.0.0-7.noarch.rpm
```

3 Checklist for Enabling Identity Tracking

Use the following checklist to verify that you complete the following tasks in order to have a complete solution with the driver.

- ❒ Ensure that you have installed the software mentioned in "System Requirements" on page 19.
- ❒ Determine where you want to install the driver shim. For more information, see Chapter 4, "Planning the Driver Shim Installation," on page 19.
- ❒ Install the driver shim. For more information, see Chapter 5, "Installing the Driver Shim," on page 23.
- ❒ Create and configure the driver. For more information, see Chapter 6, "Creating and Configuring the Driver," on page 25.
- ❒ Migrate the identity data to Sentinel For more information, see Chapter 9, "Migrating Data," on page 37.

4 Planning the Driver Shim Installation

- "System Requirements" on page 19
- "Planning the Installation" on page 19

System Requirements

You need the following software to integrate OpenText Identity Manager with Sentinel:

- Identity Manager 4.0.1 or later.

  For information about installing OpenText Identity Manager 4.0.1, see OpenText Identity Manager installation instructions for your version from the documentation site.

- Designer for Identity Manager 4.0.1 or later.

  For information about installing Designer, see Designer installation instructions for your version from the documentation site.

- Sentinel 7.0.1 or later.

  For information about installing Sentinel 7.0.1 or later, see Sentinel installation instructions for your version from the documentation site.

Planning the Installation

You can install the driver shim on either the OpenText Identity Manager system or a remote host. Figure 4-1 illustrates the two installation options. The installation includes the following components:

- **Identity Vault:** Used by OpenText Identity Manager to store data for synchronization with Sentinel. The Identity Vault is a persistent database powered by OpenText eDirectory. The vault can be viewed as a private data store for Identity Manager or as a metadirectory that holds enterprise-wide data. Data in the vault is available to any protocol supported by OpenText eDirectory, including NCP (the traditional protocol used by utilities, such as ConsoleOne and Identity Console), LDAP, and DSML.

  Since the Identity Vault is powered by OpenText eDirectory, you can easily integrate Identity Manager into your corporate directory infrastructure by using your existing directory tree as the vault. The OpenText Identity Vault runs on any platform supported by Identity Manager and communicates with the module on the connected system over a secure network link. For information on the supported platforms, see "Supported Platforms "in the *Identity Manager Installation Guide.*

- **Driver Shim (Integration Module for Sentinel):** Converts the XML based Identity Manager command and event language (XDS) to the protocols and API calls required to interact with Sentinel. This driver uses a Java based driver shim (`SentinelRESTShim.jar`.) The driver shim is an executable code and is available on the OpenText download Web site.

- **Remote Loader:** Enables a driver shim to execute outside of the OpenText Identity Manager engine. The Remote Loader is typically used when a requirement of the driver shim is not met by the Identity Manager server. For example, if Identity Manager engine is running on Linux but you want to integrate with Active Directory, the Remote Loader is used to execute the Active Directory driver shim on a Windows server.

  The Remote Loader is a service that executes the driver shim and passes information between the shim and OpenText Identity Manager engine. You can install the driver shim on the server where the Remote Loader is running. You can choose to use SSL to encrypt the connection between the Identity Manager engine and the Remote Loader.

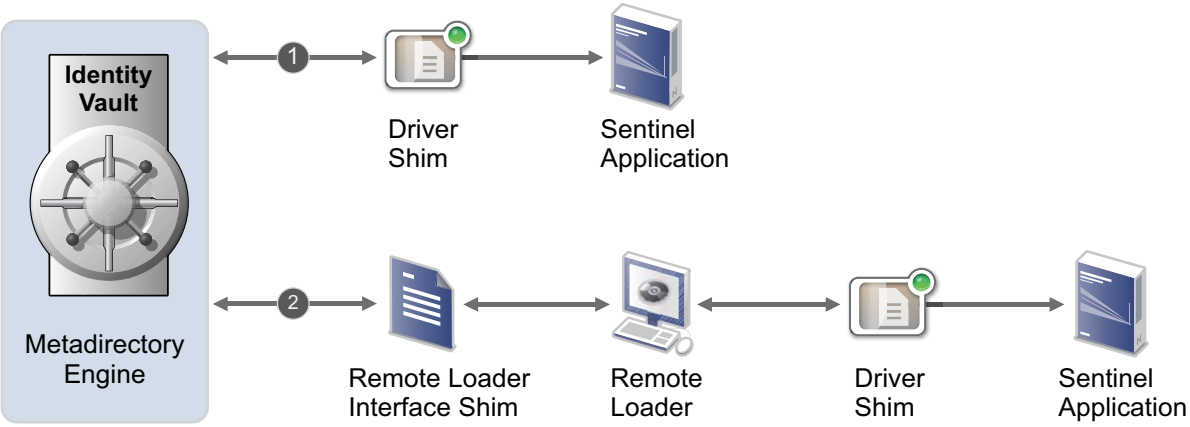
  When you use the Remote Loader with the driver shim, two network connections are established:

  - Between Identity Manager and Remote Loader
  - Between Sentinel and the driver shim

  For more information on Remote Loader, see "Exporting a Third-Party Certificate for Remote Loader (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b18xta1v.html#t45m493ku3p5)" 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)*.

The following figure illustrates the two options for installing the driver shim:

Figure 4-1 *Installing the Driver Shim*



Installing the Driver Shim on the Identity Manager System

The most common hosting for Identity Manager integration is in the Identity Manager engine.

Advantages:

- The Integration module logs the trace messages in the Identity Manager server trace log. Therefore, troubleshooting might be easier.
- No need to configure a Remote Loader instance.
- No extra TCP/IP traffic between Identity Manager and Remote Loader.

Disadvantages:

- Resource consumption on the Identity Manager server (memory, processor time).
- The requirement to restart the Identity Manager server each time the integration module is installed or updated.

Installing the Driver Shim on a Remote System

The following are the advantages and disadvantages of the installing the driver shim on a remote system:

Advantages:

- Resource consumption (memory, processor time) is in a different process, or on another host.
- You need to restart only the Remote Loader process when the integration module is updated.

Disadvantages:

- Multiple trace files. Therefore, when troubleshooting, you might need to examine trace files from both the Identity Vault process and the Remote Loader process.
- The need to configure a Remote Loader instance.
- Extra TCP/IP traffic between the Identity Vault and the Remote Loader.

5 Installing the Driver Shim

Before you create and configure the driver, you need to install the driver shim in order to be able to create and configure the driver.

- "Installing on Windows" on page 23
- "Installing on Linux" on page 24

Installing on Windows

1. Download the Integration Module for Sentinel (driver shim), `sentinel_driver_install.exe`, from the OpenText download Web site.
2. Execute the `sentinel_driver_install.exe` file on the Windows system, which is either the Identity Manger server or the Remote Loader, depending on where you want to install the driver shim.
3. Follow the installer prompts.

Once the installation is complete, ensure that you import the certificate since the driver users TLS/SSL protocol to communicate with the server. The type of certificate depends on the Sentinel server configuration:

- Self-signed public key certificate
- Trusted root certificate of the certificate authority

Self-signed Certificate

If your organization has not replaced the default Web server certificate, which is created when the Sentinel server is installed, you must obtain the self-signed certificate from the Sentinel server.

You can obtain this certificate either by using the supplied `getcert` utility or extracting it from the keystore file on the Sentinel server.

Using the getcert utility

1. Run the `getcert` utility on the system where you are running Identity Manager Designer. If the `getcert.jar` file is not located on the Identity Manager Designer system, you can either copy the `getcert.jar` file from the system on which you ran the Integration Module installer or install the getcert.jar file directly on the system using the Integration Module installer:

   Windows: Locate the `getcert.jar` file in Windows Explorer and double-click the file.

   Linux: Execute the `/opt/novell/eDirectory/lib/dirxml/util/sentinel_rest/getcert.jar` file by using the following command:

   ```
   java -jar getcert.jar
   ```

2. Specify the address and port of the Sentinel server and click **Get Certificate**.

   The certificate data is displayed.

3 Verify the certificate data and if the certificate is correct, click **Yes**.

4 Use this certificate data when prompted for the Sentinel TLS/SSL certificate while creating the driver.

For more information, see Step 4 in "Creating the Driver" on page 26.

Obtaining the certificate from the keystore file

Extract the `Webserver` certificate as the `root` user, from the following Java keystore file:

```
/etc/opt/novell/sentinel/config/.webserverkeystore.jks
```

The keystore password is `password`.

Trusted Root Certificate of the Certificate Authority

If your organization has replaced the Sentinel server default Web server certificate with a public key certificate signed by a certificate authority, such as Verisign or Entrust, you must obtain the appropriate trusted root certificate that corresponds to the certificate authority. You can obtain the trusted root certificate from your organization or the certificate authority your organization uses.

Installing on Linux

You need to install the integration module as the root user.

1 Download the Integration Module (driver shim) for Sentinel, `sentinel_driver_install_linux.bin`, from the OpenText Download Web site.

2 Execute the `sentinel_driver_install_linux.bin` file on the Linux machine, which is either the Identity Manager server or the Remote Loader, depending on where you want to install the driver shim.

- If the Linux machine has a windowing system, execute the installer in GUI mode by using the following command:

  ```
  <path>/sentinel_driver_install.bin
  ```

- If the Linux machine does not have a windowing system, execute the installer in console mode by using the following command line:

  ```
  <path>/sentinel_driver_install_linux.bin -i console
  ```

3 Follow the installer prompts.

6 Creating and Configuring the Driver

After you install the driver shim on the server where you want to run the driver, you must create the driver in the Identity Vault by using Designer.

- "Understanding the Configuration" on page 25
- "Creating the Driver" on page 26
- "Deploying the Driver" on page 27
- "Starting the Driver" on page 28
- "Verifying the Functionality" on page 28

Understanding the Configuration

You need to implement the driver by using two Designer packages: `Sentinel REST API BASE` and `Sentinel Identity Tracking`. When you import these packages, they create a driver with a set of rules and policies suitable for synchronizing identities and their associated account information with Sentinel. If your requirements for the driver are different from the default policies, you need to change them to effect the policies you want. Pay close attention to the default matching policies. The data that you trust to match users usually is different from the default. The policies themselves are commented and you can gain a greater understanding of what they do by creating a test driver and reviewing the policies with Designer.

When Identity Manager determines that a Sentinel Identity must be created from an Identity Vault User object, the driver first checks the existing Sentinel Identity objects to determine if it should use an existing Identity object should be used. The need for this arises when you install or reinstall the driver into a system that has previously tracked identity information, or when resynchronizing identity data for any purpose.

The driver performs the searches in the following order:

- **Match by Distinguished Name:** The DN and source Identity Vault name of the originating Identity Vault User object are stored as part of the Sentinel Identity object. The first search attempts to find an existing Sentinel Identity object with the same Identity Vault name and DN as the originating Identity Vault User object.
- **Match by GUID:** The driver stores the GUID value of the originating Identity Vault User object in the Sentinel Identity object **Source Identity Id** field. The second search attempts to find a Sentinel Identity object with a **Source Identity Id** value that matches the Identity Vault User object GUID value.
- **Match by User Attribute Values:** For information on matching attributes, see, Step 7.

Creating the Driver

To create and maintain Identity data, the driver must have administrator privileges. NetIQ recommends that you create a Sentinel user account with administrator privileges specifically for use with the driver. This allows you to use audit events to track the changes made by the driver in the Sentinel system.

You need to configure the driver with the name and password of a Sentinel user account so that the driver can authenticate to the Sentinel server and make changes to the Sentinel database.

Before you create the driver, import the latest packages to Designer. This driver requires the following packages:

- `NIQSENRESTB_<version>.jar`
- `NIQSENIDTRK_<version>.jar`

To import the packages:

1. Launch Designer.
2. Create a new project for the driver.

   For more information, see "Creating a Project (https://www.netiq.com/documentation/identity-manager-4.10/designer_admin/data/creating-a-project.html)"in the *Designer Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/designer_admin/data/bookinfo.html)*.
3. Select **Help** > **Check for Package Updates** to install the Sentinel driver packages.
4. In the Outline view, right-click Package Catalog and click **Import Packages**.
5. If the driver packages are available in the packages list, select the packages and click **OK**.

   or

   Click **Browse**, then browse to select the packages on the file system.

   For more information, see "Importing Packages (https://www.netiq.com/documentation/identity-manager-4.10/designer_admin/data/importpackage.html)"in the *Designer Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/designer_admin/data/bookinfo.html)*.
6. Go to **Enterprise** > **Sentinel**.
7. Verify whether **Sentinel REST API BASE** and **Sentinel Identity Tracking** packages are available.

To create the driver:

1. Drag the Sentinel application icon from the **Designer** palette > **Enterprise** > **Sentinel** folder to the Designer modeler.
2. From the **Available Packages** list, select **Sentinel REST API Base** and click **Next**.
3. From the **Select Mandatory Features** list, select **Sentinel Identity Tracking** and click **Next**.
4. Specify the following information, and then click **Next**:
   - **Sentinel account name:** Specify a name for the Sentinel account which the driver uses to perform operations in Sentinel.
   - **Sentinel account password:** Specify a password for the Sentinel account.

- **Sentinel server:** Specify the IP address of the Sentinel server. The Sentinel driver communicates with this server.
- **Sentinel port:** Specify the port for the Sentinel server. The default port is 8443.
- **Sentinel TLS/SSL certificate:** The driver uses Sentinel TLS/SSL to communicate with the Sentinel server.

  You must either obtain the Sentinel server self-signed public key certificate or the trusted root certificate of the certificate authority used to sign the Sentinel server public key certificate.

5 If you want to connect the driver to a Remote Loader, select **yes** and specify the Remote Loader connection data.

6 Specify a name for the driver and click **Next**.

7 In the following screen, perform the following steps and click **Next**:

Matching Attributes: This option allows you to find a matching Sentinel Identity object attribute values with the corresponding Identity Vault user object values. The default Identity Vault attributes are **Full Name** and **Internet EMail Address**. The standard mapping of these two Identity Vault attributes to Sentinel Identity attributes are `name` and `email`, respectively.

You can change the default attributes to any attribute or attributes that your organization finds useful. To be useful for matching, each Identity Vault object must have values for the attributes you choose.

NOTE: The values that you specify in Matching Attributes are considered when the driver's matching policies do not find a match by DN and User GUID value. For more information, see Understanding the Configuration.

Tenant ID: If you are configuring the driver for a managed security customer or if the driver is synchronized to a Sentinel server that contains data for multiple managed security customers, select **yes** and specify the numeric **Tenant Id** of the managed security customer.

TIP: The Tenant ID is available in the CUST table in the Sentinel database.

---

8 Click **Next**. Review the configuration settings and click **Finish**.

Deploying the Driver

After you create the driver in Designer, you must deploy the driver into the Identity Vault, because Designer is an offline tool.

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 Identity Vault, skip to Step 5; otherwise, specify the following information to authenticate:

- **Host:** Specify the IP address or DNS name of the server hosting the Identity Vault.
- **Username:** Specify the DN of the user object used to authenticate to the Identity Vault.
- **Password:** Specify the user's password.

4 Click **OK**.

5 Read through the deployment summary, then click **Deploy**.

6 Click **OK**.

7 Click **Define Security Equivalence** to assign rights to the driver.

The driver requires rights to objects within the Identity Vault that are involved in synchronization. The Admin user object is most often used to supply these rights. However, you might want to create an object, for example, DriversUser and assign security equivalence to that user. The DriversUser object must have the same security rights on the server as the driver. The driver needs read rights for the following attributes: Full Name, mail, Given Name, Surname, OU, Title, photo, mailstop, Telephone Number, workforceID, manager, GUID, DirXML-Accounts, and Login Disabled.

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

7b Click **OK**.

8 Click **Exclude Administrative Roles** to exclude users that should not be synchronized.

You should exclude any administrative User objects, such as 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

After you create and deploy the driver, you need to start the driver. Identity Manager is an event-driven system, so after the driver is started, it waits for events to occur.

1 In Designer, open your project.

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

Verifying the Functionality

After you deploy and configure the driver, you need to verify that the driver correctly creates and updates Sentinel Identity and Account data.

1 Ensure that you have started the driver.

2 Create a test user in the Identity Vault with all attributes required by the matching attributes you configured.

3 Verify that a corresponding Sentinel Identity is found in Sentinel > **People** browser.

For more information, see "Integrating Identity Information with Sentinel Events" in the *Sentinel User Guide*.

4 In the People browser, verify that a Sentinel account corresponding to the Identity Vault account appears under the Identity's **Profile** tab.

5 If your Identity Vault already contains objects with Identity Tracking data from other systems, such as Active Directory with the Identity Tracking package, you can use **Migrate from Identity Vault** in Identity Console to validate the configuration.

7 Activating the Driver

The driver for Sentinel contains its own activation that you receive from the customer center. The Sentinel driver requires this new activation within 90 days of creating the driver. Otherwise, the driver will be disabled.

If you create the driver in a driver set where you have activated the Sentinel driver, the driver inherits the activation. For information on activation, refer to Understanding Licensing and Activation (https://www.netiq.com/documentation/identity-manager-4.10/idm_overview_planning/data/licensing-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)*.

8 Understanding the Schema Mapping

The default Sentinel Identity Tracking packages apply a schema mapping between Identity Vault attributes and the corresponding Sentinel attributes. At a high level, information about each user identity in the Identity Vault is mapped to the USR_IDENTITY table in the Sentinel database and information from each identity's multi-value DirXML-Accounts attribute is mapped to the USR_ACCOUNT table. The driver maps each account to the incoming event stream, retrieves the associated Identity information from the Identity Vault, and sends the information about the user identity to the Sentinel database.

The following table describes how the Identity Vault attributes are mapped to the Sentinel USR_IDENTITY table and where the associated value is placed for events that match any associated accounts:

Table 8-1 *Mapping in USR_IDENTITY Table*

Identity Vault Attribute/ Metadata	Sentinel Column	Event Field and Comments
Not applicable (NA)	IDENTITY_GUID	InitiatorUserIdentityID TargetUserIdentityID Sentinel generates these fields internally.
srcDN	DN	
NA	CUST_ID	This field is set based on the tenant ID assigned to each Identity Tracking Integration Module for Sentinel, when Sentinel is receiving data for multiple tenants.
NA	VAULT_NAME	This is field is set to the eDirectory tree name.
GUID	SRC_IDENTITY_ID	Stores the Identity Vault GUID.
workforceID	WFID	pInitiatorUserWorkforceID TargetUserWorkforceID
Given Name	FIRST_NAME	
Surname	LAST_NAME	
Full Name	FULL_NAME	InitiatorUserFullName TargetUserFullName
Title	JOB_TITLE	
OU	DEPARTMENT_NAME	InitiatorUserDepartment TargetUserDepartment

Identity Vault Attribute/ Metadata	Sentinel Column	Event Field and Comments
mailstop	OFFICE_LOC_CD	
Internet Email Address	PRIMARY_EMAIL	InitiatorEmail TargetEmail
Telephone Number	PRIMARY_PHONE	
manager	MGR_GUID	Stores the Sentinel GUID that represents the identity of this person's manager. The mapping is not direct. Sentinel uses the object referenced by the Identity Vault “manager” attribute to determine the manager's Sentinel Identity object and thereby obtains the actual GUID value that forms the reference in Sentinel.
photo	PHOTO	

In addition to the Identity information, Sentinel stores information about accounts associated with this Identity. The Identity Manager drivers that are provisioning accounts to connected systems store information about those accounts in a multi-valued attribute on the DirXML-Accounts source User object. The format of each value in DirXML-Accounts is as follows:

```
<driver guid>#<account id type>#<account id>#<idv account status>#<app
account status>#<app Name>
```

The following table describes how these fields are mapped to the internal USR_ACCOUNTS table in Sentinel:

Table 8-2 *Mapping in USR_ACCOUNTS Table*

Identity Vault Value	Sentinel Column	Event Field and Comments
<account id>	USR_NAME	This field and USR_NAME are parsed out from the account information.
(calculated)	AUTHORITY	This field and USR_NAME are parsed out from the account information. This field and AUTHORITY are parsed out from the account information.
<idv account status>	BEGIN_EFFECTIVE_DATE	This value is set based on the settings of this field and the <app account status> field, plus Sentinel records a temporal record of when the account status was changed.
<app account status>	END_EFFECTIVE_DATECURRENT_F	This value is set based on the settings of this field and the <idv account status> field, plus Sentinel records a temporal record of when the account status was changed.

NOTE: If the default schema mapping does not meet your requirements, you can customize most of the schema mappings between the Identity Vault and Sentinel to suit your requirements. The framework is fully extensible to store arbitrary Identity attributes in Sentinel by using the Extended Attributes table (USR_IDENTITY_EXT_ATTR).

9 Migrating Data

If you are adding the Integration module to an Understanding Licensing and Activation Identity Vault with existing user objects and data, you need to perform an initial migration of the Understanding Licensing and Activation Identity Vault data into Sentinel.

If the Sentinel system to which you are migrating the Identity Vault data has no existing identity data, you can disable the Subscriber channel matching policies to speed up the migration process.

- "Disabling the Subscriber Channel" on page 37
- "Migrating the Data from Identity Vault to Sentinel" on page 37

Disabling the Subscriber Channel

You can temporarily disable the Subscriber channel matching policies to speed up the migration process. You can do so by setting the Global Configuration Value (GCV) in Designer.

1. Go to **Driver Properties page > GCV > Sentinel Identity Tracking Configuration**.
2. Set the **Initial Synchronization Mode** value to `true`.
3. Deploy and restart the driver.

   For more information, see "Deploying the Driver" on page 27 and "Starting the Driver" on page 28.

Migrating the Data from Identity Vault to Sentinel

You can migrate the user data with or without manager references.

Migrating Data Without Manager References

If your Identity Vault user data does not have manager references between user objects, the migration is a one-step process.

You can either manually select the objects to be migrated by using the **Migrate from Identity Vault** option in Identity Console. or allow the Identity Vault to automatically submit all objects by using the **Synchronize** option in Identity Console.

After the migration is complete, enable the Subscriber channel by setting the **Initial Synchronization Mode** to `false`.

Migrating Data with Manager References

If your Identity Vault user data has manager references between user objects the migration is a two-step process.

First, perform the steps mentioned in "Migrating Data Without Manager References" on page 37.

After the migration is complete, repeat the procedure. Repeating the procedure ensures that manager references that could not be established in the first step are resolved.

If an employee's Identity Vault object is synchronized before the employee's manager's Identity Vault object, the manager reference in Sentinel cannot be established because the manager's object does not exist in Sentinel yet. When you repeat the process, migration occurs after all objects are created in Sentinel so that all manager references can be established in Sentinel.