EVIDEN

Identity and Access Management

DirX Identity

Resolution Service Setup Guide

Version 9.0.0, Edition April 2026

Table of Contents

Preface

This manual provides an introduction to Resolution Service as a Windows Service for DXI V9. It consists of the following sections:

- Chapter 1 describes the prerequisites to install the Resolution service.
- Chapter 2 provides guidance to set up Resolution Service as a Windows Service in your environment.
- Chapter 3 provides guidance to set up Resolution Service as a Linux Service in your environment.
- Chapter 4 describes required and optional configuration properties for the Resolution Service.
- Chapter 5 describes how to configure SL4J logging.
- Chapter 6 describes how to configure Log levels for various packages.
- Chapter 7 describes how to manage and archive log files efficiently.
- Chapter 8 describes how to interact with the application's management endpoints.
- Chapter 9 describes how to interact with the application's JMX management endpoints.
- Chapter 10 describes how to disable the existing resolution adapter.
- Chapter 11 describes how to verify consumer in ActiveMq.
- Chapter 12 describes how to handle port conflicts.

DirX Identity Documentation Set

Version: 9.0.0 | Build: 29615 | Date: 2026-04-30

The DirX Identity document set consists of the following manuals:

- *DirX Identity Introduction*. Use this book to obtain a description of DirX Identity architecture and components.
- *DirX Identity Release Notes*. Use this book to understand the features and limitations of the current release. This document is shipped with the DirX Identity installation as the file **release-notes.pdf**.
- *DirX Identity History of Changes*. Use this book to understand the features of previous releases. This document is shipped with the DirX Identity installation as the file **history-of-changes.pdf**.
- *DirX Identity Tutorial*. Use this book to get familiar quickly with your DirX Identity installation.
- *DirX Identity Provisioning Administration Guide*. Use this book to obtain a description of DirX Identity provisioning architecture and components and to understand the basic tasks of DirX Identity provisioning administration using DirX Identity Manager.
- *DirX Identity Connectivity Administration Guide*. Use this book to obtain a description of DirX Identity connectivity architecture and components and to understand the basic tasks of DirX Identity connectivity administration using DirX Identity Manager.
- *DirX Identity User Interfaces Guide*. Use this book to obtain a description of the user interfaces provided with DirX Identity.
- *DirX Identity Application Development Guide*. Use this book to obtain information how to extend DirX Identity and to use the default applications.
- *DirX Identity Customization Guide*. Use this book to customize your DirX Identity environment.
- *DirX Identity Integration Framework*. Use this book to understand the DirX Identity framework and to obtain a description how to extend DirX Identity.
- *DirX Identity Web Center Reference*. Use this book to obtain reference information about the DirX Identity Web Center.
- *DirX Identity Web Center Customization Guide*. Use this book to obtain information how to customize the DirX Identity Web Center.
- *DirX Identity Resolution Service Setup Guide*. Use this book to set up the DirX Identity Resolution Service as a standalone service.
- *DirX Identity Meta Controller Reference*. Use this book to obtain reference information about the DirX Identity meta controller and its associated command-line programs and files.
- *DirX Identity Connectivity Reference*. Use this book to obtain reference information about the DirX Identity agent programs, scripts, and files.
- *DirX Identity Troubleshooting Guide*. Use this book to track down and solve problems in your DirX Identity installation.

- *DirX Identity Installation Guide*. Use this book to install DirX Identity.
- *DirX Identity Migration Guide*. Use this book to migrate from previous versions.
- *DirX Identity REST Service Guide*. Use this book to migrate from previous versions.

Notation Conventions

Boldface type
In command syntax, bold words and characters represent commands or keywords that must be entered exactly as shown.

In examples, bold words and characters represent user input.

Italic type
In command syntax, italic words and characters represent placeholders for information that you must supply.

[ ]
In command syntax, square braces enclose optional items.

{ }
In command syntax, braces enclose a list from which you must choose one item.

In Tcl syntax, you must actually type in the braces, which will appear in boldface type.

|
In command syntax, the vertical bar separates items in a list of choices.

...
In command syntax, ellipses indicate that the previous item can be repeated.

userID_home_directory
The exact name of the home directory. The default home directory is the home directory of the specified UNIX user, who is logged in on UNIX systems. In this manual, the home pathname is represented by the notation *userID_home_directory*.

install_path
The exact name of the root of the directory where DirX Identity programs and files are installed. The default installation directory is *userID_home_directory*/**DirX Identity** on UNIX systems and **C:\Program Files\DirX\Identity** on Windows systems. During installation the installation directory can be specified. In this manual, the installation-specific portion of pathnames is represented by the notation *install_path*.

dirx_install_path
The exact name of the root of the directory where DirX programs and files are installed. The default installation directory is *userID_home_directory*/**DirX** on UNIX systems and **C:\Program Files\DirX** on Windows systems. During installation the installation directory can be specified. In this manual, the installation-specific portion of pathname is represented by the notation *dirx_install_path*.

dxi_java_home
The exact name of the root directory of the Java environment for DirX Identity. This location is specified while installing the product. For details see the sections "Installation" and "The Java for DirX Identity".

tmp_path

The exact name of the tmp directory. The default tmp directory is /tmp on UNIX systems. In this manual, the tmp pathname is represented by the notation *tmp_path*.

tomcat_install_path
The exact name of the root of the directory where Apache Tomcat programs and files are installed. This location is defined during product installation.

mount_point
The mount point for DVD device (for example, **/cdrom/cdrom0**).

1. Resolution Service

1.1. Introduction

The Resolution Service is a standalone system component that provides identity and reference resolution capabilities within the DIRX Identity. It is responsible for processing incoming resolution requests and executing the internal resolution logic.

The Resolution Service serves as a modern replacement for the Resolution Adapter (since DXI version 8.10). It is designed as an independent, standalone service that can be deployed and operated separately from the Java application server. This approach enables customers to execute resolution logic in a more flexible, scalable, and observable manner, while maintaining seamless integration with existing system landscapes.

The service is delivered as a Spring Boot application and can be operated as:

- a Windows service, or
- a Linux daemon.

1.2. Purpose and Value

The primary purpose of the Resolution Service is to decouple resolution logic from the core application server and provide it as a dedicated, reusable system service.

From a system and operational perspective, the Resolution Service offers the following benefits:

Independence and flexibility

The service runs standalone and is not tied to the Java server lifecycle or deployment model.

Operational transparency

Built-in health, metrics, and monitoring endpoints allow administrators to observe system behavior, throughput, and performance.

Modern integration

The service is based on Spring Boot and Spring JMS, enabling easier integration with modern messaging infrastructures and simplifying maintenance and upgrades.

Future-proof architecture

JMS handling, monitoring, and logging are implemented using standard Spring mechanisms, supporting easier migration to future messaging technologies.

1.3. Position in the Overall Architecture

Within the overall system architecture, the Resolution Service acts as an intermediary between the messaging infrastructure and the internal resolution logic.

At a high level, the processing flow is as follows:

1. External systems or internal components send resolution requests via JMS.
2. The Resolution Service listens on the configured message queue or topic.
3. The service processes the request using the internal resolver.
4. Resolution results are applied to DIRX Identity.

The Resolution Service therefore acts as:

- a message-driven resolution engine,
- a boundary between messaging and business logic,
- a central component for identity resolution.

By isolating this responsibility into a dedicated service, the overall system becomes more modular, scalable, and easier to operate.

1.4. Resolution Flow and Message Handling

The task of resolving a user is delegated to the Resolution Service.

The Resolution Service listens for resolution requests on the JMS destination:

```
dxm.request.user.resolve
```

If the configuration parameter `topicIncludesDomain` is set to `true`, the domain name is prefixed to the topic, for example:

```
my-company.dxm.request.user.resolve
```

Each incoming message contains:

- the distinguished name (DN) of the changed user, and
- the timestamp indicating when the change occurred.

1.4.1. Internal Resolution Process

Upon receiving a message, the internal resolver performs the following steps:

1. **Resolution Lock**
   The user is locked using a so-called *resolution lock*. This ensures that no other resolution process can resolve the same user concurrently.
2. **User Resolution**
   The resolver executes the standard resolution logic, which includes:
   - calculating new or removed group assignments,

    - determining required account creations or deletions,
    - applying privilege-based changes.
3. **Concurrency Behavior**
   While a user is locked for resolution:
    - no other resolution process can resolve the same user,
    - other clients may still modify the user (for example by assigning or removing privileges).

This mechanism guarantees consistency of resolution while still allowing concurrent administrative changes.

1.5. When to Use the Resolution Service

Customers and administrators should consider using the Resolution Service when:

- they want to replace the legacy Resolution Adapter,
- they need a standalone resolution component,
- they require improved scalability and monitoring,
- they want to decouple resolution from the application server.

In most setups, the Resolution Service can be used as a direct replacement for the existing Resolution Adapter or as an additional optional component, depending on system architecture and operational requirements.

2. Setting Up the Resolution Service as a Windows Service

This document describes how to set up the Resolution Service as a Windows Service.

2.1. Preparing the Resolution Service

1. **Locate the Installation Files**: Navigate to dxi_install_path\services\resolution.
2. **Extract the ZIP file**:
    - Unzip the resolution-9.0-SNAPSHOT.zip file inside the dxi_install_path\services\resolution to a directory of your choice or in the same folder dxi_install_path\services\resolution.
    - The extracted folder should contain the following files:
        - bin
        - lib
        - application.properties
        - dxiresolution.service
        - logback-spring.xml
        - ResolutionWinService.exe
        - ResolutionWinService.xml

2.2. Prerequisites

- Disable the Resolution Adapter
- Configure the application.properties File
- Open a Command Prompt with administrative privileges in the following directory: dxi_install_path\services\resolution\resolution-9.0-SNAPSHOT\resolution-9.0-SNAPSHOT

Or navigate to the folder where you extracted resolution-9.0-SNAPSHOT.zip, depending on your setup. Once there, run the following commands.

2.3. Installing the Windows Service

- Run the following command to install the service:

```
ResolutionWinService.exe install
```

- After installation, the **Dirx Identity Resolution Service** will appear in the Services window.

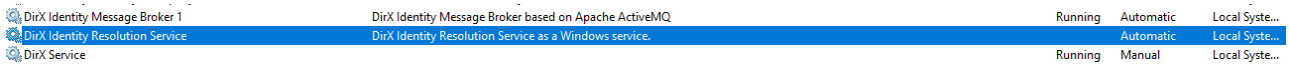

Figure 1. DirX Identity Resolution Service in Windows Services window

2.4. Managing the Windows Service

2.4.1. Starting the service

- Run the following command to start the service:

```
ResolutionWinService.exe start
```

- Open the Windows Services window, refresh it, and verify that the **DirX Identity Resolution Service** is now running

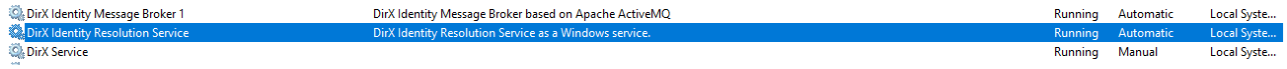

Figure 2. DirX Identity Resolution Service running in Windows Services window

- You can also verify that the Resolution Service is the only consumer by Verifying Consumer in ActiveMQ

2.4.2. Check the Service Status

- Run the following command to check the service status:

```
ResolutionWinService.exe status
```

- The output will show the current status of the service, including whether it is running or stopped.

2.4.3. Stopping the service

- Run the following command to stop the service:

```
ResolutionWinService.exe stop
```

- Alternatively, you can start and stop the service using the Windows Services window.

2.4.4. Uninstalling the service

If you want to uninstall the service first stop the service and then, follow these steps:

- Run the following command to uninstall the service:

```
ResolutionWinService.exe uninstall
```

- The service will be removed from the Windows Services window.

3. Setting up the Resolution Service as a Linux Daemon

This document describes how to set up the Resolution Service as a Linux daemon. To use the resolution service with a DXI linux installation, the process needs to be daemonize-d. It is done by configuring a daemon with a Systemd configuration file.

3.1. Prerequisites

- Disable the Resolution Adapter
- Configure the application.properties File

3.2. Execute the following steps to get the Resolution Service running:

3.2.1. Set Up the systemd Service

The commands bellow must be executed as the `root` user.

- Adjust the file `dxiresolution.service`

Please be aware that before starting the setup, the variables in the file have to be adjusted to Your system accordingly (You can find it under the installation folder of the resolution service (<dxi_install_path>\services\resolution\resolution-9.0-SNAPSHOT e.g))

User: the user that will run the service, it should be the same user that runs the DXI installation

Group: the group that will run the service, it should be the same group that runs the DXI installation

WorkingDirectory: the folder where service was extracted (<dxi_install_path>\services\resolution\resolution-9.0-SNAPSHOT e.g)

ExecStart: the command to start the service, it should be adjusted to the location path of the java version installed on your system.

- Copy the Service File

Copy the `dxiresolution.service` from `[dxi_install_path\services\resolution\resolution-9.0-SNAPSHOT\resolution-9.0-SNAPSHOT]` or from the folder where you extracted resolution-9.0-SNAPSHOT.zip, depending on your setup, to `[<DXI_INST_PATH>/etc]`.

- If you are logged in as `qameta` or Your suer configured in the service file above, use `su` to switch to the `root` user

- Make a softlink under [/etc/systemd/system]:

```
cd /etc/systemd/system
ln -s <DXI_INST_PATH>/etc/dxiresolution.service
```

- reload the systemctl:

```
systemctl daemon-reload
```

- to start the service:

```
systemctl start dxiresolution.service
```

- to see its status (it should be active, in green status):

```
systemctl status dxiresolution.service
```

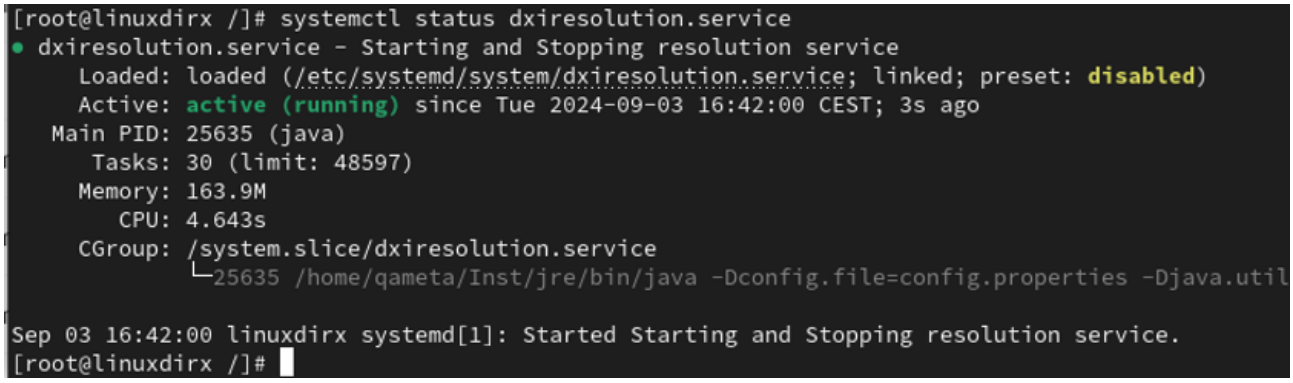

Figure 3. DirX Identity Resolution Service running as a Linux Daemon

- You can also verify that the Resolution Service is the only consumer by Verifying Consumer in ActiveMQ
  - to stop the service:

```
systemctl stop dxiresolution.service
```

- By each modification of the 'dxiresolution.service' file the steps from 'reload' above should be repeated. Do not forget that You have to adjust the copied 'dxiresolution.service' file in <DXI_INST_PATH>/etc.

4. Configuring the application.properties File

This section outlines the required and optional configuration properties for the Resolution Service, defined in the `application.properties` file.

The `application.properties` file is located at:

```
<dxi_install_path>\services\resolution\resolution-9.0-SNAPSHOT\resolution-9.0-SNAPSHOT\
```

4.1. 1. Mandatory Properties

4.1.1. LDAP Configuration

The following LDAP properties must be configured:

- `ldap.host`
- `ldap.port`
- `ldap.ssl`
- `ldap.user`
- `ldap.domain`
- `ldap.clientSSL`

4.1.2. Password Configuration

You must configure one of the following:

- `ldap.password` **OR**
- `pwd.passwordFile` along with `pwd.passwordPropertyKey`

If both `ldap.password` and `pwd.passwordFile` are configured, `ldap.password` takes precedence.

4.1.3. Optional LDAP Properties

- `ldap.keyStore`
- `ldap.keyStorePassword`
- `ldap.trustStore`
- `ldap.trustStorePassword`

- `session.configUpdateFrequency` – update frequency (in minutes) of SvcSession master configuration (e.g., object descriptions)

4.1.4. server port configuration

- `server.port`(default set to 9443) If server.port=9443 is already occupied, you'll need to choose a different available port and update your application's configuration accordingly. Here's how you can detect and set a new server port : Handling Port Conflicts.

4.2. 2. JMS Broker Configuration

If `config.use.ldap=true`, the JMS broker URL is retrieved from LDAP. In this case, setting `spring.activemq.broker-url` is not required.

To override and configure the broker URL directly:

- Set `config.use.ldap=false`
- Set the broker URL:

```
spring.activemq.broker-url=tcp://localhost:61616
```

4.3. 3. Number of Message Listeners

Listener configuration logic:

- If `jms.nrListeners` is set in `application.properties`, it takes precedence.
- If not set, the number of listeners is loaded from LDAP (see the domain entry, tab 'PrivilegeResolution' in Identity Manager, or LDAP attribute `dxr0ptions`, e.g., `resolutionadapter_nrlisteners 2`)
- If not configured in either source, the default value of 2 is used.

4.4. 4. Management and Monitoring Configuration

4.4.1. I. Health Endpoint Exposure

Configure health and monitoring endpoints for observability:

```
management.server.port=8081
management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always
management.endpoint.health.probes.enabled=true
management.endpoints.web.base-path=/resolution-service
```

```
management.endpoints.web.path-mapping.health=healthcheck
```

If port 8081 is already in use, you can select an available port by updating the application's configuration. This ensures the service can start successfully without port conflicts. Here's how you can detect and set a new server port : Handling Port Conflicts

4.4.2. II. JMS Metrics and Prometheus Integration

Enable metrics for JMS and Prometheus:

```
management.metrics.enable-jms=true

# Enable Prometheus metrics export (uncomment to activate)
# management.metrics.export.prometheus.enabled=true
```

4.4.3. III. Application Information

Expose custom metadata via the `/info` endpoint:

```
info.app.name=DirX Identity Resolution Service
info.app.description=DirX Identity Resolution Service as a Windows
service
info.app.version=1.0.0
management.info.env.enabled=true
```

5. Configuring SLF4J Logging

The Spring Boot-based resolver application generates logs from all components, including those using SLF4J (such as the `dirx-dxi-resolution` component) and those relying on Java Util Logging (such as most legacy components like `svclayer` and `storage`). By leveraging any of the options below, you can enable SLF4J logging for components that use Java Util Logging.

5.1. 1. Configure via `application.properties`

This method sets the SLF4J usage flag by reading it from the `application.properties` file and applying it as a system property.

```
# System variable to decide if SLF4J logging should be used
com.siemens.dirxcommon.logging.USE_SLF4J=true
```

5.2. 2. Configure as an Environment Variable

You can define the property in the `ResolutionWinService.xml` file:

```
<env name="com.siemens.dirxcommon.logging.USE_SLF4J" value="true" />
```

5.3. 3. Configure as a JVM System Property

Set the SLF4J property as a JVM argument in the `ResolutionWinService.xml` file:

```
<arguments>
    -Dcom.siemens.dirxcommon.logging.USE_SLF4J=true
</arguments>
```

6. Configuring Log Levels

Log levels for various packages can be configured in the `logback-spring.xml` file. Supported log levels include: `ERROR`, `WARN`, `INFO`, `DEBUG`, `TRACE`, and `OFF`.

6.1. Applicable Packages

Log levels can be set individually for the following packages:

- `solutions.dirx.identity`
- `siemens.dxr.service`
- `siemens.dxm.storage`
- `com.siemens`
- `net.atos.dirx`
- `com.dirxcloud`

6.2. Log Levels Overview

- `ERROR` – Logs errors that typically result in failed resolution.
- `WARN` – Logs potentially harmful situations that may need attention but are not necessarily errors.
- `INFO` – Logs general informational messages to show application progress.
- `DEBUG`, `TRACE`, `ALL` – Logs detailed messages for debugging and flow tracing.
- `OFF` – Disables logging entirely.

Log level configuration is managed centrally in the `logback-spring.xml` file.

6.3. Log Output and Locations

- **Warning Logs**: All warnings for each package are written to the `warning-resolutionservice.log` file.
- **All Logs**: All logs, depending on the configured levels, are written to the `resolutionservice.log` file.

Both files are located in: `<yourFolder>/resolution-9.0-SNAPSHOT/logs/`

7. Log File Management and Archiving

The Resolution Service uses a rolling file appender to manage and archive log files efficiently. This ensures that log files do not grow indefinitely and that older logs are archived systematically.

7.1. Log Archiving Configuration

The `logback.xml` configuration file defines how the log files are rotated, archived, and deleted over time. The following outlines the main log file handling:

7.2. Main Log File: `resolutionservice.log`

7.2.1. 1. Log File Location

The main logs are written to: logs/resolutionservice.log

7.2.2. 2. Rolling Policy

The log file rolls over based on both time and file size, ensuring that the logs are archived regularly.

- **Daily Rolling** – A new log file is created each day, helping to keep logs organized by date.
- **File Size Limit** – If a log file exceeds 10MB within a day, it rolls over to a new file. This prevents any single log from growing too large.

7.2.3. 3. Archived File Pattern

Archived log files are saved in the `logs/archived/` directory using the following naming convention:

```
resolutionservice.yyyy-MM-dd.i.log
```

Where:

- `yyyy-MM-dd` is the date when the log file was created.
- `i` is a counter used when multiple files are created on the same day due to file size limits.

7.2.4. 4. Maximum History

- Up to **60 days** of logs are retained.
- Any log files older than 60 days are automatically deleted.

7.2.5. 5. Total Size Cap

- The total size of all archived log files is capped at **20 GB**.
- If the cap is exceeded, the oldest files are deleted to make room for new ones.

7.3. Warning Log File (`warning-resolutionservice.log`)

7.3.1. 1. Log File Location

Warnings are logged to:

`logs/warning-resolutionservice.log`

7.3.2. 2. Rolling Policy

Similar to the main log file, the warning log file rolls over based on time and size:

- **Daily Rolling** – A new warning log file is created each day.
- **File Size Limit** – A new log file is created if the current warning log exceeds 10MB.

7.3.3. 3. Archived File Pattern

Archived warning log files are stored in the `logs/archived/` directory with the pattern:

`warning-resolutionservice.yyyy-MM-dd.i.log`

7.3.4. 4. Maximum History and Size Cap

- **Maximum History:** Up to 60 days of warning logs are retained.
- **Total Size Cap:** The total size of archived warning logs is capped at 20GB.

7.4. Summary of Log Locations

- **Current Logs:**
- `logs/resolutionservice.log`
- `logs/warning-resolutionservice.log`
- **Archived Logs:**
- `logs/archived/`
    - `resolutionservice.yyyy-MM-dd.i.log`
    - `warning-resolutionservice.yyyy-MM-dd.i.log`

This log management strategy helps keep the logs organized, manageable, and prevents excessive disk usage by automatically archiving and cleaning up old log files.

8. Management Endpoints

The application exposes various management endpoints for health checks, application info, metrics, and more. These endpoints are accessible at the base path `/resolution-service` on the port configured via `management.server.port` in `application.properties`(default is configured as port 8081).

To interact with the application's management endpoints, ensure the Resolution Service is up and running. You can test the endpoints using a command-line tool like `curl` or a REST client like Postman.

8.1. Available Endpoints

8.1.1. 1. Health Check Endpoint

Returns the health status of the application.

URL: `http://localhost:8081/resolution-service/healthcheck`

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/healthcheck'
```

8.1.2. 2. Application Info Endpoint

Displays metadata about the application, such as its name, description, and version.

URL: `http://localhost:8081/resolution-service/info`

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-service/info'
```

8.1.3. 3. Metrics Endpoint

Provides a list of application metrics, useful for monitoring system performance.

URL: `http://localhost:8081/resolution-service/metrics`

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-service/metrics'
```

8.1.4. 4. Beans Endpoint

Displays all Spring Beans currently loaded in the application context.

URL: `http://localhost:8081/resolution-service/beans`

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-service/beans'
```

8.1.5. 5. Loggers Endpoint

Lists loggers and their current logging levels, allowing runtime adjustments if necessary.

URL: `http://localhost:8081/resolution-service/loggers`

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-service/loggers'
```

9. JMX Monitoring

The Resolution Service exposes key runtime metrics via Spring Boot Actuator endpoints using Micrometer Gauges. These metrics reflect internal counters such as redelivered, failed, and successfully resolved messages.

9.1. Accessing Metrics

All metrics are available at:

`http://<host>:<port>/resolution-service/metrics`

Replace `<host>` and `<port>` with the actual service URL.The port configured via `management.server.port` in `application.properties`(default is configured as port 8081).

9.2. Available Metrics

9.2.1. 1. Resolver Messages Received

To view how many messages were received:

URL: http://localhost:8081/resolution-service/metrics/resolver.messages.received

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/metrics/resolver.messages.received'
```

9.2.2. 2. Resolver Messages Succeeded

To view how many messages were successfully processed:

URL: http://localhost:8081/resolution-service/metrics/resolver.messages.succeeded

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/metrics/resolver.messages.succeeded'
```

9.2.3. 3. Resolver Messages Failed (Permanent)

To view how many messages failed permanently:

URL: http://localhost:8081/resolution-service/metrics/resolver.messages.failedFinal

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/metrics/resolver.messages.failedFinal'
```

9.2.4. 4. Resolution Messages Redelivered

To view how many messages were redelivered:

URL: http://localhost:8081/resolution-service/metrics/resolution.messages.redelivered

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/metrics/resolution.messages.redelivered'
```

9.2.5. 5. Resolution Messages Failed (Temporary)

To view how many messages temporarily failed:

URL: http://localhost:8081/resolution-service/metrics/resolution.messages.failedTemporary

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/metrics/resolution.messages.failedTemporary'
```

9.2.6. 6. Resolution Messages Ignored

To view how many messages were ignored/skipped:

URL: http://localhost:8081/resolution-service/metrics/resolution.messages.ignored

Example `curl` command:

```
curl --location 'http://localhost:8081/resolution-
service/metrics/resolution.messages.ignored'
```

10. Disabling the Resolution Adapter for Standalone Use

- The resolution service consumes resolve messages too, therefore the resolution adapter shall be deactivated to be able to test the resolution service. This section describes how to disable the Resolution Adapter before starting the `Dirx Identity Resolution Service` .

10.1. Disabling the Resolution Adapter

You can disable the Resolution Adapter in one of the following ways:

Prerequisites : Stop the DXI Java server before making any changes.

Option 1: Rename the Adapter Folder

Rename the folder:

```
dxi_install_path/ids-j-<DOMAIN-
NR>/extensions/net.atos.dirx.dxi.adapter.resolution
```

to:

```
dxi_install_path/ids-j-<DOMAIN-
NR>/extensions/OFF_net.atos.dirx.dxi.adapter.resolution
```

Option 2: Move the Adapter Folder

Move the `[net.atos.dirx.dxi.adapter.resolution]` folder out of the extensions directory entirely:

```
mv dxi_install_path/ids-j-<DOMAIN-
NR>/extensions/net.atos.dirx.dxi.adapter.resolution
/some/backup/location/
```

After Disabling

Once you've disabled the adapter:

1. Start the DXI Java server.
2. Ensure the Resolution Adapter is no longer active.

11. Verifying the Consumer in ActiveMQ

You can verify that the **Resolution Service** is the only consumer of the `resolve` queue by following these steps:

Open the ActiveMQ Web Console.

- Access the **ActiveMQ Web Console** using the following URL:

```
http://<hostname>:<port>/admin
```

- To directly view the list of queues, navigate to:

```
http://<hostname>:<port>/admin/queues.jsp
```

Replace `<hostname>` and `<port>` with the appropriate values for your ActiveMQ instance.

Locate the `resolve` queue.

Confirm that only one consumer is listed: - This should be the **Dirx Identity Resolution Service**.

Figure 4. ActiveMQ Web Console

This confirms that no other service (such as the original adapter) is competing for messages on the `resolve` queue.

12. Handling Port Conflicts

If the configured port for the service (`server.port`) or for management endpoints (`management.server.port`) is already in use, the service may fail to start. Follow the steps below to resolve the issue for either port.

12.1. Step 1: Check If a Port Is Occupied

Use the following commands to check if the specific port (e.g., 9443 or 8081) is currently in use:

On **Windows (Command Prompt)**:

```
netstat -aon | findstr :<port>
```

On **Linux/macOS**:

```
lsof -i :<port>
```

Replace `<port>` with the port number you're troubleshooting (e.g., 9443 or 8081).

If the port is in use, these commands will show the process ID occupying it. You can then:

- Kill the process (if safe), or
- Change your application's configuration to use a different free port.

12.2. Step 2: Choose a New Available Port

Common alternative ports for HTTPS include 9444, 8443, 10443, etc. Pick a port that is not currently in use.

To check if a port (e.g., 9444) is available, run:

```
nc -zv localhost 9444
```

If the port is available, it will return "Connection refused" or "succeeded" depending on the tool used — either way, it indicates the port is free.

12.3. Step 3: Update Application Configuration

Update your application configuration to use the new port.

12.3.1. In `application.properties`

To update the **main server port**:

```
server.port=9444
```

To update the **management endpoint port**:

```
management.server.port=8090
```

12.4. Step 4: Restart the Resolution Service

After updating the port configuration, restart the Resolution Service. The application should now start successfully using the new port(s).

DirX Product Suite

The DirX product suite provides the basis for fully integrated identity and access management; it includes the following products, which can be ordered separately.

DirX Identity

DirX Identity provides a comprehensive, process-driven, customizable, cloud-enabled, scalable, and highly available identity management solution for businesses and organizations. It provides overarching, risk-based identity and access governance functionality seamlessly integrated with automated provisioning. Functionality includes lifecycle management for users and roles, cross-platform and rule-based real-time provisioning, web-based self-service functions for users, delegated administration, request workflows, access certification, password management, metadirectory as well as auditing and reporting functionality.

DirX Directory

DirX Directory provides a standards-compliant, high-performance, highly available, highly reliable, highly scalable, and secure LDAP and X.500 Directory Server and LDAP Proxy with very high linear scalability. DirX Directory can serve as an identity store for employees, customers, partners, subscribers, and other IoT entities. It can also serve as a provisioning, access management and metadirectory repository, to provide a single point of access to the information within disparate and heterogeneous directories available in an enterprise network or cloud environment for user management and provisioning.

DirX Access

DirX Access is a comprehensive, cloud-ready, scalable, and highly available access management solution providing policy- and risk-based authentication, authorization based on XACML and federation for Web applications and services. DirX Access delivers single sign-on, versatile authentication including FIDO, identity federation based on SAML, OAuth and OpenID Connect, just-in-time provisioning, entitlement management and policy enforcement for applications and services in the cloud or on-premises.

DirX Audit

DirX Audit provides auditors, security compliance officers and audit administrators with analytical insight and transparency for identity and access. Based on historical identity data and recorded events from the identity and access management processes, DirX Audit allows answering the “what, when, where, who and why” questions of user access and entitlements. DirX Audit features historical views and reports on identity data, a graphical dashboard with drill-down into individual events, an analysis view for filtering, evaluating, correlating, and reviewing of identity-related events and job management for report generation.

For more information: support.dirx.solutions/about

Legal remarks

On the account of certain regional limitations of sales rights and service availability, we cannot guarantee that all products included in this document are available through the Eviden sales organization worldwide. Availability and packaging may vary by country and is subject to change without prior notice. Some/All of the features and products described herein may not be available locally. The information in this document contains general technical descriptions of specifications and options as well as standard and optional features which do not always have to be present in individual cases. Eviden reserves the right to modify the design, packaging, specifications and options described herein without prior notice. Please contact your local Eviden sales representative for the most current information. Note: Any technical data contained in this document may vary within defined tolerances. Original images always lose a certain amount of detail when reproduced.