EVIDEN

Identity and Access Management

DirX Identity

Installation Guide

Version 9.0.0, Edition July 2026

Table of Contents

Preface

This manual describes how to install and configure DirX Identity. It consists of the following chapters:

- Chapter 1 provides a summary of the installation procedures and their requirements.
- Chapter 2 describes how to install DirX Identity on a single machine.
- Chapter 3 describes the DirX Identity installation procedure.
- Chapter 4 describes the DirX Identity configuration procedure.
- Chapter 5 describes how to install single components.
- Chapter 6 describes other installation configurations.
- Chapter 7 describes how to install the DirX Identity Windows Password Listener.
- Chapter 8 describes how to deploy the JMS-Audit handler into a DirX Identity installation.
- Chapter 9 provides detailed information about the Java environment for DirX Identity.
- Chapter 10 explains additional topics
- Appendix A describes how to set up Windows single sign-on based on Kerberos.

DirX Identity Documentation Set

Version: 9.0.0 | Build: 29615 | Date: 2026-07-14

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. Introduction

DirX Identity provides installation procedures to set up a complete DirX Identity environment. These procedures are divided into two main steps:

1. **Installation**

   This step copies the required files to the file system and updates registry entries (on Windows) and environment variables (which requires a restart on Windows). The installation process supports both local and distributed DirX Identity systems.

2. **Configuration**

   After installation, the configuration process extends the directory server schema for DirX Identity, loads initial data into DirX Identity domains, and configures all other components. The configuration procedure is tailored for local or distributed systems.

   See the section Schema and Content Handling for details.

Installation and configuration are clearly separated. First, select all components needed on a specific machine. Then, configure these components using the **Configuration Wizard**.

This chapter provides general information about installation and configuration. It does **not** cover distributed environment installation.

1.1. Version Updates and Service Packs

DirX Identity regularly introduces new versions to deliver enhancements and improvements. Between major releases, service packs are available to provide additional updates, which may include:

- New functionality for DirX Identity Manager
- Extended connectivity or provisioning configurations with new default applications
- Updated or new agents and connectors
- Documentation and help enhancements
- Fixes for critical bugs

For the latest service pack information, contact your support organization.

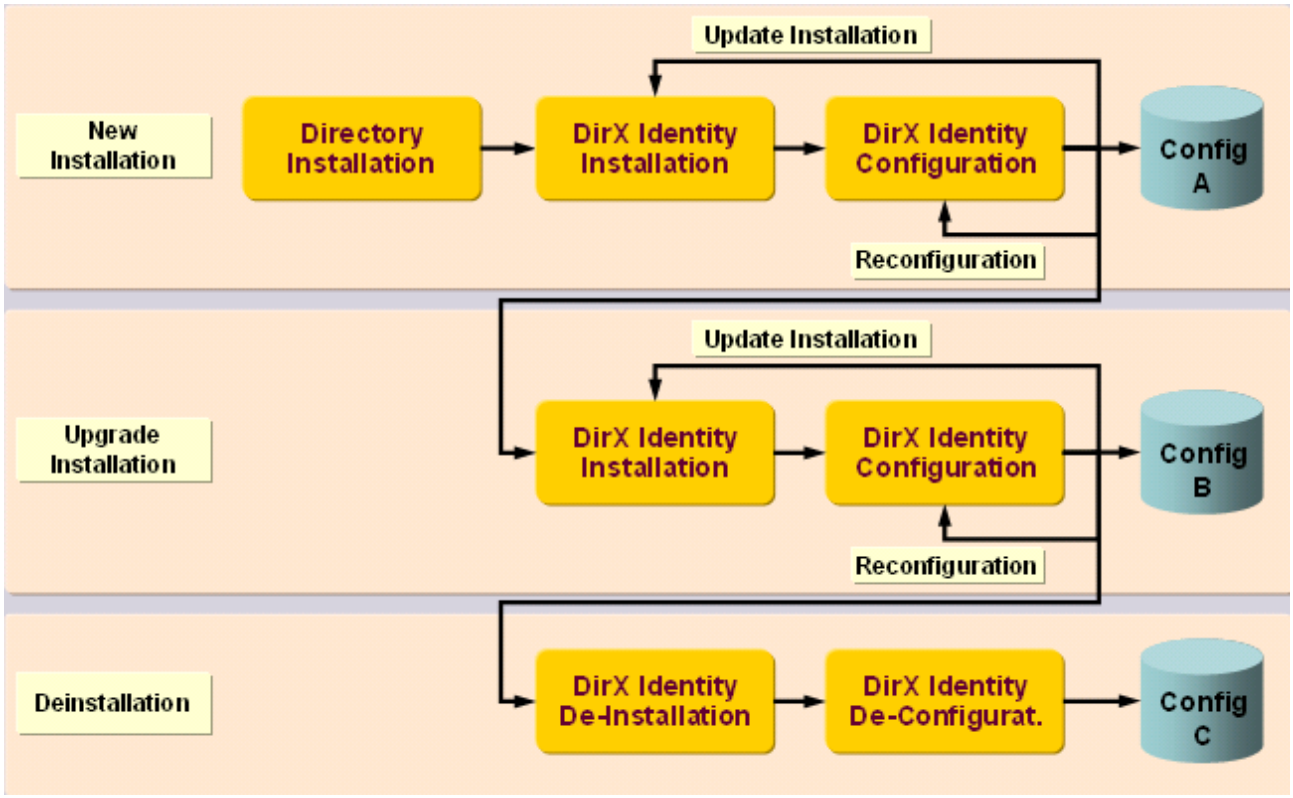
1.2. General Information

This section applies to all DirX Identity installation procedures.

1.2.1. Supported Use Cases

- **New Installation**: Install a supported directory server, then perform DirX Identity installation and configuration.

- **Update Installation**: Refresh or repair existing installations. Backup and restore files as described in the Migration Guide.
- **Upgrade Installation**: Install a new version and run configuration for automatic migration. Manual steps are detailed in the Migration Guide.
- **Uninstallation**: Remove all installed software using the uninstallation and unconfiguration routines. LDAP configuration data remains intact.



1.2.2. Supported Platforms

- **Directory Servers**: DirX Directory Server (see Release Notes for version details)
- **Operating Systems**: Microsoft Windows and Linux (see Release Notes for prerequisites and limitations)

1.2.3. Disk Space Requirements

- Temporary: 2300 MB
- Complete installation: Minimum 2500 MB

  Additional space is required for data and log files.

  See section Disk Space Calculation.

1.2.4. Firewall Configuration

Open the default ports for components you use. For example:

Service	Non SSL	SSL
Apache ActiveMQ broker	61616	61617
Apache ActiveMQ admin Web console	8161	8161
Apache ActiveMQ (JMX)	10098+10099	10098+10099
C++-based Server (SPML Service)	9900	9901
C++-based Server (proprietary JMX)	5315	5315
Java-based Server (HTTP/HTTPS Web services)	40_n_00	40_n_00
Java-based Server (JMX)	40_n_05+40_n_06	40_n_05+40_n_06
Tomcat deployments (defaults)	8080	8443

- Ports can be changed during configuration; update firewall rules accordingly.
- For IdS-J server, *n* = 0 for S1, 1 for S2, etc.
- LDAP defaults: 389 (Non SSL), 636 (SSL).
- Avoid port conflicts (e.g., Tomcat vs DirX Directory REST service).
- JMX uses two ports: second = first + 1.
- Additional ports may be required for connectors/agents.
- C++ server uses port 1111 for local key transfer (do not open in firewall).
- Windows ephemeral ports: 49152–65536.

1.2.5. Limitations

- Installation must be performed locally (remote installation is not supported)
- In distributed environments, update **all machines** to avoid interworking issues
- Before updates or upgrades:
  - Stop all workflows and services
  - Stop all DirX Identity services and UI components, including Tomcat

1.2.6. Deploying Web Services

DirX Identity provides Web Services deployable in Apache Tomcat:

- Stand-alone Tomcat or embedded in IdS-J server (default port 40000).
- Embedded deployments start/stop with IdS-J server.
- External Tomcat service operation is out of scope.

1.2.7. Linux Configuration Account Requirements

Superuser permissions are sufficient. If using DirX Identity installation account:

- **Tomcat installation conditions**

    - Account must have read/write/execute access to Tomcat directory.
    - Required subdirectories: `conf/Catalina/localhost`, `work/Catalina/localhost`.
    - Permissions: 775 (shared group) or 777 (otherwise).
- **LDAP directory conditions**:
    - Superuser required for configuration.
    - DirX home directory must be readable/executable for group members.
    - Define Linux group including DirX and DirX Identity accounts.

1.2.8. Accounts for Tomcat and DirX Identity (Linux Only)

Tomcat account must have read/write access to `password.properties` files of Web Apps (e.g., Web Center). Options:

- Install Tomcat as superuser.
- Define shared Linux group.
- Install Tomcat under DirX Identity account.

2. Installation on a Single Machine

This section explains how to install all DirX Identity components on a single machine, suitable for evaluation or small-scale environments.

2.1. Prerequisites

- Supported operating system and hardware requirements
- Java version compatible with your licensed features
- Administrative privileges on the machine

2.2. Installation Steps

1. **Install a directory server** Use native tools for your chosen directory server (e.g., LDAP). Ensure it is running before proceeding.
2. **Install Apache Tomcat** Download and install Tomcat. On Windows, select **Windows Service**. Refer to Supported Apache Tomcat installations in Release Notes for version compatibility.
3. **Run DirX Identity installation**

   Launch the installer and, in the **Choose Licensed Features Set** dialog, select the features you have licensed.

4. **Configure DirX Identity**

   Follow the configuration procedure to set up database connections and system parameters.

5. **Extend the directory schema** Apply schema extensions required for DirX Identity functionality.

2.3. Verification

Start the services and access the DirX Identity web interface to confirm successful installation.

3. Installing DirX Identity

This chapter explains how to install and uninstall DirX Identity.

To start the DirX Identity Installer, you need Java VM version 21 or newer. If Java is already set in your **Path** environment variable, you do not need to select it explicitly. We recommend using the default installation folder.

To specify a Java VM manually, use the `LAX_VM` command-line argument:

```
dirxidty.exe LAX_VM "<FULL_PATH_TO_JAVA_VM>"
```

With an update or upgrade installation the folder *<DXI_INSTALL_PATH>*/security will be deleted because the files are of no use anymore. Do not forget to make an installation folder backup.

3.1. Installation

This section describes how to run the DirX Identity installation procedure to install the DirX Identity software on a machine.

3.1.1. General Remarks

or every successful installation or uninstallation, a record is written to: *<DXI_INSTALL_PATH>***/install_history.txt**

This file includes:

- Date of installation / uninstallation
- DirX Identity version
- Name of user performing the installation / uninstallation
- Installed components

Log files:

- Installation: *<DXI_INSTALL_PATH>*`/`***DirX_Identity***`<VERSION>_Install_<TIMESTAMP>.log`
- Uninstallation: *<DXI_INSTALL_PATH>*`/`***DirX_Identity***`<VERSION>_Uninstall_<TIMESTAMP>.log`

After installation, store any configuration you want to keep in the `/conf/` folder. This folder is preserved during updates. See Configurable Content for details.

3.1.2. Windows Instructions

Windows introduces **User Access Control (UAC)**, which requires explicit confirmation for administrative tasks. Because this confirmation cannot occur during Configurator execution, do one of the following:

- Add the user running Configurator to the local administrator group.
- Start the application with **Run as Administrator**.
- Disable UAC during configuration.

Access rights:

- You may need explicit read/write permissions on installation subfolders.
- Options:
  - Install under a custom path (e.g., `C:\My_Program Files...`).
  - Use default path and set permissions manually (Properties → Security → Permissions).
  - Open a command prompt with administrative rights for file edits.

3.1.3. Linux Instructions

Ensure your Linux installation includes all required packages for DirX Identity. We recommend using an Xorg-based interface (e.g., GNOME Classic) instead of Wayland for better Java Swing compatibility.

Installation folder: *<DXI_INSTALL_PATH>*`/DirX/Identity`

Use an account different from the DirX Directory Server account. The user ID must exist before installation.

Two installation modes:

- Graphical (GUI)
- Console (default)

During GUI installation:

- Click **Cancel** anytime to exit.
- Click **Previous** to return to the previous step.

3.1.4. Prerequisites for First Installation

Before installing:

- Install required Java version (see Java Requirements).
- Install required Tomcat version if using web applications/services (see Tomcat Requirements).

If Tomcat is running, stop it manually before installation. The installer stops DXI services automatically, but Tomcat can only be stopped if its service name is default (e.g., **Tomcat11**) and only on Windows

3.1.5. Starting the Installation

3.1.5.1. Windows:

Default installation path: `%ProgramFiles%\DirX\Identity`

Steps:

1. Locate `dirxidty.exe` and run as Administrator.
2. Choose installation mode (default: GUI).

3.1.5.2. Linux:

Steps:

1. Log in as Linux user.
2. Mount installation media.
3. Open shell and navigate to: `cd /<MOUNT_POINT>/Installation/DirXIdentity/Linux/Server`
4. Run `dirxidty.bin` (default: **Console** mode, see installation mode)

3.2. Update / Upgrade Installation

Starting with version 8.10.3, full installer packages are delivered. Update or upgrade installation is supported from version 8.7 and later. For details on migrating from older versions, see Migration Guide.

You can perform an update or upgrade at any time.

Default applications are overwritten during an update. Do not use modified default applications in production environments.

During an update, the entire *<DXI_INSTALL_PATH>* folder is replaced, except for files and folders listed in Configurable Content

When upgrading DirX Identity from version 8.x to 9.0, certain configuration settings for **Provisioning** (including **SMTP** and related communication parameters), **Workflows**, **Services** are overwritten by common provisioning factory data (not only the sample domain).

Before starting the upgrade, it is recommended to back up all customer -specific configuration parameters in these areas.

> **After the upgrade**, please review and, if necessary, re-apply your customer -specific settings for these areas (for example SMTP server, ports, authentication and other mail service options, workflow-related parameters, and service endpoints) to ensure correct system operation.

3.2.1. Prerequisites for Update / Upgrade Installation

Before updating:

- Review migration-guide:ch3_preparing.pdf for files not preserved automatically.
- **Backup** all DirX Identity databases.
- **Backup** the installation folder.
- Stop all workflows (disable scheduling, shut down event managers).
- Stop all DirX Identity services and UI components, including Tomcat services.
- Install required Java version.
- Install required Tomcat version if using web applications/services.

The installer attempts to stop DXI services automatically. Tomcat can only be stopped if its service name is default (e.g., **Tomcat11**) and only on Windows. On **Linux**, stop Tomcat manually before starting the installation.

3.2.2. Starting the Update / Upgrade Installation

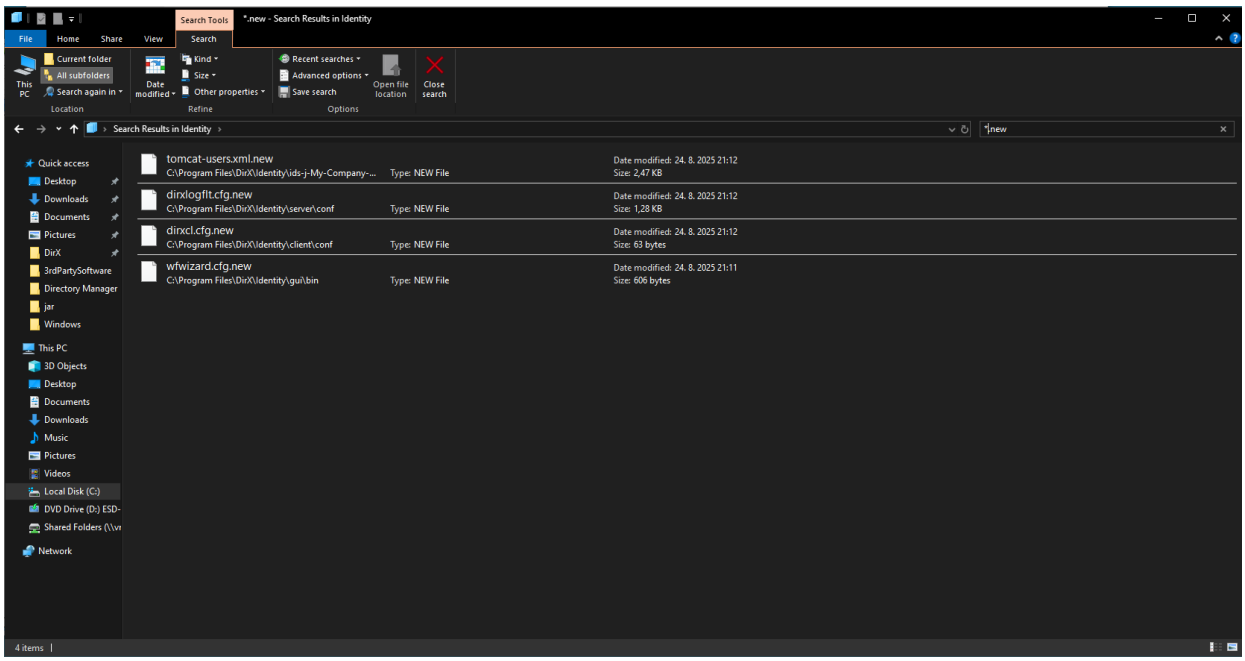
Two options:

1. Unconfigure and uninstall DirX Identity, then perform a fresh installation and configuration.
2. Install over the existing installation. Select all previously installed components. Install over the existing installation. Select all previously installed components.

3.2.3. After the Update / Upgrade Installation

After updating:

- Search for files with `.new` suffix in *<DXI_INSTALL_PATH>*. These files appear when conflicts occur during backup restoration. Review and merge changes as needed. Backup originals before overwriting.



Example of files with .new suffix after update from 8.10.12 to 9.0.0 on Windows

- Run the Configuration Wizard:
  - Mandatory options: “Connectivity Schema and Data Configuration” and “Provisioning Schema and Data Configuration” (if DirX Directory server is installed).
  - Configure each ActiveMQ Message Broker.
  - Configure the primary C++ server (first installed in distributed environments).
  - Configure each domain (includes automatic migration).
- Update Workflow Starter, Report Tool, or Eventing Tool if previously used.

Running the "Configuration Wizard" after an update is **not** the same as “Initial Configuration” (see Configuration Wizard configure command

3.3. Installation Modes

The installer supports multiple modes, selectable via the `-i` argument or the `INSTALLER_UI` property. Refer to InstallAnywhere documentation for all command-line options.

Installer is also referred to as **Setup** in the documentation (based on InstallAnywhere terminology).

3.3.1. Graphical User Interface (GUI) Mode

Runs the installer with a wizard-style interface.

Windows

```
dirxidty.exe -i swing
```

Linux

```
sh ./dirxidty.bin -i gui
```

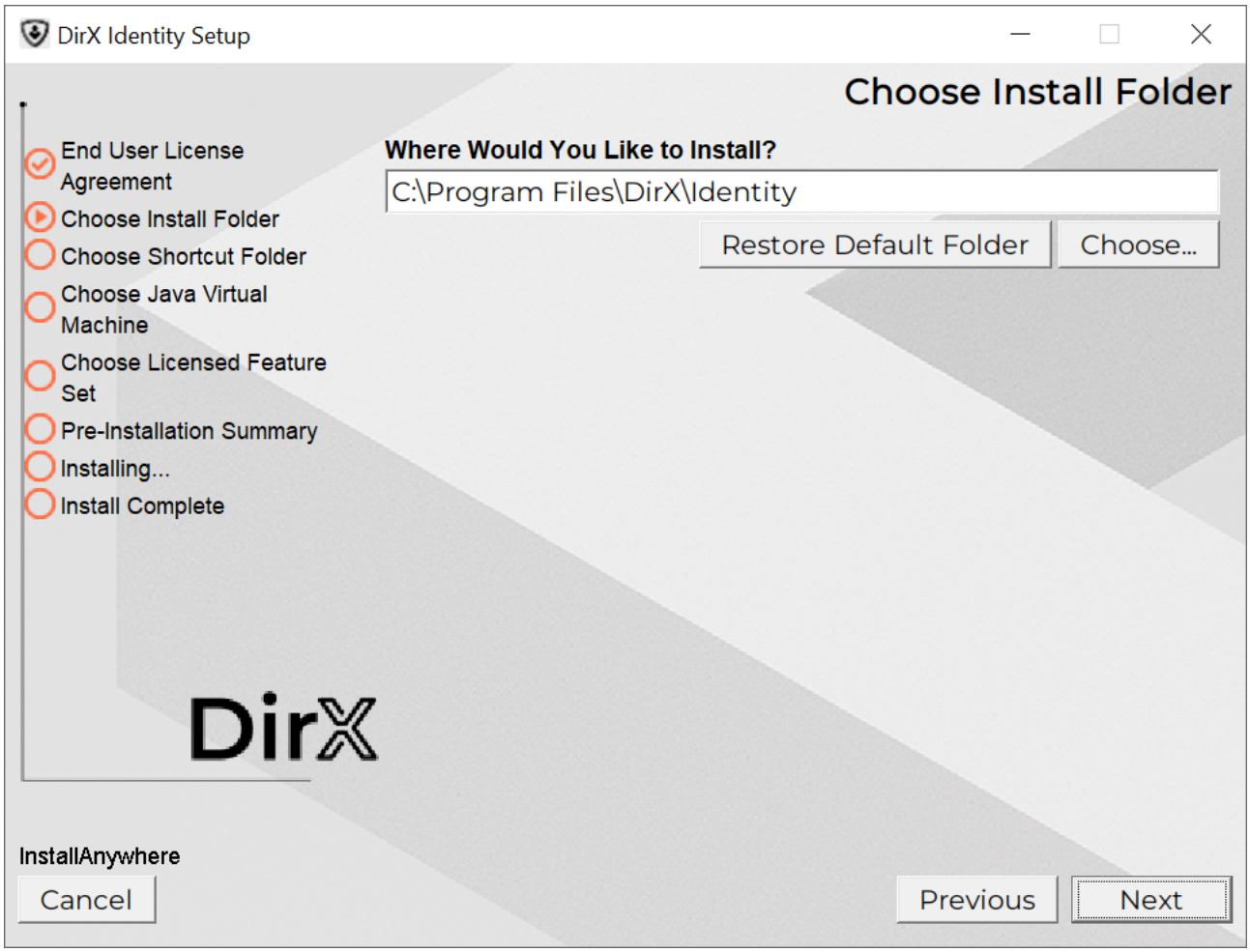
Default mode for Windows. Only essential properties are shown.

3.3.1.1. Steps

3.3.1.1.1. End User License Agreement

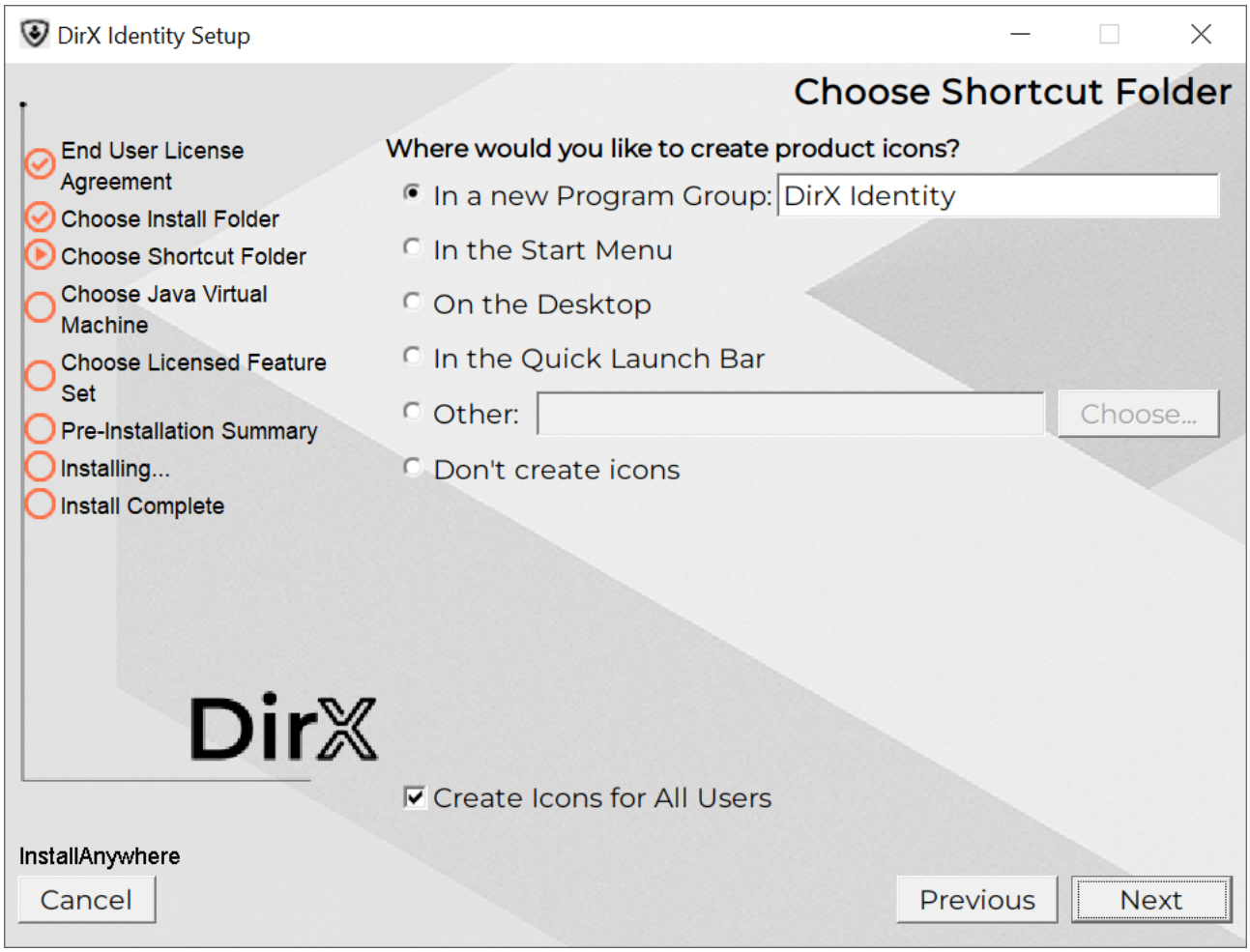
Read the license, accept terms, and click **Next**.

3.3.1.1.2. Choose Install Folder

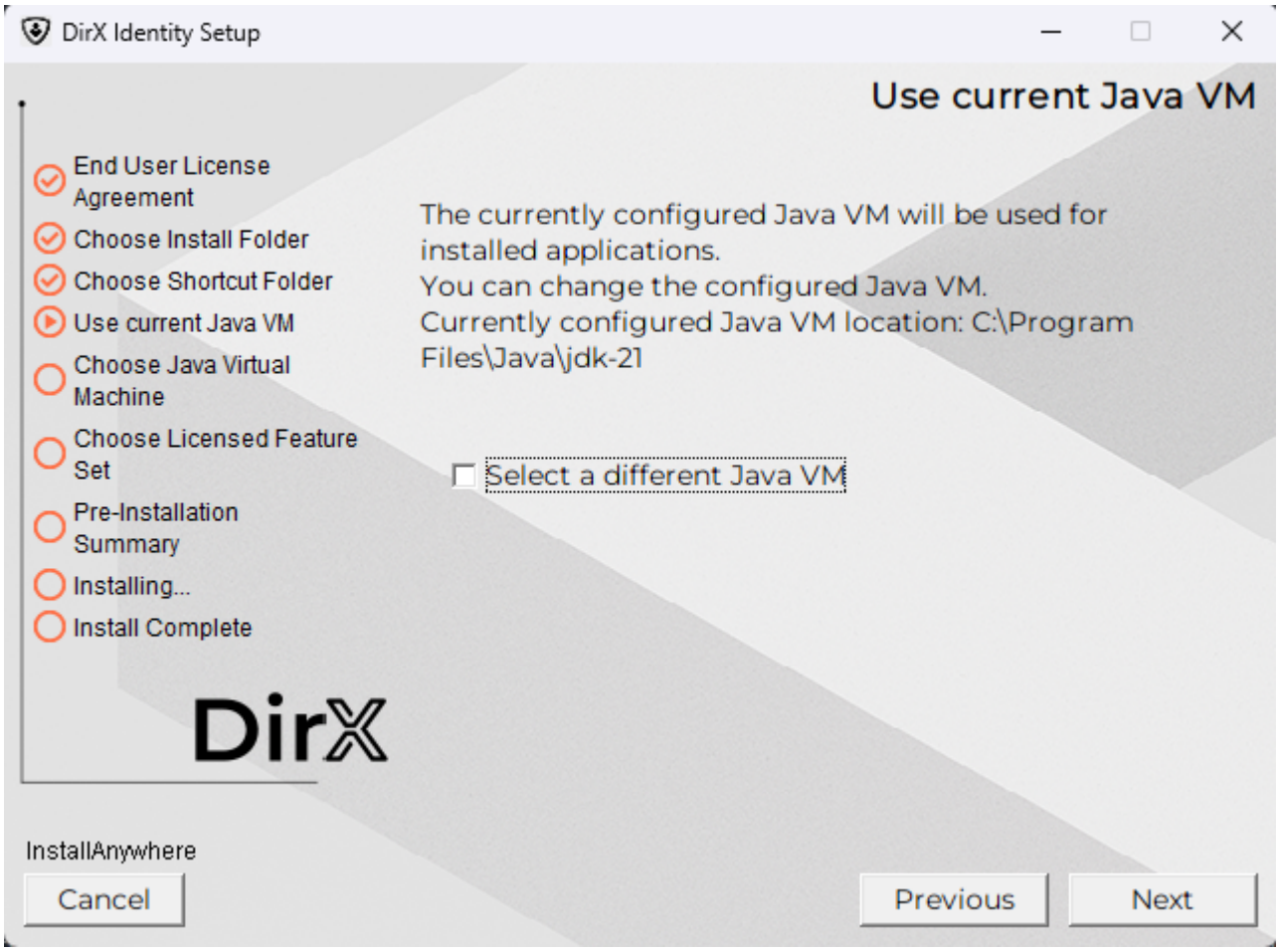


Select default or custom installation folder, and click **Next**. Default: `%ProgramFiles%\DirX\Identity`

3.3.1.1.3. Choose Shortcut Folder

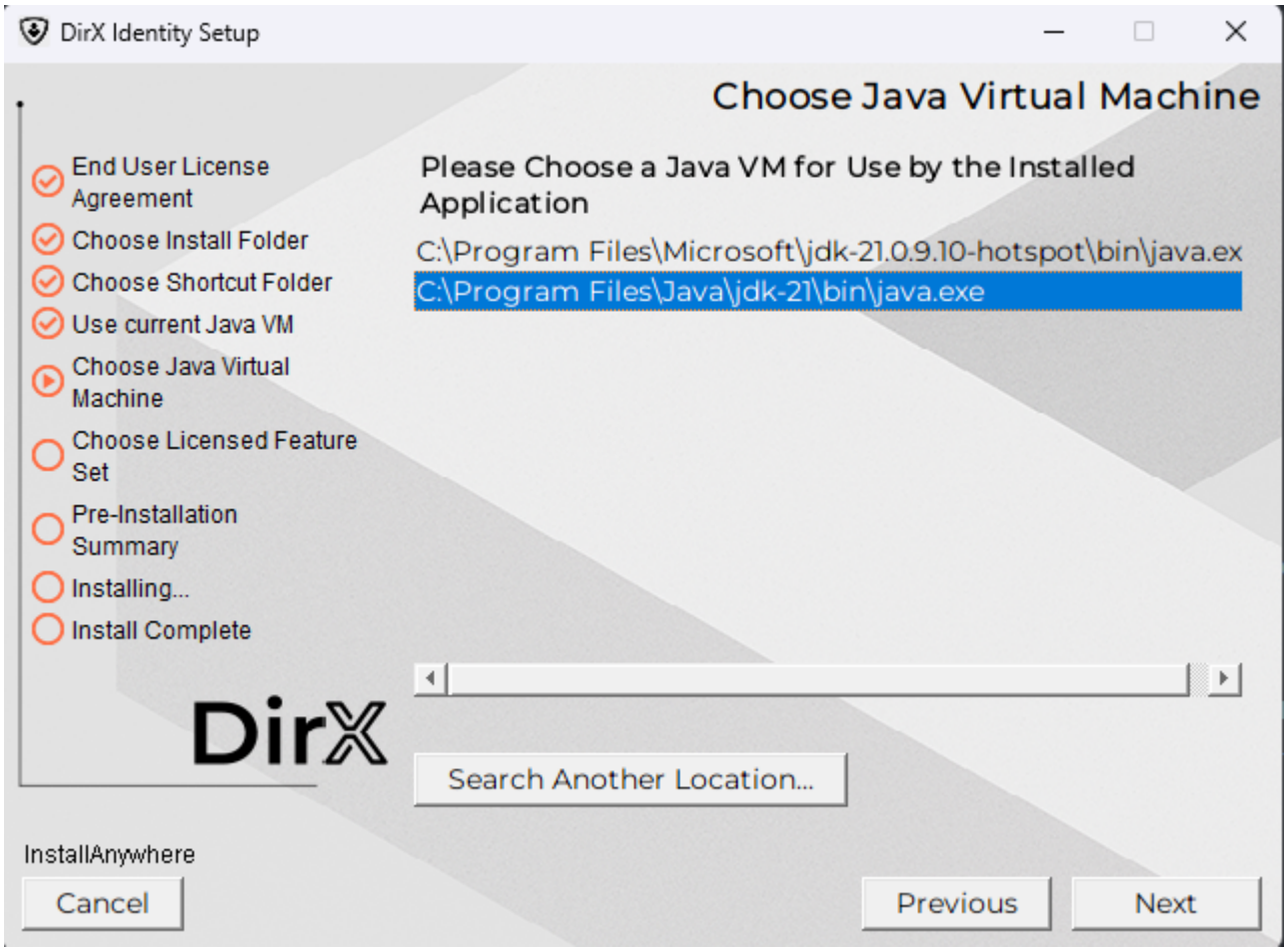


3.3.1.1.4. Use Current Java VM



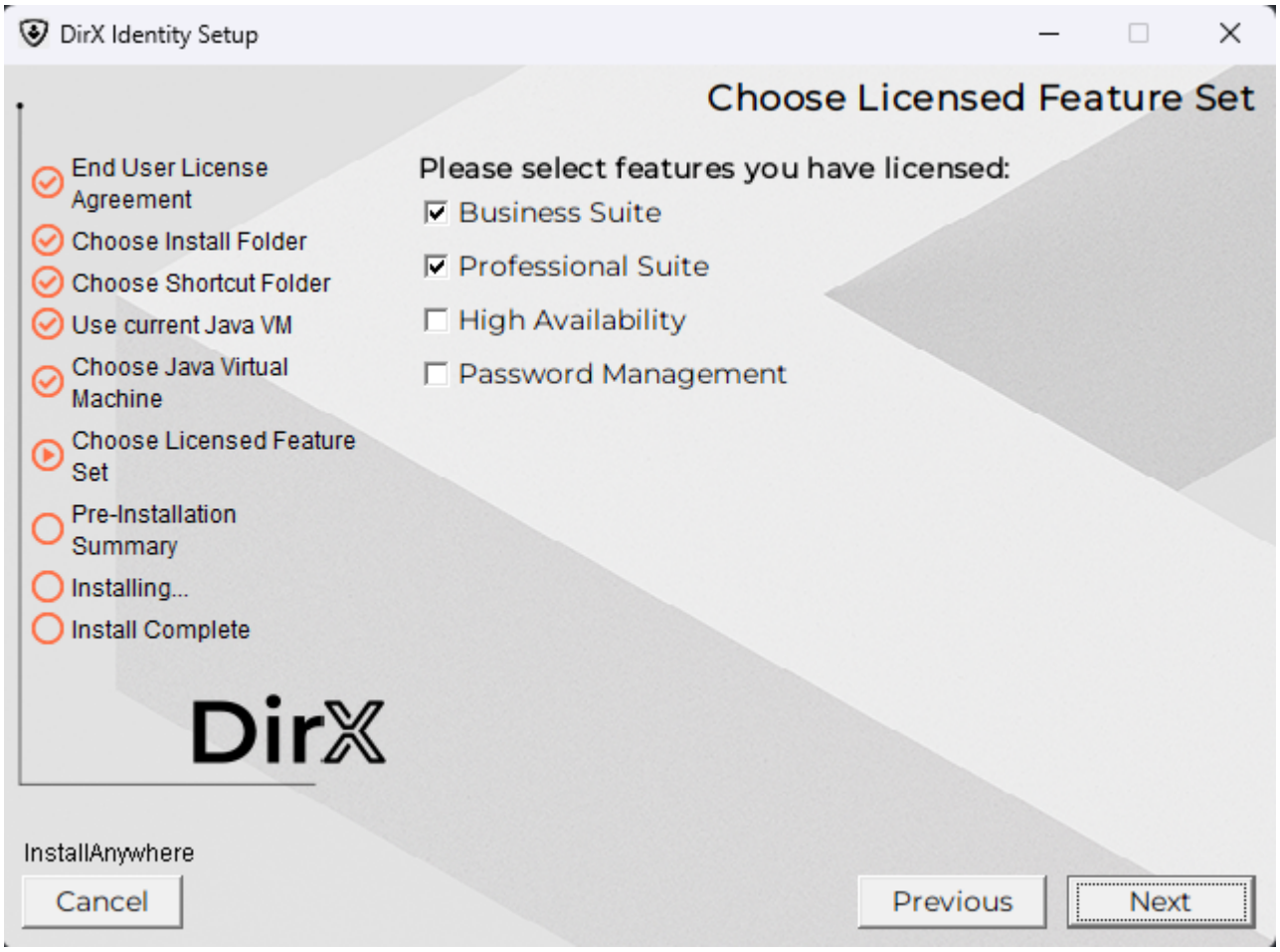
If an existing Java VM meets requirements, you can reuse it. Otherwise, select a different VM to proceed to Choose Java Virtual Machine. Click **Next**.

3.3.1.1.5. Choose Java Virtual Machine



Select Java 21 VM for DirX Identity processes not hosted by Tomcat, and click **Next**.

3.3.1.1.6. Choose Licensed Features Set

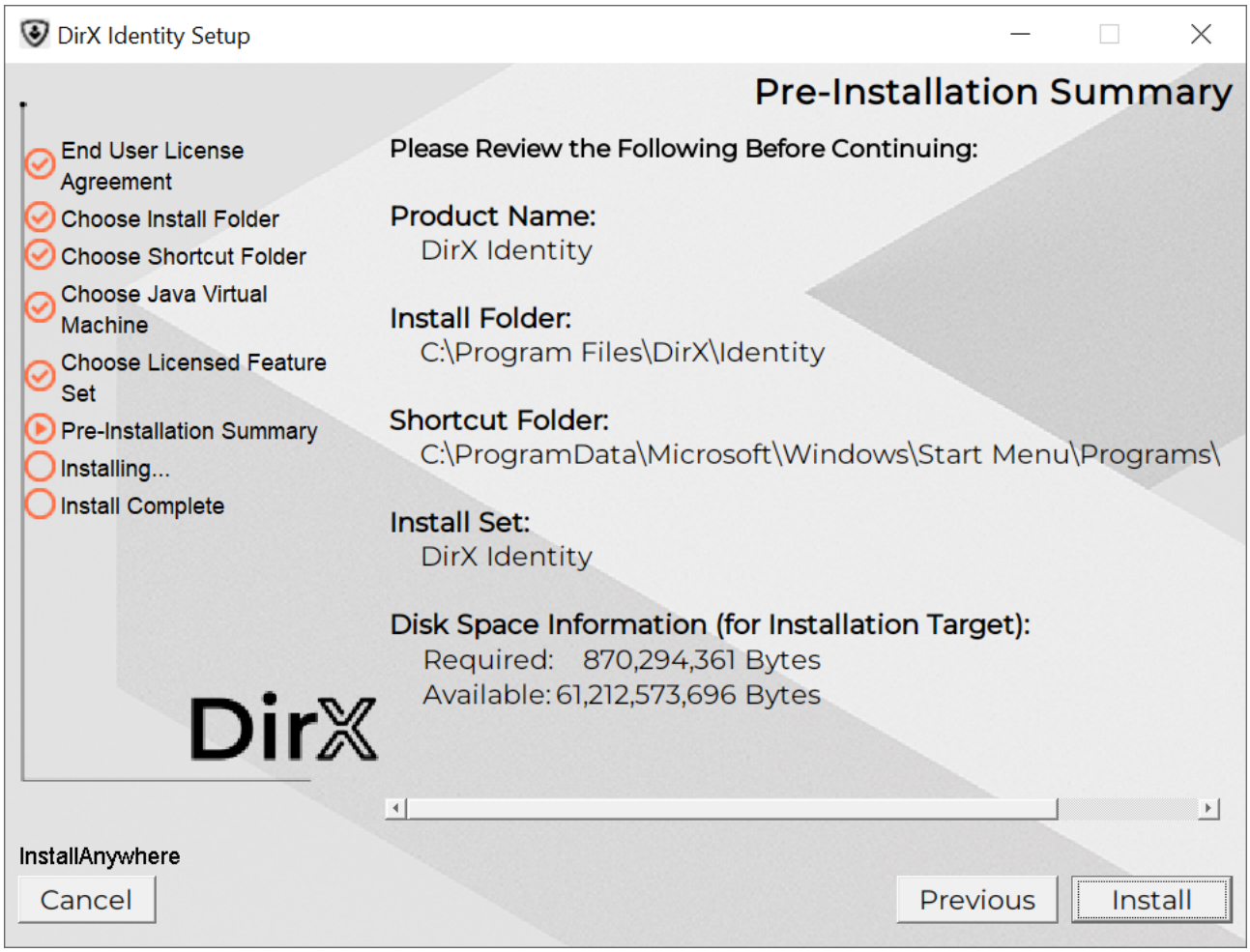


Select features based on your license:

- **Business Suite** – Core functionality and basic connectivity.
- **Provisioning Suite** – Adds provisioning capabilities.
- **High Availability** – Adds HA components.
- **Password Management** – Includes password-related components.

Review carefully. Changes require restarting the installation. Click **Next** to confirm your selection.

3.3.1.1.7. Pre-Installation Summary

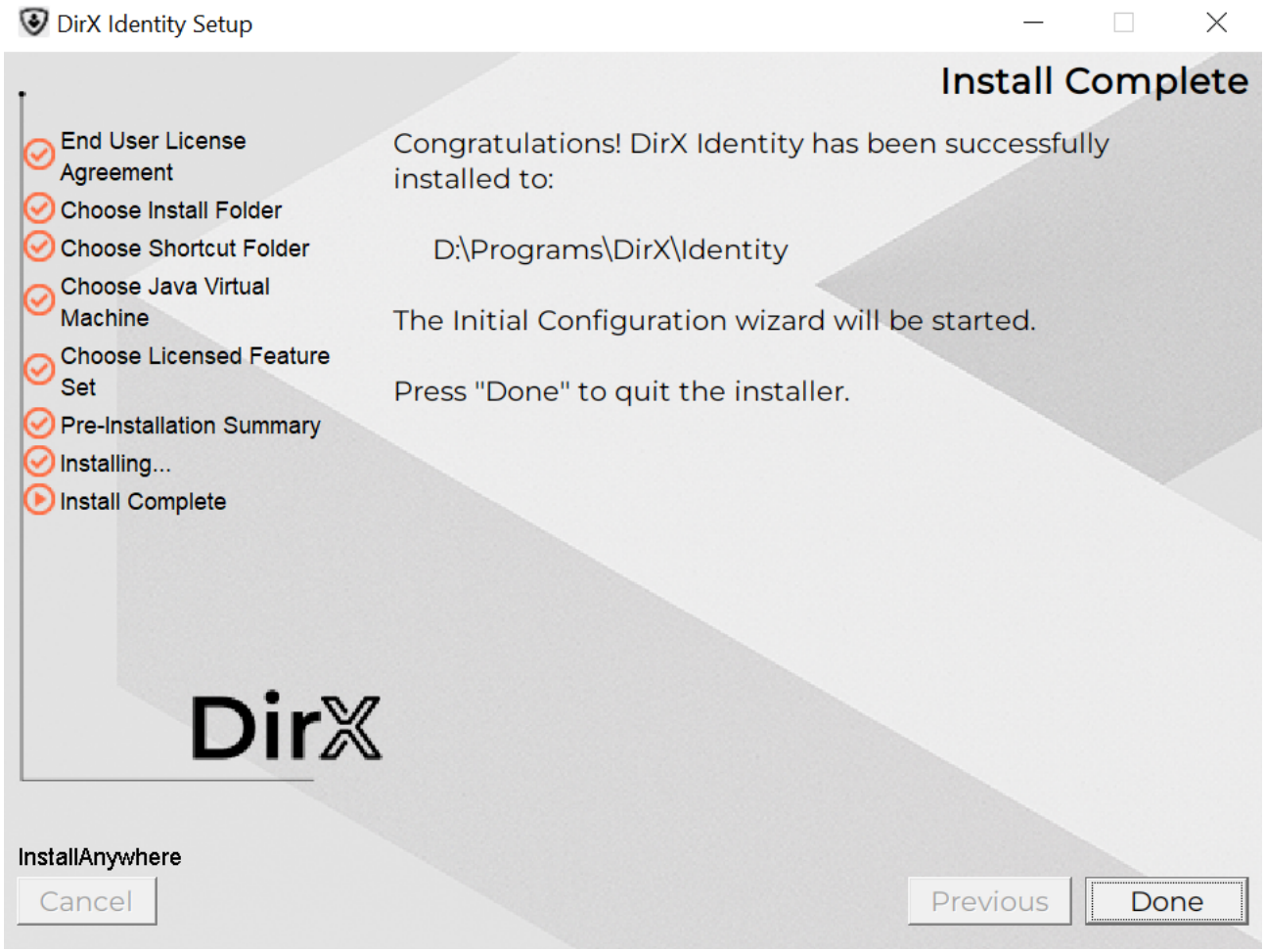


Review selections before proceeding. Click **Previous** to change any settings you have made. Otherwise, click **Install**.

3.3.1.1.8. Installing DirX Identity

Installation progress is displayed.

3.3.1.1.9. Install Complete



Click **Done** if installation succeeds. If errors occur, see troubleshooting-guide:ch3_install.pdf.

3.3.1.2. Completing the installation

After installing DirX Identity, the Configuration Wizard must be started to prepare the product for the first use. The Configuration Wizard is started automatically after a successful installation. Alternatively, it can be manually started using its Start Menu shortcut or launcher.

3.3.2. Console Mode

- Runs in command-line interface.
- Default mode for Linux.
- Respond to prompts to proceed. Type `back` to return or `quit` to cancel.

Windows

```
dirxidty.exe -i console
```

Linux

```
sh ./dirxidty.bin -i console
```

3.3.3. Silent Mode

- Runs without user interaction.
- Requires a properties file with all installation parameters.

Windows

```
dirxidty.exe -i silent -f "<PATH_TO_PROPERTIES_FILE>"
```

Linux

```
sh ./dirxidty.bin -i silent -f "<PATH_TO_PROPERTIES_FILE>"
```

3.4. Installation Properties

You can use a properties file to preset all installation values for any mode. For reference, check the `dirxidty.properties` file — it lists all properties and descriptions.

Place the file in the same folder as the installer or specify it via the `-f` argument:

```
dirxidty.exe -f "<PATH_TO_PROPERTIES_FILE>"
```

Example: Silent installation properties file (comments explain each property):

```
…

#################################################################
#########
#
# DirX Identity installation properties
#
#################################################################
#########

#################################################################
#########
#
```

```
# Release information
#
# Version=${version}
# Build=${buildnumber}
# InstallationType=${installationtype}
# CreationDate=${date}
#
################################################################################
#########

################################################################################
#########
#
# Installer created with InstallAnywhere by ${ia_vendor}
# ${ia_version}
#
################################################################################
#########

################################################################################
#########
#
# InstallAnywhere installation properties
#
################################################################################
#########

# UI mode for the installer
# INSTALLER_UI=[SILENT | CONSOLE | GUI]

# Default for Windows: GUI
# Default for Unix: CONSOLE

################################################################################
#########
#
# DirX Identity specific installation properties
#
################################################################################
#########

# Installation path for DirX Identity
```

```
# PROP_USER_INSTALL_DIR=<path>

# Default for Windows:
# PROP_USER_INSTALL_DIR=$PROGRAMS_DIR$$/$DirX$/$Identity

# Default for Linux:
# PROP_USER_INSTALL_DIR=$PROGRAMS_DIR$$/$DirX$/$Identity

# Note for Windows:
# If an existing installation path for DirX Identity is found in the
registry
# then this path will be used for the installation, and it cannot be
overridden
# with the PROP_USER_INSTALL_DIR property

#
-------------------------------------------------------------------
-------

# Shortcut group for DirX Identity
# PROP_USER_SHORTCUTS=<shortcut group>

# Default:
# PROP_USER_SHORTCUTS=$WIN_COMMON_PROGRAMS_MENU$$/$DirX Identity

# Note for Windows:
# If an existing shortcut group for DirX Identity is found in the
registry
# then this shortcut group will be used for the installation, and it
cannot be
# overridden with the PROP_USER_SHORTCUTS property

#
-------------------------------------------------------------------
-------

# Selected licenses for DirX Identity

# Apply selected licenses: the selected license values will be
applied if
# PROP_SET_LICENSES is set to 1
PROP_SET_LICENSES=1
```

```
# Selected licenses
# <license>=[1 | 0]
# 1: License will be selected
# 0: License will be not selected

PROP_LIC_Business_Suite=1
PROP_LIC_Professional_Suite=1
PROP_LIC_HighAvailability=0
PROP_LIC_PwdMgmt=0

#
-------------------------------------------------------------------
-------

# Selected Java environment
# PROP_SELECTED_JRE=<path to existing JDK or JRE>

# Example for Windows:
# PROP_SELECTED_JRE=C\:\\Program Files\\Java\\java-21
# Example for Linux:
# PROP_SELECTED_JRE=/opt/java-21

#
-------------------------------------------------------------------
-------

# Restart Windows - applicable for (un-)installation in silent mode
# Note:
# The following property can be used to force restarting the computer
after the
# installation has been completed

# PROP_RESTART_NEEDED=YES

#
-------------------------------------------------------------------
-------

# Skip configuration after restart
# Note:
# The following property can be used to skip starting the
Configuration Wizard
```

```
# after the next restart of the computer

# PROP_SKIP_CONFIG_AFTER_RESTART=YES

#
-----------------------------------------------------------------------
-------
```

3.5. Configurable Content

Since version 8.10.2, the installation process works by removing the existing installation and installing the new version. To prevent loss of configurable and custom content (plus files like logs), the installation has a defined set of files and folders that are preserved during the update/upgrade. The installation process creates a backup of these files and folders before removing the existing installation, and restores them after installing the new version.

To reduce the risk of losing important data, it is recommended to create a backup of the whole installation folder before performing an update/upgrade installation.

We encourage you to put all custom content into the folders defined in the list below, and to avoid putting custom content into other folders. This will ensure that your custom content is preserved during update/upgrade installations.

Folder **<DXI_INSTALL_PATH>/conf** is the recommended location for all custom content.

The following files (paths with extensions) and folders relative to the <DXI_INSTALL_PATH> are preserved during update/upgrade:

Asterisk ***** represents any string of characters.

- configuration.ini
- install_history.txt
- Atos_DirX_Identity_*Install_*.log
- DirX_Identity_Install_*.log
- *-configuration.tpl
- /client/conf/*.db
- /client/conf/*.cfg
- **/conf/**
- /gui/profiles/* *(except those delivered with the product)*
- /gui/cacerts
- /gui/bin/*.cfg

- /gui/DirXjdiscover.settings
- /ids-j-*/
- /messagebroker/
- /logs/
- /server/conf/dxmmsssvr.ini
- /server/conf/server.crt
- /server/conf/*.cfg
- /server/log/
- /services/instances/*
- /ssl/*.cer
- /ssl/*.crt
- /ssl/*.csr
- /ssl/*.der
- /ssl/*.ext
- /ssl/*.jks
- /ssl/*.key
- /ssl/*.pem
- /ssl/*.p12
- /ssl/conf
- /ssl/identity-keystore
- /ssl/identity-truststore
- /ssl/identity-truststoreAll
- /ssl/provisioningservices-keystore-*
- /ssl/provisioningservices-truststore
- /status/
- /web/instances/*
- /work/

There is also a list of GLOB patterns that might be part of the backed up folders, but are not preserved in the backup because they are not expected to contain any custom content. These patterns are:

- **/*.org
- **/*.org.*
- **/*.template

Configurable content whose location differs in the new version is also preserved, moved to the new location automatically during installation.

3.5.1. Notable changes in configurable content location

- *(9.0.0)* All instances of the web applications are moved to a new folder *<DXI_INSTALL_PATH>*/web/instances. These include:
  - *<DXI_INSTALL_PATH>*/provisioningWebServices/provisioningServlet-* → *<DXI_INSTALL_PATH>*/web/instances/ProvisioningService-*
  - *<DXI_INSTALL_PATH>*/restServices/DirXIdentityRestServices/DirXIdentityRestService-* → *<DXI_INSTALL_PATH>*/web/instances/DirXIdentityRestService-*
  - *<DXI_INSTALL_PATH>*/restServices/ServerAdminRestServices/ServerAdminRestService-* → *<DXI_INSTALL_PATH>*/web/instances/ServerAdminRestService-*
  - *<DXI_INSTALL_PATH>*/web/BusinessUserInterface-* → *<DXI_INSTALL_PATH>*/web/instances/BusinessUserInterface-*
  - *<DXI_INSTALL_PATH>*/web/webCenter-* → *<DXI_INSTALL_PATH>*/web/instances/webCenter-*
  - *<DXI_INSTALL_PATH>*/web/pwdManagement-* → *<DXI_INSTALL_PATH>*/web/instances/pwdManagement-*
- *(9.0.0) <DXI_INSTALL_PATH>*/GUI folder is renamed to *<DXI_INSTALL_PATH>*/gui

3.6. Custom Libraries

You can extend DirX Identity with custom JAR libraries for:

- Identity Manager
- PolicyAgent, ServiceAgent, HistoryAgent
- WebCenter
- SCIM Rest Service (DirXIdentityRestService)
- IdSJ Server

Place JAR files in: `<DXI_INSTALL_PATH>/lib/java/custom/`

Alternatively, define environment variable: `DIRXIDENTITY_CUSTOMLIB_PATH` to point to an external folderto point to an external folder.

The specified folder (if it's located in the installation) is not deleted during upgrade/update installation, so any custom libraries placed there are preserved.

3.7. Environment Variable Settings

The installer updates environment variables:

- PATH → adds *<DXI_INSTALL_PATH>*/**bin**
- DIRXMETAHUB_INST_PATH and DIRXIDENTITY_INST_PATH → set to *<DXI_INSTALL_PATH>*

- DXI_JAVA_HOME → set to *<DXI_JAVA_HOME>*

Windows:

- Creates `setdxienv.bat` in *<DXI_INSTALL_PATH>* with Java settings.

Linux:

- `DXI_JAVA_HOME` will be set to *<DXI_JAVA_HOME>*
- Updates user profile and creates backup (e.g., `~/.profile.NUMBER`). Remove backups manually

3.8. Uninstallation

Uninstallation removes:

- DirX Identity from PATH
- Variables: DIRXMETAHUB_INST_PATH, DIRXIDENTITY_INST_PATH
- All non-configurable files

Windows:

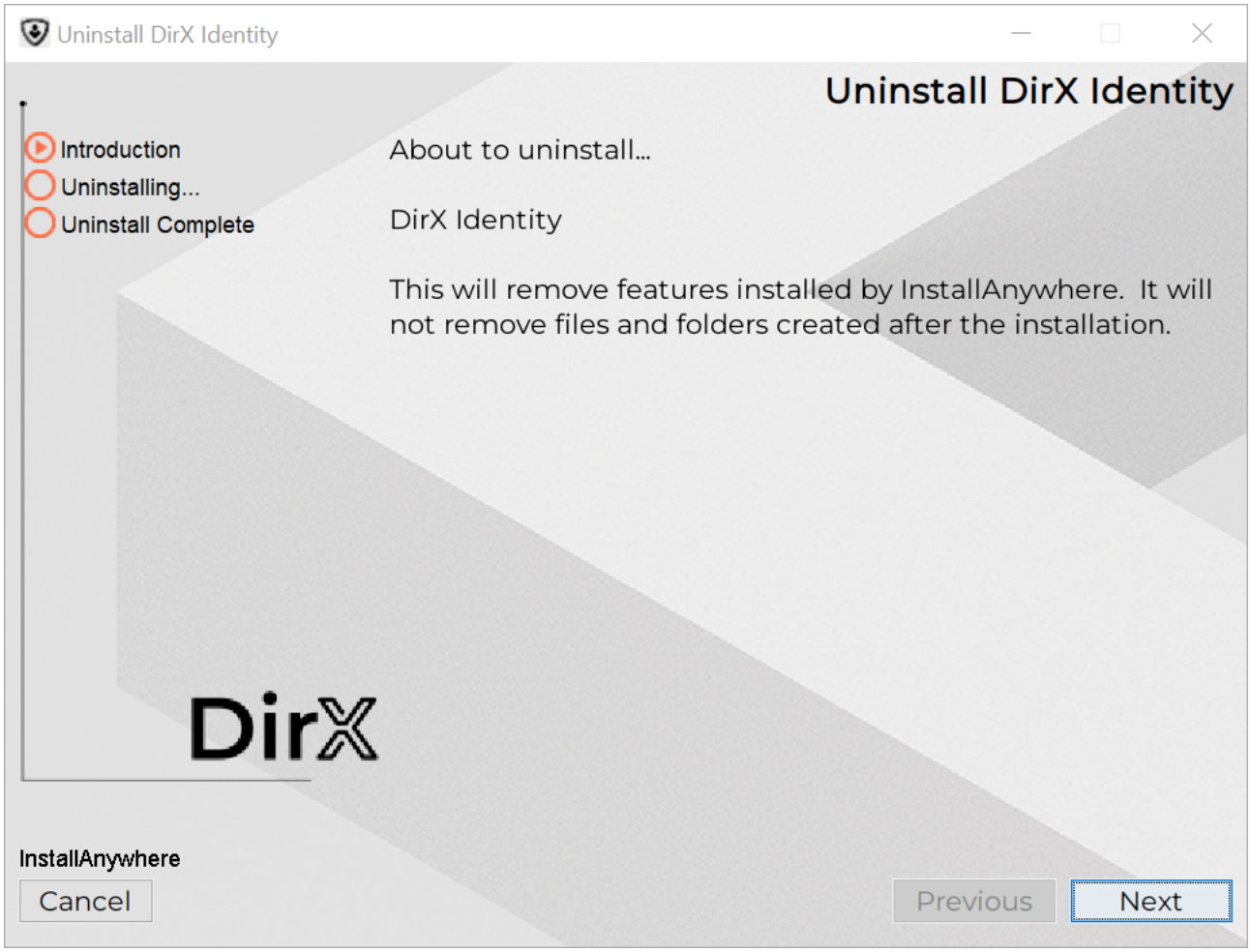
- Open **Settings → Apps → Installed apps**, or **Add/Remove Programs**.
- Find **DirX Identity** and click **Uninstall**.

Linux:

- Log in as installation user.
- Navigate to *<DXI_INSTALL_PATH>*/UninstallerData
- GUI mode: `sh ./Uninstall_DirX_Identity -i gui`
- Console mode: `sh ./Uninstall_DirX_Identity`

If un-configuration was not performed, the wizard runs in un-configuration mode first.

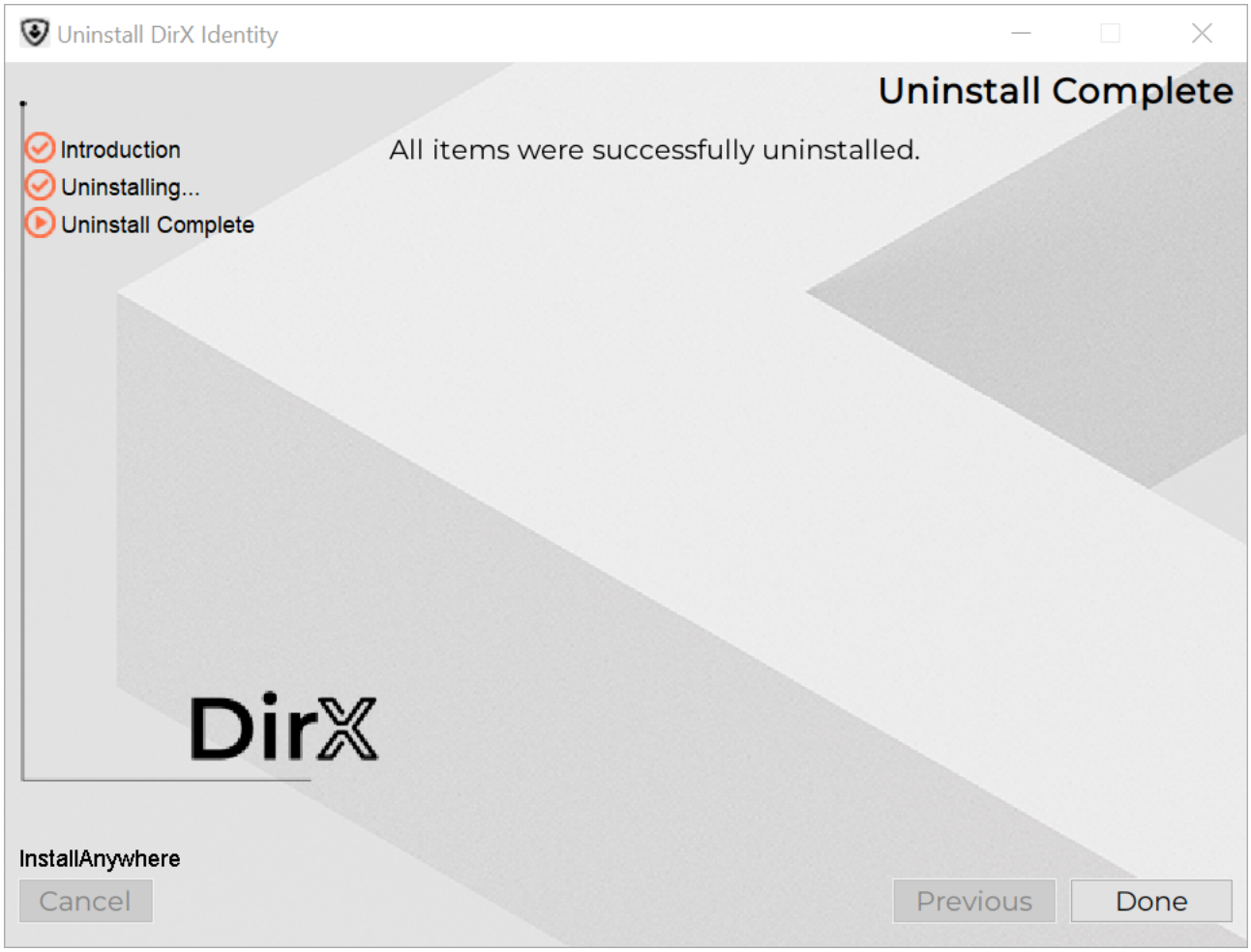
3.8.1. Uninstall DirX Identity



- Click **Next** to start uninstallation.
- Click **Cancel** to exit.

After uninstallation, database removal is no longer possible. Run configuration tool before uninstalling.

3.8.2. Uninstall Complete



Click **Done** to finish

3.9. Additional instructions for Linux

If the installation account uses **bash** as login shell:

- Ensure `.profile` runs at each login and graphical session start.
- Add:
  - `. ~/.profile` to `.bashrc`
  - `. ~/.bashrc` to `.bash_profile` (if not present)

4. Configuring DirX Identity

This chapter explains how to configure DirX Identity across all supported platforms using the Configuration Wizard. The wizard provides a guided process for initial setup, upgrades, reconfiguration, and unconfiguration.

4.1. Configuration Wizard

The Configuration Wizard is a Java Swing application that guides you through configuring DirX Identity components.

4.1.1. Purpose

You can use the wizard to:

- Perform an **initial configuration** (automatically launched after Windows installation).
- Apply **upgrade configuration** after an update.
- **Reconfigure** settings at any time (e.g., create a new domain, modify values).
- **Unconfigure** components during uninstallation or on request.

4.1.2. Location

Script location: `<DXI_INSTALL_PATH>/bin`

4.1.3. Scripts

- Configuration: `Configuration.bat` (Windows), `Configuration.sh` (Linux)
- Unconfigure: `UnConfiguration.bat` (Windows), `UnConfiguration.sh` (Linux)
- Initial configuration: `InitialConfiguration.bat` (Windows), `InitialConfiguration.sh` (Linux) - since version 9.0.0, these act the same as Configuration scripts

4.1.3.1. Arguments

- `action` – `configure` (default) or `unconfigure`
- `mode` – `normal` (default) or `silent`
- *`<IDSJ_PROPERTIES_FILE>`* – path to Java-based Server properties file

4.1.4. Before You Start

4.1.4.1. DirX Directory

DirX Directory is required for DirX Identity to work. Ensure that **DirX Directory service** is running.

4.1.4.2. FQDN configuration

The Deployment Tool requires FQDN to be setup properly. The FQDN set in the system **has to match** the FQDN (host name) you want to set for the components during configuration, otherwise they will not be configured. It will probably result in Deployment Tool not running some tasks (maybe none at all).

4.1.4.2.1. Windows

On Windows operating system, check the FQDN configuration:

1. Run Registry Editor system application
2. Search for key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`
3. Ensure values named `Hostname` and `Domain` have data set. The FQDN is constructed by joining these two values with a dot (`Hostname.Domain`), e.g. with hostname `my-server` and domain `my.domain.net`, the FQDN will be `my-server.my.domain.net`.

4.1.4.2.2. Linux

On Linux operating system, check the FQDN configuration:

1. Run `hostname --fqdn`.
2. Ensure it returns a valid FQDN (not *`localhost`*).
3. If incorrect, edit `/etc/hosts` so the first entry for 127.0.0.1 or ::1 is the valid FQDN.

If you cannot set up the FQDN as described above, you may override the native way of obtaining it.
For Configuration Wizard, set a property in the properties file named `deploymentTool.fqdnOverride=<FQDN>`, where `<FQDN>` is the desired FQDN value.
For Deployment Tool, use the command-line argument `--fqdn -override=<FQDN>`.
Remember that this needs to be setup every time you run these programs. It's recommended to fix the FQDN properly in the operating system instead.

4.1.5. Starting the Configuration Wizard

You can start the wizard using:

- Scripts listed above
- Windows Start Menu shortcut: **Start → DirX Identity → Configuration Wizard**

Windows

For Windows 7 (Vista), Windows 2008 and later, **UAC** may block registry writes. To avoid issues:

- Add the user to the local administrator group
- OR start the application with ***Run as Administrator***
- OR temporarily disable **UAC**

Linux

Superuser permissions are usually required. If conditions in "Determining the Account for Configuration on UNIX" are met, you can use the installation account.

Linux

Older Linux environments sometimes required setting: `LD_ASSUME_KERNEL=2.4.19`. This is **deprecated and incorrect**. If active, **remove or unset this variable** before starting the wizard.

4.1.6. Properties file

The wizard stores configuration properties in an `.ini` file. Default path: *`<DXI_INSTALL_PATH>`*`/`**`configuration.ini`** (see Commands and Arguments for details)

4.1.6.1. Property Types

Type	Description
String	Standard text value
Boolean	`true` or `false` (stored as 1 or 0)
Path	File system path
Port	Integer port number
Version	Semantic version (major.minor.patch)
LDAP DN	Distinguished Name (escape '=' as '\=')
Credentials	Username and password (KeePass only)
Password	**Deprecated** (use Credentials)
PIN	**Deprecated** (use Credentials)
Size	Memory/disk size (e.g., `4 GByte`)

During normal Configuration:

- If you want to quit early, you will be asked if you want to save the changes made so far to this file. See more in Navigation and UI.

4.1.6.2. Password Policies

In some steps, password fields require a certain level of strength before you can continue. If a password does not meet these requirements (policies), hover over the password field to view the policy.

If you have not yet configured policies, you will be asked to do so in a dialog after clicking **Next** in Welcome Step. You can also configure these policies in the properties file using the following properties (default values in parentheses):

- `PasswordPolicy.MinPasswordLength` (8) - Minimum length of the password
- `PasswordPolicy.MaxPasswordLength` (64) - Maximum length of the password
- `PasswordPolicy.MinUpperCaseCharacters` (1) - Minimum number of uppercase characters required
- `PasswordPolicy.MinLowerCaseCharacters` (1) - Minimum number of lowercase characters required
- `PasswordPolicy.MinNumericCharacters` (1) - Minimum number of numeric characters required
- `PasswordPolicy.MinSpecialCharacters` (1) - Minimum number of special characters required

Validation rules:

- All policies must have values defined.
- Minimum must be at least 1.
- Maximum must be at least 1.
- Minimum cannot exceed maximum.
- The sum of all minimum character types cannot exceed the maximum length.

If invalid configuration is detected, the error is logged and the application quits.

These password policy properties apply **only to the Configuration Wizard** and are **not stored in the DirX Identity configuration**.

4.1.6.3. Installed and Configured Components

After installation, you may find properties with the prefix: `InstalledComponents.` These indicate which components were installed and help determine what should be configured.

After initial configuration, properties with the prefix: `ConfiguredComponents.` show which components have already been configured.

Do not modify or delete these properties manually. They may be used by other parts of DirX Identity.

Password policy validation is **not performed** on existing accounts during re-configuration.

4.1.7. Default Credentials File

Default credentials are stored in: *<DXI_INSTALL_PATH>/deployment/default/**dxi-credentials.kdbx***.

Protected by:

- Master password (in **dxi-resource**)
- Key file (**dxi-credentials.keyx**)

If the wizard detects the default credentials file, it becomes **read-only**.

4.1.8. Logging

Each run creates a log file: *<DXI_INSTALL_PATH>/logs*/ConfigurationWizard_<START_TIMESTAMP>.log. Use this file for troubleshooting or attach it to support tickets.

All the log settings are defined in the file *<DXI_INSTALL_PATH>/bin/deployment/confwizard/conf/**logback.xml***, but you should not have a need to modify it.

4.2. Commands and Arguments

Supported commands:

- `configure` – Starts configuration mode (default)
- `unconfigure` – Starts unconfiguration mode

Example for configure:

```
Windows: <DXI_INSTALL_PATH>\bin\Configuration.bat
Linux:   <DXI_INSTALL_PATH>/bin/Configuration.sh
```

The **Initial Configuration** is run automatically after installation on Windows platforms.

If you have integrated the DirX Identity start/stop scripts into the Linux operating system (see the section Integrating Start/Stop Scripts into the Linux Operating System), you must undo these integration actions (see the subsection Undoing the Integration) before uninstalling the product.

The Configuration Wizard can be started by scripts described above. If needed (mainly for silent configuration), the following arguments can be used:

- ***-d***, ***--dxi-installation-path*** *<PATH>* - Specifies the path where DirX Identity is

installed. If not specified, it is read from the environment variable *DIRXIDENTITY_INST_PATH*.

- ***-f***, ***--wizard-description-file*** *<PATH>* - Specifies the path to the wizard description file. If not specified, the default file *<DXI_INSTALL_PATH>/bin/deployment/confwizard/**wizard.xml*** is used. **It should not be changed**.
- ***-p***, ***--properties-file*** *<PATH>* - Specifies the path to the INI properties file. If not specified, the default file *<DXI_INSTALL_PATH>/**configuration.ini*** is used.
- ***-c***, ***--credentials-file*** *<PATH>* - Specifies the path to the credentials file. The supported formats are KeePass 2 **`.kdbx`** file or properties **`.ini`** file *(deprecated)*. Used here. If not specified, the default file is *<DXI_INSTALL_PATH>/**configuration.ini***.
- ***-m***, ***--master-password*** *<PASSWORD>* - Specifies the master password to open the KeePass database file. Not used if the credentials file is a properties **`.ini`** file. **Applicable only for the silent configuration, it is always requested in normal mode**.
- ***-k***, ***--key-file*** *<PATH>* - Specifies the path to the key file **`.key`** or **`.keyx`** of the KeePass database. Only to be used if the KeePass database requires it. Not used if the credentials file is a properties **`.ini`** file.
- ***-j***, ***--idsj-properties-file*** *<PATH>* - Specifies the path to the Java-based Server properties file. If not specified, the properties file is used.
- ***-i***, ***--interface*** *<VALUE>* - Specifies the interface to run the Configuration in. By default, it is **normal**. Otherwise, you can use **silent** for silent configuration - more about it here.

4.2.1. Exit Codes

The Configuration Wizard returns the following exit codes:

- `0` - Configuration completed successfully, or canceled by the user without errors (before Deployment Tool execution)
- 1 - Exception occurred during configuration (before Deployment Tool execution).
- 2 - Missing or invalid required data in the properties file (mostly during silent configuration).
- 3 - `BackendException` occurred during backend execution (Deployment Tool). The Deployment Tool exit code is not passed through.
- 4 - `ExecutionException` or `InterruptedException` occurred during backend execution (e.g., process canceled).

These arguments can only be changed in the scripts and cannot be provided as arguments to the scripts.

4.2.2. Navigation and UI

The Configuration Wizard is built on the common DirX Java UI framework *jDiscover*,

specifically its Wizard library.

4.2.2.1. Layout and Navigation

- Steps are listed on the **left panel**, in top-to-bottom order. Colors indicate the status of each step.
- Navigation is sequential; you cannot skip steps because some depend on previous values.

4.2.2.2. Buttons

- **Next** – Moves to the next step (disabled if data is missing or invalid).
- **Previous** – Returns to the previous step (data is preserved).
- **Finish** – Enabled only when all steps are complete; ends the configuration process.
- **Cancel** – Aborts configuration at any time. If clicked (or the window is closed), a confirmation dialog appears with:
  - **Exit with save** – Saves changes made so far and exits.
  - **Exit without save** – Discards changes and exits.
  - **Continue** – Closes the dialog and resumes configuration.
- **Help** – Opens relevant documentation in a browser. A small button next to it lets you choose the browser (auto-detected).

4.3. Steps Overview

4.3.1. Welcome Step

This is the first step of the Configuration Wizard. Here you specify the **credentials file** used for passwords.

Inputs:

- **Credentials file path** Path to the credentials file. It must exist or be created manually during configuration. Supported formats:
  - `*.kdbx` – KeePass 2 database file (secure, preferred). For details, visit the Official KeePass website.
  - `*.ini` – Standard INI properties file (not secure).

For KeePass files:

- Credentials must be stored in a group named one of: DirX Identity, dirx-identity, DXI, dxi
- The group must be a direct child of the **root** node.
- Each entry must have an **unprotected** String field named Key with the property key as its value. Example: **`hubadmin.credentials`** See Default

Credentials File for examples.

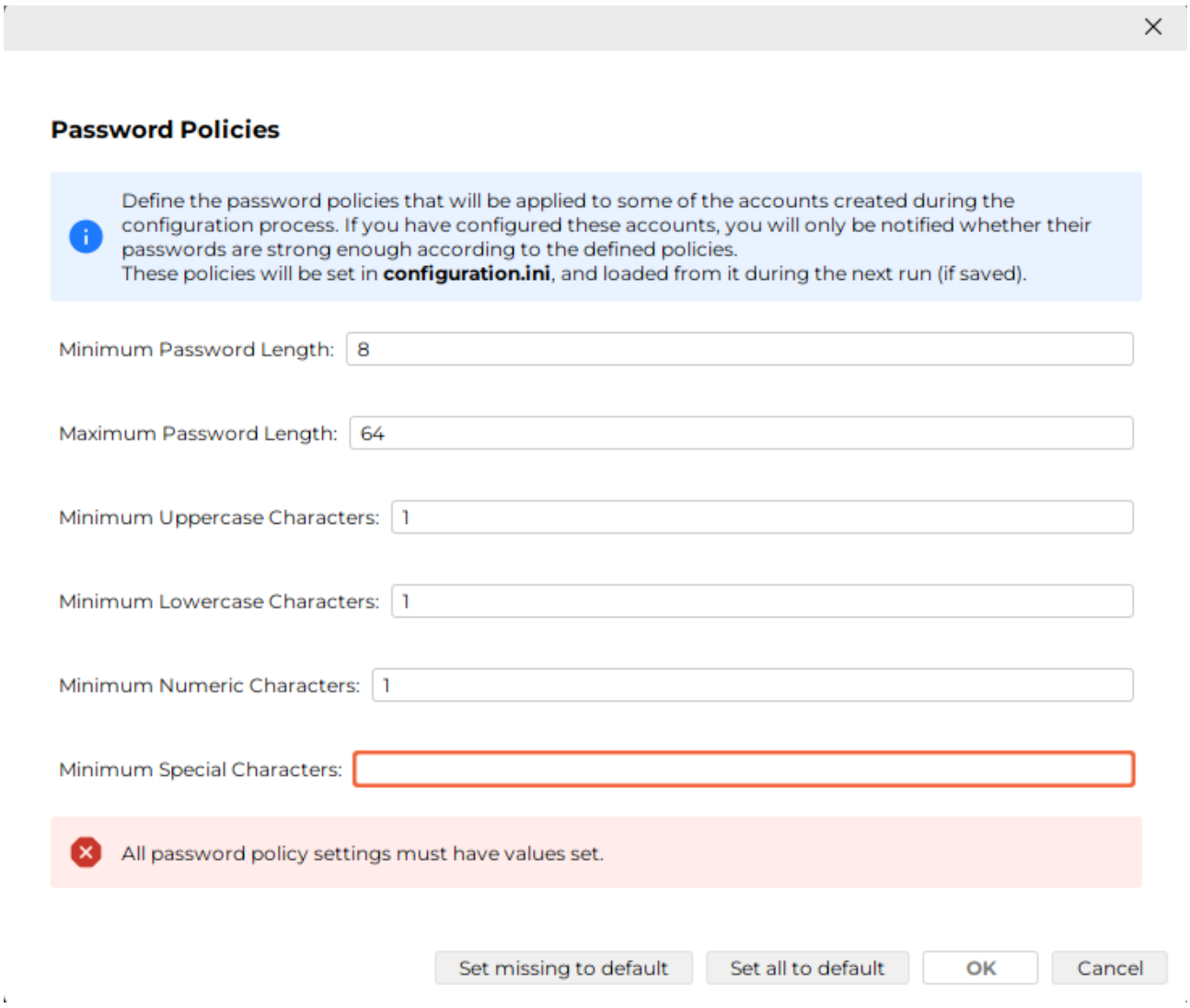
- **Master password** Displayed only for KeePass files. Enter the master password for the database.
- **Key file path *(optional)*** Displayed only for KeePass files. Enter the path to the key file (**`.key`** or **`.keyx`**) if required.
- **Unlock button** Attempts to open the KeePass database with the provided information. Once successful, the button is disabled until another credentials file path is specified.

`.ini` format is insecure (passwords stored in clear text). It is supported for backward compatibility only. Default path: *`<DXI_INSTALL_PATH>/`**`configuration.ini`***. Migrate to KeePass as soon as possible.

After proceeding to the next step, the credentials file cannot be changed without restarting the Configuration Wizard.

4.3.1.1. Password Policies dialog

If password policies are not yet configured (missing in the properties file), a dialog appears after clicking **Next**.



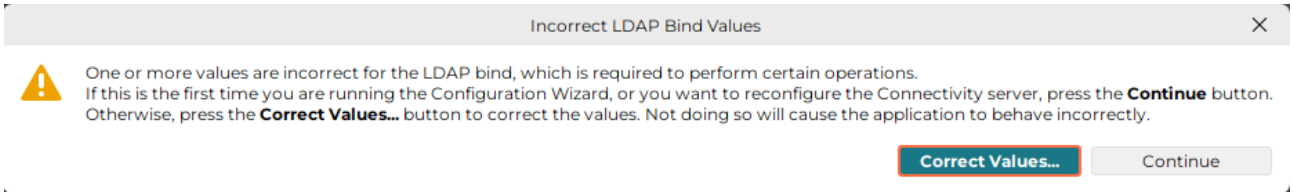
You cannot continue ("OK" button is disabled) until you set valid policies, see Password Policies for details. You can use button "Set missing to default" to fill in default values for any missing policies, or use "Set all to default" to reset all policies to default values.

4.3.2. Configuration Options

This step allows you to select which components should be configured during this run. Each option may add dependent steps automatically, even if those steps are not explicitly selected.

4.3.2.1. Possible Warning Dialog

At the beginning of this step, you might see the following dialog:



This occurs when the administrator account: `cn=admin`, `dxmC=DirXmetahub` could not bind

to the Connectivity LDAP server. The issue is usually caused by missing or incorrect values in the properties file. Check the following properties:

- `connectivityStore.host`
- `connectivityStore.port`
- `connectivityStore.ssl`
- `hubadmin.credentials`
- `hubadmin.user` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property instead)*
- `hubadmin.password` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property instead)*

4.3.2.2. How to Proceed

- If you are performing **Initial Configuration**, click **Continue** to proceed. The account does not exist yet and will be created during configuration.
- For other cases:
  - Click **Correct Values...** to open a dialog where you can fix the incorrect properties.
  - After correcting the values, click **Authenticate** to retry the LDAP bind.
  - If successful, the "OK" button becomes enabled, allowing you to proceed.

4.3.2.3. Options

Some of the options may be marked as ***(MANDATORY)***. These options are required for the configuration to be successful and cannot be deselected (for example Connectivity and Provisioning configuration during Initial Configuration).

4.3.2.4. Available Options

The following options can be selected.

Selecting an option may automatically add **dependent steps**, even if those steps are not explicitly chosen. Dependencies are described under each option.

- **Connectivity Schema and Data**
  - Property key: `option.dxm_schema` (Boolean)
  - Dependant steps: *Directory Server for Connectivity*, *DirX Identity Administrators*, *System-wide Configuration*
- **Provisioning Schema and Data**
  - Property key: `option.dxr_schema` (Boolean)

    Dependant steps: *Directory Server for Connectivity*, *Directory Server for Provisioning*,

    - *DirX Identity Administrators*
- **ActiveMQ Message Broker**
    - Property key: `option.MessageBroker` (Boolean)
    - Dependant steps: *Directory Server for Connectivity*, *Directory Server for Provisioning*, *DirX Identity Administrators*, *System-wide Configuration*, *ActiveMQ Message Broker Configuration*, *ActiveMQ Message Broker Service*
- **C++-based Server**
    - Property key: `option.idsc` (Boolean)
    - Dependant steps: *Directory Server for Connectivity*, *Directory Server for Provisioning*, *DirX Identity Administrators*, *System-wide Configuration*, *C-based Server Configuration_*, *_C-based Service*, *HCL Notes Client (Windows)*, *ODBC and SAP JCO Libraries (Linux)*
- **Domain**
    - Property key: `option.domain` (Boolean)
    - Dependant steps: *Directory Server for Connectivity*, *Directory Server for Provisioning*, *DirX Identity Administrators*, *Domain Configuration*
- **Java-based Server**
    - Property key: `option.idsj_server` (Boolean)
    - Dependant steps: *Directory Server for Connectivity*, *Directory Server for Provisioning*, *DirX Identity Administrators*, *Domain Configuration*, *System-wide Configuration*, *Java-based Server Configuration*, *Java-based Service*
- **DirX Identity Manager**
    - Property key: `option.configureManager` (Boolean)
    - Dependant steps: *Directory Server for Connectivity*, *Directory Server for Provisioning*, *DirX Identity Administrators*, *Domain Configuration*
- **Web Center**, **Web Center for Password Management**, **Provisioning Web Service**, **Server Admin REST Service**, **Identity REST Service**, **Business User Interface**
    - Property key (respectively): `option.WebCenter`, `option.WebCenterPwdMgmt`, `option.ProvisioningWebService`, `option.ServerAdminRestService`, `option.RESTService`, `option.BusinessUserInterface` (Boolean)
    - Dependant steps: *Directory Server for Provisioning*, *Domain Configuration*, *Tomcat Configuration*

4.3.3. Directory Server for Connectivity

4.3.3.1. LDAP Server Configuration

Host

Property key: `connectivityStore.host` (String)
Default value: resolved **localhost**

The FQDN (fully qualified domain name) of the directory server where the Connectivity database resides.

Port

Property key: `connectivityStore.port` (Port)
Default value: **389** (non-SSL), **636** (SSL)

The port number of the directory server where the Connectivity database resides.

SSL

Property key: `connectivityStore.ssl` (Boolean)
Default value: **false**

Switch controlling if you want to connect with SSL (checked) to the directory server where the Connectivity database resides. Make sure you have entered the appropriate port.

4.3.3.2. Administrator account

This is the account that has rights to make changes to the directory server schema. During Initial Configuration, this account does not exist yet, and it will be created. It must meet the password policies set. During next runs, it exists, and therefore you must enter the correct password and press **Authenticate** to verify the credentials.

Property key: `connectivityStore.credentials` (Credentials)
Default value: in the default passwords KeePass database.

Username

Property key: `connectivityStore.user` (LDAP DN) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: **`cn=admin,o=My-Company`** (of My-Company domain)

The name (LDAP DN). The password of the default user cannot be changed).

Password

Property key: `connectivityStore.password` (Password) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: in the default passwords KeePass database.

The password of the administrator.

4.3.3.3. DirX Directory

This is the application for the LDAP server. Only **DirX Directory** is supported.

Installation Path

Property key: `connectivityStore.directory_inst_path` (Path)

Default value: resolved from environment variable *DIRX_INST_PATH* or Windows registry.

The path to the DirX Directory installation folder. The path is validated based on the contents of the folder.

Version

Property key: `connectivityStore.version` (Version)
Default value: resolved from installation path.

This property is **read-only**. It is resolved when the installation path changes, either from one of its programs, or from the registry on Windows.

Build

Property key: `connectivityStore.build` (Version)
Default value: resolved from installation path.

This property is **read-only**. It is resolved when the installation path changes, either from one of its programs, or from the registry on Windows.

If you are using SSL for the first time to bind to the directory server with the Configurator, you must import the test CA certificate of the LDAP server into the trust store *<DXI_JAVA_HOME>***/lib/security/cacerts** of the JRE for DirX Identity before you run the Configuration Wizard. (See the chapter "Setting up the Java-based Configuration Wizard" in the *DirX Identity Connectivity Administration Guide* for details.) If DirX Identity was not installed before, and you cannot use the DirX Identity Manager to import certificates into trust stores, then you must use *<DXI_JAVA_HOME>***/bin/keytool.exe** to import the certificate.

4.3.4. DirX Directory Server for Provisioning

This step is the same as the step **Directory Server for Connectivity** described above, but it is used to configure the directory server where the Provisioning database resides. Additionally, the default values may be resolved from the Connectivity configuration.

It uses the following property keys (see the mentioned step for details):

- `provisioningStore.host`
- `provisioningStore.port`
- `provisioningStore.ssl`
- `provisioningStore.credentials`
- `provisioningStore.user` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
- `provisioningStore.password` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*

- `provisioningStore.directory_inst_path`
- `provisioningStore.version`
- `provisioningStore.build`

4.3.5. DirX Identity Administrators

This step manages the three main **DirX Identity administrator accounts**.

4.3.5.1. Purpose

- During **Initial Configuration**, these accounts do not exist yet. They will be created and stored in LDAP.
- In subsequent runs, these accounts already exist. You must enter the correct passwords and click **Authenticate** to verify credentials.

All passwords must comply with the configured password policy (see Properties file for details).

4.3.5.2. Administrator cn=admin,dxmC=DirXmetahub

This account is used to access the DirX Identity Connectivity configuration during Configuration Wizard runs.

Property key: `hubadmin.credentials` (Credentials)
Default value: in the default passwords KeePass database.

Username

Property key: `hubadmin.user` (LDAP DN) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: **`cn=admin,dxmC=DirXmetahub`**

This property is **read-only** and cannot be changed.

Password

Property key: `hubadmin.password` (Password) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: in the default passwords KeePass database.

The password of the administrator.

4.3.5.3. Administrator cn=server_admin,dxmC=DirXmetahub

This account is used by DirX Identity C++-based Server and the supervisor to access the Connectivity configuration.

Property key: `svcadmin.credentials` (Credentials)
Default value: in the default passwords KeePass database.

Username

Property key: `svcadmin.user` (LDAP DN) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: **`cn=server_admin,dxmC=DirXmetahub`**

This property is **read-only** and cannot be changed.

Password

Property key: `svcadmin.password` (Password) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: in the default passwords KeePass database.

The password of the administrator.

4.3.5.4. Administrator cn=SystemDomain,cn=DirXmetaRole-SystemDomain

This account is used as supervisor for the Provisioning configuration.

Property key: `roleadmin.credentials` (Credentials)
Default value: in the default passwords KeePass database.

Username

Property key: `roleadmin.user` (LDAP DN) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: **`cn=SystemDomain,cn=DirXmetaRole-SystemDomain`**

This property is **read-only** and cannot be changed.

Password

Property key: `roleadmin.password` (Password) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: in the default passwords KeePass database.

The password of the administrator.

4.3.6. ActiveMQ Message Broker Configuration

This step allows you to configure the Message Broker Configuration, which is based on Apache ActiveMQ.

Host Name

Property key: `MessageBroker.host` (String)
Default value: resolved **localhost**

The FQDN of the system this server runs on. If SSL is used or planned to be used for the system, specify the host name in the same form - short or fully-qualified - as for **generating**

the server certificate for this host. In a wide area network a fully-qualified host name could be required for SSL to work properly.

Display Name

Property key: `MessageBroker.displayname` (String)
Default value: **Message Broker 1**

This property is **read-only** in format *Message Broker n*, where *n* is the number of Message Brokers configured for your (possibly distributed) installation, incremented by one for a new one. It is either the display name of an existing Message Broker or a new Broker name.

Port

Property key: `MessageBroker.port` (Port)
Default value: **61616**

The port for message transfer.

Secure Port

Property key: `MessageBroker.secure_port` (Port)
Default value: **61617**

The port for secure message transfer.

Admin (Web Console) Port

Property key: `MessageBroker.admin_port` (Port)
Default value: **8161**

The port for accessing the ActiveMQ Web Console. You can access the Web Console after configuration at the URL `http://<HOST_NAME>:<ADMIN_PORT>/admin`, e.g. http://localhost:8161/admin.

JMX Port

Property key: `MessageBroker.jmx_port` (Port)
Default value: **10098**

The port for the JMX monitoring *(n)*. Note that the port ***(n+1)*** is also configured and used.

Message repository

Property key: `MessageBroker.repository` (Path)
Default value: ***`<DXI_INSTALL_PATH>/messagebroker/data/kahadb`***

The path to the message repository (KahaDB). It must be specified as an absolute path. Alternatively, on a Windows system, you can specify it as a UNC path referring to a shared folder.

4.3.7. ActiveMQ Message Broker Service Account

This step allows you to configure the ActiveMQ Message Broker service. It offers similar configuration as in the Windows Services app, with Linux lacking some parts.

The Configurator asks under which account the ActiveMQ Message-Broker service should run.

Log on as account (Windows only)

You can choose from:

- **System account** option (default)
- **This (Custom) account** option, where you can enter your preferred account. The following properties can be set:
  - Domain
    - Property key: `MessageBroker-service.domain` (String)
    - The domain can be left empty to use the local computer
  - Credentials
    - Property key: `MessageBroker-service.credentials` (Credentials)
  - User
    - Property key: `MessageBroker-service.account` (String) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
  - Password
    - Property key: `MessageBroker-service.password` (Password) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*

To use the System account in silent configuration, leave all the above properties empty.

Startup type

Property key: `MessageBroker.start_type_automatic` (Boolean)
Default value: **true** (Automatic)

The dropdown corresponds to the startup type of the Windows service, but it is used on Linux too. The options are:

- Automatic (Delayed Start)
- **Automatic**
- **Manual**
- Disabled

You cannot select **Automatic (Delayed Start)** nor **Disabled**. For compatibility, the property

is a Boolean, where **true** means **Automatic** and **false** means **Manual**.

Start service after configuration

Property key: `MessageBroker.start_after_configuration` (Boolean)
Default value: **true** (checked)

If checked, the service is attempted to be started after Configuration Wizard (Deployment Tool) finishes.

After pressing **Next**, the specified account is checked for validity and the rights to create files in the directory *<DXI_INSTALL_PATH>*. To perform these checks, the account under which this configuration procedure runs (the account you are logged in) must have the advanced user rights "Act as part of the operating system" and "Replace a process level token". If this is not the case, error is displayed. We recommend aborting the configuration at that point and performing this procedure:

- Cancel the configuration.
- Grant the required rights to the user.
- Reboot your computer.
- Run the DirX Identity Configurator again.

4.3.8. C++-based Server Configuration

This step allows you to configure the DirX Identity **C++-based Server**.

Primary DirX Identity C Server.

Property key: `IdS-C.primary` (Boolean)
Default value: **true**

This checkbox is displayed **only if you are configuring a C Server on a machine different from the one hosting the Connectivity Database**.

- **When to check the box:** Select this option if you want to configure the **primary C Server** for this host name.
- **When not checked:** If the box remains unchecked, the system will configure a **secondary C Server** for this host name.

When reconfiguring an existing primary or secondary C Server:

- Ensure the **host name matches exactly** the previous configuration (the suggested value is taken from the properties file).
- Do **not** switch between short and fully qualified host name formats.

Host

Property key: `IdS-C.host` (String)
Default value: resolved **localhost**

The FQDN of the system this server runs on. If SSL is used or planned to be used for the system, specify the host name in the same form - short or fully-qualified - as for **generating the server certificate** for this host.

Port for key transfer

Property key: `IdS-C.port` (Port)
Default value: **1111**

The port for secure connections between the DirX Identity C++-based server and the DirX Identity Java-based agents if encryption mode is to be used.

Work path

Property key: `path.work` (Path)
Default value: ***<DXI_INSTALL_PATH>/work***

The path to a directory, which will be used to store the files during a run of an activity. It must be specified as an absolute path. Alternatively, on a Windows system, you can specify it as a UNC path referring to a shared folder.

Status path

Property key: `path.status` (Path)
Default value: ***<DXI_INSTALL_PATH>/status***

The path to a directory, which will be used to store the permanent files during a run of an activity. It must be specified as an absolute path. Alternatively, on a Windows system, you can specify it as a UNC path referring to a shared folder.

The Configurator asks you to select a work path directory and a status path directory. It displays the default locations in the fields provided.

We recommend setting the work and status paths on separate disks in production environments. DirX Identity is designed to ignore a full status area disk, but cannot ignore a full disk where the work area is located.

4.3.9. C++-based Service Account

This step is the same as the step **ActiveMQ Message Broker Service Account** described above, but it is used to configure the DirX Identity C++-based Server service.

It uses the following property keys (see the mentioned step for details):

- `IdS-C-service.domain`
- `IdS-C-service.credentials`
- `IdS-C.start_type_automatic`
- `IdS-C.start_after_configuration`
- `IdS-C-service.account` *(deprecated since 9.0.0, usable only from configuration.ini file,*

use `.credentials` property with KeePass database instead)

- `IdS-C-service.password` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*

4.3.10. Domain Configuration

This step allows you to configure your domain. It offers two options:

- Customer Domain (**default**)
  - Property key: `option.cust_domain` (Boolean)
- Sample Domain
  - Property key: `option.sample_domain` (Boolean)

You can configure any number of customer domains, but only one at a time. If you want to configure another customer domain, you must run the Configuration Wizard again.

The sample domain is a complete and fully working example of a domain named **My-Company**. For more information, see the *DirX Identity Tutorial Guide*.

4.3.10.1. Domain

Name

Property key: `domain` (String)
Default value: **My-Company** (of sample domain) or empty (of customer domain)

The name of the domain. Characters dollar sign (**$**), asterisk (*****), question mark (**?**), hash sign (**#**), comma (**,**), plus sign (**+**) and equals sign (**=**) are not allowed.

Technical name

Property key: `tech_customer_domain` (String)
Default value: suggested based on the domain name.

You can change the suggested technical name if needed. This name is used for:

- Creating the folder: *<DXI_INSTALL_PATH>*`/idsj-`*`technical_domain`*`-S`*`n`* This folder corresponds to the **n-th Java-based Server** for the specified domain.
- Generating service names related to Java-based Servers for that domain.

For technical reasons, the name must follow strict rules:

- Allowed characters:
  - Letters: `A–Z`, `a–z`
  - Digits: `0–9`

    - Symbols: minus sign (-) and underscore (_)
- No spaces or special characters beyond the above.

4.3.10.2. Administrator *cn=DomainAdmin,cn={DOMAIN}*

This account has full control over the customer domain tree.

Property key: `domainadmin.credentials` (Credentials)
Default value: in the default passwords KeePass database

Username

Property key: `domainadmin.user` (LDAP DN) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: **`cn=DomainAdmin,cn=`*`{DOMAIN}`*** (**`cn=DomainAdmin,cn=My-Company`** for the sample domain).

This property is **read-only** and cannot be changed manually. The {DOMAIN} is replaced by the provided domain name.

Password

Property key: `domainadmin.password` (Password) *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
Default value: in the default passwords KeePass database (for the sample domain).

The password of the administrator. If the domain does not yet exist in the directory, it will be stored in the LDAP for future logins.

4.3.11. System-wide Configuration

This dialog is only displayed when either **ActiveMQ Message Broker** or **Java-based Server** or **C++-based Server** is selected.

Activate High Availability

Property key: `systemwide.ha` (Boolean)
Default value: **false** (unchecked)

Activate the High Availability functionality system-wide, which is selectable **only** if the HA license is installed. The Directory Server for Connectivity step must also be selected for this option to be available.

Use SSL

Property key: `systemwide.ssl` (Boolean)
Default value: **false** (unchecked)

Use SSL for all connections to the ActiveMQ Messaging Server(s) and to the Java Servers running in the system. The Directory Server for Connectivity step must also be selected for this option to be available.

SSL cannot be set on new installations because some certificate and keystore- and truststore-generating scripts as described in Securing Identity Server Connections with SSL must be run first.
When preparing the above-mentioned scripts for generating certificates and key and trust store, be sure that you specify the local machine name (hostname) the same way - either in the short form or in the fully-qualified name form, which is recommended in a wide area network - as you intend to do in the configuration steps for the ActiveMQ Message Broker, Java-based Server and C++-based Server. They all write the specified hostname into the same Connectivity system object, which must match the name contained in the server certificate for that machine.

If SSL is selected, the following options are displayed:

Keystore Password

Property key: `systemwide.keystore_pwd` (Password)
Default value: in the default passwords KeePass database.

The passwords are written to the *<DXI_INSTALL_PATH>*`/ssl/password.properties` file. Be sure that you specify the same passwords as you did when generating the store.

If encryption mode is configured at the Connectivity **Configuration** object or the `cn=server_admin,dxmC=DirXmetahub` object contains a certificate, the Configurator asks you to set:

Pin

Property key: `systemwide.pin` (PIN)
Default value: in the default passwords KeePass database.

Used for reading the private key from the **server_admin** object.

Previous Pin

Property key: `systemwide.previous_pin` (PIN)
Default value: in the default passwords KeePass database.

Used for reading the previous private key from the **server_admin** object.

The PINs are written to the *<DXI_INSTALL_PATH>*`*/ssl/password.properties*` file. Be sure that you specify the same PINs as you did when generating the certificate and private key for the Connectivity **server_admin** object with the `dirxgenpse` tool.

If client signature is configured at the Provisioning **Domain** object, the Configurator asks you to set:

Signature Pin

Property key: `systemwide.signature_pin` (PIN)
Default value: in the default passwords KeePass database.

Used for reading the private key from the **DomainAdmin** user object.

The domain-specific signature PIN is written to the *<CURRENT_JAVA_SERVER_FOLDER>*/private/**password.properties** file (if a Java-based Server is also configured in this configuration run). Be sure that you specify the same PIN as you did when generating the certificate and private key for the Provisioning **cn=DomainAdmin,cn=*<DOMAIN_NAME>*** object with the **dirxgenpse** tool.

The **password.properties** file is always written in this step, no matter whether ssl, encryption or client signature is set. If one of them is not set and so the Configurator does not ask for the related password or PIN, the default values are written to the **password.properties** file.

4.3.12. Java-based Server

This step allows you to configure a DirX Identity Java-based Server to a specific domain, which was set in the Domain Configuration step. You can configure multiple servers per domain.

Server to configure

Property key: NONE
Default value: Java Server whose values are loaded from the properties (Configuration) file

The drop-down list contains three types of entries:

- ***CONFIGURATION FILE*** - There is only a single entry of this type, which is always present and selected by default. It represents the Java-based Server whose values are loaded from the properties file. Used mainly for the silent configuration.
- ***LDAP*** - There can be any number of these entries (also none). They are loaded from the LDAP and filtered for currently selected domain. You can select one of these to reconfigure existing server.
- ***NEW*** - There is only a single entry of this type, which is always present. This should be selected if you want to configure new server in normal configuration mode. Some of the values are calculated automatically based on the existing servers in the LDAP.

You are not allowed to update a Java-based Server for the domain specified in **Domain Configuration** if that Java-based Server is already configured for another domain.

Display name

Property key: `IdS-J.server_name`
Default value: resolved by the values, described below.

The name consists of the *technical domain name*, the incremented number based on the amount of servers configured for this domain (*n*), and the host name the server runs on. The full format is ***<TECHNICAL_DOMAIN_NAME>*-Sn-*<HOST_NAME>***.

There are multiple checks for the consistency of the name done when you try going to the next step. If the name is not consistent, an error message is displayed and you cannot continue.
This may happen when your host can be accessed via different names (and also if you want to use **localhost**). In these cases, you can deactivate the checks by setting the following property in the properties file: `IdS-J.relaxed_name_check=1`. **Take extra care when doing so.**

Host

Property key: `IdS-J.host` (String)
Default value: resolved **localhost**

FQDN of the system on which the Java-based Server runs. Can be literal **localhost**, but then the check described above must be deactivated. If SSL is used or planned to be used for the system, specify the host name in the same form - short or fully-qualified - as for **generating the server certificate** for this host.

System heap size

Property key: NONE
Default value: resolved from the system

This property is **read-only**. It shows the total amount of memory available on the system in gigabytes (GByte) resolved from the system.

IdS-J heap size

Property key: `IdS-J.heap_size` (Size)
Default value: **2** GByte

The heap size for the Java-based Server. The value must be specified in gigabytes.

IdS-J HTTP(S) Port

Property key: `IdS-J.port` (Port)
Default value: **40*n*00**, where *n* is the number of Java-based Servers configured for this domain, incremented by one for a new one, starting from 0.

IdS-J JMX Port

Property key: `IdS-J.jmx_port` (Port)
Default value: **40*n*0*m***, where *n* is the number of Java-based Servers configured for this domain, incremented by one for a new one, starting from 0, and *m* is by default **5**.

The port for JMX monitoring. **Note that the port (*m*+1) is also configured and used.**

Logging files folder path

Property key: `IdS-J.log_path` (Path)
Default value: **`../logs`**

The path to the folder where the Java-based Server writes its warning and server logging files. This path is also used for classloading logging and other items.

4.3.13. Java-based Service

This step is the same as the step **ActiveMQ Message Broker Service Account** described above, but it is used to configure the DirX Identity C++-based Server service.

It uses the following property keys (see the mentioned step for details):

- `IdS-J-service.domain`
- `IdS-J-service.credentials`
- `IdS-J.start_type_automatic`
- `IdS-J.start_after_configuration`
- `IdS-J-service.account` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*
- `IdS-J-service.password` *(deprecated since 9.0.0, usable only from configuration.ini file, use `.credentials` property with KeePass database instead)*

At the end of this step, the Configuration Wizard always saves a template file containing the Java-based Server properties. The default path to the file is *<DXI_INSTALL_PATH>* /***<IDSJ-DISPLAY-NAME>*-configuration.tpl**. Here is an example of a template file:

```
IdS-J.heap_size=2 GByte
IdS-J.host=HOSTNAME
IdS-J.port=40000
IdS-J.jmx_port=40005
IdS-J.protocol=http
IdS-J.log_path=../logs
IdS-J.server_name=My-Company-S1-HOSTNAME
IdS-J.start_after_configuration=1
IdS-J.start_type_automatic=1
IdS-J-service.domain=mydomain
IdS-J-service.account=myaccount
IdS-J-service.password=<PASSWORD>
// for silent configuration only:
domain=My-Company
tech_customer_domain=My-Company
domainadmin.user=cn=DomainAdmin,cn=My-Company
domainadmin.password=<PASSWORD>
```

You can create these files yourselves to configure several Java-based Servers automatically

in silent mode. For this purpose, create one configuration file for each Java-based Server in `<DXI_INSTALL_PATH>/deployment/conf/idsj`. Then run the wizard for each Java-based Server, specifying the name of the Java-based Server configuration file as the third parameter to the script files, or modifying the relevant variables in those script files. You can also specify the Domain properties in this file, otherwise they will be read from the properties file.

4.3.14. Tomcat Configuration

This step allows you to configure the Web Container Configuration, which is based on Apache Tomcat. You can find the supported versions in the release notes.

When upgrading to a new Tomcat version, please install it into the same installation folder as the existing version. The current deployment architecture supports only a single Tomcat instance. Changing the installation folder does not create a new instance; therefore, the web applications will not be redeployed and the new Tomcat version will not be activated.
This limitation will be addressed in a future release.
If you need to install Tomcat in a different folder, first back up the configuration files from the existing installation directory. After installing the new version, manually copy these configuration files from the backup into the new installation directory and redeploy the web applications accordingly.

▾ *details for upgrading (external) Apache Tomcat*

Sometimes it is necessary to update the external Tomcat because of security reasons.

You can do this but you should stick on the major Tomcat version, which is for all DXI 9.0 versions **Tomcat 11**.

In this section it is assumed, that the tomcat is installed in the default directory folder:

`C:\Program Files\Apache Software Foundation\Tomcat 11.0`.
In the following this folder path is replaced with ***<tomcat-installation-folder>***.

- Download a new tomcat '32-bit/64-bit Windows Service Installer' from http://tomcat.apache.org/download-11.cgi (for DXI 9.0)
- Stop the current tomcat service.
- It would be a good decision to backup the complete folder *<tomcat-installation-folder>* to a save place. But at least - and this is mandatory - backup the following files and folders to some save place, e.g. `C:\Inst\`:
  - *`<tomcat-installation-folder>`*`\conf\Catalina\`
  - *`<tomcat-installation-folder>`*`\dxilib\`
  - *`<tomcat-installation-folder>`*`\conf\server.xml`
  - any additional changed configuration files inside *`<tomcat-installation-folder>`*`\conf\` (look at the dates of the files)

    - the log files (if you need them) from *<tomcat-installation-folder>*\logs
- Uninstall completely the current Tomcat using 'Control Panel –> Programs and Features'
- The folder *<tomcat-installation-folder>* should now be gone
- Install the new Tomcat with the default configuration. You can deactivate for security reasons the options `Documentation` and `Manager` while installation.
- Stop the tomcat service, if it was started.
- Copy the folders from the backup back to
  <tomcat-installation-folder>\conf\Catalina\
  <tomcat-installation-folder>\dxilib\
- Edit the file *<tomcat-installation-folder>*\conf\server.xml and add the changes from the file server.xml from the backup into the new server.xml. Mainly it is the section starting with '<Connector port="8443"' (or some other port you are using). A diff tool e.g. from Notepad++ could be helpful.

Only Windows:

- Run:
  <tomcat-installation-folder>\bin\Tomcat11w.exe
- Switch to the tab 'Java' and add the following two lines to 'Java Options':
  `-Dorg.apache.catalina.connector.RECYCLE_FACADES=false`
  `-Dcom.sun.jndi.ldap.object.disableEndpointIdentification=true`
  (This second line is only neceassary, if your LDAPS (DirX Directory) server certificate does not contain the correct name of the server. If you have a correct name inside the certificate, it should not be necessary to add this line - for security reasons.)
- Add the following extensions to the end of the 'Java Classpath':
  ; *<tomcat-installation-folder>*\dxilib*;
  <DXI_INSTALL_PATH>\lib\java\custom*
- Click on 'OK'
- Start the tomcat service

Only Linux:

- Edit the file: *<tomcat-installation-folder>*/bin/catalina.sh
  and add below the line
  `# Control Script for the CATALINA Server`
  this line:
  `JAVA_OPTS="$JAVA_OPTS`
  `-Dcom.sun.jndi.ldap.object.disableEndpointIdentification=true`
  `-Dorg.apache.catalina.connector.RECYCLE_FACADES=false"`
  (The part `-Dcom.sun.jndi.ldap.object.disableEndpointIdentification=true` is only necessary, if your LDAPS (DirX Directory) server certificate does not contain the correct name of the server. If you have a correct name inside the certificate, it should not be necessary to add this line - for security reasons.)

- Start the tomcat service

The web container here will be configured for the domain specified in **Domain Configuration**, for the web applications you have selected in the Options step. These applications are:

- DirX Identity Web Center (*webCenter*)
- DirX Identity Web Center for Password Management (*pwdManagement*)
- DirX Identity Provisioning Web Service (*ProvisioningService*)
- DirX Identity REST Service (*DirXIdentityRestService*)
- DirX Identity Business User Interface (*BusinessUserInterface*)

These applications are deployed to the folder *<DXI_INSTALL_PATH>*`/web/instances`. Each application is deployed to its own folder, named after the application. The format of the name is *<APP_NAME>*`-`*<TECHNICAL_DOMAIN_NAME>*, where the *<APP_NAME>* is the name above in the parentheses, and *<TECHNICAL_DOMAIN_NAME>* is the technical domain name of the currently specified domain. For example, the Web Center for My-Company domain would be deployed to *<DXI_INSTALL_PATH>*`/web/instances/webCenter-My-Company`.

The DirX Identity Web Center can be set up with other Web servers, too. Contact your support group to get more information about configuration with your specific Web server.

4.3.14.1. Server Configuration

Host

Property key: `tomcat.host` (String)
Default value: resolved **localhost**

This property is **read-only**. It shows the FQDN of the system this server runs on.

Port

Property key: NONE
Default value: **8080** (non-SSL), **8443** (SSL)

This property is **read-only**. It changes dynamically depending on whether system-wide SSL is used or not.

SSL

Property key: NONE
Default value: **false** (non-SSL), **true** (SSL)

This property is **read-only**. It changes dynamically depending on whether system-wide SSL is used or not.

4.3.14.2. Apache Tomcat

Installation Path

Property key: `tomcat.path` (Path)
Default value: resolved from environment variable *`CATALINA_HOME`* or *`CATALINA_BASE`*. Otherwise, on Windows from the registry, and on Linux from several standard paths.

The path to the Apache Tomcat installation folder. The path is validated based on the contents of the folder.

Version

Property key: `tomcat.version` (Version)
Default value: resolved from scripts in the installation path.

This property is **read-only**. It is resolved when the installation path changes, either from one of its programs, or from the registry on Windows.

4.3.14.3. Service

Service Name (Windows only)

Property key: `tomcat.service_name` (String)
Default value: **Apache Tomcat** or **Tomcat** with the version number appended, for example **Tomcat11**.

The name of the Windows service.

Executable Path (Windows only)

Property key: `tomcat.service_executable_path` (Path)
Default value: resolved based on the installation path.

The path to the **Tomcat service application** executable file. This file is located in the `<TOMCAT_INSTALL_PATH>/bin` folder and by default is named after the service name, the format is *`<TOMCAT_SERVICE_NAME>`*`.exe`, so for the service named Tomcat11, it would be `Tomcat11.exe`. This field was introduced to allow you to select a different file if you have renamed the service executable file. If you have not renamed it, you should accept the default value.

There is also the **Tomcat monitor application** executable file in the same folder, which is named after the service name too, but has ***w*** appended. For the service named Tomcat11, it would be `Tomcat11w.exe`. Be sure to select the correct file. You can find more about it here.

Start after Configuration

Property key: `tomcat.start_after_configuration` (Boolean)
Default value: **true** (checked)

If checked, the service is attempted to be started after Configuration Wizard (Deployment Tool) finishes.

4.3.15. HCL Notes Client

This step is only displayed on Windows platforms when **C++-based Server Configuration** is selected.

Path

Property key: `path.notes` (Path)
Default value: **`C:\HCL\Notes`**

The path to the HCL Notes Client installation folder. The folder does not need to exist yet, but the path format is validated. You can install the HCL Notes Client after the configuration procedure.

Reboot the machine after the configuration is completed before using connectivity to the HCL Notes system.

4.3.16. ODBC Library Path

This step is only displayed on Linux platforms when **C++-based Server Configuration** is selected.

Path

Property key: `path.odbc` (Path)
Default value: **`/opt/odbc/lib`**

The path to the ODBC library files. The folder does not need to exist yet, but the path format is validated. You can install the ODBC library files after the configuration procedure.

4.3.17. SAP ECC UM Library Paths

This dialog is only displayed on Linux platforms when **C++-based Server Configuration** is selected.

SAP JCO library path

Property key: `path.sap.um` (Path)
Default value: **`/opt/sapjco`**

The path to the SAP JCO library folder. The folder does not need to exist yet, but the path format is validated. You can install the SAP JCO library file after the configuration procedure.

SAP JCO JAR file

Property key: `path.sap.jco` (Path)

Default value: **`/opt/sapjco/sapjco.jar`**

The path to the SAP JCO JAR file. The file does not need to exist yet, but the path format is validated. You can install the SAP JCO JAR file after the configuration procedure.

4.3.18. Pre-Configuration Summary

This is the last step before the configuration procedure starts. You can review all your inputs here. If everything is correct, click **Next** to start the configuration procedure.

Pressing **Next** starts the configuration procedure immediately. The properties and credentials files are also saved automatically at this point. If you want to change something, press **Back** to return to the previous steps.

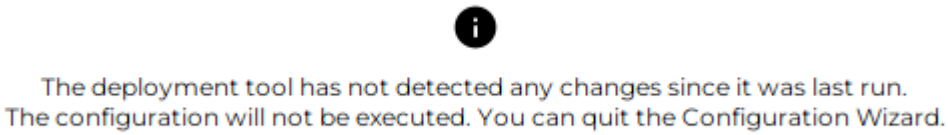
4.3.19. Configuration in Progress

This step displays the progress of the configuration & deployment procedure. It shows all the tasks that are performed in the order they are performed. Colors indicate the status of each step.

The configuration is performed by the Deployment Tool, which is started in the background. This may take few seconds. The Configuration Wizard monitors its progress and updates the display accordingly. The Tool is started with following command: *<DXI_INSTALL_PATH>*`/bin/dxictl(.sh|.bat) configure -i` *<PROPERTIES_FILE>* `-c` *<CREDENTIALS_FILE>* `-m` *<MASTER_PASSWORD>* `-k` *<KEY_FILE>*. The arguments are the same as described in the Commands and arguments, taken from what you have provided for the Configuration Wizard.

A KeePass database with the passwords used during the configuration will be created with each run - the path is *<DXI_INSTALL_PATH>*`/deployment/conf/`**`lastrequest.kdbx`**. This database can be used to retrieve the passwords used during the last configuration run. If you have used a KeePass database for credentials, the master password and key file (if any) used to open that database will also be used to create this database. If you have used an INI file for credentials, the default master password (see Default Credentials File) will be used to create this database.

The tasks are the ones described in the Deployment Tasks. Since Deployment Tool supports Persisted State, you may see the following message:



This means that no changes were detected between the current persisted state and the provided configuration file, so no tasks were performed.

Otherwise, the configuration ends when all the steps finish successfully or when one of the

steps fails. At that point, the button to view the log file becomes active. Clicking it opens the Deployment Tool log file in the application text editor.

It's possible that the Deployment Tool doesn't calculate the architecture changes correctly, resulting in the message above. The calculation is based on the FQDN of the machine where the Deployment Tool was run, and where it's to be run. See FQDN configuration for more details.

You can set property `deploymentTool.force=1` in the properties file to force all tasks to be performed again. The property and value is automatically removed from properties file after the run.
Use this option as last resort, as it wipes out the persisted state and forces all tasks to be performed again, which may lead to data loss.

If all steps are performed successfully, you have completely configured DirX Identity.

Don't forget to set the correct passwords for all the pre-configured accounts in the DirX Identity database. See Managing Administrative Accounts for details.

4.4. Deployment Tool

The Deployment Tool is a command-line application that performs the actual configuration of DirX Identity based on the provided architecture YAML or configuration INI file. It is used by the Configuration Wizard, but can also be used on its own to integrate into your own deployment processes.

It is located in the folder *<DXI_INSTALL_PATH>*`/bin` and is named ***`dxictl.bat`*** on Windows systems and ***`dxictl.sh`*** on Linux systems. See Commands and arguments for the usage. An example call to configure DirX Identity is *<DXI_INSTALL_PATH>*`/bin/dxictl.bat configure -a` *<ARCHITECTURE_FILE>* `-c` *<CREDENTIALS_FILE>* `-m` *<MASTER_PASSWORD>* `-k` *<KEY_FILE>*.

4.4.1. Persisted State

The state of the configuration is persisted in the integrity protected file *<DXI_INSTALL_PATH>/deployment/conf/architecture.bin* file. This file is created automatically when the Deployment Tool is run for the first time. It is used to keep track of the current state of the deployment, so that subsequent runs can determine which tasks need to be performed.

Every time the Deployment Tool is run with the **`configure`** command, it compares the provided architecture YAML or configuration INI file with the persisted state in the `architecture.bin` file. Based on this comparison, it determines which deployment tasks need to be performed to bring the system to the desired state. This means that if you run the Deployment Tool with unchanged configuration, it will not perform any tasks, as the system is already in the desired state.

Each run is also recorded in integrity protected file *<DXI_INSTALL_PATH>/deployment/conf/requests.bin*.

Make sure that the `architecture.bin` file is not deleted or modified manually, as this can lead to inconsistent state and unexpected behavior during subsequent runs of the Deployment Tool.

4.4.2. Logging

For each run of the Deployment Tool, a log file is created in the folder *<DXI_INSTALL_PATH>/logs*, with name pattern `Deployment_<START_TIMESTAMP>.log`. You can use this log file to troubleshoot any problems that may occur during the configuration process, or attach it to the support ticket if you cannot resolve the problem yourselves. The error is usually at the end of the file.

4.4.3. Commands and arguments

The application has several commands that perform different deployment related tasks. These commands are:

4.4.3.1. configure

Configures DirX Identity based on the provided architecture YAML or configuration INI file. The configuration INI file option is used when running the Deployment Tool from Configuration Wizard, but can also be used on its own. See Tasks for the possible deployment tasks.

- ***-a***, ***--architecture-file*** *<PATH>* **(required)** - Specifies the path to the YAML architecture file.
- ***-i***, ***--configuration-ini-file*** *<PATH>* **(required)** - Specifies the path to the INI configuration file. This option is exclusive with the ***-a*** option, meaning that you can use either one of them, but not both at the same time.
- ***-c***, ***--credentials-file*** *<PATH>* **(required)** - Specifies the path to the credentials file. The supported format is KeePass 2 **.kdbx** file.
- ***-m***, ***--master-password*** *<PASSWORD>* **(required)** - Specifies the master password to open the KeePass database file.
- ***-k***, ***--key-file*** *<PATH>* - Specifies the path to the key file **.key** or **.keyx** of the KeePass database. Only to be used if the KeePass database requires it.
- ***-u***, ***--user*** *<USER>* **(required - Linux only)** - Specifies the DirX Identity technical user.
- ***-g***, ***--group*** *<GROUP>* **(required - Linux only)** - Specifies the DirX Identity technical group.
- ***-s***, ***--simulate*** - Runs the configuration in simulation mode, only determining which tasks to run. No changes are made to the system.
- ***--fqdn-override*** *<FQDN>* - Overrides the FQDN of the system where the Deployment Tool is run. This is useful when the FQDN cannot be resolved correctly. See FQDN

configuration for more details.

- ***--force*** - Forces all tasks to be performed, ignoring the persisted state. **Use this option as last resort, as it wipes out the persisted state and forces all tasks to be performed again, which may lead to data loss.**

When run from a console, the output will contain a line for each task that is performed, along with its status [OK] or [FAIL]. In simulation mode, the output will contain a line for each task that would be performed, along with status [OK].

4.4.3.2. unconfigure

Unconfigures all DirX Identity components from the system.

- ***-c***, ***--credentials-file*** *<PATH>* **(required)** - Specifies the path to the credentials file. The supported format is KeePass 2 **.kdbx** file.
- ***-m***, ***--master-password*** *<PASSWORD>* **(required)** - Specifies the master password to open the KeePass database file.
- ***-k***, ***--key-file*** *<PATH>* - Specifies the path to the key file **.key** or **.keyx** of the KeePass database. Only to be used if the KeePass database requires it.
- ***-s***, ***`--simulate*** - Runs the unconfiguration in simulation mode, only determining which tasks to run. No changes are made to the system.
- ***--fqdn-override*** *<FQDN>* - Overrides the FQDN of the system where the Deployment Tool is run. This is useful when the FQDN cannot be resolved correctly. See FQDN configuration for more details.

4.4.3.3. export

Exports the persisted architecture from the system to a YAML architecture file.

- ***-a***, ***--architecture-file*** *<PATH>* **(required)** - Specifies the path to the YAML architecture file to export to.

4.4.3.4. start

Starts all DirX Identity services on the system.

4.4.3.5. stop

Stops all running DirX Identity services on the system.

There are additionally some global options that can be used with any command:

- ***-h***, ***--help*** - Displays help about the application or a specific command.
- ***-V***, ***--version*** - Displays the version of the Deployment Tool.
- ***-I***, ***--indent*** *<NUMBER>`* - Specifies the number of spaces to use for indentation in the console, default is 0.

4.4.4. Exit Codes

The Deployment Tool returns the following exit codes:

- `0` - Success
- 1 - General error
- 2 - `DeploymentException` occurred
- -2 - Missing argument `-c` (Credentials file)
- -3 - Missing argument `-m` (Master password)
- -4 - Missing argument `-a` (Architecture file)

4.4.5. Tasks

4.4.5.1. Stop Services Task

Stops all running DirX Identity services on the system. If no services are running, the task is skipped.

4.4.5.2. Connectivity Schema Task

The schema is installed by running command *<DXI_INSTALL_PATH>*`/deployment/schema/dirx-ee/dirxadm SchemaTool.adm` (`dxmEEdirxadm` on Linux) against the local LDAP DirX Directory server.

The files with prefix `dirxabbr-` are copied from *<DXI_INSTALL_PATH>*`/deployment/schema/role-dirx/SystemDomain` and *<DXI_INSTALL_PATH>*`/client/conf` to the *<DXD_INSTALL_PATH>*`/client/conf` folder in the local DirX Directory installation. The file `dirxcl.cfg` is copied in opposite direction. File `dirxabbr-ext.DirXmetahub.Admin` is deleted in the Directory folder.

4.4.5.3. Connectivity Data Task

Imports the archive *<DXI_INSTALL_PATH>*`/deployment/factory-data/dirx-identity-conn-<VERSION>.l2fs` into the local LDAP DirX Directory server using the LDAP2FS tool. This archive contains the initial data for the Connectivity system domain.

The Configuration Objects of `dxmC=DirXmetahub` are created and modified, including attributes like `dxmHAactive` for High Availability and `dxmSSL` for SSL.

4.4.5.4. Provisioning Schema Task

If Connectivity Schema is not installed, performs the copy of `dirxabbr-` files as described in Connectivity Schema Task.

Installs schema by running command *<DXI_INSTALL_PATH>*`/deployment/schema/dirx-ee/dirxadm Setup.adm` (`dxmEEdirxadm` on Linux) in working directory *<DXI_INSTALL_PATH>*`/deployment/schema/role-dirx/SystemDomain` against the local

LDAP DirX Directory server.

4.4.5.5. Provisioning Customer Domain Task

Installs schema by running command *<DXI_INSTALL_PATH>*/deployment/schema/dirx-ee/dirxadm Setup.adm (dxmEEdirxadm on Linux) in working directory *<DXI_INSTALL_PATH>*/deployment/schema/role-dirx/CustomerDomain against the local LDAP DirX Directory server.

Imports the archive *<DXI_INSTALL_PATH>*/deployment/factory-data/dirx-identity-prov-<VERSION>.l2fs into the local LDAP DirX Directory server using the LDAP2FS tool. This archive contains the initial data for Provisioning of the customer domain.

If Password Management is licensed, imports the archive *<DXI_INSTALL_PATH>*/deployment/factory-data/dirx-identity-prov-pwmgmt-<VERSION>.l2fs into the local LDAP DirX Directory server using the LDAP2FS tool. This archive contains the initial data for Password Management of the customer domain.

At last, some domain objects are modified, domain admin is added and licenses are updated.

4.4.5.6. Provisioning Sample Domain Task

Performs the same actions as described in Provisioning Customer Domain Task, but for the sample domain only. Additionally, agent schema is installed.

4.4.5.7. Provisioning System Domain Task

Imports the archive *<DXI_INSTALL_PATH>*/deployment/factory-data/dirx-identity-prov-system-<VERSION>.l2fs into the local LDAP DirX Directory server using the LDAP2FS tool. This archive contains the initial data for Provisioning of the system domain.

The Configuration Objects of dxmC=DirXmetahub and cn=DirXmetaRole-SystemDomain are modified.

4.4.5.8. Message Broker Task

Templates the folder *<DXI_INSTALL_PATH>*/messagebroker.template to *<DXI_INSTALL_PATH>*/messagebroker, customizing the configuration files based on the provided architecture.

Updates System and Service (with ports) entries in LDAP, and created the entry for the Message Broker in the Connectivity Configuration Object.

4.4.5.9. Message Broker Service Task

Creates and customizes the service start/stop scripts in **<DXI_INSTALL_PATH>*/etc/ (Linux) or Windows Service (Windows) for the DirX Identity Message Broker component. If Custom account is selected, the service is created to run under the specified account (Windows only).

4.4.5.10. IdS-C Task

Writes the TCL property files `basic.input.tcl` and `components.input.tcl` to the folder `_<DXI_INSTALL_PATH>/`, customizing them based on the provided architecture.

Templates the file `_<DXI_INSTALL_PATH>/server/conf/dxmmsssvr.ini`.

On Windows, sets HCL Notes path into Registry. On Linux, templates ODBC and SAP ECC UM library scripts.

4.4.5.11. IdS-C Service Task

On Windows, installs the service *<DXI_INSTALL_PATH>*`/bin/dxmsvr.exe`, with custom account if provided.

On Linux, templates the start/stop script *<DXI_INSTALL_PATH>*`/etc/S99dmsvr`.

4.4.5.12. IdS-C Dependencies Task

Only valid for Windows installations. Installs the required Microsoft Visual C Redistributable packages v9.0 (2008) and v14 (2015+) and PCMX application for the DirX Identity C-based Server component. If applications are installed already, they are not reinstalled.

4.4.5.13. IdS-J Task

Templates the folder *<DXI_INSTALL_PATH>*`/ids-j.template` to *<DXI_INSTALL_PATH>*`/ids-j-<IDSJ_SERVER_NAME>`, customizing the configuration files based on the provided architecture.

Updates System and Service (with ports) entries in LDAP, and created the entry for the IdS-J container and entry in the Connectivity Configuration Object.

4.4.5.14. IdS-J Service Task

On Windows, installs the service *<DXI_INSTALL_PATH>/<IDSJ_FOLDER>*`/ids-j.exe`, with custom account if provided.

On Linux, templates the start/stop script *<DXI_INSTALL_PATH>*`/etc/S99dmsvrj`.

4.4.5.15. DirX Identity Manager Task

Creates the file *<DXI_INSTALL_PATH>*`/gui/dxi.cfg`, customizing it based on the provided architecture.

Creates the server profiles

- *<DXI_INSTALL_PATH>*`/gui/profiles/DirXmetahub-<CONN_ID>.server` for Connectivity,
- *<DXI_INSTALL_PATH>*`/gui/profiles/<DOMAIN_NAME>.server` for each Provisioning domain,

- *<DXI_INSTALL_PATH>*/gui/profiles/data.server for Provisioning Sample domain.

4.4.5.16. Web Container Task

The supported web container is Apache Tomcat.

On Windows, updates Java options for Tomcat service in registry and extends its classpath.

On Linux, creates the file *<TOMCAT_INSTALL_PATH>*/bin/setenv.sh with custom Java options and extends its classpath.

The classpath is extended to include folder *<TOMCAT_INSTALL_PATH>*/dxilib/ and value of Custom Libraries path.

4.4.5.17. Web Applications Task

A common task for all web applications and services. Templates the instance into folder *<DXI_INSTALL_PATH>*/web/instances/<APP_NAME>-<TECHNICAL_DOMAIN_NAME>. Moves endorsed files to the web container. Deploys the web application to the web container.

The web applications are:

- DirX Identity Web Center
- DirX Identity Web Center for Password Management
- DirX Identity Provisioning Web Service
- DirX Identity REST Service
- DirX Identity Business User Interface
- DirX Identity Server Admin REST Service

4.4.5.18. Start Services Task

Starts the DirX Identity services that were installed and configured on the system, with "Start after Configuration" option set.

This task is run even if one of the previous tasks failed.

4.5. LDAP2FS Tool

The LDAP2FS tool is a command-line application that allows you to export and import LDAP directory data to and from the file system archive. It is located in the folder *<DXI_INSTALL_PATH>_/bin*, and is named ***ldap2fs.bat*** on Windows systems and ***ldap2fs.sh*** on Linux systems. See Commands and arguments for usage. An example call to import a basic archive is *<DXI_INSTALL_PATH>*/bin/ldap2fs.bat import -f *<ARCHIVE_FILE>* -H *<LDAP_HOST>* -p *<LDAP_PORT>* -D *<BIND_DN>* -w *<BIND_PASSWORD>*.

The file system archive file .l2fs is a compressed file that contains the LDAP entries in LDIF format, along with any binary data associated with those entries. The archive can be used for backup purposes, data migration, or for transferring data between different LDAP

directory servers.

You can find default archives in the folder *<DXI_INSTALL_PATH>*`/deployment/factory-data`.

4.5.1. Commands and arguments

The application has two commands, both with the same set of arguments:

4.5.1.1. import

Imports LDAP directory data from an archive file to the LDAP directory.

4.5.1.2. export

Exports LDAP directory data from the LDAP directory to an archive file.

- ***-f***, ***--fs-path*** *<PATH>* **(required)** - Specifies the path to the archive file.
- ***-H***, ***--ldap-host*** *<HOST>* - Specifies the LDAP server host. Default is **localhost**.
- ***-p***, ***--ldap-port*** *<PORT>* - Specifies the LDAP server port. Default is **389**.
- ***-S***, ***--ldaps*** - Use LDAPS to connect to the LDAP server. Default is not to use LDAPS.
- ***-D***, ***--bind-dn*** *<DN>* - Specifies the bind DN to use for authentication. Default is empty.
- ***-w***, ***--bind-password*** *<PASSWORD>* - Specifies the bind password to use for authentication. Default is empty.
- ***-F***, ***--settings-file-path*** *<PATH>* - Overrides path of settings file.
- **-m_**, ***--mode*** *<MODE>* - Override mode during import/export with the specified value. The possible values are:
  - **`full`** - Full import/export of all entries.
  - **`selective`** - Partial import/export of specified entries only.
- ***-r***, ***--root-dn*** *<DN>* - Override root DN during import/export with the specified value.
- ***-e***, ***--select_entries*** *<DN1;DN2;...>* - Override selected entries during import/export with the specified values, separated by semicolons.
- ***-s***, ***--select-subtrees*** *<DN1;DN2;...>* - Override selected subtrees during import/export with the specified values, separated by semicolons.
- ***-c***, **`--context-prefix`** *<PREFIX>* - Override context prefix (e.g. cn=Domain) during import/export with the specified value.
- ***-v***, ***--verbose*** - Print verbose output to stdout.
- ***-h***, ***--help*** - Displays help about the application or a specific command.
- ***-V***, ***--version*** - Displays the version of the LDAP2FS Tool.

4.5.2. Exit Codes

The LDAP2FS Tool returns the following exit codes:

- `0` - Success or entry kept (during delete)
- 1 - Aborted
- 2 - Finished with errors
- -1 - Invalid settings and/or arguments

4.6. Integrating Start/Stop Scripts into the Linux Operating System

For Linux platforms, the following DirX Identity service components are not necessarily stopped during system shutdown and not necessarily started during system start:

- Message Broker
- C++-based Server
- Java-based Servers

This section describes how to use the DirX Identity integration utility to integrate start and stop scripts for these DirX Identity components into Linux and how to undo the integration if necessary.

4.6.1. Using the Integration Utility

Use the **updrcs-linux.sh** utility located in *<DXI_INSTALL_PATH>***/etc** to integrate or unintegrate the start/stop scripts created for the listed DirX Identity components in **/etc/init.d**. The default names for these scripts are:

- **dmsvr** for the C++-based Server
- **dmmbrk-***number* for a DirX Identity Message Broker
- **dmsvrj-***technical_domain_name***-S***number* for a Java-based Server

It is unlikely but possible that a script with a default name already exists in this folder which does not belong to DirX Identity. The utility automatically detects this kind of naming conflict. In this case, the utility can be customized by modifying the value of the shell variable **`UNIQUESUFFIX`** (default: empty string) in order to append a suffix to the default names. To integrate these scripts, perform the Linux command **`chkconfig -add`** with the related script names.

The scripts are similar to the scripts **dmmbrk-*** and **S99*** in *<DXI_INSTALL_PATH>***/etc**. For instance, the script **dmsvr** on SuSE platforms is a concatenation of the suitable INIT-V information in *<DXI_INSTALL_PATH>***/etc/suse/S99dmsvr.txt** *<DXI_INSTALL_PATH>***/etc/S99dmsvr**. Here the related placeholders (like **@dirx@)** are substituted so that they reflect the dependencies correctly. For RedHat platforms, the related file in *<DXI_INSTALL_PATH>*/etc/redhat* is used.

The script **updrcs-linux.sh** uses the technical domain name rather than the original domain name when handling domain-specific components.

For the Message Broker and the Java-based Servers, the script prompts the user to specify whether the related component needs configuration or un-configuration because this cannot be determined from **configuration.ini**.

4.6.2. Performing the Integration

If the DirX Identity components whose start/stop scripts are to be integrated use a co-located DirX Directory Server installation as the configuration and/or Provisioning store, the following prerequisites must be satisfied:

- DirX Directory has been configured so that it is started when entering runlevels 3 or 5 and stopped during shutdown. See the DirX Directory documentation for Linux platforms for further details.

For SuSE, this prerequisite means:

- A start/stop script *dirx_script_name* for DirX (for example, **dirx**) must exist in **/etc/init.d**.
- The Linux command **`chkconfig --list`** *`dirx_script_name`* is successful.

You must be superuser in order to verify these prerequisites.

If these prerequisites are satisfied, perform these steps:

- Login as superuser.
- Using a shell, navigate to *<DXI_INSTALL_PATH>*/etc*.
- Execute the command **`./updrcs-linux.sh`**. For SuSE Linux platforms, *`dirx_script_name`* must be supplied as an input argument if the DirX Identity installation uses a co-located DirX installation.
- The script displays the exit code on screen. An exit code indicates successful execution of the script. A log file **updrcs-linux.sh.log** is also written.

The integration is now complete. The relevant DirX Identity components are started during system startup (in the order listed in "Integrating Start/Stop Scripts into the Linux Operating System") listed and stopped during system shutdown (in reverse order).

Running this script is required whenever the configuration has been changed with respect to these DirX Identity components.

4.6.3. Undoing the Integration

Before uninstalling DirX Identity, the integration must be undone:

- Log in as superuser.
- Using a shell, navigate to *<DXI_INSTALL_PATH>***/etc**.
- Execute the command `./updrcs-linux.sh -cleanup` and then check the exit code

and the log file.

4.6.4. Silent Mode

The Silent Mode is a way to perform the configuration and un-configuration of DirX Identity without user interaction, which is useful for automated deployments.

Since 9.0, using Deployment Tool is preferred way to perform silent configuration and un-configuration.
The silent mode of the Configuration Wizard is still supported, offering more validation before running the Deployment Tool. See Commands and Arguments for details.

Preparing for Silent Configuration:

- If you have previously run the Configuration Wizard, backup the Properties file and credentials file from the installation folder.
- Fill out the Properties file with the desired configuration settings. Select the components to be configured based on properties in the Configuration Options step. You can also run the Configuration Wizard in normal mode to generate the file, exiting with save at the Pre-Configuration Summary step.
- Create (or copy existing) credentials file, fill out the required credentials for the selected components. The **Connectivity Admin** must always be provided. Check the Default Credentials File for details.
- If necessary, create the Java-based Server configuration file (see Java-based Service step).

For a Silent Unconfiguration:

- Run Deployment Tool or Configuration Wizard with the *unconfigure* command.

5. Configuring Single Components

This chapter provides information on how to configure components separately, including:

5.1. Hints for Installing the DirX Directory Server

- Install the directory using default settings.
- If working with an empty database, create an LDAP configuration and an administrator object before running DirX Identity installations (see the DirX Directory administration documentation for details).
- If using the example database, enter:
  - **`cn=admin,o=my-company`**
  - Password: **`dirx`** when prompted for the directory administrator during installation.

f you installed DirX Directory after uninstalling the software without a new DirX Identity setup, restore the database and copy: `dirxabbr-ext.DirXmetahub...` and `dirxabbr-ext.DirXmetaRole...` files from *`<install_path>`*`/`**`client/conf`** to *`<dirx_install_path>`*`/`**`client/conf`**.

DirX Identity with shadow agreements

- Check the indexing requirements (see Indexed Attributes).Insufficient indexes will cause configuration failure!
- If shadow agreements exist, exclude DirX Identity configuration from these agreements.
- If shadowing starts at the root context prefix, DirX Identity configuration data will also be shadowed. Ensure schema parts are present on all shadows.

- Alternatively, separate Connectivity Configuration from the Provisioning database on another DirX Directory server.

5.2. Hints for Configuring the Tomcat Server

Ensure Tomcat has sufficient maximum memory:

- **Windows**: Open **Configure Tomcat** → **Java** tab → set **Maximum memory pool**.
- **Linux**: add at the beginning of Tomcat's startup script **startup.sh**: `JAVA_OPTS=-Xmx512m; export JAVA_OPTS` (Example sets 512 MB)

Restart Tomcat for changes to apply.

5.3. Configuring Connectivity Schema and Data

Steps for separate installation:

1. Ensure a directory server exists (use native tools).
2. Run installation (see Installing DirX Identity):
   - In **Choose Licensed Features Set**, select licensed features.
3. Run configuration (see Configuring DirX Identity):
   - Select **Connectivity Schema and Data Configuration**.
4. Extend directory schema as required (see Indexed Attributes).

5.4. Configuring Provisioning Schema and Data

Follow similar steps as Connectivity:

1. Ensure a directory server exists.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **Provisioning Schema and Data Configuration**.
4. Extend schema as needed.

5.5. Message Broker

1. Prerequisites: Prerequisite: Connectivity Configuration must exist.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **ActiveMQ Message Broker Configuration**.

5.6. C++-based Identity Server

1. Prerequisites: Connectivity Configuration and Message Broker.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **C++-based Server Configuration**.
   - First server type = `primary` (runs Status Tracker); others = `secondary`.
   - Meta Controller (metacp) installs automatically.

Only one C++-based Identity Server per computer.

5.7. Java-based Identity Server

1. Prerequisites: Connectivity Configuration and Message Broker.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **Java-based Server Configuration**.

All servers are `primary` (contain repository). Secondary servers not supported.

5.8. DirX Identity Manager

1. Prerequisite: Connectivity Configuration.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **Manager Configuration**.

5.9. DirX Identity Web Center

1. Prerequisites:
   - Connectivity Configuration
   - Provisioning Configuration (sample or customer domain)
   - Supported Tomcat installation
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **Web Center Configuration**.

5.10. Business User Interface

1. Prerequisites:
   - Connectivity Configuration
   - Provisioning Configuration (sample or customer domain)
   - Supported Tomcat installation
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **Business User Interface Configuration**.

5.11. DirX Identity Connectivity Packages - Agents

1. Prerequisite: Connectivity Configuration.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **C++-based Server Configuration**.

Installed C++-based Server type = **`secondary`**.

5.12. DirX Identity Connectivity Packages - Connectors

1. Prerequisite: Connectivity Configuration.
2. Install DirX Identity (select licensed features).
3. Configure:
   - Select **Java-based Server Configuration**.

6. Other Installation Configurations

This chapter provides information about other installation configurations:

- A sample distributed installation on several machines
- Password Management (including Web Center for Password Management)
- Configurations using DirX Identity High Availability Features

6.1. Distributed Installation

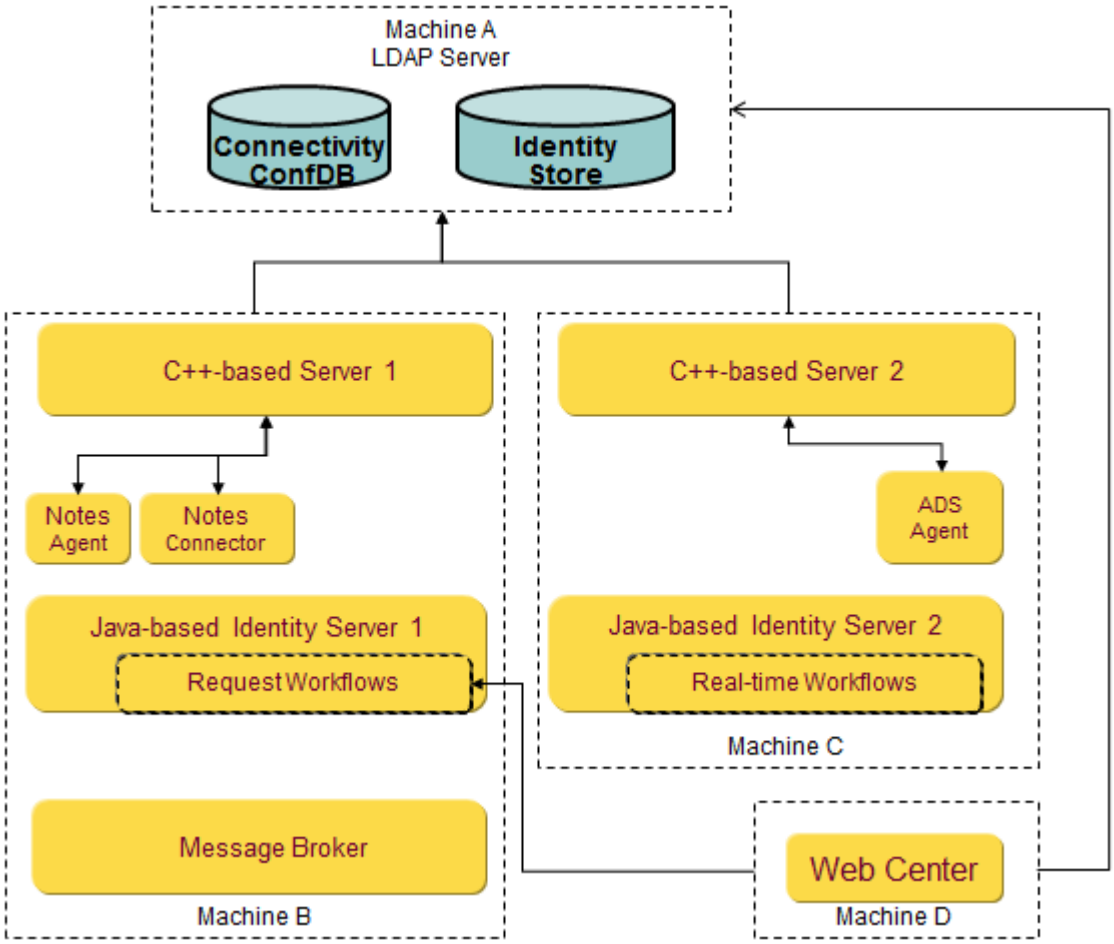
DirX Identity components can be distributed across different machines.

If DirX Identity is installed in a distributed environment, **ensure all machines are updated to the same software version** to avoid severe interworking issues.

You can verify the installed version on each machine by checking the `install_history.txt` file in the installation directory.

6.1.1. Example Distributed Setup

- **Machine A**:
  - DirX Directory with Connectivity configuration
  - Identity Store (users)
- **Machine B**:
  - Message Broker for all DirX Identity components
  - C++-based Identity Server 1 (hosts Notes connector and Notes agent)
  - Java-based Identity Server 1 (handles request workflows)
- **Machine C**:
  - C++-based Identity Server 2 (controls Active Directory agent)
  - Java-based Identity Server 2 (runs real-time workflows for provisioning and password synchronization)
- **Machine D**:
  - Web Center



Distributed Installation

6.1.1.1. License Requirements

- **Business Suite**
- **Professional Suite** (required for request workflows)
- Connectivity packages for Active Directory and Lotus Notes

6.1.1.2. Installation Steps

1. **Machine A** – Install Connectivity Configuration:
    - Follow steps in:
        - "Hints for Installing the Directory Server"
        - "Configuring Connectivity Schema and Data"
        - "Configuring Provisioning Schema and Data"
2. **Machine B** – Install Message Broker, Servers, Manager, and Notes connectivity:
    - Run installation (see Installing DirX Identity)
    - In **Choose Install Set** dialog, select:
        - Message Broker
        - C++-based Server

        - Java-based Server
        - Manager
        - IBM Connectivity Package
3. **Machine C** – Install additional Servers, Manager, and Microsoft connectivity:
    - Run installation (see Installing DirX Identity)
    - In **Choose Install Set** dialog, select:
        - C++-based Server
        - Java-based Server
        - Manager
        - Microsoft Connectivity Package
4. **Machine D** – Install Web Center:
    - Follow steps in DirX Identity Web Center

The configuration procedure writes details about installed servers and connectivity packages into the DirX Identity Connectivity configuration.

6.2. Password Management including specialized Web Center

This section explains how to install the password management feature described in Use Case Description Password Management.

License required: Password Management for installing **Web Center for Password Management**.

6.2.1. Installation Steps (Single Machine)

1. Install a directory server using native tools.
2. Install Tomcat:
    - Select NT Service on Windows platforms
    - Check release notes for supported Tomcat versions and Java environment
3. Run DirX Identity installation (see Installing DirX Identity):
    - In **Choose Licensed Features Set** dialog, select at least **Password Management**
    - In **Choose Install Set** dialog, select components as per “Installation” in *Use Case Description Password Management*
    - Include **Web Center for Password Management Configuration**
4. Run DirX Identity configuration (see Configuring DirX Identity):
    - In **Configuration Options** dialog, select steps as per “Configuration” in *Use Case Description Password Management*

- Complete all configuration dialogs

For details, refer to Use Case Description Password Management.

6.3. Configurations using DirX Identity High Availability Features

Setting up High Availability configurations requires understanding Use Case Description High Availability. Before starting, ensure you have the following licenses:

- **Business Suite**
- **High Availability**
- Connectivity packages for all systems you plan to deploy

6.3.1. Planning Considerations

A robust High Availability setup should include:

- Multiple **C++-based Servers**
- Multiple **Java-based Servers** per provisioning domain (deploy these servers across multiple hosts for redundancy)
- Identical connectivity packages installed on each host
- Two **Message Brokers** on different machines with a shared message repository
- Shared folders for:
- Message repository
- Each Java-based Server repository
- Any additional components required for productive use

6.3.2. Installation and Configuration Steps

1. **Install DirX Identity on each relevant host** (see Installing DirX Identity):
   - In the **Choose Licensed Features Set** dialog:
     - Select all licensed features (must be identical across all hosts)
     - Consistency is critical because this information is stored in multiple locations and must match across the distributed configuration
   - In the **Choose Install Set** dialog:
     - Select components according to the section “Configuration” in Use Case Description High Availability
2. **Configure DirX Identity on each relevant host** (see Configuring DirX Identity):
   - In the **Configuration Options** dialog:
     - Select components as described in the “Configuration” section of Use Case Description High Availability

3. **Perform additional configuration steps**:
    - Follow the instructions in “Configuration” in Use Case Description High Availability to implement your High Availability scenario
    - Ensure shared resources (message repository and Java-based Server repositories) are properly configured and accessible across hosts
    - Validate connectivity packages and Message Broker setup for redundancy

6.3.3. Key Recommendations

- Keep all licensed feature selections consistent across hosts
- Use shared storage for message repositories to avoid data loss
- Deploy multiple servers per domain to ensure failover capability

7. Installing the Windows Password Listener

This chapter explains how to install the **DirX Identity Windows Password Listener** on Windows Server 2016 (Long-Term Servicing Channel) or Windows Server 2019 (Desktop Experience).

7.1. Overview

The Windows Password Listener is required only if you intend to capture user passwords for password synchronization. Install it on all machines that host an Active Directory installation.

- No other DirX Identity components are required on the same machine.
- A Java environment is needed only during installation and uninstallation. For uninstall, any Java accessible via the system PATH variable is sufficient.

The installation creates configuration files:

- `libdxmEventListenerAds.ini`
- `options.ini`

These files contain default values. During upgrades, existing values are reused or updated based on user input.

Upgrade installations are supported from versions **8.9** onward.

7.2. Key Points

- For multiple domain controllers, you can perform one installation, create a response file, and reuse it for unattended installations (see Unattended (Silent) Installation/Uninstallation).
- The listener uses **ActiveMQ messaging** from a DirX Identity installation. During setup, you must provide:
  - Host name and port of the messaging service
  - SSL option for secure connections

If SSL is enabled:

- Copy the following files from the DirX Identity `ssl` folder to the Password Listener `ssl` folder:
  - `ca-crt.pem`
  - `client-key.pem`

- `password.properties`

The `password.properties` file cannot be encrypted by the Password Listener. Use the encrypted file from the Identity installation. If passwords change, copy the updated encrypted file again.

For SSL setup details, refer to **Securing Identity Server Connections with SSL** in the Connectivity Administration Guide.

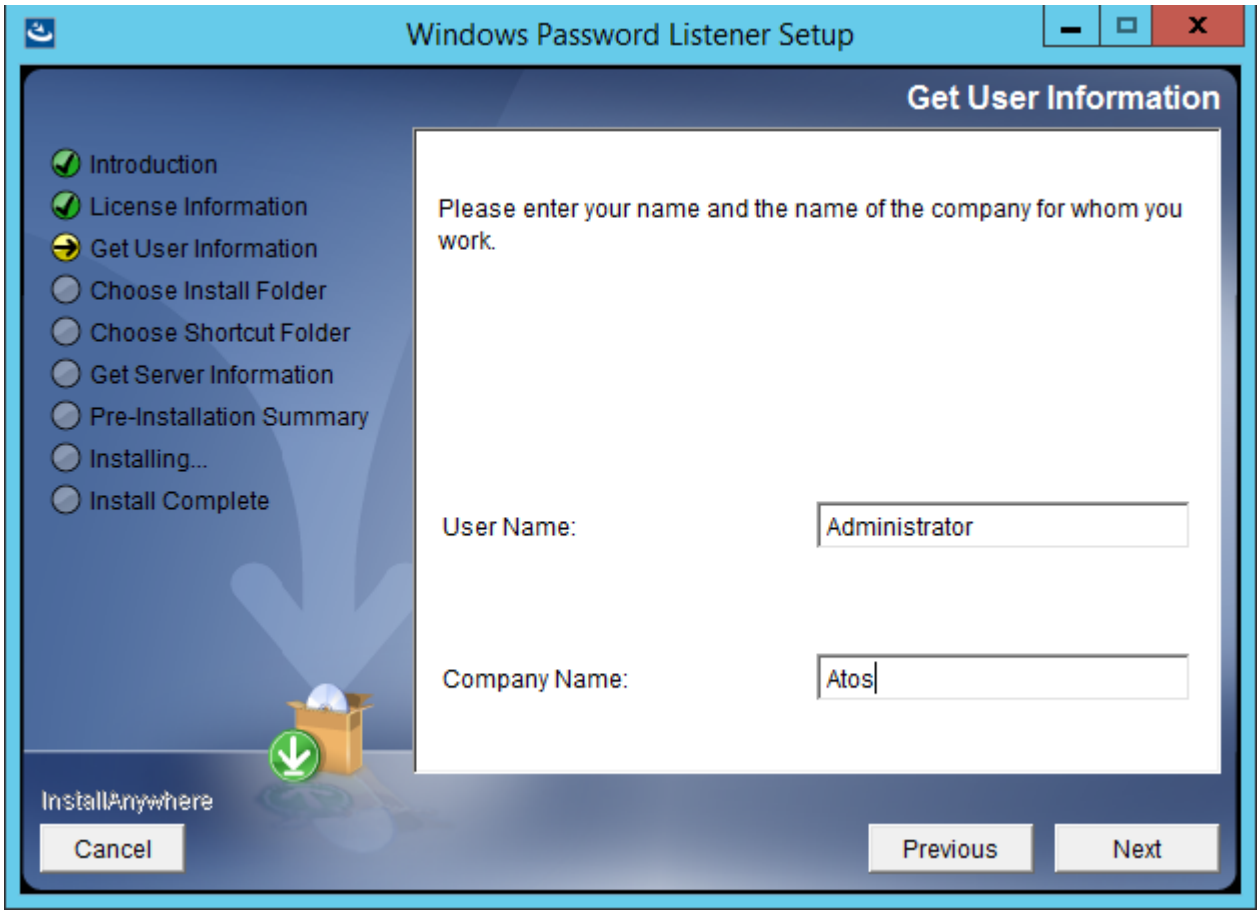
7.3. Installing the Windows Password Listener

- Run **setup.exe** from the DirX Identity DVD.
- In the dialog:
  - Click **DirX Identity**.
  - Click **Install DirX Identity Windows Password Listener**.

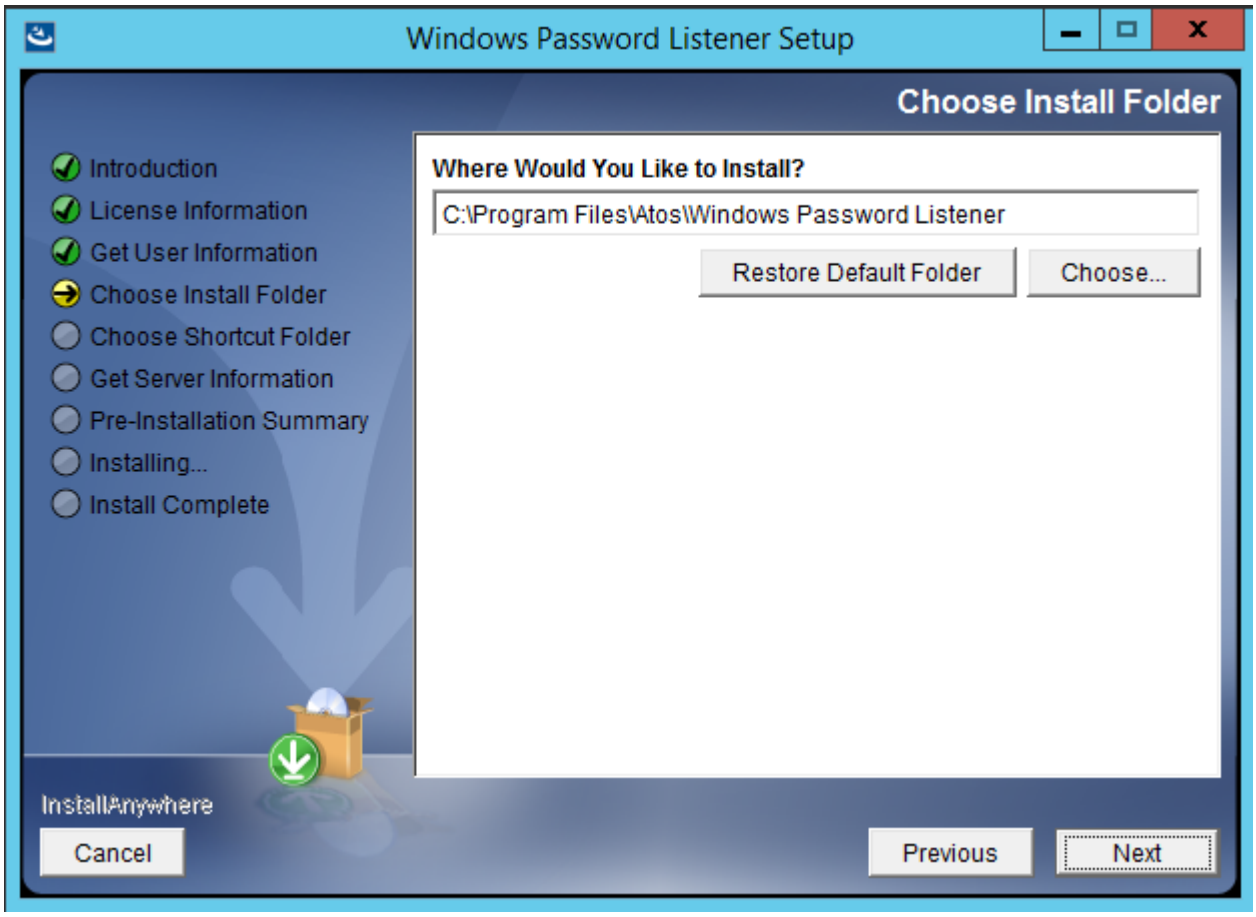
- Click **Next**.

You can click **Cancel** anytime to exit or **Previous** to return to the previous dialog.

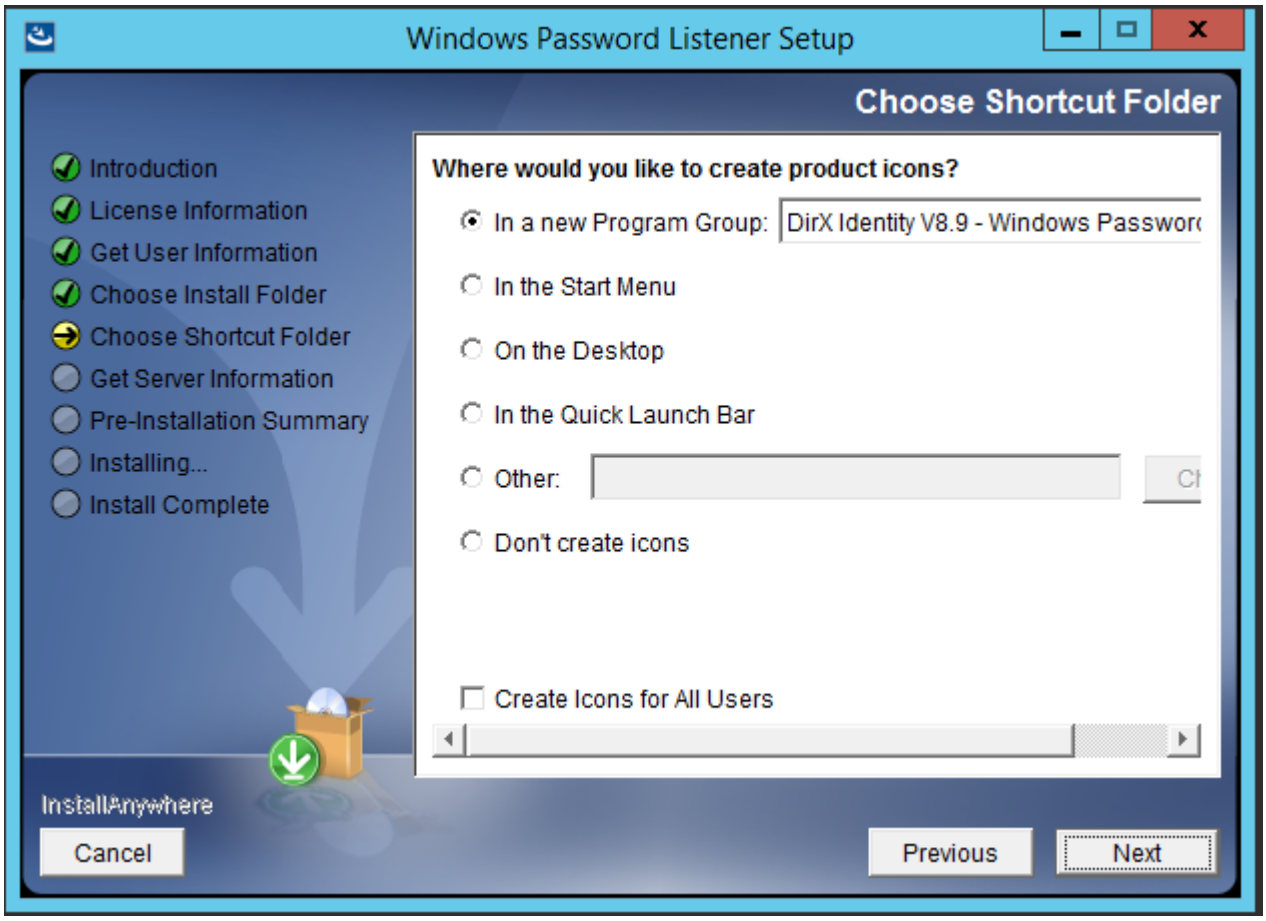
- Setup displays a License Information dialog.
- Read the licensing information, and then click **Yes** and then click **Next**. Clicking **No** cancels the installation.



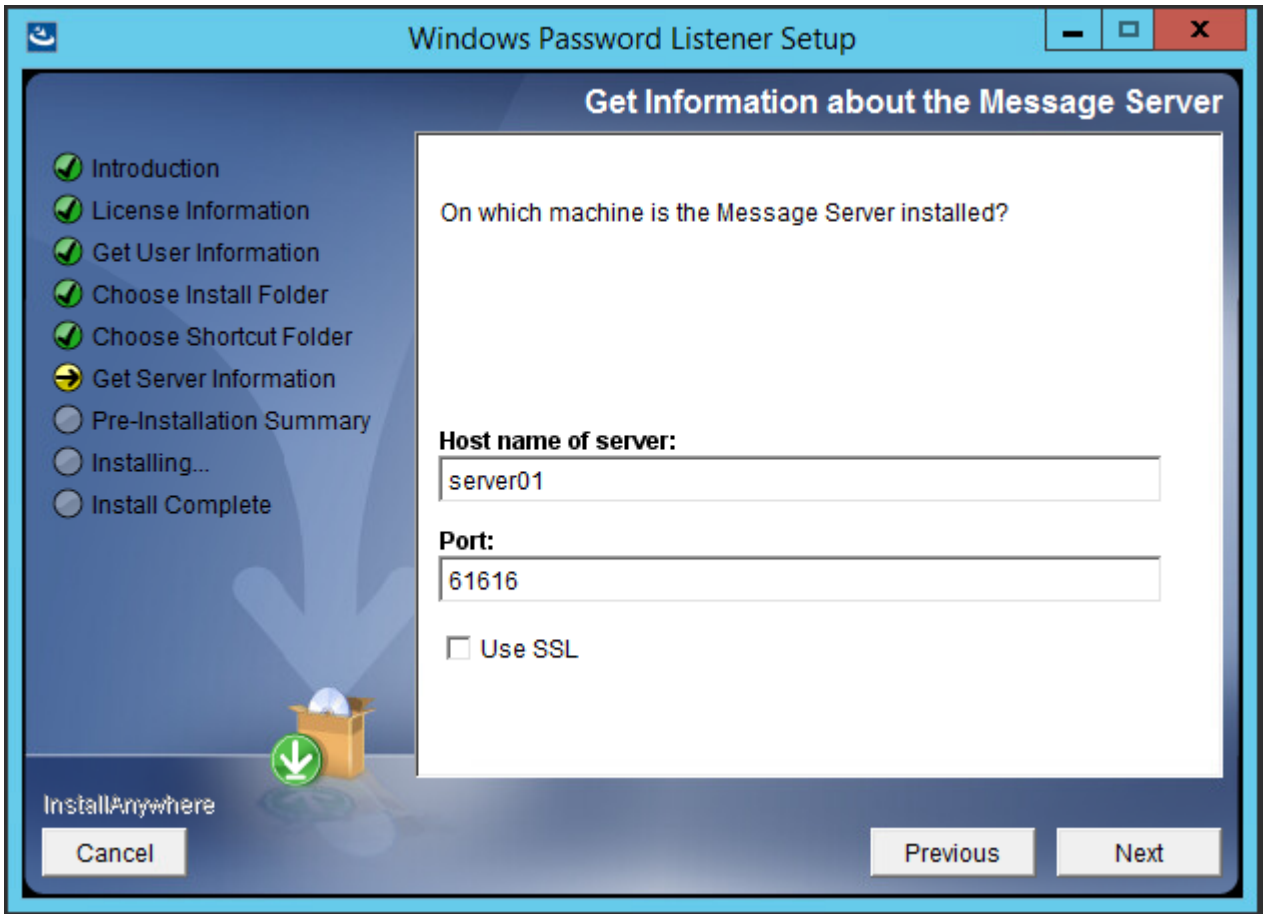
- Setup displays a Customer Information dialog.
- Enter your name and your company name in the fields provided, and then click **Next**.



- Setup asks you to select an installation directory. It displays the default location in the field provided.
- In this dialog, you can:
- Click **Next** to select the default location.
- Click **Choose** to select a different directory, and then click **Next**.

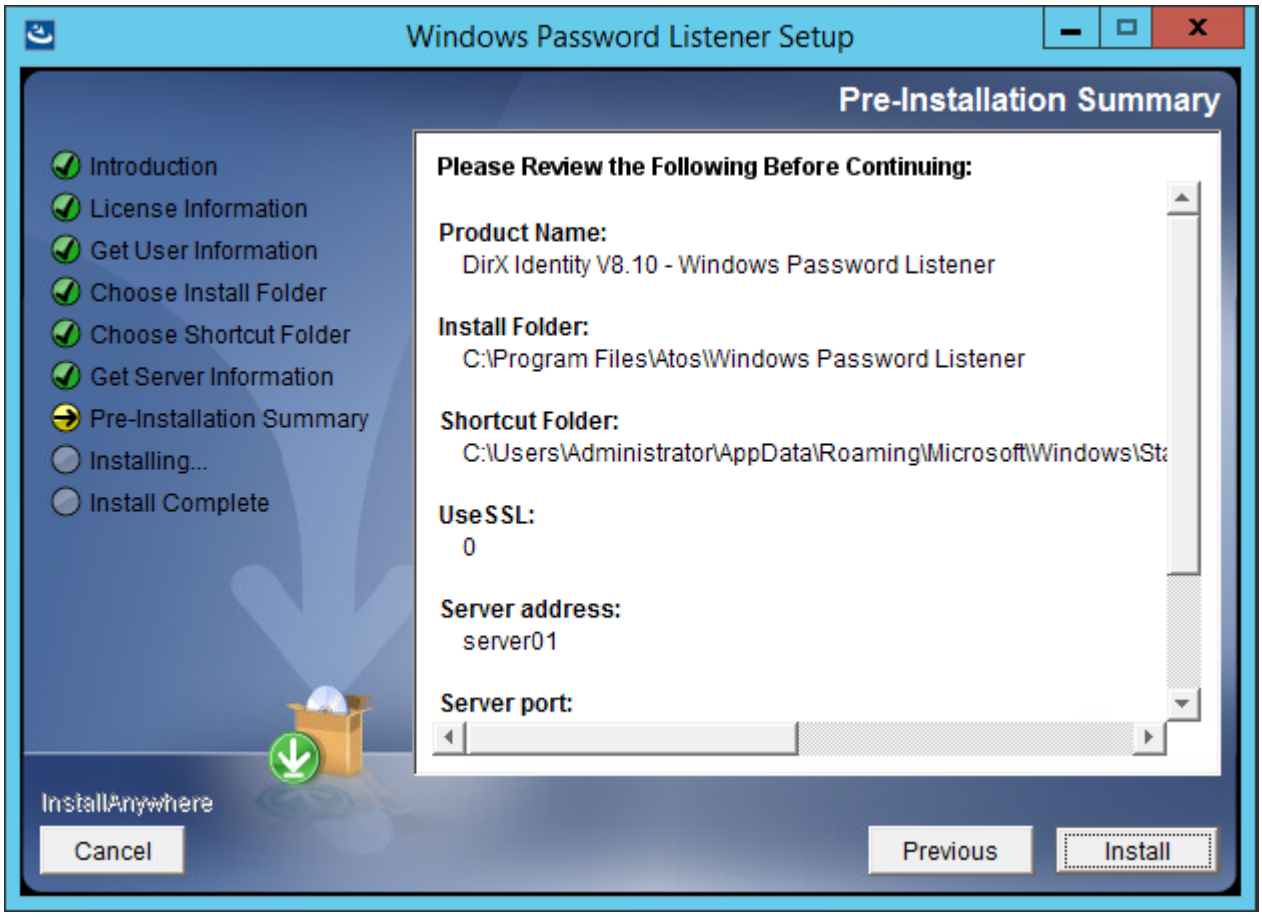


- Setup asks you where you would like to create product icons. It displays the default location in the field provided.
  Note: If you upgrade from an older version, the currently used Program Group name is shown **In a new Program Group:** (in the picture above, from version V8.9). Otherwise, a default name is shown.
- In this dialog, you can:
- Click **Next** to select the default location.
- Select a radio button of another predefined location and then click **Next**.



- Setup asks you for information about the message server.
- Enter the host name of the machine on which the message server is installed and the port number of the message server in the fields provided.
- Check **Use SSL** if you want to use an SSL connection to the server.
- Click **Next**.

Be sure that your DNS (domain name service) works correctly if you use symbolic names for the host name. If you are not sure, use a TCP/IP address instead. If you use SSL, you must give the exact same server name of the messaging server that is used in its server certificate.



- Setup displays the installation selections you have made and asks you to review them.
- Click **Previous** to change any settings you have made. Otherwise, click **Install**.

- Setup displays the installation status.
- When Setup completes the installation, it displays the following dialog.

- Click **Done** to exit Setup.DirX Identity Windows Password Listener installation is now complete.
- Wait for restart of the computer.The restart is necessary to register the Windows Password Listener Plugin DLL correctly into the LSASS service of the Windows domain controller.

7.4. Unattended (Silent) Installation/Uninstallation

You can install the Windows Password Listener without any user interaction by creating a silent setup.

Follow these steps:

- Copy the contents of the folder **DirXIdentity/WinPWListener** from the DVD to a local folder on your machine.
- Edit and customize the file **dirxidty_wpl.properties** to match your environment.
- Run the installation program from the local folder.
- Verify the installation and check for any errors.

Below is an example of the **dirxidty_pwl.properties**:

```
#########################################################################
```

```
#########
# DirX Identity - Windows Password Listener install properties for
# InstallAnywhere
#########################################################################
#########
# Release Information
# Release=8.10
# Version=8.10
# Build=nn
# CreationDate=YYYYMMDD
#########################################################################
#########
# installer created with InstallAnywhere by Flexera
# InstallAnywhere 2021 Build 6526
#########################################################################
#########
# InstallAnywhere install properties
#########################################################################
#########
# UI mode for the installer
# INSTALLER_UI=[SILENT | CONSOLE | GUI | SWING | AWT]

# default for Windows: swing
# default for Unix: console

################################################
# Note for Windows:
# if INSTALLER_UI is set to swing, than installer does not prompt
with a
# dialog "Not enough space...", if necessary
################################################

#INSTALLER_UI=swing
#########################################################################
#########
# own DirX Identity - Windows Password Listener install properties
#########################################################################
#########
#-------------------------------------------------------------------
---------
#Get User Information
```

```
#--------------------
#PROP_DX_USER_INFORMATION_1=<userName>
# default:
#PROP_DX_USER_INFORMATION_1=<login user>

#PROP_DX_USER_INFORMATION_2=<companyName>
# default:
#PROP_DX_USER_INFORMATION_2=<>

#-----------------------------------------------------------------------
---------
#Choose Install Folder
#--------------------
# PROP_USER_INSTALL_DIR=<path>

# default for Windows:
# PROP_USER_INSTALL_DIR=$PROGRAMS_DIR$$/$Atos$/$Windows Password
Listener

# Note for Windows:
# If an install path for Windows Password Listener is found in the
registry,
# then this path will be taken!

# PROP_USER_INSTALL_DIR=$PROGRAMS_DIR$$/$Atos$/$Windows Password
Listener
#-----------------------------------------------------------------------
---------
#Choose Shortcut Folder
#----------------------
# PROP_USER_SHORTCUTS=<program group>

# default:
# PROP_USER_SHORTCUTS=$WIN_COMMON_PROGRAMS_MENU$$/$Atos DirX
Identity$/$Windows Password Listener

# Note for Windows:
# If a program group for DirX Identity is found in the registry,
# than this program group is taken!

#-----------------------------------------------------------------------
---------
```

```
#Get Information about the Message Server
#--------------------------------------
#PROP_DX_LOCAL_HOST=<hostName>

#PROP_DX_PORT=<port>
# default:
#PROP_DX_PORT=61616

#PROP_DX_Q_MGR=<Q-Name>

# use SSL
#PROP_DX_USE_SSL=[0|1]
# default:
#PROP_DX_USE_SSL=0
#-----------------------------------------------------------------------
---------

#Get Service Account Information
#------------------------------
#PROP_DX_ACCOUNT=<accountName>
#PROP_DX_USER_PASSWORD=<password>
#
-----------------------------------------------------------------------
---------
# sleep time
# installation / uninstallation is waiting, when the service will be
removed
#PROP_DX_SLEEP_TIME=<msec>

# default:
#PROP_DX_SLEEP_TIME=10000
#
-----------------------------------------------------------------------
---------

# Restart Windows - (De)-Installation in silent modus
# Note:
# When you want to force a reboot, if necessary,
# you can set the following variable

# PROP_RESTART_NEEDED=YES
#
```

```
-------------------------------------------------------------------------------
---------
```

8. Installing the JMS-Audit Handler

The JMS-Audit Handler is not installed or updated automatically during standard setup. This section explains how to manually deploy and configure it.

Prerequisite: A working DirX Audit installation.

8.1. Deploying the JMS-Audit Handler

The JMS-Audit Handler is included with the DirX Identity product. Locate it in:

product_media_DirX Identity`/Additions/jmsAuditHandler`

Steps:

1. Open the folder `jmsAuditHandler`. It contains:
   - `com.siemens.idm.audit.jms.zip` – includes `server.xml` and required JAR files.
   - `lib` subfolder – contains `com.siemens.idm.audit.jms.JmsAuditLogHandler.jar` (the plug-in).
2. Ensure you use the handler version matching your DirX Identity installation.
3. If upgrading, remove old JAR files from `lib` to avoid conflicts.
4. Deploy the handler to **each DirX Identity Java-based server**:

   dxi_install_path`/ids-j-`*domain*`-S`*n*`/extensions/com.siemens.idm.audit.jms`
   Unzip `com.siemens.idm.audit.jms.zip` into this folder.

8.2. Configuring the JMS-Audit Handler

Configure the handler using **DirX Identity Manager**:

1. Navigate to:
   - **Connectivity** → **Expert** view
   - Select the LDAP entry for the DirX Identity Java-based server (IdS-J)
   - Open the **Status and Auditing** tab
2. Settings:
   - **Enable JMS-based Auditing** – Activates JMS auditing and disables file-based auditing.
   - **Message Broker URL** – Must match the DirX Audit Configuration Wizard:
     - Non-SSL: `tcp://host:30666`
     - SSL: `ssl://host:30667`
   - **JMS Queue** – Matches the queue configured in DirX Audit Wizard:

- Example: `dxt.<tenantID>.dxi`

- **User and Password** – Use a broker user with write access:
  - Default writer: `dxt-<tenantID>-writer`
- **Audit Trail Folder** – Temporary storage if JMS is unavailable:
  - Default: ${IDM_HOME}
  - Relative paths are based on `<dxi_install_path>/ids-j-<domain>-S<n>/bin`

If the JMS Audit handler cannot send audit records to the message server, it stores them temporarily in the Audit Trail Folder (one message per file). Once the connection is restored, the handler sends and deletes these files automatically.

8.2.1. SSL Configuration

To enable SSL, import the broker's CA certificate into the Java VM truststore:

Default truststore location: `<JRE_folder>/lib/security/cacerts`

Command (Windows example):

```
%JAVA_HOME%bin\keytool -importcert -trustcacerts -keystore cacerts
-storepass* cacerts-pwd -alias ca-alias *-file ca.crt
```

8.3. Fallback Behavior

If the handler cannot find its configuration in IdS-J, it falls back to **`server.xml`** in the extension folder.

- Default LDAP configuration activates file-based auditing (multiple records per file).
- Manage these files to prevent disk space issues.

8.4. Additional Recommendation

You can use the same ActiveMQ message broker for DirX Identity and DirX Audit to simplify installation and reduce the number of components.

9. The Java for DirX Identity

Many essential DirX Identity components are implemented in Java. Therefore, at least one Java installation is required to use the product.

The Java for DirX Identity is Java 21, which is used by core components such as:

- Java-based servers
- DirX Identity Manager
- Configuration utility

Some components require deployment into an external Tomcat running with Java 21. The Java instance used by Tomcat depends on its configuration.

Unlike previous releases, this version requires an external Java installation and does not provide an embedded Java environment.

9.1. Advantages of a Customer-Supplied Java Installation

- External Tomcat for DirX Identity applications can use the same Java installation.
- The installation can also be used for internet browsers or browser-based applications.
- Updating Java is straightforward using official update packages. See Security Updates for Java in DirX Identity for details.

9.2. Requirements for Java in DirX Identity

A customer-supplied Java must meet the following requirements:

- Must implement Java Platform, Standard Edition (Java SE).
- Version must be 21.x.y.
- Must be a 64-bit distribution.
- Must be TCK-tested (Technology Compatibility Kit for Java).

Supported Java products include:

- Oracle Java SE 21 (LTS)
- Microsoft Build of OpenJDK 21 (LTS)
- Adoptium Eclipse Temurin JDK-21

9.3. Security Updates for Java in DirX Identity

9.3.1. Updating a Customer-Supplied Java Runtime Environment

Update the Java environment using an official downloadable update for the appropriate version.

General procedure:

1. Stop all DirX Identity services and close all programs.
2. If `<DXI_JAVA_HOME>/lib/security/cacerts` contains custom certificates, back up this file outside `<DXI_JAVA_HOME>`.
3. Download and install the Java update. Options for installation path:
   a. Use the current `<DXI_JAVA_HOME>` path for consistency. **Not recommended** if the path already includes `java-21`.
   b. Use the default path (e.g., `C:\Program Files\Java\java-21`). This requires updating `<DXI_JAVA_HOME>` in DirX Identity.
4. Restore custom certificates to the updated `cacerts` file.
5. If the installation path changed, follow Managing a Relocated Customer-Supplied Java.
6. Verify the update by running:
   a. `.\java -version` (Windows)
   b. `./java -version` (UNIX)
7. Restart services.

9.3.2. Managing a Relocated Customer-Supplied Java

If the customer-supplied Java installation has been moved due to an update, perform the following steps:

1. **Update configuration files to reflect the new Java location:**
   - **Windows only:** Update *<DXI_INSTALL_PATH>/setdxienv.bat* to set `DXI_JAVA_HOME`. Use Windows path notation. Example: `SET DXI_JAVA_HOME=C:\Program Files\Java\java-21`
   - **UNIX only:** Update *<DXI_INSTALL_PATH>/.dirxmetarc* to set `DXI_JAVA_HOME`. Use UNIX path notation. Example: `DXI_JAVA_HOME=/opt/java-21`
   - **All platforms:** Update *<DXI_INSTALL_PATH>/configuration.ini* for `dxi.java.home`:
     - Windows example (with escaped characters): `dxi.java.home=C\:\\Program Files\\Java\\java-21`
     - UNIX example: `dxi.java.home=/opt/java-21`
2. **Reconfigure components:** Run the Configuration Wizard to update the Message Broker and Java-based servers.
3. **Update Tomcat configuration (if applicable):** If Tomcat was configured to use Java for DirX Identity, adjust its settings to point to the relocated Java installation

10. Additional Topics

This chapter provides additional information useful for understanding and administering the DirX Identity system.

If some steps fail during configuration:

- Check the log file to identify which parts did not run.
- Correct the error and restart the Configuration Wizard.
- You may select only those steps that did not finish successfully.

10.1. Disk Space Calculation

To calculate the required disk space for a DirX Identity installation, consider the following:

1. Space for DirX Identity (see Disk Space Requirements).
2. Additional space for auditing.
3. Space for work and status areas where DirX Identity stores temporary and permanent files.

The next sections discuss disk space requirements for auditing and for work/status areas.

10.1.1. Space for Auditing

The required space for auditing depends on:

- Regularly remove auditing information from this area. Either import it into a database or store it on backup devices.
- Audit only the absolutely necessary objects in your Identity Store.
- Configure only the necessary attributes for these objects.
- Check whether signed audit records are really needed. If the audit area is secure, this may not be necessary. Signed records require about double the space and slow down real-time workflows considerably.

Recommendation: Reserve enough space for auditing and write audit information to a separate disk to avoid impacting server operations.

10.1.2. Space for Work and Status Areas

This can only be estimated and depends on:

- Files configured to be stored in the status area (by default, all files are stored).
- Amount of data (number and size of entries) to be synchronized.
- Frequency of scheduled workflows, configured status expiration time, and status compression mode.

Because status information can easily fill your disk, note:

- DirX Identity ignores a full disk in the status area but cannot handle a full disk in the work area.

Guidelines:

- Keep work and status areas on different disks.
- Mark only important files for automatic storage in the status area (e.g., trace and report files, not large data files).
- Set individual status expiration times for each workflow:
  - Example:**
    - 1 month for weekly workflows
    - 1 week for daily workflows
    - 1 day for workflows running every 10 minutes
- Set individual status expiration times for all Java-based Identity Servers.

These steps help make your system more reliable and resource-efficient.

10.2. Schema and Content Handling

During configuration, DirX Identity prepares the LDAP directory according to its requirements:

- Extends the LDAP directory schema for the Connectivity Configuration tree.
- Imports basic content into that tree, including DirX Identity Default Applications.

To run workflows, specific object classes and attributes for each agent type are required. Extend the schema for joined data in the Identity Store with agent-specific schema parts.

10.2.1. Setup of the Schema for Connectivity Configuration

DirX Identity extends the schema of the defined LDAP directory with object and attribute definitions needed for correct operation.

10.2.2. Basic Content Extension

DirX Identity writes all preconfigured objects (workflow, activity, job, connected directory definitions, etc.) from LDIF files to the LDAP directory. Based on this, administrators can configure their own objects and synchronizations.

10.2.3. Target System Specific Schema Extensions

Setting up the Identity Store schema for target systems should be carefully planned. Only set up required object classes and attributes for high performance and easy handling.

- DirX Identity automatically extends the schema if you select the Sample Domain.

- A minimal set of attributes and object classes is defined for all target systems requiring LDAP schema extensions.
- Not all target systems require schema changes.
- Using additional attributes in the Sample Domain requires manual schema extension.

For customer domain schema extension, DirX Identity provides complete sets of attribute and object class extensions for each supported target system type.

Backup your directory before running any scripts! Schema extensions cannot be reversed.

Steps:

1. Open the directory *DXI_INSTALL_PATH*`/schema/tools`.
2. Open the subdirectory for your directory type: `dirx-ee` for DirX Directory installation.
3. Copy the entire `Customer Domain` subdirectory and name it `Customer Domain.orig`.
4. Update schema definitions in the `Customer Domain` subdirectory as required. Additional steps for directory type `dirx-ee` (DirX V8.3 or higher):
   - Select the LDIF file of your DirX Identity Connectivity package in the `ldif` subdirectory (e.g., `dirx.nt.ldif`).
   - Remove unwanted attributes by deleting "MODIFY" records referring to `attributeTypes`.
   - Remove attributes from object class definitions by deleting LDAP attribute names from "MODIFY" records referring to `objectClasses`.
   - If indexes were defined for attributes, remove them from `dbconfig_opt` statements in the `dirxadm` script of your Connectivity package (e.g., `DirXmetahub-schema.Nt.adm` for NT).
5. Run the script `agent-schema.bat` (Windows) or `agent-schema.sh` (Linux) under `schema/tools`.
   - Enter the DirX Identity administrator password (`admin`).
   - Select the DirX Identity Connectivity package to install this part of the schema extension (each package must be selected separately* Select the DirX Identity Connectivity package to install this part of the schema extension (each package must be selected separately).
   - Choose whether to create attribute indexes.
6. Check the trace.txt file at the end for errors (the exit codes at the end should be 0).

10.2.4. Indexed Attributes

DirX Identity requires a set of indexes. The minimum number of indexes is 84, the maximum number of indexes is 137 (all target system schema extensions performed).

This information is especially important to set up DirX correctly.

10.2.4.1. DirX Identity Connectivity Configuration (17 attribute indexes)

dxmActive
dxmActivityStatusData-DN
dxmC
dxmDisplayName
dxmEndTime
dxmExitCode
dxmExpirationTime
dxmName
dxmOkStatus
dxmOrigWorkflow-DN
dxmResult
dxmScheduleName
dxmStartTime
dxmStatusExpirationTime
dxmType
dxmWarningStatus
dxmWorkflowInstID

10.2.4.2. DirX Identity Provisioning Configuration Extensions (67 attribute indexes)

dxmOprEventDivision
dxmOprMaster
dxmOprOriginator
dxmOprTriggerOrigin
dxmPwdLastChange
dxrAccessRightLink
dxrApproverLink
dxrApproverPotentialLink
dxrAssignedAccounts
dxrAssignedGroups
dxrAssignFrom
dxrAssignTo
dxrAssignmentLink
dxrCurrentParticipants
dxrDeleteDate
dxrDisableStartDate
dxrDisableEndDate
dxrEndDate
dxrErrorExpDate
dxrError
dxrExpirationDate
dxrGroupLink
dxrGroupMemberAdd
dxrGroupMemberDelete
dxrGroupMemberIgnore
dxrGroupMemberImported

dxrGroupMemberOK
dxrInheritedPrivilegeLink
dxrInheritedUserFacetPrivilegeLink
dxrIsActive
dxrIsExtensionGroup
dxrIsInconsistent
dxrName
dxrNeedsApproval
dxrNextApprovalDate
dxrObjectComplete
dxrObjectType
dxrOperationImp
dxrPeerTS
dxrPermissionLink
dxrPrimaryKey
dxrPrivilegeLink
dxrPrivilegesGrantedLink
dxrPwdChangedTime
dxrPwdChangeState
dxrReference
dxrResourceGroupLink
dxrResourceLink
dxrRoleID
dxrRoleLink
dxrRPvalues
dxrStartDate
dxrState
dxrSubjectLink
dxrSubjectGroupLink
dxrTBA
dxrToDo
dxrToPeer
dxrTSState
dxrTSStateExtended
dxrType
dxrUID
dxrUsedBy
dxrUserAssignementPossible
dxrUserLink
employeeNumber
uniqueMember

10.2.4.3. DirX Identity Connectivity Package Schema Extensions (18 attribute indexes)

If you install the sample domain, you need 18 additional indexes.

A) ADS: (7)

dxmADsComputerName
dxmADsDNSdomainName

dxmADsDomain
dxmADsForest
dxmADsGuid
dxmADsSamAccountName

B) Exchange 5.5: (2)

dxmEXcn
dxmEXrfc822Mailbox

C) Notes: (4)

dxmLNfullName
dxmLNlistName
dxmLNnoteID
dxmLNshortName

D) ODBC: (4)

dxmODBCdatabaseName
dxmODBCdatabaseType
dxmODBCfirstName
dxmODBClastName

E) SAP/R3-UM: (1)

sapUsername

10.2.4.4. DirX Identity Agent Schema Extensions for a Customer Domain (52 attribute indexes max)

For each Connectivity Package schema extension, you need the corresponding number of indexes. This list shows the maximum number delivered with each default set. If you extended the schema with fewer attributes, the number of indexes is lower.

A) ADS: (6)

dxmADsComputerName
dxmADsDNSdomainName
dxmADsDomain
dxmADsForest
dxmADsGuid
dxmADsSamAccountName

B) Exchange: (8)

dxmEXcn
dxmEXdescription
dxmEXemployeeNumber
dxmEXgivenName
dxmEXname
dxmEXrdn

dxmEXrfc822Mailbox
dxmEXsn

C) HDMS: (11)

dxmHDbuilding
dxmHDchristianName
dxmHDcompany
dxmHDcountry
dxmHDdmsid
dxmHDlocation
dxmHDname
dxmHDorg1
dxmHDorg2
dxmHDorg3
dxmHDsortName

D) Notes: (10)

dxmLNcomment
dxmLNemployeeID
dxmLNfirstName
dxmLNfullName
dxmLNinternetAddress
dxmLNlastName
dxmLNlistDescription
dxmLNlistName
dxmLNnoteID
dxmLNshortName

E) ODBC: (4)

dxmODBCdatabaseName
dxmODBCdatabaseType
dxmODBCfirstName
dxmODBClastName

F) SAPR3/HR: (4)

dxmSAPR3HRcommonName
dxmSAPR3HRgivenName
dxmSAPR3HrpersonnelNumber
dxmSAPR3HRsurName

G) SAP/R3-UM: (1)

sapUsername

Appendix A: DirX Identity Web Center - Kerberos Authentication

This document describes how to set up Kerberos authentication for Web Center via the SPNEGO mechanism.

A.1. Introduction

Some terms used in this document:

- **Host** - The simple name of the host with the Tomcat server Web Center is deployed into.For example, “alpha“.
- **Domain** - The fully qualified domain name of the host; for example, “beta.com“.
- **Service Principal Name** - The service principal name is http/host.domain@DOMAIN, for example “http/alpha.beta.com@BETA.COM“.
- ***tomcat_install_path*** - The installation folder of the Tomcat server, for example “C:/Program Files/Apache/Tomcat 1116”.Note that “*tomcat_install_path*“ is just a notation used in this document.Always replace it with the real path name of the Tomcat installation folder.

Restrictions:

- Kerberos authentication does not work if browser and Tomcat run on the same machine.

A.2. Windows Prerequisites

For Kerberos on Windows, you must create a Windows user account, register the service principal name for that account, and create a keytab for the service principal name.This section describes how to perform these steps on a Windows Server 2012 R2. The details might be different on other server versions like Windows Server 2016 and 2019.

A.2.1. Creating an Active Directory User Account

Create a user account for Kerberos in Active Directory via program “Active Directory Users and Computers”.Assign a “User logon name” and a password.Check the flag “Password never expires”.

Kerberos on Windows works with one of three different encryption modes:

- **RC4-HMAC-NT** – The default 128-bit encryption.
- **AES256-SHA1** – AES256-CTS-HMAC-SHA1-96 encryption. This is the recommended encryption mode.
- **AES128-SHA1** – AES128-CTS-HMAC-SHA1-96 encryption.

When using AES encryption for Kerberos (as recommended), open the created user

account and select tab “Account”. In the options list, check the flag “This account supports Kerberos AES 128-bit encryption” or “This account supports Kerberos AES 256-bit encryption” as appropriate.

In the following examples, we assume the logon name is “beta\kerberosUser”.

A.2.2. Registering the Service Principal Name

Use the windows tool **setspn** to register the service principal name for the created Kerberos account:

```
setspn -u -s <servicePrincipalName> <accountName>
```

For example

```
setspn -u -s http/alpha.beta.com@BETA.COM beta\kerberosUser
```

Note that the service principal name must not be registered for more than one account. You can check this with

```
setspn -q <servicePrincipalName>
```

For example

```
setspn -q http/alpha.beta.com@BETA.COM
```

You can unregister a service principal name with

```
setspn -d <servicePrincipalName> <accountName>
```

After having registered the service principal name, open the Kerberos user account again and select the new tab “Delegation”. Select the option “Trust this user for delegation to any service (Kerberos only)”.

A.2.3. Creating the Keytab File

Use the Windows utility **ktpass** to create a keytab for the service principal name. The keytab is used on the Tomcat server to perform Kerberos authentications.

```
ktpass /princ <servicePrincipalName>
       /ptype KRB5_NT_PRINCIPAL
       /mapuser <accountName>
       /out <keytabFileName>
       /crypto <AES256-SHA1 or AES128-SHA1 or RC4-HMAC-NT>
       /pass *
       /kvno 0
```

For example:

```
ktpass /princ http/alpha.beta.com@BETA.COM
       /ptype KRB5_NT_PRINCIPAL
       /mapuser beta\kerberosUser
       /out alpha.keytab
       /crypto AES256-SHA1
       /pass *
       /kvno 0
```

You are prompted for the password of the Kerberos user twice.

Note that you must create a new keytab file each time you change the Kerberos account in Active Directory.

Now open the Kerberos user account again and select tab “Account”.The “User logon name” should have changed to the service principal name.

A.3. Tomcat Configuration

In this section, we create some configuration files on the Tomcat server.We suggest putting the files into the folder *tomcat_install_path***/conf** but you can choose any other folder as well.Make sure that all the created files are readable by the Tomcat service.

A.3.1. Keytab

Copy the keytab file to *tomcat_install_path***/conf**.In the subsequent examples, we assume the keytab file is copied to *tomcat_install_path***/conf/alpha.keytab**.

A.3.2. File krb5.conf

Create file *tomcat_install_path*/conf/krb5.conf* with the following content:

```
#
# Kerberos configuration file for running JGSS applications.
# Adapt the entries to your environment.
#

[libdefaults]
    default_realm = <DOMAIN>
    permitted_enctypes = aes256-cts aes128-cts
    allow_weak_crypto = false
    kdc_timesync = 0
    kdc_default_options = 0x40000010
    clockskew = 300
```

```
    check_delegate = 0
    ccache_type = 3
    kdc_timeout = 60000

[realms]
    <DOMAIN> = {
        kdc = <keyDistributionCenter>:<keyDistributionCenterPort>
    }

[domain_realm]
    .<domain> = <DOMAIN>
```

Note that some lines have been splitted for better readability.

Replace each occurrence of

- <DOMAIN> with the fully qualified domain name in uppercase letters.
- <domain> with the fully qualified domain name in lowercase letters.
- <keyDistributionCenter> with the IP address or fully qualified host name of the Kerberos key distribution center, which is part of the Windows domain controller.
- <keyDistributionCenterPort> with the port number of the Kerberos key distribution center, which is usually 88.

Adjust the encryption types if necessary. They must include the encryption type of the created Windows user account. For example, add “rc4-hmac” if necessary.

The **krb5.conf** file for our sample data is:

```
#
# Kerberos configuration file for running JGSS applications.
# Adapt the entries to your environment.
#

[libdefaults]
    default_realm = BETA.COM
    permitted_enctypes = aes256-cts aes128-cts
    allow_weak_crypto = false
    kdc_timesync = 0
    kdc_default_options = 0x40000010
    clockskew = 300
    check_delegate = 0
    ccache_type = 3
    kdc_timeout = 60000
```

```
[realms]
    BETA.COM = {
        kdc = dc.beta.com:88
    }

[domain_realm]
    .beta.com = BETA.COM
```

A.3.3. File krb5-jaas.conf

Create file *tomcat_install_path***/conf/krb5-jaas.conf** with the following content:

```
/**
 * JAAS configuration file for JGSS.
 * Modify the entries to suit your environment.
 */

/* Needed for SPNEGO/Kerberos */
com.sun.security.jgss.krb5.accept {
    com.sun.security.auth.module.Krb5LoginModule required
        debug=true
        doNotPrompt=true
        useKeyTab=true
        keyTab="<keytabPathName>"
        storeKey=true
        principal="<servicePrincipalName>"
    ;
};
```

Replace

- <servicePrincipalName> with the service principal name.
- <keytabPathName> with the full path name of the keytab file.

The **krb5-jaas.conf** file for our sample data is:

```
/**
 * JAAS configuration file for JGSS.
 * Modify the entries to suit your environment.
 */
```

```
/* Needed for SPNEGO/Kerberos */
com.sun.security.jgss.krb5.accept {
    com.sun.security.auth.module.Krb5LoginModule required
        debug=true
        doNotPrompt=true
        useKeyTab=true
        keyTab="tomcat_install_path/conf/alpha.keytab"
        storeKey=true
        principal="http/alpha.beta.com@BETA.COM"
    ;
};
```

A.3.4. Java System Properties

Assign the path names of the Kerberos configuration files to Java system properties of the Tomcat service:

```
-Djava.security.krb5.conf=<krb5PathName>
-Djava.security.auth.login.config=<krb5jaasPathName>
```

Replace

- <krb5PathName> with the full path name of file **krb5.conf**.
- <krb5jaasPathName> with the full path name of file **krb5-jaas.conf**.

The values for our sample data are

```
-Djava.security.krb5.conf=tomcat_install_path/conf/krb5.conf
-Djava.security.auth.login.config
                         =tomcat_install_path/conf/krb5-jaas.conf
```

You can also enable some output for debugging purposes which will be written to some Tomcat log files:

```
-Dsun.security.krb5.debug=true
-Dsun.security.jgss.debug=true
```

To set the Java system properties for a Tomcat running as a Windows service, open the service's "Configure Tomcat" item in the Windows start menu; another way to start the Configuration Wizard is to run the program **tomcat9w.exe** in Tomcat's **bin** folder manually. In the Configuration Wizard, select the Java tab and then add the properties to the list of

Java Options.

If running Tomcat from a batch script, assign the system properties to the environment variable “CATALINA_OPTS” before starting Tomcat, for example:

```
set KRB5_FILE=-Djava.security.krb5.conf=
               tomcat_install_path/conf/krb5.conf
set KRB5_JAAS_FILE=-Djava.security.auth.login.config =
               tomcat_install_path/conf/krb5-jaas.conf
SET KRB5_DEBUG=-Dsun.security.krb5.debug=true
SET JGSS_DEBUG=-Dsun.security.jgss.debug=true
SET CATALINA_OPTS=%KRB5_FILE% %KRB5_JAAS_FILE% %KRB5_DEBUG%
%JGSS_DEBUG%
```

Note that some lines have been splitted for better readability.

A.3.5. Copying the SPNEGO Jar File

Copy the SPNEGO jar file **spnego-tomcat11.jar** delivered with DirX Identity to Tomcat’s lib folder *tomcat_install_path***/lib**.

A.3.6. Increasing the Maximum HTTP Header Size

Tomcat limits the maximum size of HTTP headers by default to 4kb and discards any request with larger headers. Kerberos tickets are transferred from the browser to the server in an HTTP header. For Windows users with many groups the Kerberos ticket may become quite large so that the header size easily exceeds the default maximum size.

Therefore, you should increase the maximum HTTP header size permitted by Tomcat to 32kb or 64kb by adding the property maxHttpHeaderSize to your HTTP and/or HTTPS connector configuration in the file *tomcat_install_path***/conf/server.xml**.

```
<Connector port="8080" ...
    maxHttpHeaderSize="65536" .../>
<Connector port="8443" ...
    maxHttpHeaderSize="65536" .../>
```

A.4. Web Center Configuration

A.4.1. Activating SPNEGO Authentication

A.4.1.1. File web.xml

Uncomment the security settings in the deployment descriptor **WEB-INF/web.xml**:

```
<!-- SPNEGO Windows single sign-on security settings -->
<security-constraint>
    <web-resource-collection>
        <web-resource-name>Restricted Area</web-resource-name>
        <url-pattern>*.do</url-pattern>
        <url-pattern>*.jsp</url-pattern>
        <url-pattern>/saveFile</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>SpnegoAuthenticatedUser</role-name>
    </auth-constraint>
</security-constraint>

<security-role>
    <description>SPNEGO authenticated users</description>
    <role-name>SpnegoAuthenticatedUser</role-name>
</security-role>
```

A.4.1.2. File webCenter-*dxiDomain*.xml

Add the Kerberos authentication elements “Realm” and “Valve” to Web Center’s context descriptor file **webCenter-***dxiDomain***.xml**.

The file is located in the folder *tomcat_install_path**/conf/***engineName/hostName*, where *engineName* and *hostName* depend on your Tomcat installation. In most cases, the location is *tomcat_install_path***/conf/Catalina/localhost*.

```
<Context ...>
  ...
  <!-- Kerberos authentications -->
  <Realm
      className="com.siemens.symphonia.spnego.SPNEGOVoidRealm"/>

  <Valve
     className="com.siemens.symphonia.spnego.SPNEGOAuthenticator"
     servicePrincipalName="<servicePrincipalName>"
     changeSessionIdOnAuthentication="false"/>
</Context>
```

Replace

- <servicePrincipalName> with the service principal name.

The context descriptor file for our sample data is:

```
<Context ...>
  ...
  <!-- Kerberos authentications -->
  <Realm
      className="com.siemens.symphonia.spnego.SPNEGOVoidRealm"/>

  <Valve
     className="com.siemens.symphonia.spnego.SPNEGOAuthenticator"
     servicePrincipalName="http/alpha.beta.com@BETA.COM"
     changeSessionIdOnAuthentication="false"/>
</Context>
```

A.4.2. Configuring User Mapping

A Kerberos authentication provides the authenticated user's Windows account name and the fully qualified Windows domain name. The task is to map these data to a user in the DirX Identity database. Usually this is done by searching for a target system matching the domain name, and then by looking for an account matching the Windows account name in that target system. If the Windows credentials are stored in a user attribute, an alternative way is to search the user directly in the user tree.

The user mapping configuration is done in the deployment descriptor **WEB-INF/web.xml**.

Note that since **web.xml** is an XML file, ampersands in configuration values must be replaced with the corresponding XML character entity reference "&".

Some configuration parameters accept a regular expression as value. The syntax for regular expressions is described in the Java API documentation of Java class "java.util.regex.Pattern". The part of a pattern marked in bold letters shows the captured substring.

A.4.2.1. Activating the SSO Header Filter

Enable the filter by uncommenting it:

```
<!-- SSO header filter definition -->
<filter>
  <filter-name>SSOHeaderFilter</filter-name>
  <display-name>SSO Header Filter</display-name>
  <description>
    Evaluates single sign-on information passed in HTTP headers
  </description>
```

```
  <filter-class>
    com.siemens.webMgr.filter.SSOHeaderFilter
  </filter-class>
  ...
</filter>
```

Set the filter's authentication type to "SPNEGO/Kerberos":

```
<init-param>
    <param-name>authType</param-name>
    <param-value>SPNEGO/Kerberos</param-value>
</init-param>
```

Set the filter's header name to "RemoteUser":

```
<init-param>
    <param-name>headerName</param-name>
    <param-value>RemoteUser</param-value>
</init-param>
```

Once you have Kerberos authentication with Web Center running, you can switch the log level back to "error".

Other initialization parameters are described in the following sections.

Activate the mappings for the SSO header filter by uncommenting them:

```
<!-- SSOHeaderFilter mappings -->
<filter-mapping>
    <filter-name>SSOHeaderFilter</filter-name>
    <url-pattern>*.do</url-pattern>
</filter-mapping>

<filter-mapping>
    <filter-name>SSOHeaderFilter</filter-name>
    <url-pattern>*.jsp</url-pattern>
</filter-mapping>

<filter-mapping>
    <filter-name>SSOHeaderFilter</filter-name>
    <url-pattern>/saveFile</url-pattern>
```

```
</filter-mapping>
```

While the filter definition sequence is arbitrary, the filter mapping sequence is not. Insert the mappings of the SSO header filter directly before the mappings of the Session filter.

A.4.2.2. User Mapping via Target System and Account

A.4.2.2.1. Extracting Account and Domain Name

The Windows credentials are passed in the format *accountName*@*fullDomainName*. The first task is to extract the Windows account name and domain name from the credentials string.

Set the type of the SSO header filter to “account”:

```
<init-param>
    <param-name>type</param-name>
    <param-value>account</param-value>
</init-param>
```

By default, the extracted account name is the substring of the Windows credentials preceding the first @ character

```
<init-param>
    <param-name>accountRegExpr</param-name>
    <param-value>([^@]*)@.*</param-value>
</init-param>
```

By default, the extracted domain name is the substring between the first @ character and the next dot, which gives the simple domain name:

```
<init-param>
    <param-name>domainRegExpr</param-name>
    <param-value>[^@]+@([^.]+)(?:[.].*)?</param-value>
</init-param>
```

To extract the fully qualified domain name, use

```
<init-param>
    <param-name>domainRegExpr</param-name>
    <param-value>[^@]+@(.*)</param-value>
```

```
</init-param>
```

For example, if the provided Windows credentials are “smith@beta.com”, the extracted account name is “smith”, while the domain name is “beta” for the first domain pattern, and “beta.com” for the second domain pattern.

A.4.2.2.2. Finding the DirX Identity Target System

The target system is by default the one whose attribute “dxrTSDomainName” matches the domain name extracted in the previous step.

You can configure base and filter for the search operation via context parameters:

```
<context-param>
  <param-name>
      com.siemens.webMgr.auth.targetSystemBase
  </param-name>
  <param-value>cn=TargetSystems,cn=<dxiDomain></param-value>
</context-param>

<context-param>
  <param-name>
    com.siemens.webMgr.auth.targetSystemFilter
  </param-name>
  <param-value>
    (&amp;(objectclass=dxrTargetSystem)(dxrTSDomainName=%DOMAIN))
  </param-value>
</context-param>
```

Replace

- <dxiDomain> with your DirX Identity domain.

The placeholder “%DOMAIN” is replaced at runtime with the extracted Windows domain.

The search filter for the the Windows credentials “smith@beta.com” is

```
(&(objectClass=dxrTargetSystem)(dxrTSDomainName=beta.com))
```

when using the first of the above domain patterns, and

```
(&(objectClass=dxrTargetSystem)(dxrTSDomainName=beta))
```

when using the second pattern.

A.4.2.2.3. Finding the DirX Identity Account

The account is by default the one whose attribute dxmADsSamAccountName matches the extracted account name. The search base is the target system found in the previous step. If no target system was found, the search extends over all target systems.

You can configure the filter for the search operation via a context parameter:

```
<context-param>
  <param-name>
      com.siemens.webMgr.auth.accountFilter
  </param-name>
  <param-value>
      (&amp;(objectclass=dxrTargetSystemAccount)
          (dxmADsSamAccountName=%ACCOUNT))
  </param-value>
</context-param>
```

The placeholder “%ACCOUNT” is replaced at runtime with the Windows account extracted from the provided Windows credentials, the placeholder “%DOMAIN” (which is not used by default) with the extracted Windows domain.

The search filter for our sample Windows credentials “smith@beta.com” is

```
(&(objectClass=dxrTargetSystemAccount)
  (dxmADsSamAccountName=smith))
```

A.4.2.2.4. Finding the DirX Identity User

The user is the one referenced by attribute “dxrUserLink” of the DirX Identity account found in the previous step.

You can configure a different attribute via a context parameter:

```
<context-param>
  <param-name>
      com.siemens.webMgr.auth.accountUserLink
  </param-name>
  <param-value>
      dxrUserLink
  </param-value>
```

```
</context-param>
```

A.4.2.3. User Mapping via User Attribute

A.4.2.3.1. Extracting the Windows Credentials

The Windows credentials are passed in the format *accountName*@*fullDomainName*. The first task is to extract the part of the credentials that is needed to find the corresponding DirX Identity user.

Set the type of the SSO header filter to “user”:

```
<init-param>
    <param-name>type</param-name>
    <param-value>user</param-value>
</init-param>
```

The value to be extracted for the user filter is either the complete Windows credentials with account and fully qualified domain name, or a substring thereof.

The first example extracts the complete credentials with account and fully qualified domain name:

```
<init-param>
    <param-name>userRegExpr</param-name>
    <param-value>(.*)</param-value>
</init-param>
```

To extract the credentials with simple domain name, use:

```
<init-param>
    <param-name>userRegExpr</param-name>
    <param-value>([^@]+@[^.]+)(?:[.].*)?</param-value>
</init-param>
```

To extract the Windows account only, use:

```
<init-param>
    <param-name>userRegExpr</param-name>
    <param-value>([^@]*)@.*</param-value>
</init-param>
```

For example, if the provided Windows credentials are “smith@beta.com”, the extracted substring is “smith@beta.com” for the first pattern, “smith@beta” for the second pattern, and just “smith” for the third one.

Finding the DirX Identity User

You can configure base and filter for the search operation via context parameters:

```
<context-param>
  <param-name>
      com.siemens.webMgr.auth.userBase
  </param-name>
  <param-value>cn=Users,cn=<dxiDomain></param-value>
</context-param>

<context-param>
  <param-name>
    com.siemens.webMgr.auth.userFilter
  </param-name>
  <param-value>
    (&amp;(objectclass=dxrUser)(<attrName>=%USER_ID))
  </param-value>
</context-param>
```

Replace

- <dxiDomain> with your DirX Identity domain.
- <attrName> with the name of the user attribute holding the Windows credentials.

The placeholder “%USER_ID” is replaced at runtime with the Windows credentials extracted in the previous step.

The search filter with attribute name “description” for the Windows credentials “smith@beta.com” is

```
(&(objectClass=dxrUser)(description=smith@beta.com))
```

when using the first of the above user patterns,

```
(&(objectClass=dxrUser)(description=smith@beta))
```

when using the second pattern, and

```
(&(objectClass=dxrUser)(description=smith))
```

when using the third pattern.

A.5. Browser Settings

A.5.1. Internet Explorer / Chrome / Edge

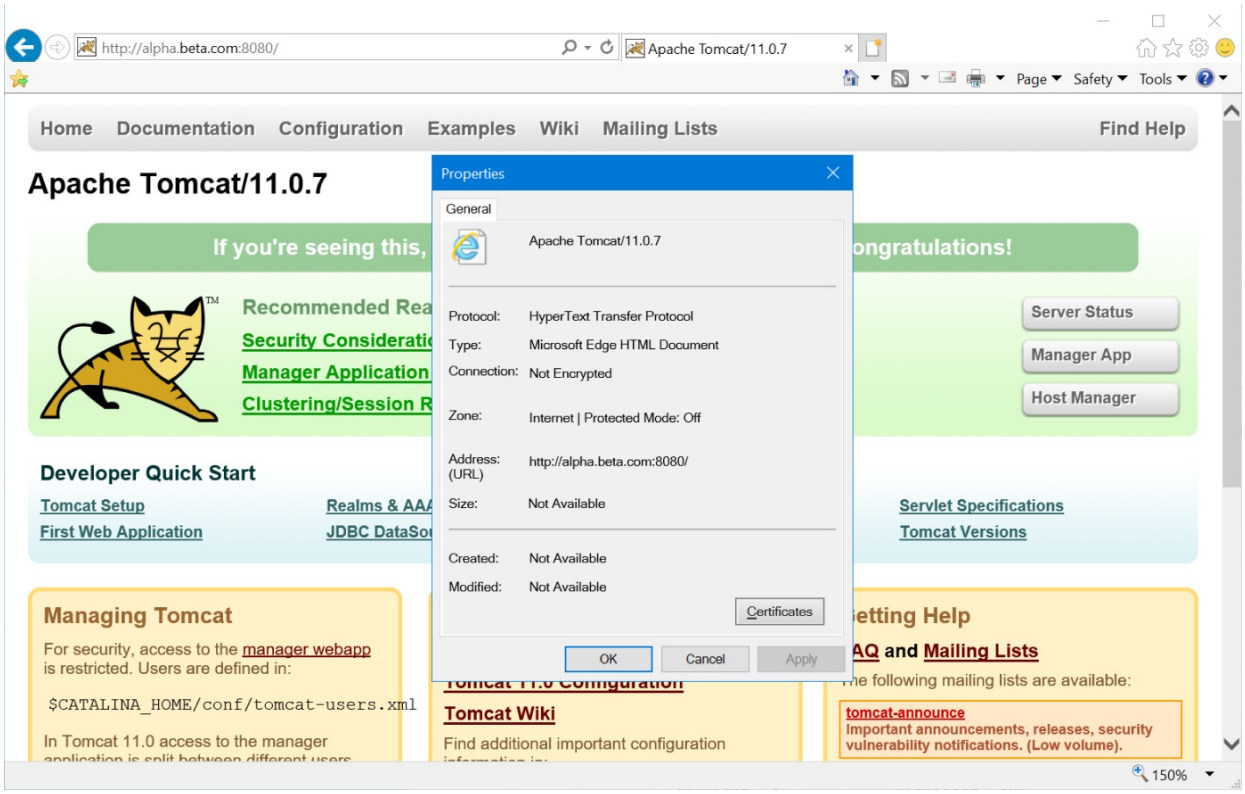
This section describes how to configure an Internet Explorer for Windows domain authentication based on Kerberos.This configuration requires Internet Explorer 11 respective Internet Explorer mode in Edge.

Kerberos authentication in Google Chrome and Microsoft Edge should work out-of-the-box if you have configured it for Internet Explorer.

A.5.1.1. Configuring Local Intranet Sites

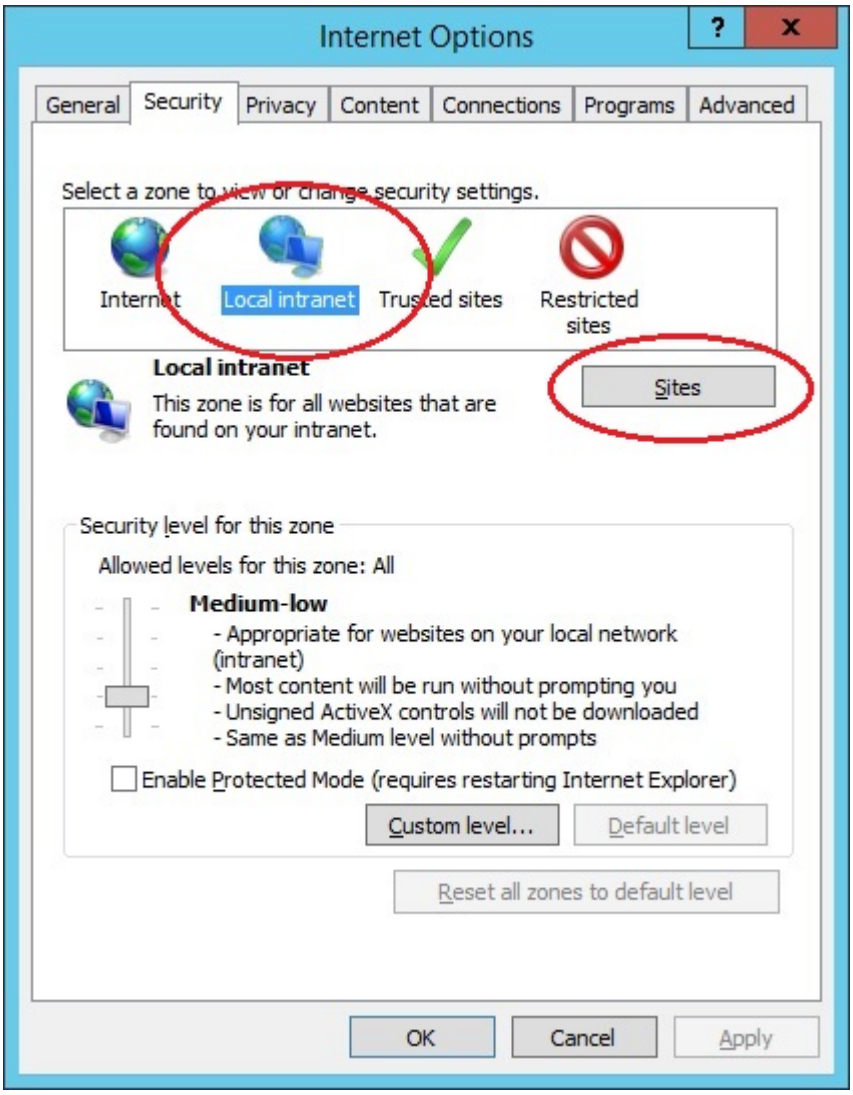
Internet Explorer requires servers to be configured in the category “Local intranet” to run Windows domain authentication based on Kerberos.

- Check if the Tomcat server belongs to the browser category “Local intranet” when accessing it.Go to the home page of your Tomcat, select item **Properties** from the **Page** menu, and check property **Zone**.

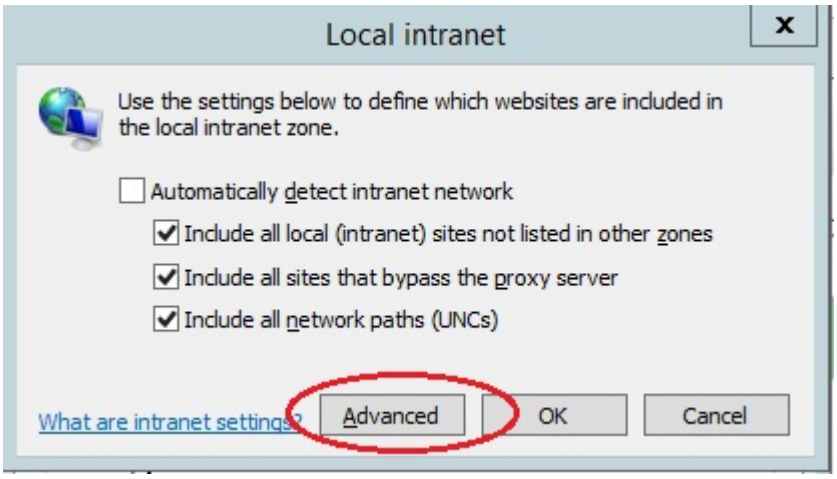


If “Local intranet” is shown when accessing Tomcat, you can skip the configuration steps for local intranet sites.

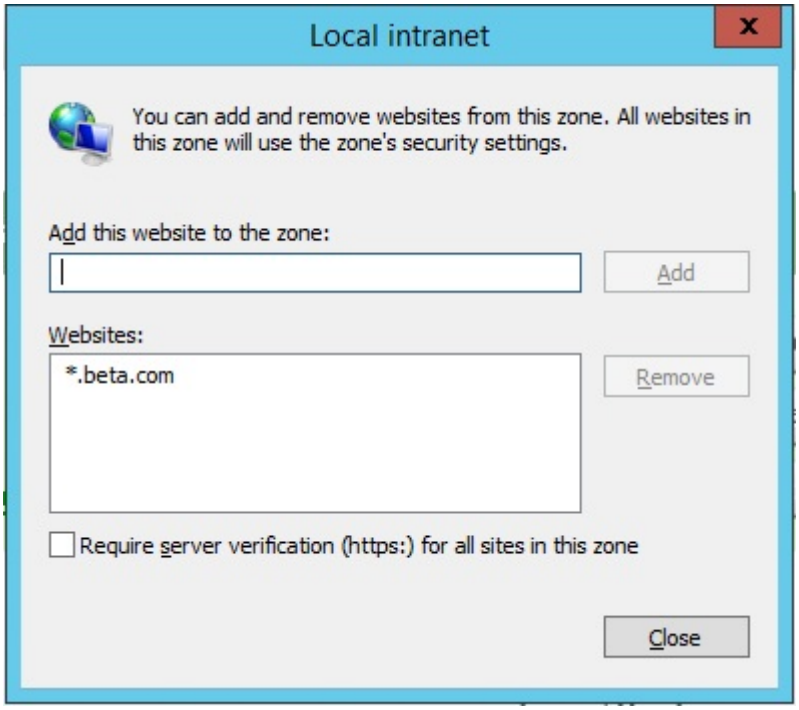
- Configure local intranet sites:
  - In Internet Explorer, click **Tools**, and then click **Internet Options**.
  - Click the **Security** tab.
  - Click **Local intranet**.
  - Click **Sites.**



  - Ensure that **Include all sites that bypass the proxy server** is checked, then click **Advanced.**



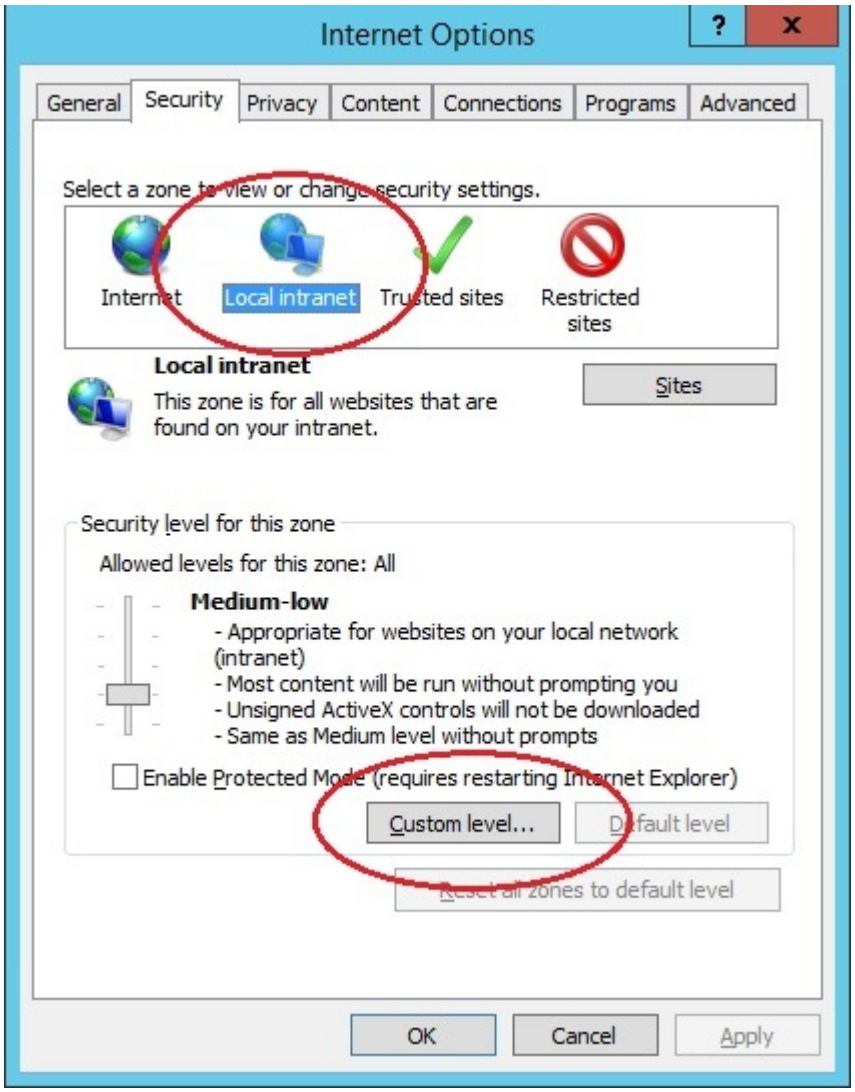
- In the Local intranet (Advanced) dialog box, enter all relative domain names that will be used for Tomcat servers with enabled Windows domain authentication (for example, *.beta.com).



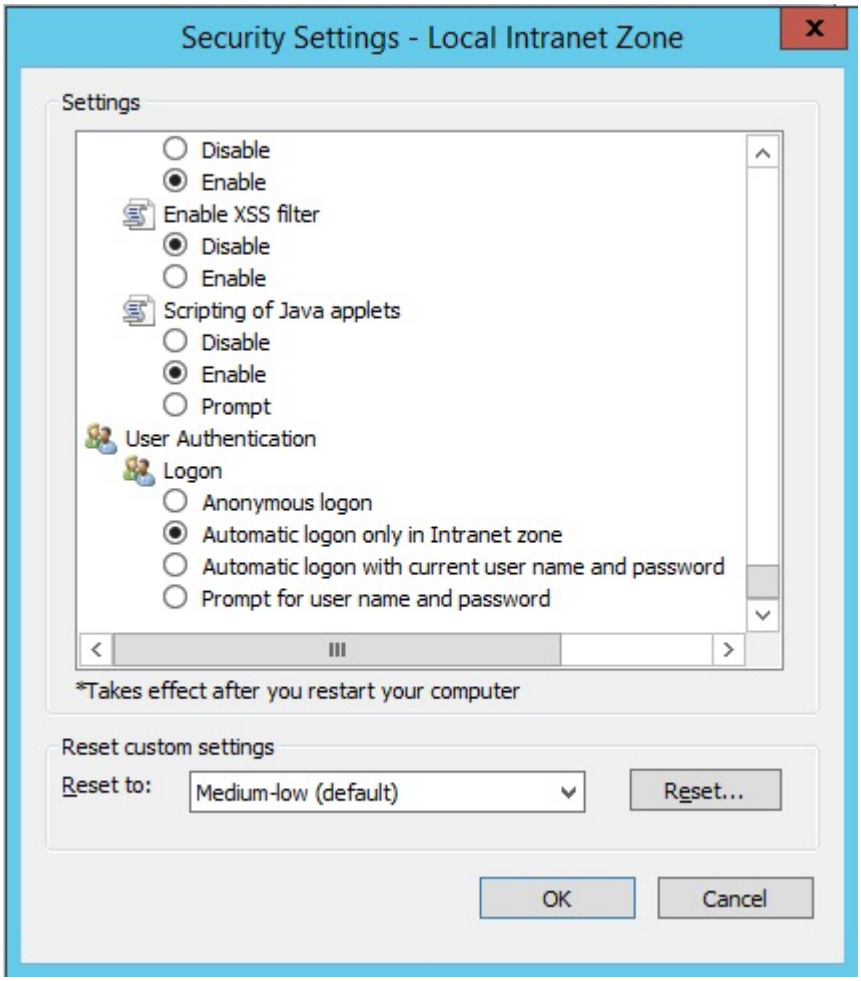
- Click **Close** to close the dialog boxes.

A.5.1.2. Configuring Intranet Authentication

- Next, click the **Security** tab, click **Local intranet**, and then Click **Custom Level**.



- In the **Security Settings** dialog box, scroll down to the **User Authentication** section of the list.

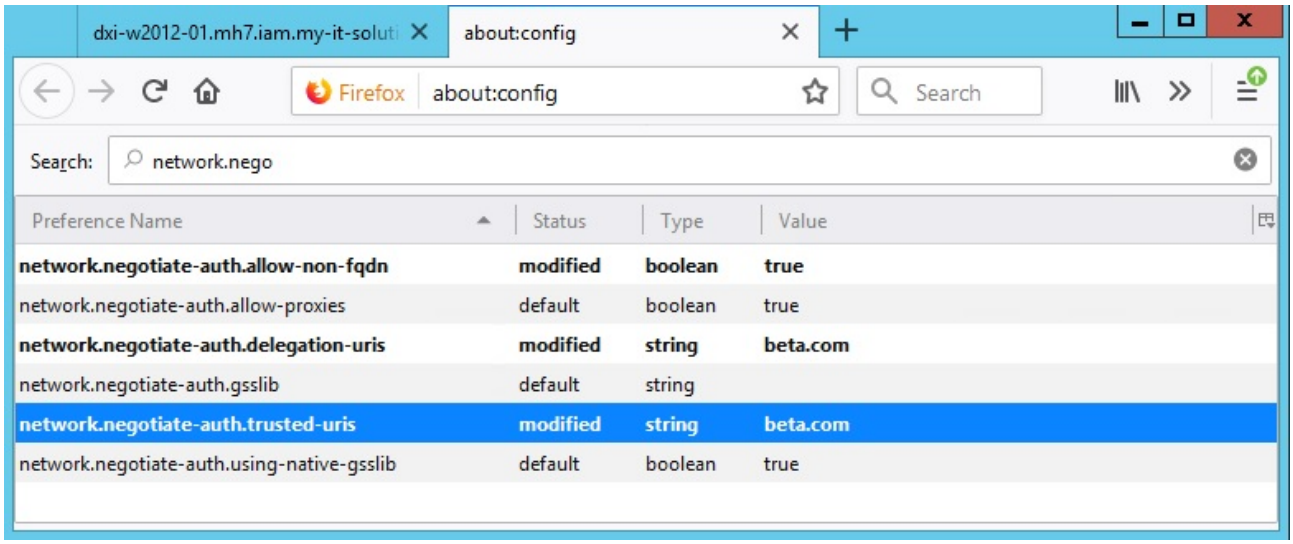


- Select **Automatic logon only in Intranet zone**. This setting enables integrated Windows domain authentication in the Internet Explorer.
- Click **OK** to close the Security Settings dialog box.

A.5.2. Firefox

This section describes how to configure Firefox for Windows domain authentication based on Kerberos.

- Open Firefox and enter “about:config” in the browser’s address bar.
- Enter “network.negotiate” in the search bar.search bar.
  - Add the fully qualified domain name to parameter “network.negotiate-auth.trusted-uris”.
  - Add the fully qualified domain name to parameter “network.negotiate-auth.delegation-uris”.
  - Set “network.negotiate-auth.allow-non-fqdn” to “true”.



A.6. Testing

Restart Tomcat for the configuration changes to become effective.

Open a browser and enter the URL

```
http://<tomcatHost>:<tomcatPort>/webCenter-<dxiDomain>
```

Replace

- <tomcatHost> with the fully qualified host name of Tomcat.
- <tomcatPort> with the HTTP port of Tomcat.
- <dxiDomain> with your DirX Identity domain.

For example:

```
http://alpha.beta.com:8080/webCenter-My-Company
```

If everything works fine, you will be authenticated to Web Center and see Web Center's home page.

If it works, test again using the simple host name:

```
http://alpha:8080/webCenter-My-Company
```

Note that Kerberos usually doesn't work if browser and Tomcat run on the same machine.

A.7. Logging and Debugging

A.7.1. Kerberos Ticket Evaluation

Logging for the standard Java classes evaluating and verifying Kerberos tickets and the keytab file is activated via

- Two Java system properties of the Tomcat service:

```
-Dsun.security.krb5.debug=true
-Dsun.security.jgss.debug=true
```

- A debug parameter in file **krb5-jaas.conf**:

```
com.sun.security.jgss.krb5.accept {
    com.sun.security.auth.module.Krb5LoginModule required
        debug=true
        ...
```

For details see section “Tomcat Configuration”.

The logs are written to Tomcat’s standard output, which is usually redirected to a Tomcat log file (like *tomcat_install_path*/**logs/tomcat11-stdout.***date***.log**) or the Tomcat console window.

Logging for the ticket evaluation classes in jar file **spnego-tomcat11.jar** is activated by setting the general log level in file **WEB-INF/web.xml:**

```
<context-param>
    <param-name>com.siemens.webMgr.log.level</param-name>
    <param-value>2</param-value>
</context-param>
```

The logs are also written to Tomcat’s standard output.

A.7.2. User Mapping

Logging for the components involved in mapping the Windows credentials to a DirX Identity user is activated via the general log level in file **WEB-INF/web.xml:**

```
<context-param>
    <param-name>com.siemens.webMgr.log.level</param-name>
    <param-value>2</param-value>
```

```
</context-param>
```

The logs are written to Tomcat's standard output.

A.7.3. Viewing Tickets

Use the Windows command "klist" to display the list of currently cached Kerberos tickets.

Listing the currently cached tickets:

```
klist
```

Deleting the tickets:

```
klist purge
```

A.8. Links

- An overview on Kerberos by MIT:
  https://web.mit.edu/kerberos
- Configuring Kerberos Authentication in Different Browsers:
  http://woshub.com/enable-kerberos-authentication-in-browser/#h2_2
- Configuring Kerberos Authentication in Firefox:
  https://developer.mozilla.org/en-US/docs/Mozilla/Integrated_authentication
- The Windows command **klist**:
  https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/klist
- The Windows server command **ktpass**:
  https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ktpass

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.