EVIDEN

Identity and Access Management

DirX Identity

Certification Campaigns

Version 9.0.2, Edition November 2026



All product names quoted are trademarks or registered trademarks of the manufacturers concerned.

© 2026 Atos Group

All Rights Reserved

Distribution and reproduction not permitted without the consent of Atos Group.

Table of Contents

Copyright ii
Preface 1
DirX Identity Documentation Set 2
Notation Conventions 4
1. Overview 6
1.1. Use Cases 6
1.1.1. User Certification Campaigns 6
1.1.2. Privilege Certification Campaigns 6
1.1.3. Certification Campaigns with Risk Governance 7
1.1.4. Continuous Access Certification via Re-approval Workflows 7
1.1.5. Scheduled Access Certification Campaigns via Re-approval Workflows 7
1.2. How a Certification Campaign Works 7
1.2.1. Certification Campaign Pre-Requisites 8
1.2.2. Creating a Certification Campaign 8
1.2.3. Certification Campaign Notifications 8
1.2.4. Starting a Certification Campaign 9
1.2.5. Certifying Users or Privileges 10
1.2.6. Finishing a Certification Campaign 11
1.2.7. Recurring Certification Campaign 11
1.2.8. Generating Reports 12
2. Preparing for a Certification Campaign 13
2.1. Configuration Steps 13
2.1.1. 1. Configure the Email Service 13
2.1.2. 2. Configure the Certificate Service 13
2.1.3. 3. Configure the Resource Family 14
2.1.4. 4. Create a Schedule 14
2.1.5. 5. Enable the Workflow 14
3. Creating a Certification Campaign 16
3.1. Steps to Create a New User Certification Campaign 16
3.1.1. Mandatory Fields 16
4. Starting a Certification Campaign 18
4.1. What the Campaign Controller Does 18
4.2. Handling Failures 18
4.2.1. Approver Actions 18
5. Monitoring and Finishing a Campaign 19
5.1. Campaign Workflow Overview 19
5.2. Certification Task Workflow 20
5.3. Campaign Completion 21
6. Customizing with User Hooks 22

6.1. Documentation and Sample Source Code . . . 22
6.2. Select Approvers with a “Find Approvers” User Hook . . . 22
6.2.1. Before – Creating the Certification Task in the User Hook . . . 23
6.2.2. After – Changing Default Attributes . . . 23
6.2.3. Find Approvers – Adding Additional Approvers . . . 23
6.3. Select Campaign Subjects with a “Find Subjects” User Hook . . . 24
6.4. Limit Resources with a “Limit Resources” User Hook . . . 25
6.5. Send emails with the “Send Email” User Hook . . . 26
6.6. Override Campaign Creation with “Campaign Creator” . . . 26
7. Generating Certification Campaign Reports . . . 28
7.1. About the Default Report Templates . . . 28
7.2. Producing a Report . . . 28
7.3. Adding or Extending Reports . . . 29
8. Continuous Certification by Re-Approval . . . 30
8.1. About this Use Case . . . 30
8.2. Setup and Configuration . . . 31
8.2.1. Setting up the Domain Object . . . 31
8.2.2. Setting up the Privileges . . . 32
8.2.3. Setting up the Workflows . . . 32
8.2.3.1. InitializeReapproval . . . 32
8.2.3.2. StartReapproval . . . 33
8.3. Running the Use Case . . . 34
8.4. Alternative or Extended Configurations . . . 34
8.4.1. Certifying Other Privilege Types . . . 34
8.4.2. Running Separate Campaigns . . . 34
9. Scheduled Certification by Re-Approval . . . 35
9.1. About this Use Case . . . 35
9.2. Setup and Configuration . . . 36
9.2.1. Setting up the Domain Object . . . 36
9.2.2. Setting up the Privileges . . . 37
9.2.3. Setting up the Workflows . . . 37
9.2.3.1. InitializeReapproval . . . 37
9.2.3.2. StartReapproval . . . 38
9.3. Running the Use Case . . . 38
9.4. Alternative or Extended Configurations . . . 39
9.4.1. Certifying Other Privilege Types . . . 39
9.4.2. Setting Individual Re-Approval Dates or Re-Approval Periods . . . 39
10. Hints and Limitations . . . 40
10.1. Manual and Automatic Assignments . . . 40
10.2. Certification Campaign Logging . . . 40
10.3. Java-based Server Workflow Load . . . 40
Legal Remarks . . . 43

Preface

This document describes how to set up and run certification campaigns for **users** and **privileges**.

A user or privilege certification campaign does not use request workflows. In the current DirX Identity version, this functionality is visible in configuration settings for **users** and **privileges**. The request workflow-based campaign technique is deprecated and will be removed with the next DirX Identity version.

This document consists of the following chapters.

- Overview provides an overview of the use cases.
- Preparing for a Certification Campaign describes how to prepare for user/privilege certification campaigns.
- Creating a Certification Campaign describes how to create a user/privilege certification campaign.
- Starting a Certification Campaign describes how to start a user/privilege certification campaign.
- Monitoring and Finishing a Campaign describes how to monitor and finish a user/privilege certification campaign.
- Customizing with User Hooks describes how to customize user/privilege certification campaigns with user hooks.
- Generating Certification Campaign Reports describes how to create reports for a user/privilege certification campaign.
- Continuous Certification by Re-Approval describes continuous certification via re-approval workflows.
- Scheduled Certification by Re-Approval describes scheduled certification via re-approval workflows.
- Hints and Limitations provides information about hints and limitations.

DirX Identity Documentation Set

Version: 9.0.2 | Date: 2026-11-02 | Timestamp: 13:57:36 UTC

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. Overview

DirX Identity provides a comprehensive role model to control user access rights to resources in connected systems.Features such as access policies, segregation of duties (SoD) and approval workflows help to secure the assignment of access rights.Compliance requirements require regular certification or re-certification of these assignments.DirX Identity provides several mechanisms to support these compliance processes.

Be aware that certification can only be applied to manually-assigned privileges.Rule-based assignments and the assignments made from business object inheritance are not subject to user/privilege certification campaigns because DirX Identity assumes that a rule or an inheritance definition is designed to make the correct assignments.It is not possible to "adjust" the rule or the inheritance definition later on by manually removing specific assignments.

1.1. Use Cases

This document describes several use cases in detail.Be aware that other use cases are possible that are not described in this document.

1.1.1. User Certification Campaigns

This use case performs a certification campaign for a subset of users, referred to as the **subjects** of the campaign.

The users are selected by applying an LDAP search filter configured for the campaign. By default, all manual privilege assignments must be certified. By implementing a “Find Subject” user hook, customers can implement their own method of finding users for the campaign. The set of privileges to be certified (called the **resources**) can be reduced by configuring an LDAP filter or by implementing a “Limit Resources” user hook.

For each user, one or more approvers are automatically defined. The default implementation selects the user’s manager. A “Find Approvers” user hook can override this setting and select a custom set of approvers per user.

1.1.2. Privilege Certification Campaigns

This use case performs a certification campaign for a subset of privileges, typically roles. In this case, the privileges are the certification **subjects**.

The privileges are selected by applying an LDAP search filter configured for the campaign. By default, all manual user-privilege assignments must be certified. By implementing a “Find Subject” user hook, customers can implement their own method of finding privileges for the campaign. The set of users to be certified (called the resources) can be reduced by configuring an LDAP filter or by implementing a “Limit Resources” user hook.

For each privilege, one or more approvers are automatically defined. The default implementation selects the privilege’s owner. A “Find Approvers” user hook can override this setting and select a custom set of approvers per privilege.

1.1.3. Certification Campaigns with Risk Governance

In addition to certification campaigns above, customers can create certification campaigns for **subjects** with a high risk. If the **Risk Governance** feature is enabled, the customer can extend the LDAP search filter for adding users based on risk values; for example a user certification campaign for the finance department, for users with a high risk, or a privilege certification campaign for the role finance administrator only for users with a medium risk or above.

1.1.4. Continuous Access Certification via Re-approval Workflows

This use case works with DirX Identity's built-in re-approval feature. In this scenario, the approval for selected and critical privileges is repeated at the specified time. You can either run the same workflow that was run to approve the privilege or a specific workflow for re-approval. If the approvers reject the assignment, the privilege is removed from the user. This method works individually per assignment.

You can combine the re-approval of selected privileges with timing conditions. If you run the **InitializeReapproval** and the **StartReapproval** workflows on a daily basis, these workflows check the timing conditions and then start re-approval workflows as necessary.

1.1.5. Scheduled Access Certification Campaigns via Re-approval Workflows

This use case works with the built-in re-approval feature. In this scenario, the approval of all critical privileges is scheduled for a specific time. You can either run the same workflow that was run to approve the privilege or a specific workflow for re-approval. If the approvers reject the assignment, the privilege is removed from the user. This method works individually per assignment. The difference from the previous use case is that the parameters are set to run all re-approvals at the same time. Flag the critical privileges for re-approval and then set the re-approval date at the domain.

In this case, it makes sense to run the **InitializeReapproval** and the **StartReapproval** workflows only once at the correct time to start all re-approval workflows in parallel.

1.2. How a Certification Campaign Works

This section describes some technical details that help to understand the certification campaign use cases. First we describe how the feature works and which components are involved. Next, we explain how all these components are integrated in the system and how to calculate the certification campaign end date.

You can skip this section and read it later on if you need deeper understanding of this fairly complex feature.

Several processes are necessary to perform a certification campaign in DirX Identity. They are described in the following sections.

1.2.1. Certification Campaign Pre-Requisites

A certification campaign has the following DirX Identity configuration pre-requisites:

- **Certification Campaign Controller workflow** (called the campaign controller) – this workflow needs to be active, needs a schedule so that it runs at least once per day and must be deployed on exactly one Java-based Server (resource family Certification Campaign).
- **Email service** – the campaign controller sends various notifications during the campaign and relies on the general email service to be configured correctly.

1.2.2. Creating a Certification Campaign

A certification campaign administrator creates a campaign entry in state **Campaign is in preparation (PREPARING)** and then:

- Select the certification type (user/privilege)
- Sets the start and approval period.
- Sets the filter for the subjects (the users or privileges) or alternatively a “Find Subjects” user hook.
- Sets a resource (privilege or user) filter or a “Limit Resources” user hook if some privileges of a user or some users assigned a privilege shall be excluded.
- Sets a “Find Approvers” user hook if the approver is not the default.
- (Optional) Sets “Recurring Certification Campaign” to indicate that the campaign is to be run periodically and sets the time period after which the campaign will be run again in “Interval”.

Next, the administrator must enable the notifications to be used and adapt their templates, especially subject and body.

1.2.3. Certification Campaign Notifications

From the campaign’s start until its end, the Certification Campaign Controller can send different mail notification for each campaign phase. When a campaign is created, the mail notification templates are copied from **Certification Campaigns** → **_Default** → **Notifications** container to the newly created campaign **Notifications** container. The administrator can enable or disable these notifications (**General** → **Is Active**), and adjust the notification content (for example, subject, body).

The following mail notification templates are available (listed here in alphabetical order):

- **Approval Remind** – Send a notification to approvers to remind them about their tasks. This notification is configurable with **Remainder** notification values available in the Certification Campaign entry (**Days before due date** and **Interval between remainders (hours)**).
- **Approval Start** – Send a notification to approvers when they get new certification tasks.
- **Approval Timeout** – Send a notification to approvers when one of their certification

tasks has expired.

- **Assignment Reject** – Send a notification to certified users when at least one of their privilege assignments has been rejected.
- **Campaign End** – Send a notification to the campaign owner when a campaign ends.
- **Campaign Start** – Send a notification to the campaign owner when a campaign starts.
- **No Approver** – Send a notification to the campaign owner when no approver was found for a user or privilege to be certified.
- **Prepare Error** – Send a notification to the campaign owner when starting a campaign failed.



When the certification campaign is restarted (manually or as part of the recurring certifications) the Notifications container and its content are not modified.

1.2.4. Starting a Certification Campaign

When the start date of the campaign in state **Campaign is in preparation (PREPARING)** is reached, the Certification Campaign Controller workflow starts the campaign.

The following figure provides an overview of the campaign phases and notifications to be sent:

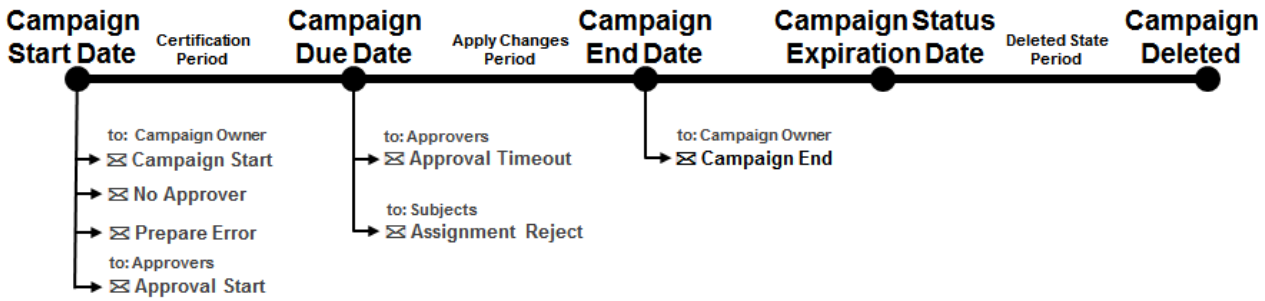


Figure 1. Certification Campaign Phases and Notifications

At start-time, the campaign controller validates the parameters of the campaign. Any warnings or errors are stored in the **Logs** field of the campaign entry.

- **Approval Period.**
- **Due Date**: can be empty, and it will be calculated at the start of the campaign: **Start Date** plus **Approval Period**. If **Due Date** is set, this value is used for the current campaign run. If it is a recurring certification campaign, the value will be set to the default value: **Start Date** plus **Approval Period** - on the next start of the campaign
- **Status Expiration Date**: if available, must be after the **Due Date**. If not available, the controller logs a warning. At the end of the campaign, it will calculate a default value **End Date** plus 30 days.
- **User Filter** and **Privilege Filter**:

For a user campaign, both a **User Base** and a **User Filter** must be set or a “Find Subjects” user hook must be specified.

For a privilege campaign, both a **Privilege Base** and a **Privilege Filter** must be set or a “Find Subjects” user hook must be specified.

If the validation fails, the controller sets the campaign state to **Campaign failed to start (FAILED.PREPARED)** and sends a notification “**Prepare Error**” to the campaign owner. The administrator - who should be the campaign owner - can then fix the problems, set the state to **PREPARING** and then start the Certification Campaign Controller workflow.

If the start settings are correct, the campaign controller searches the users (subjects of the campaign), collects their manual privilege assignments (the resources of the certification), potentially removes them according the privilege filter or the “Limit Resources” user hook and sets the approver(s) per user. For each user, it stores the result in a certification entry underneath the campaign entry in a child container **User Certification**. If a “Find Approvers” user hook provides more than one approver for the user, it creates a subtree of certification entries. For more details, see the section “Select Approvers with a Find Approvers User Hook”.

For a privilege campaign, the certification entries are stored in a child container **Privilege Certification**. Each certification entry represents a privilege to be certified and its assignments are the users of this privilege. The approver is responsible for certifying the privilege.

For each certification entry (that is, for each user or privilege to be certified), the campaign controller sends an **Approval Start** notification if available.

If a certification entry does not have at least one approver, its state is set to **FAILED.PREPARE** and the campaign controller sends a notification **Prepare Error** to the campaign owner. If a certification entry does not have any manual assignments, the state will be directly set to **RUNNING** and certification entry will be automatically closed at the end of the approval period. The campaign controller also sends a **No Approver** notification to the campaign owner.

When all certification entries have been created, the campaign controller sets the campaign state to **Campaign is running (RUNNING)** and then sends a **Campaign Start** notification to the campaign owner.

1.2.5. Certifying Users or Privileges

Approvers are notified by an **Approval Start** notification. When they open DirX Identity Web Center, they can see the certification campaigns in which they are involved in their start page and can immediately navigate to their certification tasks, or they can use DirX Identity Business User Interface Certification Campaign feature to approve or reject certification tasks. For each entry to be certified (user or privilege), they can see all manual assignments and can decide whether to accept or reject each individual assignment. If an assignment has an end date or a role parameter (for example, the manager for a project), the approver can delete or modify the end date or the parameter. These changes are stored underneath the certification entry and are applied at the end of the campaign. Approvers can store their decisions at any time and then continue with the open tasks later on. When

the approver saves the decisions for all assignments of a certification entry, DirX Identity Web Center or DirX Identity Business User Interface sets the state of the entry to **APPROVAL.FINISHED**.

The Certification Campaign Controller workflow should be scheduled to run regularly. In addition to starting campaigns, it also monitors running campaigns.

When it detects a finished certification entry, the campaign controller starts downstream approvals for that entry when available. When a certification task is approaching the due date, it sends reminder notifications (type **Approval Timeout**). When a certification task has reached the due date, the campaign controller sets its state to **FAILED.EXPIRED**. If a downstream approval exists, it is started.

The administrator can change the due date or state of single certification entries or the entire campaign at any time, as necessary. So they can react in a flexible way to any errors.

1.2.6. Finishing a Certification Campaign

When the **Due Date** of the certification campaign is reached (manually set by the campaign administrator or automatically calculated at the start of the campaign)**,** the campaign controller starts to apply the changes. The actions depend on the “revoke privileges” settings made in the campaign entry. Rejected, changed and ignored (uncertified) assignments can simply be left as they are, removed from the users or evaluated by an approval workflow (only when set accordingly per privilege). When an assignment is removed, the campaign controller sends an **Assignment reject** notification to the users.

If an error occurs during the apply change process, the campaign controller sets the state of the certification entry to **FAILED.APPLIED.CHANGES**. Otherwise, the controller sets the state to **FINISHED** and sets the **End Date**.

At the end, the campaign controller sets the campaign state to **Campaign finished successfully (SUCCEEDED)**, sets the **End Date**, calculates the **Status Expiration Date** if required (the date at which the campaign will be moved to state **DELETED**), and sends a **Campaign End** notification.

When the **Status Expiration Date** is reached, the campaign controller sets the campaign state to **Campaign is marked for deletion (DELETED)**. A subsequent “Cleanup Objects” workflow will physically delete the entire subtree for the campaign later on.

1.2.7. Recurring Certification Campaign

When values are set for **Recurring Certification Campaign**, the certification campaign will be restarted after the time period specified in **Interval**.This option applies only to campaigns that have finished with success.The campaign controller will check if the new start date is reached (**Start Date** from previous campaign plus **Interval**) and will move the current closed campaign to the **_Archive** container in a folder with the campaign name and start date (for example, Finance High Risk Users Certification 2016-07-01) and will start the new campaign.The **Due Date** for the new campaign is calculated based on **Start Date** and **Approval Period** from the initial campaign.Any eventual manual modification of the

Due Date during the previous campaign will be ignored.

To stop a recurring certification campaign, the administrator must clear all values from **Interval**.

1.2.8. Generating Reports

DirX Identity provides default reports on certification campaigns.You can generate a report on a campaign at any time, especially after the campaign has finished.

To generate a report, in DirX Identity Manager, right-click a campaign entry, select **Report** and then select one of the available reports from the list provided. See their descriptions to get more information on their content.

2. Preparing for a Certification Campaign

Before starting a certification campaign, ensure the following prerequisites are met:

- The workflow for creating and monitoring certification campaigns is active.
- The workflow runs on exactly one Java-based server.
- The general email service is configured.

2.1. Configuration Steps

2.1.1. 1. Configure the Email Service

The Certification Campaign Controller workflow uses the central **SendMail** workflow to send emails via **SMTP**.

To enable this service:

1. Navigate to **Provisioning** → **Workflows** → **Services** → **SMTP**.
2. Provide correct values for:
   - **SMTP host**
   - **Port** (external SMTP server)
3. Check the **Map mail address** field:
   - For production, ensure it is blank.
   - Do **not** contain the text **dummy** (this disables the SMTP workflow).
4. If sending encrypted emails:
   - Enable **Encrypt Email**.
   - Decide on the **Send on Encryption Failure** flag.



For detailed instructions, consult the context-sensitive help.

2.1.2. 2. Configure the Certificate Service

If you plan to send encrypted emails via the **SendMail** workflow:

1. Define **IP Address** and **Port** if retrieving user certificates from an external LDAP directory.
2. If certificates are stored in the Provisioning View user tree:
   - Leave **IP Address** empty.
3. Provide values for:
   - **Search Parameter** fields
   - **Refresh Interval** for the internal certificate table

Refer to the context-sensitive help for more details.

2.1.3. 3. Configure the Resource Family

The Certification Campaign Controller workflow requires the resource family **CertificationCampaign** to run on a Java-based server.

To configure:

1. Navigate to **Connectivity** → **Expert View** → **Configuration** → **Java Servers** → *yourdomain* → *yourdomain - yourhost* .
2. In the **Resource Families** tab:
   - Move **CertificationCampaign** from **Available** to **Selected**.
   - Set **Thread number** to **1**.
3. Click **Save**.

2.1.4. 4. Create a Schedule

The Certification Campaign Controller workflow is triggered by a schedule (not events). It performs actions such as:

- Starting campaigns
- Monitoring campaigns
- Applying changes
- Finishing and deleting campaigns
- Sending notifications

To create a schedule:

1. Navigate to **Connectivity** → **Expert View** → **Schedules**.
2. Create a schedule for the workflow:
   - **Workflows** → **Default** → **Identity Store** → **CertificationCampaignController**.

     For details, consult the context-sensitive help and DirX Identity documentation.

2.1.5. 5. Enable the Workflow

The Certification Campaign Controller workflow is disabled by default.

To enable:

1. Navigate to **Connectivity** → **Workflows** → **Default** → **Identity Store** → **CertificationCampaignController**.
2. In the **General** tab:
   - Enter edit mode.

- Check **Is Active**.

3. Creating a Certification Campaign

A certification campaign is represented by a campaign entry in the **Provisioning** domain.

3.1. Steps to Create a New User Certification Campaign

1. Navigate to the sub-view **Certification Campaigns** in the view group **Provisioning**.
2. Right-click an appropriate parent container and select:

   New → CertificationCampaign

3. Provide values for the following mandatory fields:

3.1.1. Mandatory Fields

- **Name** The name of the campaign. This will be displayed to approvers and included in emails.
- **Type** The campaign type. Choose one of:
  - **User Certification**
  - **Privilege Certification**
- **Owner** A user in the domain who will be considered the campaign owner.
- **Reminder Notifications** Define:
  - The period before the campaign due date when reminders should start.
  - The interval at which reminders are sent to approvers.
- **Apply Changes** Specify how rejected assignments are handled at the end of the campaign:
  - **Do not revoke any rejected privileges** – No assignments are removed.
  - **Revoke all manually rejected privileges that are rejected or left uncertified** – Removes both explicitly rejected and ignored assignments.
  - **Revoke only rejected privileges that were manually assigned** – Removes only assignments explicitly rejected by the approver.
  - **Review the revocation of all manually assigned privileges that are rejected or left uncertified** – Starts an approval workflow for all manual assignments not explicitly accepted. These assignments are removed if approval is required.
- **Status** Must be set to:
  - `Campaign is in preparation (PREPARING)`
- **Start Date** The date when the campaign should start. To start immediately, set this date in the past.
- **Approval Period** The duration of the approval period. This determines the **Due Date** and should allow sufficient time for approvers to complete certifications.

- **Due Date** Calculated as:
  - `Start Date + Approval Period` The administrator can adjust this during the campaign.
- **Status Expiration Date** Optional. Specifies when certification LDAP entries should be physically deleted. If left blank, the workflow sets a default expiration date of:
  - `Current Date + 30 days`
- **User Base** and **User Filter** Define the subjects of the certification.



You can create a Certification Campaign for users with a specific risk level if the **Risk Governance** feature is enabled in the **Domain Configuration** tab.

Add search parameter for the LDAP attribute: **`dxrRskLevel` Values:** `0` – Normal risk **1 – Low risk** 2 – Medium risk ** 3 – High risk

Example filter:
`(&(objectClass=dxrUser)(|(ou=Finances))(dxrRskLevel>=2))`

This query retrieves all users from the **Finance** organizational unit with medium risk level.

- **Privilege Filter Base** and **Privilege Filter** Define the privileges to be certified:
  - Leave blank to certify all manual assignments of the subjects.
  - Specify values to restrict privileges to those matching the filter.

These fields are mandatory to start a certification campaign. ``

4. Starting a Certification Campaign

When the start date of a campaign in state **Campaign is in preparation (PREPARING)** is reached, the **Certification Campaign Controller** workflow starts the campaign.

The workflow is normally triggered according to the defined schedule. However, you can also start it manually using **DirX Identity Manager**:

1. In the **Connectivity** view, navigate to the workflow entry.
2. Select **Run Workflow** from the context menu.

4.1. What the Campaign Controller Does

The Campaign Controller workflow performs the following actions:

- Collects the users to be certified along with their privilege assignments.
- Identifies approver(s) for each user.
- Stores certification tasks in the campaign subtree.
- Notifies the approvers.

4.2. Handling Failures

If something fails — for example, an approver cannot be found for a user — the workflow:

- Creates the certification task.
- Sets its state to **FAILED.PREPARE**.
- Notifies the campaign owner.

The campaign owner can then:

- Manually fix the issue.
- Change the state to **PREPARED** or **RETRY.PREPARE**.

When the Certification Campaign workflow runs again, it re-evaluates the task and, if successful, changes the state to **RUNNING**.

4.2.1. Approver Actions

Approvers can work on their tasks and certify user assignments using:

- **DirX Identity Web Center**
- **DirX Identity Business User Interface**

5. Monitoring and Finishing a Campaign

The following diagram provides an overview of the states of a campaign:

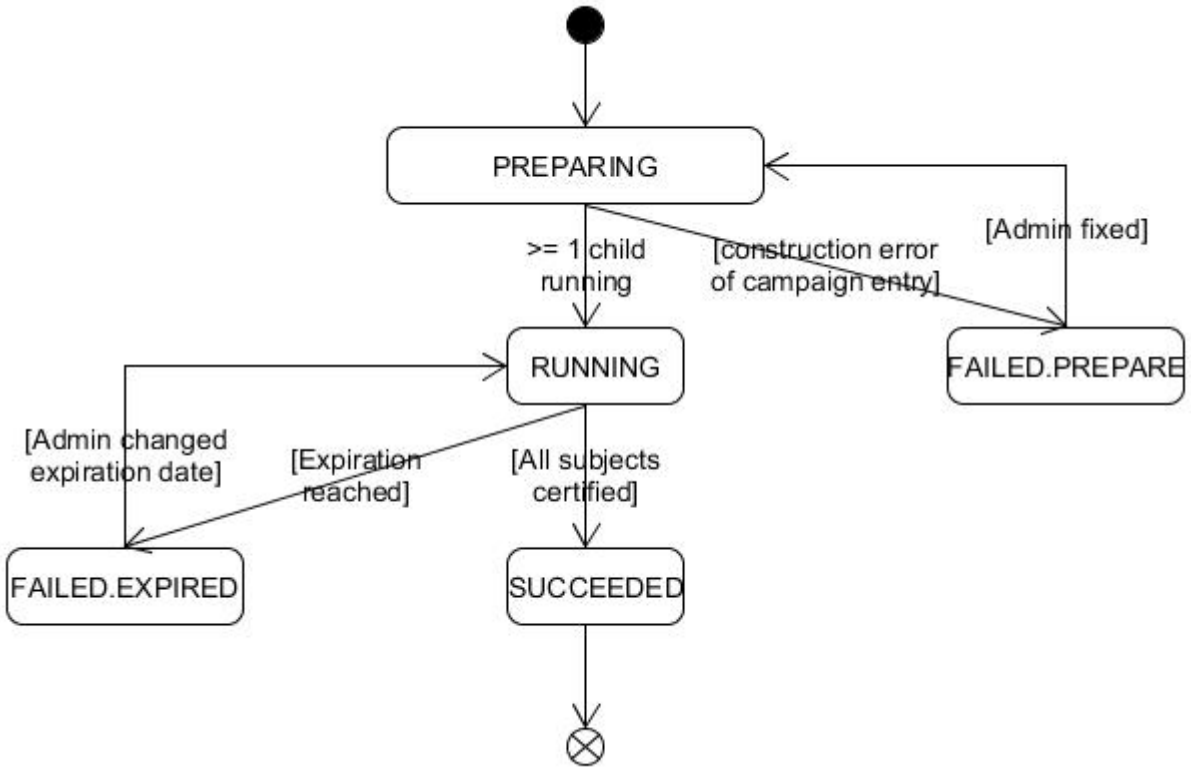


Figure 2. Certification Campaign States

5.1. Campaign Workflow Overview

- When the Certification Campaign workflow successfully creates at least one certification task, the campaign state becomes **RUNNING**.
- If no task is created, the state is set to **FAILED.PREPARE**.
  - The administrator (campaign owner) can fix any issues and manually set the state to **PREPARING**.
  - On the next run, the workflow attempts to start the campaign.
- When the campaign reaches its **Due Date**:
  - The workflow applies changes from all finished certification tasks.
  - If no task is finished, the campaign state changes to **SUCCEEDED**.
  - In both cases, the administrator can extend the campaign by:
  - Updating the due date.
  - Setting the state back to **RUNNING**.

The following diagram shows the states of a single certification task:

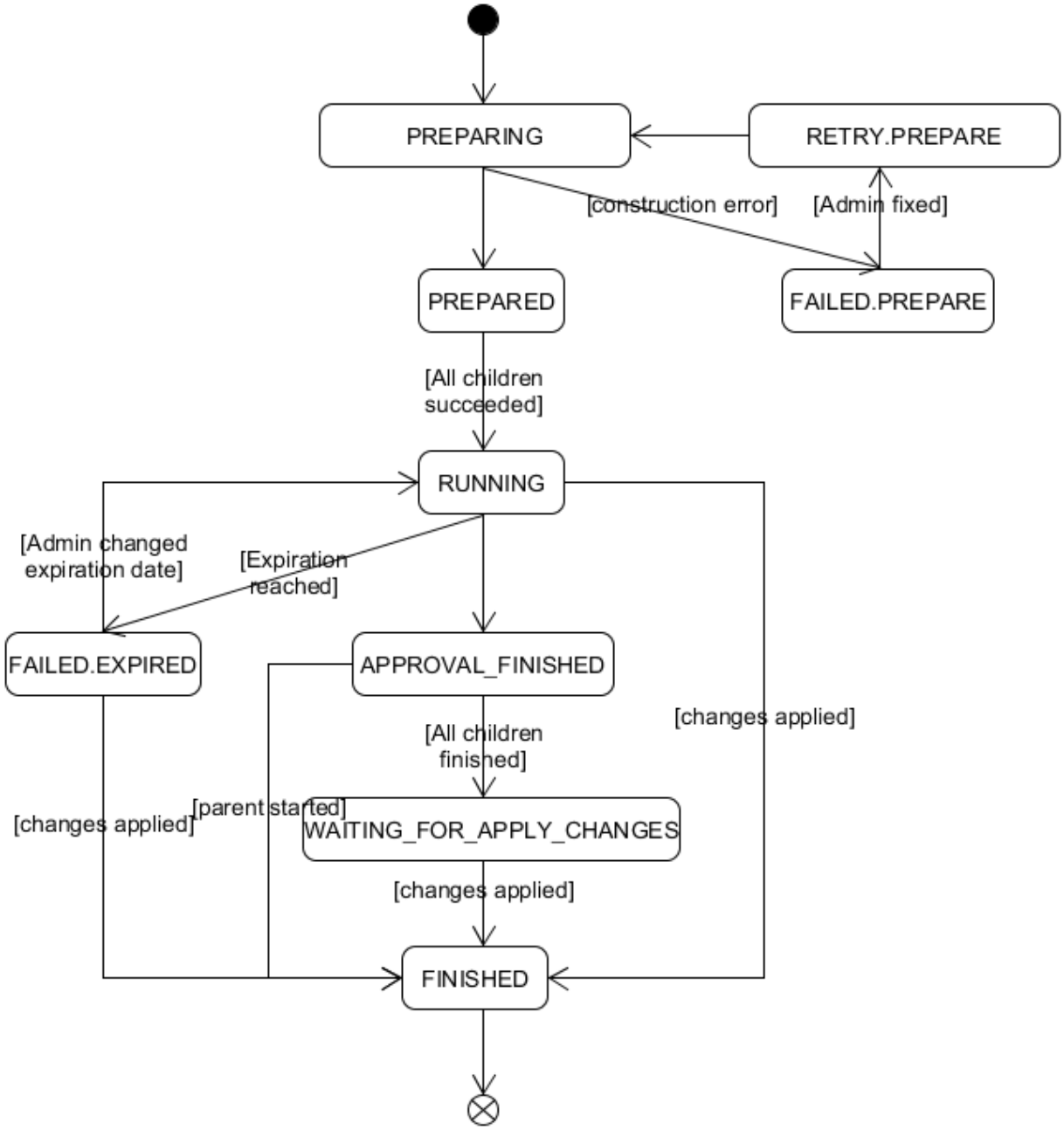


Figure 3. Certification Task States

5.2. Certification Task Workflow

- If a task cannot be created successfully:
  - The state is set to **FAILED.PREPARE**.
  - The administrator can fix issues and set the state to **RETRY.PREPARE**.
  - On the next run, if the task starts successfully, the state becomes **RUNNING**.
- The **PREPARED** state:

    - Temporary state used when multiple approvers require additional tasks.
- When the approver certifies all assignments:
    - The task state changes to **APPROVAL.FINISHED**.
- If the task reaches its **Due Date** without completion:
    - The state changes to **FAILED.EXPIRED**.
    - The administrator can:
    - Extend the due date.
    - Reset the state to **RUNNING** to allow completion.

5.3. Campaign Completion

- When the campaign reaches its **Due Date**:
    - All running tasks are set to **FAILED.EXPIRED**.
    - The workflow applies changes based on the **Apply Changes** field.
    - After changes are applied, certifications move to **FINISHED**.
- Special states:
    - **WAITING.FOR.APPLY.CHANGES**: Transient state (currently unused).
    - If uncertified assignments remain at the due date:
        - The certification moves to **FINISHED**.
    - If changes cannot be applied:
        - The certification moves to **FAILED.APPLY.CHANGES**.

6. Customizing with User Hooks

This section gives hints for customizations with user hook implementations.



if you configure user hooks, you must provide their implementation in Java ***.jar** libraries and deploy them to the relevant IdS-J Server into the sub-folder **confdb/jobs/ CertificationCampaign/lib**.After you complete these tasks, make sure you re-start the IdS-J Server.

6.1. Documentation and Sample Source Code

You can find the Java documentation of the user hooks interfaces and samples on the DirX Identity DVD in the folder:

Documentation/DirXIdentity/CertificationCampaign

The sample source code is delivered in the folder:

Additions/CertificationCampaign

6.2. Select Approvers with a “Find Approvers” User Hook

The **Find Approvers** user hook allows you to change the default approvers for certifications. For a user certification campaign, the default approver is the **manager** of the user, and for a privilege certification campaign, the default approver is the **owner** of the privilege. You can override this default implementation with a custom **Find Approvers** user hook.

The **FindApprovers** user hook must implement the **FindApproversUserhook** Java interface. This interface contains the following methods:

```
void setContext(FindApproversContext context);
```

```
FindApproversContext getContext();
```

```
boolean before();
```

```
boolean after();
```

```
TreeNode<StorageObject> findApprover(FindApproversContext context)
```

```
throws NoApproverException;
```

The method setContext is used by the campaign controller to give the user hook access to some DirX Identity context objects:

`GlobalContext` object – an object that can be used to access different attributes from the IdS-J Server.

`Campaign` object - a `StorageObject` that contains details about the current campaign.

`Certification` object - a `StorageObject` that contains details about the current certification task.

`Subject` object - a `StorageObject` that contains details about the current subject to be certified (for example, in a user certification campaign, it is a `SvcUser` object).

Inside the user hook, you can change objects, especially the Certification (task). For example, you can store the due date in the attribute `dxrEndDate`. If you do so, you must store your changes to LDAP on your own.

6.2.1. Before – Creating the Certification Task in the User Hook

The method `before()` is executed before all controller actions and the method `after()` is called after the controller finishes all the actions for the current certification object. If these methods return false, the default implementation of the Certification Campaign controller will not be executed.

This is especially important for the method `before()`. If it returns false, the controller assumes that the complete certification task object has been created by the user hook. In this case, the user hook must find the approvers, the assignments to certify per subject (user), set the state and the start and end dates, and at the end save the certification task object(s).

6.2.2. After – Changing Default Attributes

When the method `after()` is called, the certification object has already been created and saved to LDAP. The user hook can then override some attribute values and must store its changes in LDAP.

6.2.3. Find Approvers – Adding Additional Approvers

You can also use the **Find Approvers** user hook to add more approval tasks (called **sub-certifications**) for the given subject. In this way, you can even create a hierarchy of approval tasks by several approvers, as illustrated in the following diagram:

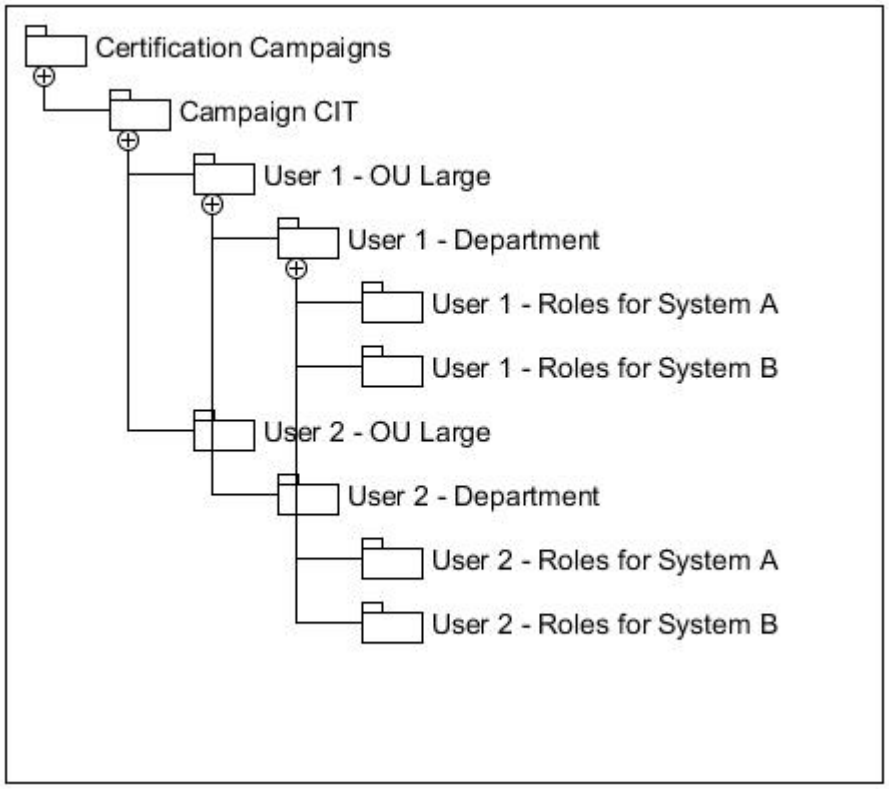


Figure 4. Certification Campaign Sub-certification Structure

To create this kind of structure for a certification, you must implement the following method inside the **FindApprovers** user hook:

`TreeNode`*`StorageObject`* `findApprover(FindApproversContext context)`

This method returns a tree structure of `StorageObjects`. The controller creates a certification object (which reflects one approval task) for each approver in the tree structure. The tasks are processed according the campaign field **Approver Sequence** top down or bottom up. Each sub-certification follows the same states as a normal certification. The last approval task determines the state of the whole subject certification.

6.3. Select Campaign Subjects with a “Find Subjects” User Hook

A user hook that implements the `FindSubjectsUserhook` Java interface can select the subjects. The subjects are users for a user certification and privileges for a privilege certification. This action is helpful when an LDAP filter is not sufficient to identify the subjects of the campaign.

You need to implement the following methods:

```
FindSubjectsContext getContext()
```

```
setContext(FindSubjectsContext context)
```

```
List<String> findSubjects(FindSubjectsContext context)
```

The first two methods have the same functionality as in the **FindApprovers** user hook. You can use setContext method to store the FindSubjectContext object.

The method `findSubjects` is called before the controller starts to find the subjects for the certification campaign. You can use this user hook method to override this functionality and return your own list of subjects (a list of DNs) instead. The `FindSubjectContext` provides access to a `GlobalContext` object and to the `Campaign` object.

6.4. Limit Resources with a “Limit Resources” User Hook

A user hook that implements the `LimitResourcesUserhook` Java interface can reduce the list of resources to be certified. The resources are privileges for a user certification and users for a privilege certification.

The default implementation considers the resource base and filter. If they are empty, all resources must be certified. If they have values, the resources must be descendants of the resource base (including the resource base itself), and their attributes must match the LDAP filter.

If the LDAP filter is not sufficient, you can implement this user hook. If you do, both resource base and resource filter are ignored.

You need to implement the following methods:

```
LimitResourcesContext getContext()
```

```
setContext(LimitResourcesContext context)
```

```
List<String> limitResources(LimitResourcesContext context)
```

The first two methods have the same functionality as the **FindApprovers** user hook. You can use the setContext method to store the LimitResourcesContext object.

The method `limitResources` is called before the controller starts to find resources for the certification campaign. You can use this user hook method to override this functionality and return your own list of resources (a list of DNs) instead. The `LimitResourcesContext`

provides access to the `GlobalContext` object and to the `Campaign` object.

6.5. Send emails with the “Send Email” User Hook

With a user hook that implements the `SendEmailUserhook` Java interface, you can send notifications your own way.

The user hook needs to implement the following methods:

```
SendEmailContext getContext()
```

```
setContext(SendEmailContext context)
```

```
boolean onSendEmail (SendEmailContext context)
```

```
boolean before()
```

```
boolean after()
```

The first two methods have the same functionality as the **FindApprovers** user hook. You can use `setContext` method to store the `SendEmailContext` object.

The `SendEmailContext` method contains the following objects: GlobalContext object, Campaign object, a list with Certification objects used for sending emails, a list with NotificationProcessor objects which can be used to process the email templates.

The `before()` and `after()` methods are called before and after sending emails. If these methods return false, the default implementation from the Certification Controller will not be executed.

With the `onSendEmail` method, you can override the default implementation of the Certification Campaign controller. If you don’t want to execute the default sendEmail implementation, the user hook method must return false.

6.6. Override Campaign Creation with “Campaign Creator”

With an implementation of the `CampaignCreatorUserhook` Java interface, you can completely override the creation of a campaign: find the subjects, select the assignments and calculate the approvers.

You need to implement the following methods:

```
CampaignContext getContext()
```

```
setContext(CampaignContext context)
```

```
boolean before()
```

```
boolean after()
```

The first two methods have the same functionality as the **FindApprovers** user hook. You can use the setContext method to store the CampaignContext object.

The method `before()` is called by the controller when it is ready to start creation of the certifications task entries. If it returns false, the controller skips the default implementation and assumes that all the certification tasks have been created and stored by the user hook.

If the method `after()` returns false, the default implementation is skipped, which means that the notification about the started campaign will not be sent.

7. Generating Certification Campaign Reports

This chapter explains how to create and manage reports for a certification campaign.

7.1. About the Default Report Templates

DirX Identity provides several default report templates that you can use immediately. These are located under:

Provisioning → **Domain Configuration** → **Reports** → **Default** → **CertificationCampaigns**

Each report typically includes the following files (example: **Campaign with all properties**):

- **Campaign with all properties** – Contains:
  - Search base and search filter for the report
  - List of all attributes to retrieve from the LDAP directory
  - Export formats (HTML, XML, RAW for XSLT transformation)
- **Settings for Producer/Selector** – Defines:
  - Search filter
  - Sort criteria
  - Report name
  - Search level
  - Attributes to retrieve from LDAP
- **Settings for Templates** – Includes flags to show or hide fields in the report.
- **Templates** – XSLT markup for the report layout.

7.2. Producing a Report

To generate a report:

1. Right-click on a certification campaign entry.
2. Select **Report**. The **Run report** dialog appears, listing available reports for certification campaigns.
3. Choose a report from the list.

At the bottom of the dialog, you'll find: - A description of the report - Parameter fields

Leave the **search base** and **scope** as they refer to the selected campaign folder. Set **Type** to **HTML** for a user-friendly output format. Optionally: - Change the output file name - Check **Output to viewer** to preview the report in the dialog window.



Select **XML** as the output format if: - You need all available attributes and values - You plan to process the file later with an XML tool

In **HTML** format, some attributes may be hidden during XSLT transformation based on template settings.

7.3. Adding or Extending Reports

You can create a new report by copying and modifying an existing one.

Example: Copy **Campaign with Failed.Prepare certificates** and adjust it to include certifications where applying changes failed (state: **FAILED.APPLY.CHANGES**).

To update the search filter, modify the following line in **Settings for Producer/Selector**:

```
<var name="SearchFilter" value="(|(objectClass=
dxrCertificationCampaign)(&amp;(objectClass=dxrCertificationEntry)(|(
dxrState=FAILED.PREPARE)(dxrState=FINISHED))))"/>
```

to

```
<var name="SearchFilter" value="(|(objectClass=
dxrCertificationCampaign)(&amp;(objectClass=dxrCertificationEntry)(dx
rState=FAILED.APPLY.CHANGES))))"/>
```

This updated report will now include only certification tasks where changes could not be applied successfully.

8. Continuous Certification by Re-Approval

This chapter describes continuous certification via re-approval workflows. This is a highly effective method that distributes load with regard to your employees and your DirX Identity system and that works with maximum efficiency.

8.1. About this Use Case

This use case leverages DirX Identity's re-approval feature. Privileges are flagged for re-approval and - depending on the timing conditions in place - a re-approval workflow is started for a specific assignment before the assignment's end date is reached. The following figure illustrates this use case.

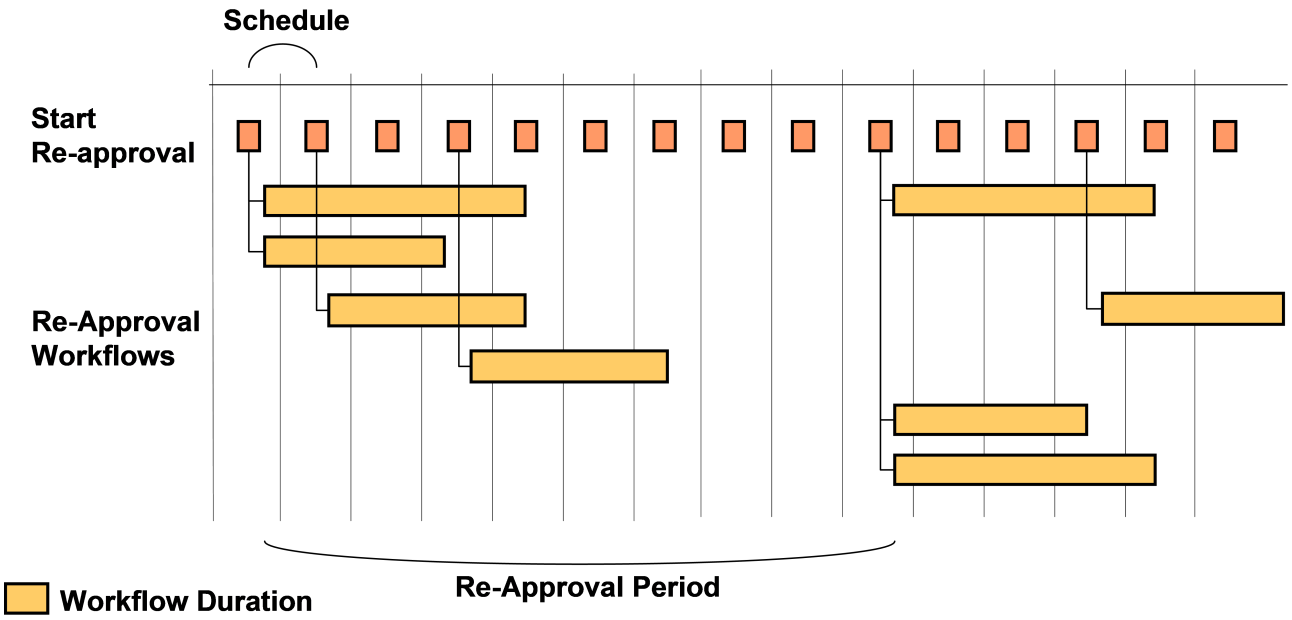


Figure 5. Certification by Re-Approval

The approvers of these workflows receive an e-mail notification informing them that they need to re-approve a specific assignment. They must decide whether the user keeps the privilege or whether the privilege should be removed.

The workflows are continuously started from a daily **StartReapproval** workflow. It checks for privilege assignments that are to be re-approved. To ensure that all assignments are correctly flagged, an **InitializeReapproval** workflow should be started in advance. The following figure illustrates the timing conditions.

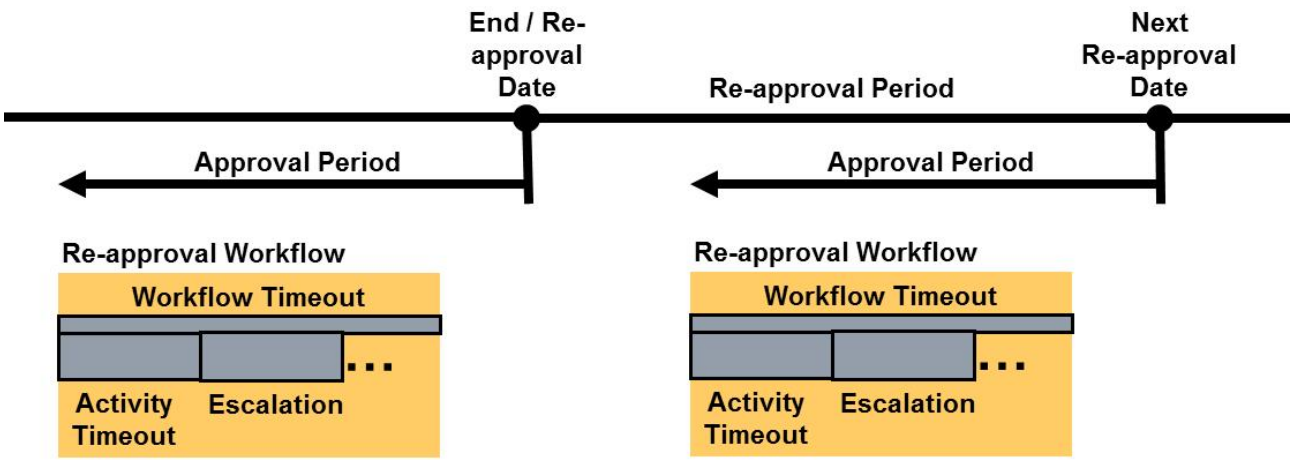


Figure 6. Certification by Re-Approval Timing

In the figure:

- The **end date** at the assignment marks the date where the next re-approval has to be finished. If this date is reached and the approval was accepted, the user keeps the assignment. If the approval was rejected or not performed in time, the user will lose the assignment at this date.
- The next re-approval date is calculated from the current end date of the assignment plus the **Re-approval Period** value at the privilege (if this value is not available, the default value is taken from the domain object) or according to the **Re-approval Date** set at the privilege (if this value is not available, the default value is taken from the domain object). For a complete definition of the calculation, see the section "About Re-Approval" in Chapter 1.
- The timeout of the approval activity is individual per re-approval workflow. When this time expires, one or more escalations can occur depending on the workflow's configuration.
- The workflow has its own individual **Workflow Timeout**, which should be longer than the sum of all possible activity timeouts including the escalations.

8.2. Setup and Configuration

This use case requires you to set up the following items:

- The relevant flags at the domain object
- The re-approval properties at the privileges
- The workflows

8.2.1. Setting up the Domain Object

Select the **Domain Configuration** view, click the top-level node and then select the **Timing** tab.

In the **Approval** area, you should set three attributes that are relevant for re-approval:

- Set the **Approval period** to four (4) weeks to give the approvers enough time to react to the re-approval request. The re-approval workflow is started at the time at which the re-approval must occur minus this approval period value. For example, if the re-approval is set to the end of May and you select a four-week period, the re-approval workflow is started at the beginning of May. If no approval period is configured at the domain, a default value of 14 days applies.
- The **Re-approval period** is a default value that takes effect when the corresponding field at the privilege is not set. Typical values for re-approval periods are from three (3) months to two (2) years. The default value is three months.
- Leave the **Re-approval date** empty because we want to run individual re-approvals for each assignment.

See the online help for an explanation of these settings.

8.2.2. Setting up the Privileges

To keep it simple, we only set up role re-approval here. You can configure privileges of any other type if that is what's required.

For each role requiring re-approval, set its **Requires re-approval** flag and then set the other parameters in one of the following ways:

- If you want to have the same behavior for all privileges, leave the **Re-approval date** and **Re-approval period** empty. In this case, the domain settings apply.
- If you have privileges requiring a shorter or longer re-approval period, set an individual **Re-approval period** at the privilege.

If you want to use different re-approval workflows, set a link to the appropriate one in the **Workflow** field for each role. Make sure that the activity, escalation and workflow timeouts match the approval period.

Perform this procedure for all roles you want to be re-approved.

8.2.3. Setting up the Workflows

You need to set up two workflows for re-approval. The next sections describe these tasks.

8.2.3.1. InitializeReapproval

The **InitializeReapproval** workflow sets for all existing privilege assignments whether an end date is set according to the defined conditions at the privilege and the domain object. You should run this workflow either manually if you changed the re-approval conditions at privileges or - and we recommend this method - regularly each night to be sure that all assignments are correctly set up for re-approval.

Open the workflow's wizard in the **Connectivity** view group:

- In the **Rule Search Parameters** tab, you can see that the **InitializePrivilegeForReapproval** consistency rule is run.

- Check this rule in the **Provisioning** → **Policies** → **Rules** → **Default** → **Consistency Rules** → **Reapproval** folder.
- With the flag **updateConfiguredReapprovalDates** in the **General** tab set, you can automatically shift the re-approval dates configured at the domain and/or the considered privileges: If the re-approval date lies in the past or within the approval period, it is shifted to the future by one re-approval period.
- Check the filter conditions in the **Filter** tab. The **Search Base** works on the entire domain and the **Search Filter** is set to:
  (dxrneedsreapproval="TRUE" and (objectclass="dxrRole" or objectclass="dxrPermission" or objectclass="dxrTargetSystemGroup"))
  which means that it searches for all privileges that have the dxrNeedsReapproval flag set.

Don't forget to copy the rule and the workflow to be sure that changes in the configuration remain and are not overwritten by the next DirX Identity product update.

This workflow is set up as a separate workflow that runs exactly one consistency rule (InitializePrivilegeForReapproval). Alternatively, you can run it together with other consistency rules in your custom policy execution workflow.

8.2.3.2. StartReapproval

The **StartReapproval** workflow should run regularly once per night. It checks whether an assignment is flagged for re-approval and whether the end date is reached. Then it starts the defined re-approval workflow.

Open the workflow's wizard in the **Connectivity** view group:

- In the **Rule Search Parameters** tab, you can see that the **StartWorkflowsForReapproval** consistency rule is run.
- Check this rule in the **Provisioning** → **Policies** → **Rules** → **Default** → **Consistency Rules** → **Reapproval** folder.
- Check the filter conditions in the **Filter** tab. The **Search Base** works on the entire domain and the **Search Filter** is set to:
  (objectclass="dxrAssignment" and dxrneedsreapproval="TRUE"
  and dxrEndDate<="$(approvaldate)" and (not (dxrInApproval="TRUE") or not (dxrInApproval=*)))
  which means that it searches for all assignments that have the dxrNeedsReapproval flag set and whose end date is reached and where no re-approval workflow is already started.

Don't forget to copy the rule and the workflow to be sure that changes in the configuration remain and are not overwritten by the next product update.

This workflow is set up as a separate workflow that runs exactly one consistency rule (StartWorkflowsForReapproval). Alternatively, you can run it together with other consistency rules in your custom policy execution workflow.

8.3. Running the Use Case

Run the configured workflows once and test the result. Be sure to run it in the sequence InitializeReapproval and then StartReapprovalWorkflows.

After completion, view the status of the workflow and check the trace file for errors.

View the Request Workflow Monitor area to check for the newly created re-approval workflows.

If everything works well, set up schedules to run the two workflows regularly. We recommend that you set up a hierarchical workflow that defines the sequence and then run this workflow daily.

8.4. Alternative or Extended Configurations

This section gives hints for alternative or extended configuration of the use case.

8.4.1. Certifying Other Privilege Types

In this use case, we ran re-approval only on roles. If required, you also can run re-approval at permission or group level. Set the corresponding flags and fields at these types of privileges.

You can also run a certification on any mix of roles, permissions and groups.

8.4.2. Running Separate Campaigns

If you need to re-approve a huge number of privileges, think about setting up multiple workflows. Set up InitializeReapproval and StartReapprovalWorkflows workflows with different search bases and filters.

If you run separate campaigns, set up the corresponding schedules and try to distribute the load.

9. Scheduled Certification by Re-Approval

This chapter describes scheduled certification via re-approval workflows. This is another use case that you can perform using the re-approval feature.

9.1. About this Use Case

This use case leverages DirX Identity's re-approval feature. Privileges are flagged for re-approval and a central definition of the re-approval date is set at the domain. This use case defines a campaign that is run at the same time for all flagged privileges. The following figure illustrates this use case.

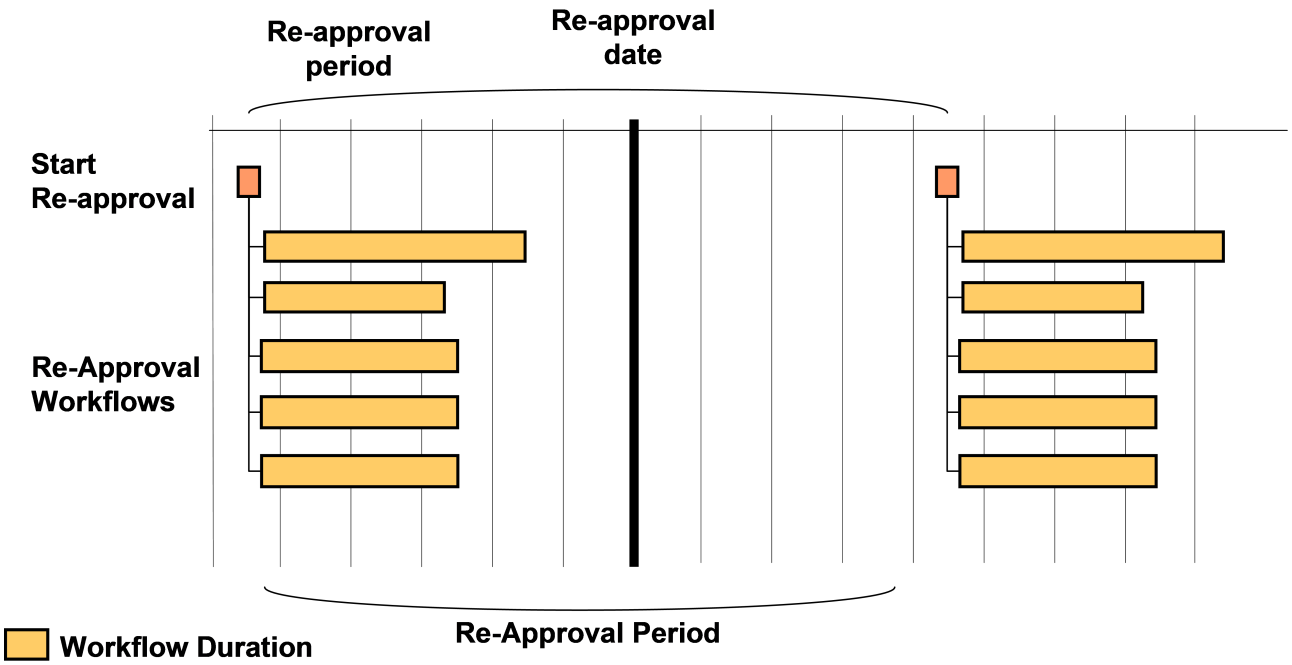


Figure 7. Certification by Re-Approval

The approvers of these workflows receive an e-mail notification informing them that they must re-approve a specific assignment. They need to decide whether the user keeps the privilege or whether the privilege shall be removed.

It makes sense to run the **InitializeReapproval** and **StartReapproval** workflows only once at the beginning of the campaign. This action starts all necessary workflows for the campaign. The following figure shows the timing condition details.

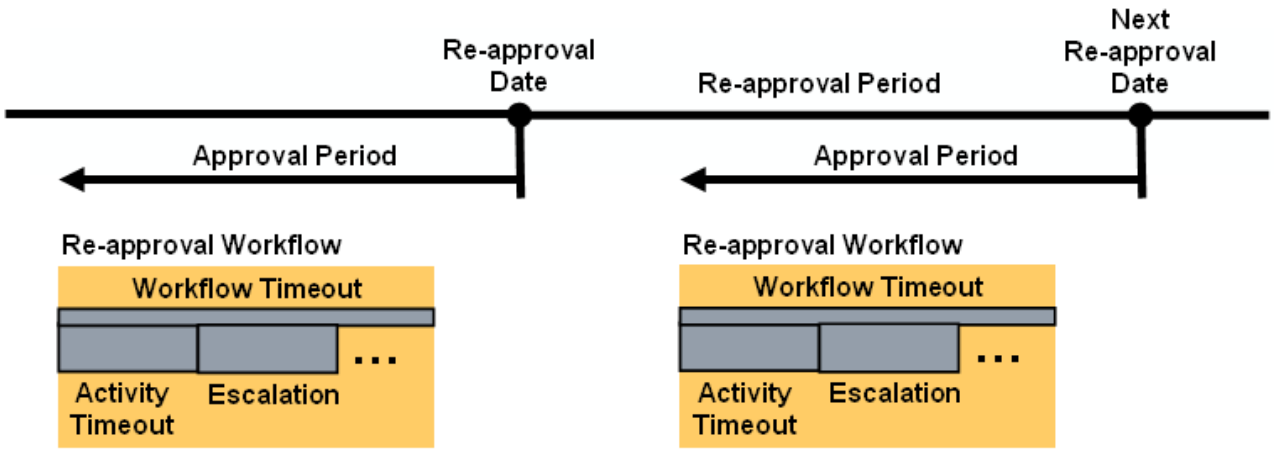


Figure 8. Certification by Re-Approval Timing

In the figure:

- We assume that the **Re-approval Date** is only set at the domain, which means that it is used for all privilege assignments that are flagged for re-approval. All re-approval workflows are started at the same time. If the re-approval date is reached and the approval was accepted, the user keeps the assignment. If the approval was rejected or no decision was taken, the user will lose the assignment on this date.
- We also assume that there is only a central definition of the **Re-approval Period** at the domain. This setting defines when the next re-approval will take place.
- The timeout of the approval activity is individual per re-approval workflow. When this time expires, one or more escalations can occur depending on the workflow's configuration.
- The workflow has its own individual **Workflow Timeout**, which should be longer than the sum of all possible activity timeouts including the escalations.

9.2. Setup and Configuration

This use case requires you to set up the following items:

- The relevant flags at the domain object
- The re-approval properties at the privileges
- The workflows

9.2.1. Setting up the Domain Object

Select the **Domain Configuration** view, click the top level node and then select the **Timing** tab.

In the **Approval** area, you should set three attributes that are relevant for re-approval:

- The **Reapproval period** is a default value that takes effect when the corresponding fields at the privileges are not set. Because we want to run a certification campaign each year, we set the value to one (1) year. The default value is 3 months.

- Because we intend to run a campaign at the end of March each year, we set the **Reapproval date** to the 31st of March.
- Set the **Approval period** to four (4) weeks to give the approvers enough time to react to the re-approval request. The re-approval workflow is started at the time when the re-approval must occur minus this approval period value. If the re-approval is set to end of March and you select a four-week period, the re-approval workflow is started at the beginning of March. If no approval period is configured at the domain, a default value of 14 days applies.

See the online help for an explanation of these settings.

9.2.2. Setting up the Privileges

To keep it simple, we only set up role re-approval for this use case. You can configure privileges of any other type if that is required.

Perform these steps for each privilege that requires re-approval:

- Set the **Requires re-approval** flag.
- If you want to use different re-approval workflows, set a link to the appropriate one in the **Workflow** field for each role. Make sure that the activity, escalation and workflow timeouts match the approval period.

All other parameters are taken from the domain definitions, so no further configuration is necessary.

9.2.3. Setting up the Workflows

You need to set up two workflows for re-approval. The next sections describe these tasks.

9.2.3.1. InitializeReapproval

The **InitializeReapproval** workflow sets, for all existing privilege assignments, whether an end date is set according to the defined conditions at the privilege and the domain object. You should run this workflow only once - in this case, at the beginning of March. The start of the workflow must be slightly later than the intended **Re-approval date** (31st of March) minus the **Approval period** (4 weeks).

Open the workflow's wizard in the **Connectivity** view group:

- In the **Rule Search Parameters** tab, you can see that the **InitializePrivilegeForReapproval** consistency rule is run.
- Check this rule in the **Provisioning** → **Policies** → **Rules** → **Default** → **Consistency Rules** → **Reapproval** folder. You can use the flag **updateConfiguredReapprovalDates** in the **General** tab to automatically shift the re-approval date that is configured at the domain: If the re-approval date lies in the past or within the approval period, it is shifted to the future by one re-approval period.
- Check the filter conditions in the **Filter** tab. The **Search Base** works on the entire domain and the **Search Filter** is set to:

(dxrneedsreapproval="TRUE" and (objectclass="dxrRole" or objectclass="dxrPermission" or objectclass="dxrTargetSystemGroup"))
which means that it searches for all privileges that have the dxrNeedsReapproval flag set.

We recommend that you copy the rule and the workflow to be sure that changes in the configuration remain and are not overwritten by the next product update.

This workflow is set up as a separate workflow that runs exactly one consistency rule (InitializePrivilegeForReapproval). You could also run it with other consistency rules in your custom policy execution workflow.

9.2.3.2. StartReapproval

The **StartReapproval** workflow should run only once directly after the **InitializeReapproval** workflow. It checks whether an assignment is flagged for re-approval and whether the end date is reached, which is the case for all assignments. Then it starts the defined re-approval workflow.

Open the workflow's wizard in the **Connectivity** view group:

- In the **Rule Search Parameters** tab, you can see that the **StartWorkflowsForReapproval** consistency rule is run.
- Check this rule in the **Provisioning** → **Policies** → **Rules** → **Default** → **Consistency Rules** → **Reapproval** folder. You can use the flag **updateConfiguredReapprovalDates** to automatically shift the re-approval date configured at the domain and/or the considered privileges: If the re-approval date lies in the past or within the approval period, it is shifted to the future by one re-approval period. Because the **InitializeReapproval** workflow performed this task, there is nothing to do.
- Check the filter conditions in the **Filter** tab. The **Search Base** works on the entire domain and the **Search Filter** is set to:
(objectclass="dxrAssignment" and dxrneedsreapproval="TRUE"
and dxrEndDate<="$(approvaldate)" and (not (dxrInApproval="TRUE") or not (dxrInApproval=*)))
which means that it searches for all assignments that have the dxrNeedsReapproval flag set and whose end date is reached and where no re-approval workflow is already started.

We recommend that you copy the rule and the workflow to be sure that changes in the configuration remain and are not overwritten by the next product update.

This workflow is set up as a separate workflow that runs exactly one consistency rule (InitializePrivilegeForReapproval). You could also run it with other consistency rules in your custom policy execution workflow.

9.3. Running the Use Case

Run the configured workflows at the defined time (the beginning of March). Be sure to run **InitializeReapproval** first and then run **StartReapprovalWorkflows**. We recommend that

you build a combined workflow that starts both workflows in sequence.

After completion, view the status of the workflow and check the trace file for errors.

View the Request Workflow Monitor area to check for the newly created re-approval workflows.

9.4. Alternative or Extended Configurations

This section gives hints for alternative or extended configuration of this use case.

9.4.1. Certifying Other Privilege Types

In this use case, we only ran re-approval on roles. If required, you can also run re-approval at the permission or group level. Set the corresponding flags and fields at these types of privileges.

You can also run a certification on any mix of roles, permissions and groups.

9.4.2. Setting Individual Re-Approval Dates or Re-Approval Periods

You may want to set individual timing parameters for specific privileges.

For example, you could set an individual re-approval date at the privilege to perform re-approval on that date. If you do not set the re-approval period at the privilege, the value from the domain is taken.

You could also set the re-approval period at the privilege to enforce shorter or longer re-approval cycles.

If you set these individual parameters at the privileges, you should schedule the **InitializeReapproval** and **StartReapproval** workflows to run on a daily basis. Otherwise, re-approvals are set but the execution is not performed.

10. Hints and Limitations

This chapter provides additional information on certification campaigns.

10.1. Manual and Automatic Assignments

Certification campaigns require certification only for **manually assigned privileges**.

Rejecting an automatically assigned privilege does not make sense because it will be reassigned as soon as:

- Provisioning rules for the user are evaluated, or
- The user's access rights are recalculated for any reason.

Therefore:

- **Rule-based assignments** and **assignments inherited from business objects** cannot be certified.

However, approvers may still want to **view all assignments**, including automatically assigned ones, during certification. For this reason:

- These assignments are displayed in the Web Center or Business User Interface.
- Approvers cannot revoke them directly, but they can **flag them for revocation** and provide a reason.
- This feedback is stored with the certification task and can be:
- Reviewed by the certification administrator.
- Included in reports.
- Used to inform relevant stakeholders for improving automatic rules.

10.2. Certification Campaign Logging

To obtain detailed information about campaign execution, increase the log level:

1. Open the **Admin** Web application.
2. Navigate to: **Java Server → Logging → Set log levels**.
3. Add the following Java packages and set their value to ALL for full debug information:
    - `com.siemens.idm.jobs.campaign`
    - `com.dirxcloud.dxi.campaign`

10.3. Java-based Server Workflow Load

Starting and finishing a certification campaign involves multiple tasks:

- **At the start**:
  - Create a certification task for each subject.
  - Identify approvers.
  - Send start notifications.
- **At the end**:
  - Check all certification tasks.
  - Identify revoked assignments.
  - Recalculate access rights for affected users.
  - Send notifications.

These operations generate **high load** on the IdS-J Server and may impact parallel provisioning or approval processes.

Recommendations:

- Allocate sufficient CPU and memory to affected servers.
- Reduce other processes during these periods.

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

Eviden is a registered trademark © Copyright 2026, Atos Group – All rights reserved.

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.