EVIDEN

Identity and Access Management

DirX Directory

Corporate Directory

Version 9.1, Edition June 2025

Table of Contents

Preface

This manual provides information on how to configure and use the Corporate Directory Web Application.

DirX Directory Documentation Set

DirX Directory provides a powerful set of documentation that helps you configure your directory server and its applications.

The DirX Directory document set consists of the following manuals:

- *DirX Directory Introduction*. Use this book to obtain a description of the concepts of DirX Directory.
- *DirX Directory Administration Guide*. Use this book to understand the basic DirX Directory administration tasks and how to perform them with the DirX Directory administration tools.
- *DirX Directory Administration Reference*. Use this book to obtain reference information about DirX Directory administration tools and their command syntax, configuration files, environment variables and file locations of the DirX Directory installation.
- *DirX Directory Syntaxes and Attributes*. Use this book to obtain reference information about DirX Directory syntaxes and attributes.
- *DirX Directory LDAP Extended Operations*. Use this book to obtain reference information about DirX Directory LDAP Extended Operations.
- *DirX Directory External Authentication*. Use this book to obtain reference information about external authentication.
- *DirX Directory Supervisor*. Use this book to obtain reference information about the DirX Directory supervisor.
- *DirX Directory Plugins for Nagios*. Use this book to obtain reference information about DirX Directory plugins for Nagios.
- *DirX Directory Disc Dimensioning Guide*. Use this book to understand how to calculate and organize necessary disc space for initial database configuration and enhancing existing configurations.
- *DirX Directory Guide for CSP Administrators*. Use this book to obtain information about installing, configuring and managing DirX Directory in the context of a Certificate Provisioning Service operating in accordance with regulations like the German “Signaturgesetz”.
- *DirX Directory Release Notes*. Use this book to install DirX Directory and to understand the features and limitations of the current release.

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.

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*.

1. Overview

DirX Corporate Directory is a web application built on the Angular platform. It is intended to showcase the use of the DirX Directory REST API service. This application is provided "as is" with the possibility to make minor adjustments (e.g., exchanging logo, colors, ...) and with configuration options (e.g., fields, filters, ...) for straightforward deployments. Future development may add new functionality to the application.

The DirX Corporate Directory application requires installation of DirX Directory 9.0.

The DirX Corporate Directory application provides:

- A configurable form for searching records in DirX Directory.
- A configurable table that shows a result of an LDAP search.
- A configurable detail view of an entry found by an LDAP search.

1.1. Third-party Software

The list of third-party software used in DirX Corporate Directory application, including the respective licenses and versions can be found in these files:

- *installation_media*/**Installation/***os_platform*/**DirXDirectory/DirXCorporateDirectory/dirx-corporate-directory-1.0.***build_number***.zip/spa/3rdpartylicenses.txt**
- *installation_media*/**Installation/***os_platform*/**DirXDirectory/DirXCorporateDirectory/dirx-corporate-directory-1.0.***build_number***.zip/spa/plugins/dirx-corporate-directory/3rdpartylicenses.txt**

2. Installation and Deployment

2.1. Requirements

The DirX Corporate Directory application is built on the Angular technology. It is a JavaScript client that consumes the DirX Directory REST API service. The application must be deployed in a web container and must have access to a DirX Directory REST server.

2.1.1. Supported Internet Browsers

The DirX Corporate Directory application supports these types of internet browsers:

- Microsoft Edge 117.0 (64-bit) or newer
- Mozilla Firefox 115.3.1esr (64-bit) or newer
- Google Chrome 117.0 (64-bit) or newer

2.1.2. Supported Web Servers

The DirX Corporate Directory application requires a customer-supplied installation of Apache Tomcat running with Java SE 11 or later:

- Apache Tomcat 10.1 version 10.1.7 or higher

Use an installed Java SE 11 (or later) version with the latest security patches installed. It is the customer's responsibility to download and install available Java SE security patches in a timely manner.

Although the application does not depend on a specific web server and can be deployed on a different one, we strictly recommend using Apache Tomcat 10.1, which is used for testing. Using a web server that is different from the recommended one may cause unexpected problems to occur in the application.

We strongly recommend that you run the DirX Corporate Directory application via the **HTTPS** protocol. See the **Tomcat** documentation for details; for example, Apache Tomcat 10.1 (10.1.15) - SSL/TLS Configuration How-To

2.1.3. DirX Directory

The DirX Corporate Directory application is a consumer of the DirX Directory REST API and therefore requires installation of:

- DirX Directory 9.0

2.2. Installation / De-Installation

2.2.1. Installation

To deploy the application into Apache Tomcat, decompress the entire content of the file *installation_media*/**Installation/***os_platform*/**DirXDirectory/DirXCorporateDirectory/dirx-corporate-directory-1.0.***build_number***.zip**|**.tar.gz** (**.tar.gz** when *os_platform* is Linux) into the directory *tomcat_home*/**webapps**.

The application's root folder is then *tomcat_home*/**webapps/spa**.

The application is accessible through an internet browser on the URL: *tomcat_URL*/**spa/**

The default context path of the application is **/spa** and equals the name of the root folder of the application's installation.
If an application is already deployed with this context path in the Tomcat server, or you want to deploy the application with a different context path, you can rename the root folder of the decompressed content (**spa**). In this case, the application is then accessible through an internet browser on the URL: *tomcat_URL/renamed_folder/*

To avoid an **'HTTP Status 404 – Not Found'** error in the application when using the refresh button, configure the rewrite valve in the Apache Tomcat installation.

Edit the file *tomcat_home*/**conf/context.xml** and add this line into the **Context** xml tag:

<Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />

The rewrite valve implements the URL rewrite functionality in a way that is very similar to mod_rewrite from the Apache HTTP Server.

2.2.2. De-Installation

To uninstall the software, remove the directory *tomcat_home*/**webapps/spa**.

2.2.2.1. RewriteValve

If RewriteValve was configured during the installation, edit the file *tomcat_home* /**conf/context.xml** and remove this line in the **Context** xml tag:

```
<Valve className="org.apache.catalina.valves.rewrite.RewriteValve"/>
```

2.3. Upgrade

Each patch version is released as a complete installation bundle. For the DirX Corporate Directory application, all patches are cumulative: each patch contains all the changes made by previous patches. This means you don't need to download any other patches to achieve

the latest state of the application.

To upgrade your current version, please follow these steps:

1. Back up the application root folder.
   This folder may differ depending on your installation process. For the default installation, it is *tomcat_home*/**webapps/spa**.
   If you deployed the application in a different application context path by renaming the **spa** folder, please back up the renamed folder.
2. Uninstall the current version. See the “De-Installation” part in the “Installation / De-installation” section in this document.
3. Install the new patch version. See the “Installation” part in the “Installation / De-installation” section in this document.
4. Update the configuration in the new installed version. See the “Configuration“ section in this document.

2.4. Configuration

The application’s default configuration must be updated for it to function properly. At a minimum, the path attribute in the rest configuration object and the baseObject attribute in the searchParameters configuration object must be provided. Check the Attributes configuration and attribute references whether they met your directory configuration. For details, see the ch3_configuration.pdf chapter.

After each configuration update, clear the browser’s cache to get the application’s current configuration.

3. Configuration

This section describes how to configure the DirX Corporate Directory web application.The application's configuration is stored in the file *install_path***/plugins/dirx-corporate-directory/assets/config/app-config.json**.

For example, *install_path* looks like C:\Program Files\Apache Software Foundation\Tomcat 10.1\webapps\spa.

The configuration file is in human-readable form in JSON format and contains these parts:

- REST API Configuration
- Bind Operation
- Search Request Parameters
- Attributes
- Quick search
- Extended search
- Expert search
- Person details
- Results table
- Logger
- Translations
- Page Headers and Paragraphs

3.1. REST API Configuration

REST API configuration is stored in the root of the configuration object in the "rest" JSON attribute.This configuration object defines the DirX Directory server instance with which the application communicates.

This configuration object contains these configurable attributes:

- **path** (string, required) - Defines a URL to the DirX Directory REST API.

```
"rest": {
    "path": "https://dxd-hostName:dxd-rest-server-port/dxd/ldap/v1"
}
```

We strictly recommend using the **HTTPS** protocol instead of HTTP for security reasons.If HTTPS is used, please ensure that the user's browser trusts the certificate that is used by the DirX Directory REST API.

> Default ports for the DirX Directory REST API are: **HTTP**: 8880, **HTTPS:** 8443

3.2. Bind Operation

Bind operation configuration is stored in the root of the configuration object in the “bind” JSON attribute.This attribute is optional.This configuration object defines bind operation-specific settings.If it is not defined, the application takes into account its default value with the “anonymous“ attribute set to **false**.

```
"bind": {
    "anonymous": false
}
```

The configuration object contains the following attributes:

- **anonymous** (boolean, required) - Whether the application UI supports login as an anonymous user.Supported values are **true** and **false**.When **true**, ** the login page contains a button for login as an anonymous user (there is no need to provide a username and password).

3.3. Search Request Parameters

Search request parameters configuration is stored in the root of the configuration object in the “searchParameters” ** JSON attribute.This object defines the base parameter values used for LDAP searches in the application.

This configuration object contains the following attributes:

- **baseObject** (string, required) - The name of the base object entry (or possibly the root) relative to where the search is to be performed.There is no default value and baseObject must be configured.See RFC 4511, section 4.5.1.1.
- **scope** (string, required) - The scope of the search to be performed.Supported values are **base** (base object), **one** (single level) and **sub** (whole subtree).For details, see RFC 4511, section 4.5.1.2.
- **sizeLimit** (string, optional) - The maximum number of entries to be returned as a result of the search.A value of zero in this field indicates that there are no client-requested size limit restrictions in effect for the search.The default value is 0. Servers may also enforce a maximum number of entries to return.See RFC 4511, section 4.5.1.4.
- **timeLimit** (string, optional) - The maximum time (in seconds) allowed for a search.A value of zero in this field indicates that there are no client-requested time limit restrictions in effect for the search.Servers may also enforce a maximum time limit for the search.The default value is 60. See RFC 4511, section 4.5.1.5.
- **filter** (string, required) - The conditions that must be satisfied for the search to match a given entry.This attribute usually corresponds to the “{{searchFilter}}” constructed in the application but can also extend it; for example, to restrict the search result only to

entries of the given object class.The default value is “(&(objectClass=person){{searchFilter}})”.For details, see RFC 4511, section 4.5.1.7.

```
"searchParameters": {
    "baseObject": "o=My-Company",
    "scope": "sub",
    "sizeLimit": "0",
    "timeLimit": "60",
    "filter": "(&(objectClass=person){{searchFilter}})"
}
```

3.4. Attributes

Attributes configuration is stored in the root of the configuration object in the “attributes” JSON attribute.This object defines the set of all LDAP attributes that the application can use.Attributes are referenced by application components like quick search, expert search, extended search, and so on.

The value is an array of attribute configuration objects.

3.5. Quick search

The Quick search component’s configuration is stored in the root of the configuration object in the “quickSearch“ JSON attribute.The Quick search component’s functionality depends on this configuration.

This configuration object contains this attribute:

- **ldapName** (string, required) - A reference to the attribute configuration object with the same ldapName.This attribute is used in the quick search.

3.6. Extended search

The Extended search view’s configuration is stored in the root of the configuration object in the “extendedSearch“ JSON attribute.The Extended search form component’s functionality and view depend on this configuration object.The view consists of groups that have defined attributes according to which user can perform the search.

This configuration object contains this attribute:

- **groups** (array, required) - A list of group configuration objects.The search filter view is divided into these groups.

3.7. Expert search

The Expert search view's configuration is stored in the root of the configuration object in the "expertSearch" JSON attribute.The Expert search form component's functionality and view depend on this configuration object.The view consists of groups that have defined attributes according to which user can perform the search.

This configuration object contains this attribute:

- **groups** (array, required) - A list of group configuration objects.The search filter view is divided into these groups.

3.8. Person details

The Person details view's configuration is stored in the root of the configuration object in the "detail" JSON attribute.The Person details view's functionality and view depend on this configuration object.The view consists of groups that have defined attributes which values are presented in the detail.

This configuration object contains this attribute:

- **groups** (array, required) - A list of group configuration objects.The search view is divided into these groups.

3.9. Results table

The Results table view configuration is stored in the root of the configuration object in the "result" JSON attribute.The Results table component's functionality and view depend on this configuration object.The configuration object contains attributes that define the columns in the result table.

This configuration object contains these configuration attributes:

- **tableParameters** (TableParameters, required) - A result table parameters object.
- **attributes** (array, required) - An array of Attribute link objects.

3.10. Table of Configuration Objects

3.10.1. Attribute Configuration Object

Each attribute configuration object contains these attributes:

- **displayName** (string, required) - The attribute's internationalization (i18n) translation key.
- **ldapName** (string, required) - The attribute's name in the LDAP directory.
- **type** (string, required) - The attribute type.Supported types are: **string**, **number**, **boolean**, **datetime**.

- **defaultSearchOperator** (string, required) - The default search operator pre-set in the user interface extended / expert search view.The supported value is one of: **begins**, **ends**, **contains**, **equals**.
- **searchOperators** (array, required) - The set of supported search operations.Supported values are: **begins**, **ends**, **contains**, **equals**.Configured values are used in a selection list in an Expert search view.
- **sortable** (boolean, required) - Whether results in the table are sortable according to this attribute.Supported values are **true** and **false**.
- **editable** (boolean, required) - Whether this attribute is editable in the Person details view.Supported values are **true** and **false**.
- **multiValued** (boolean, required) - Whether this attribute can be multi-valued. Supported values are **true** and **false**.
- **validationParameters** (object, optional) - The validation parameters configuration object used in expert / extended search and in edit mode for input values validation.

Be careful when configuring the **sortable** attribute. *** The sorting feature requires corresponding indexes to be created in the DirX Directory server. The attribute's *INITIAL** index is required. The **PRESENT** index is recommended for reverse sorting. Missing indexes may cause performance issues.

Be careful when configuring the **defaultSearchOperator** and **searchOperators** attributes. Missing corresponding indexes in a DirX Directory server instance can cause performance issues.

```
"attributes": [
    {
        "displayName": "SURNAME",
        "ldapName": "sn",
        "type": "string",
        "defaultSearchOperator": "begins",
        "searchOperators": [
            "begins"
            "ends",
            "contains",
            "equals"
        ],
        "sortable": true,
        "editable": false,
        "multiValued": true,
        "validationParameters": {
            "minLength": 1
```

```
        }
    },
    ...
]
```

3.10.2. Validation Parameters Configuration Object

The validation parameters configuration object contains these configurable attributes:

- **maxLength** (number, optional) - The maximum length of the attribute value.
- **minLength** (number, optional) - The minimum length of the attribute value.
- **pattern** (string, optional) - A regular expression condition which must be met.

3.10.3. Group Configuration Object

The group configuration object contains these configurable attributes:

- **displayName** (string, required) - The group's i18n translation key.
- **attributes** (array, required) - An array of attribute links.

```
{
    "displayName": "PERSONAL_INFO",
    "attributes": [
        {
            "ldapName": "sn"
        },
        {
            "ldapName": "givenName"
        }
    ]
}
```

3.10.4. Attribute Link Object

The attribute link configuration object consists of these attributes:

- **ldapName** (string, required) - A reference to an attribute configuration with the same **ldapName**.This attribute is used in the quick search.
- **link** (Link, optional) - A link configuration object.

```
{
    "ldapName": "sn",
```

```
    "link": {
        "actionType": "paramSearch"
    }
}
```

3.10.5. Link Configuration Object

The link configuration object consists of these attributes:

- **actionType** (string, required) - The operation to be performed when a user clicks on the link.Supported values are: **paramSearch** (sets the attribute value in the search form and performs a search), **dnSearch** (searches for entry with this dn and shows it in the detail view).

```
{
    "actionType": "paramSearch"
}
```

To display a working link, the attribute must support the **equals** operator.
See the section on Attributes configuration.

3.10.6. Table Parameters Configuration Object

The table parameters configuration object consists of these attributes:

- **pageSize** (string, required) - The count of entries shown in the results table on one page.
- **sort** (Sort configuration object, required) - The sort properties of the result table.

```
"tableParameters": {
    "pageSize": "15",
    "sort": {
        "ldapName": "sn",
        "dir": "asc"
    }
}
```

3.10.7. Sort Configuration Object

The sort configuration object consists of these attributes:

- **ldapName** (string, required) - The LDAP attribute by which the result table is sorted.
- **dir** (enum, required) - The sort direction of the column that represents the attribute specified in **ldapName**.Supported values: **asc**, **desc**.

3.11. Logger

Each Logger configuration object contains these attributes:

- **level** (enum, required) - The NGXLogger level.Valid values are **0** for TRACE, **1** for DEBUG, **2** for INFO, **3** for LOG, **4** for WARN, **5** for ERROR, **6** for FATAL and **7** for OFF.For details, see the NGXLogger documentation:
  - https://github.com/dbfannin/ngx-logger/blob/ac55c7257dddcba376342c880141df4bae49e9a4/docs/config.md
  - https://github.com/dbfannin/ngx-logger/blob/ac55c7257dddcba376342c880141df4bae49e9a4/src/lib/types/logger-level.enum.ts

```
"logger": {
    "level": 4
}
```

The default logger level in the application is set to level **4** (**WARN**).This setting can be overridden in the configuration by defining the desired value for the “level“ attribute.

3.12. Translations

The application supports two languages:

- **English**
- **German**

The client application contains translation files in JSON structure for both languages.These files are stored in the folder:

*install_path***/plugins/dirx-corporate-directory/assets/i18n/** :

- **en.json** - English translation file.
- **de.json** - German translations file.

If you are adding a new attribute into the configuration, the **displayName** value is taken as a translation key whose value is searched in the translation file.If this translation key is not found in the translation file, the value of the attribute is shown in the user interface.Translation keys for attributes are stored in the

DXD_CD.CONFIGURATION.ATTRIBUTES object in the corresponding translation file.

For example: for the **surname** attribute, the **displayName** attribute is set to “SURNAME”.This key is searched in the translation file.For the English language, the “Surname” value is returned and the “Nachname“ value is returned for the German language.

3.13. Page Headers and Paragraphs

Each view in the application contains its header text and paragraphs describing the basic functionality of the view.

It is possible to update the text of these headers and paragraphs directly in the translation files.

Translation keys for headers are stored in the **DXD_CD.PAGE.HEADERS** object in the corresponding translation file.

Possible keys are:

- **EXTENDED_SEARCH** - The header for the Extended search view.
- **EXPERT_SEARCH** - The header for the Expert search view.
- **RESULT_DETAIL** - The header for the Person details view.
- **RESULT_TABLE** - The header for the Results table view.

Translation keys for paragraphs are stored in the **DXD_CD.PAGE.PARAGRAPHS** object in the corresponding translation file.

Possible keys are:

- **EXTENDED_SEARCH** - The header for the Extended search view.
- **EXPERT_SEARCH** - The header for the Expert search view.
- **RESULT_DETAIL** - The header for the Person details view.
- **RESULT_TABLE** - The header for the Results table view.

It is possible to avoid showing view headers and paragraphs. In this case, please delete the **DXD_CD.PAGE** object in the corresponding translation file.

4. User Interface

The DirX Corporate Directory application provides a user interface implemented by the Angular web framework.

The interface consists of these components:

- Core Functionality
- Login view
- Quick search
- Extended search
- Expert search
- Results table
- Person details

Some of the functionality and look and feel of these components is configurable.

4.1. Core Functionality

Core functionality is provided by the DirX Core application, which can provide access to multiple applications from the DirX product suite.For the DirX Corporate Directory application, the DirX Core application is preconfigured and provides access only to this application.

The core functionality consists of:

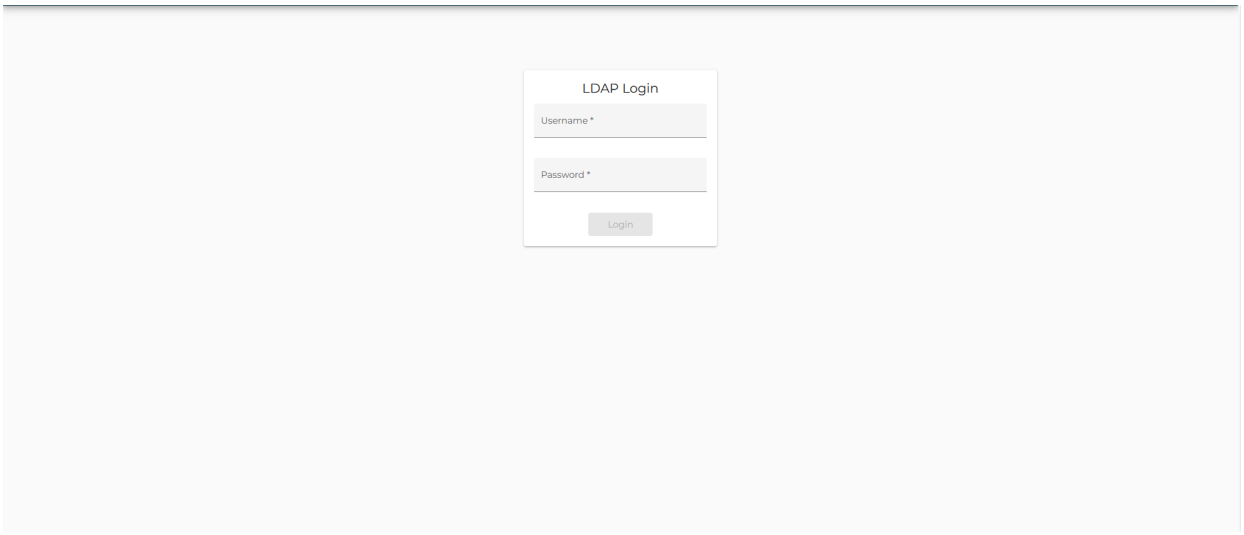
- **Hamburger button** - Click on the hamburger button located in the upper left corner to activate the menu.

  The activated menu contains the following actions:

  - **Applications** - Navigates to the home page.
  - **Settings** - Navigates to the settings view where the current language setting can be changed.
  - **Help** - Opens the user interface documentation in a new tab.
  - **About** - Opens a view with information about the application.
  - **Log out** - Is visible only if the user is logged in.This action logs out the current user.
- **Breadcrumb navigation** - A graphical control element that is used as a navigational aid in user interfaces and on web pages.It is the panel located under the application header and can be used for easier navigation in the application.

- **Application home page** - The main view of the selected application.



4.2. Login view

The Login view provides the form used for LDAP-based authentication.The form contains two input fields:

- Username - The user's DN.
- Password - The user's password.

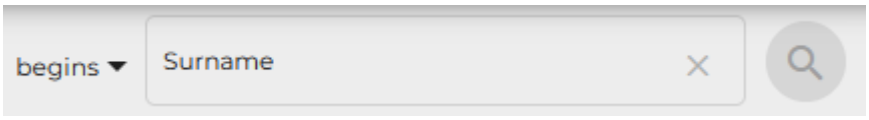
To log in, use the **Login** button or press the **Enter** key on the keyboard.If the authentication process is successful, the application displays the search form.

Depending on the bind operation configuration, the form can also provide the **Anonymous** button, which allows the user to be authenticated as anonymous without providing a username and a password.See the section "Bind Operation" in "Configuration".

4.3. Quick search

This component is used for quick LDAP searches according to one preconfigured attribute.The set of operators is configurable, too.See the section "Quick search" in "Configuration".

To perform the LDAP search, click the search button (the magnifying glass icon).The application then navigates to the Results table view where the search results are presented.The Person details view is directly displayed if there is only one result entry.



4.4. Extended search

This component is used for an extended LDAP search using a combination of attribute values.The user interface consists of input fields organized in groups.The groups and their attribute sets are preconfigured via the application configuration file.See the section "Extended search" in "Configuration".

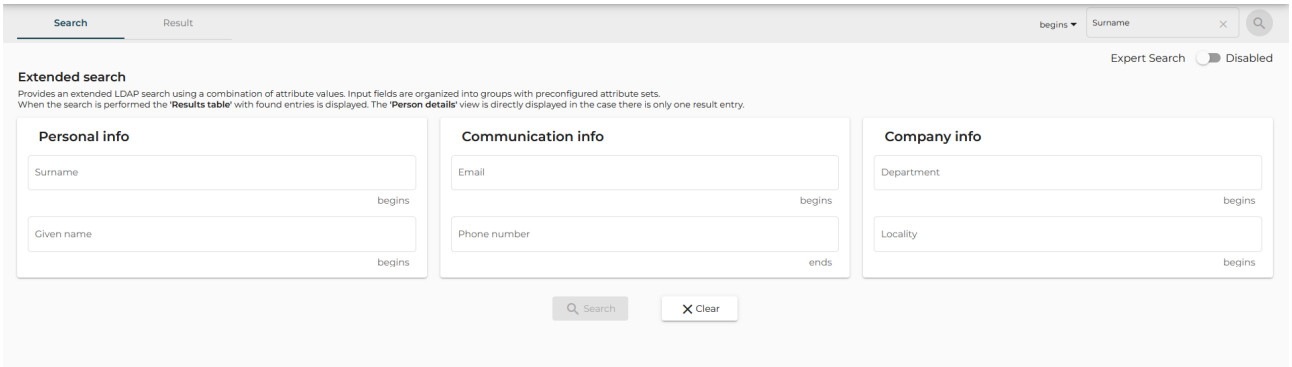
Each input field has its preconfigured search operator like **begins**, **ends**, **contains** and **equals**.In this view, the user is not allowed to change the operator.

Logical conjunction (and) is applied on all provided values in input fields.If no value is provided, the attribute is omitted from the LDAP search.

To perform an Extended search, press the **Enter** key on the keyboard or click the **Search** button (the magnifying glass icon).

Clicking the **Clear** button clears all the input fields.

Enabling **Expert Search** navigates to the Expert search view.



4.5. Expert search

This component is used for an expert LDAP search using a combination of attribute values.The user interface consists of input fields organized in groups.The groups and their attribute sets are preconfigured via the application configuration file.See the section "Expert search" in "Configuration".

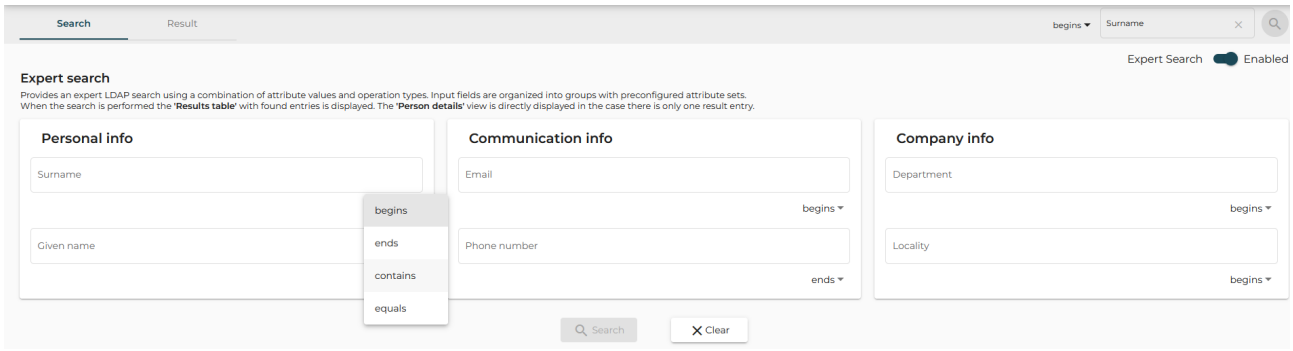
Each input field has its preconfigured search operator like **begins**, **ends**, **contains** and **equals**.In this view, the user is allowed to change the operator from the predefined set.The default operator and the set of available operators is configurable.

Logical conjunction (and) is applied on all provided values in input fields.If no value is provided, the attribute is omitted from the LDAP search.

To perform the Expert search, press the **Enter** key on the keyboard or click the **Search** button (the magnifying glass icon).

Clicking the **Clear** button clears all the input fields.

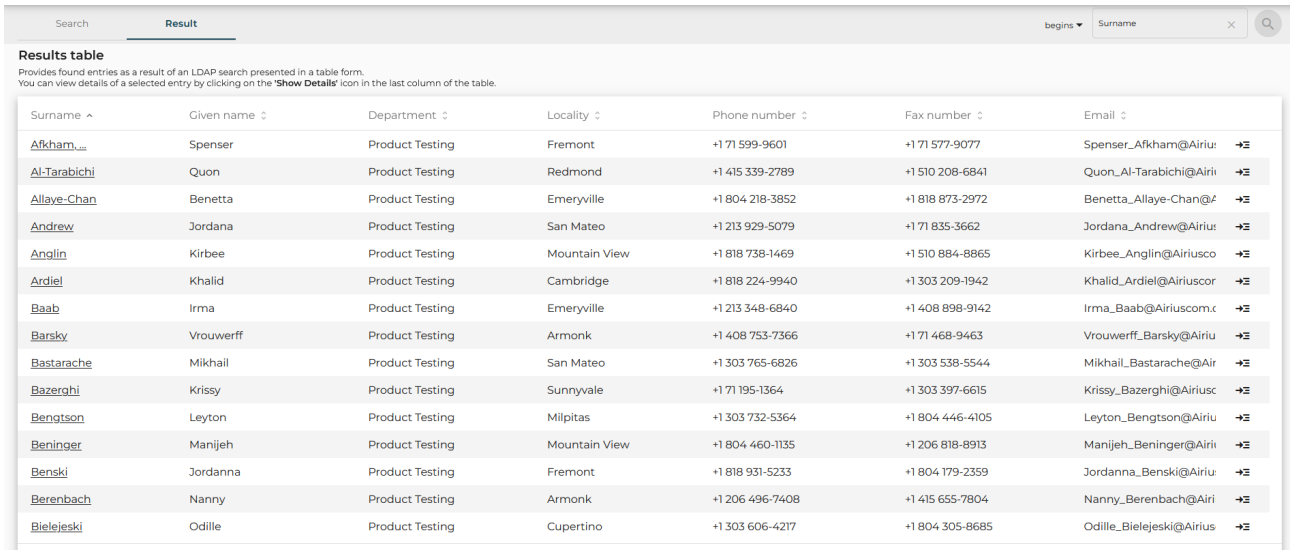
Disabling **Expert Search** navigates to the Extended search view.



4.6. Results table

The Results table view lists the entries that resulted from the LDAP search.The entries are presented in a table and paged and ordered according to the configuration.The last column contains a button which navigates to the Person details view.

The column set, page size and ordering are configurable.See the section "Results table" in "Configuration".



When the value is underlined, the configured search action (see "Link" in "Configuration") can be triggered by clicking on this value.Information about an action is displayed in a tooltip (shown on a mouse hover).
If the attribute is multivalued and more values are present, the action is triggered only on the first value.

4.7. Person details

The Person details view represents detailed information on the searched entry.When the search returns only one entry, the application navigates directly to the Person details view

and skips the Results table view.

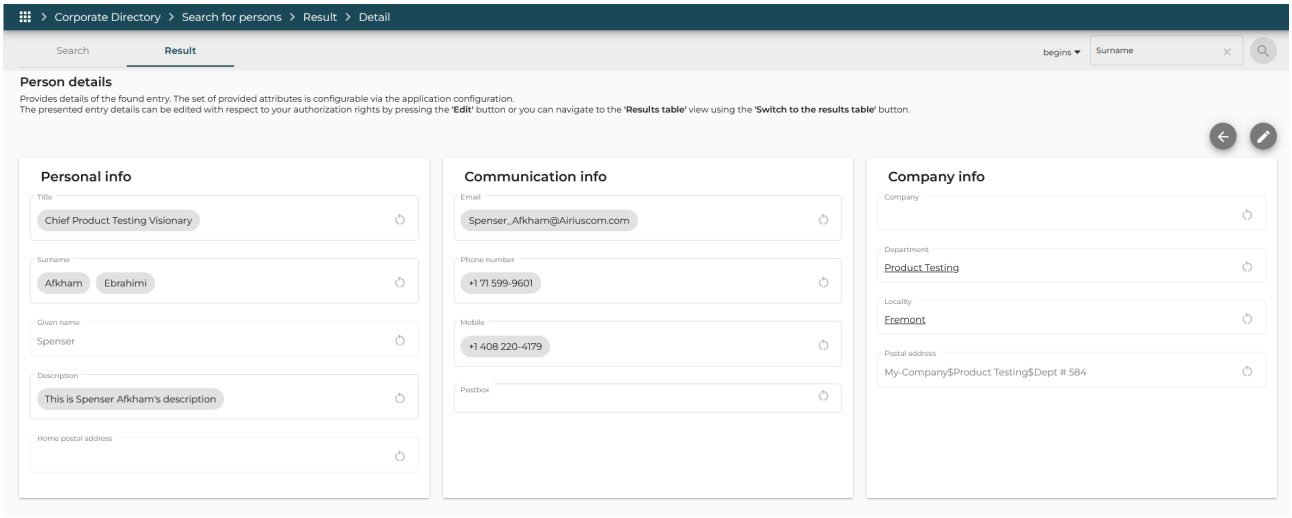
The view is organized in groups containing a set of fields.Groups and their attribute sets are configurable via the application configuration file.See the section "Person details" in "Configuration".

Click the **Switch to results table** button (left arrow icon) to navigate to the Results table view.

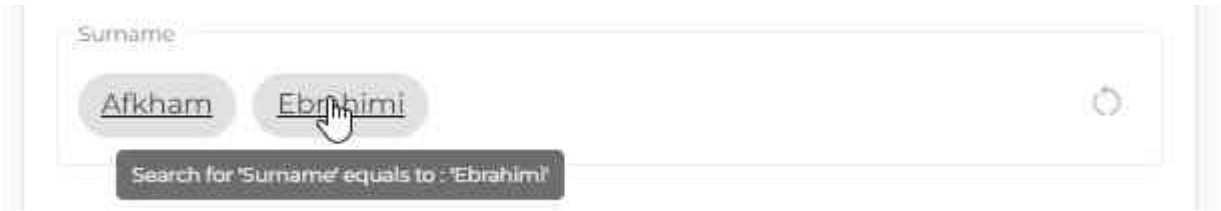
Click the **Edit** button (the pencil icon) to switch the form to edit mode.The user must have sufficient access rights to save modifications of the entry in the directory.

Click the **Save** button to save changes to the form.

To revert to the initial state of the selected entry (before editing), click the **Reset All** button. Changes are discarded and edit mode is switched off.



When the value is underlined, the configured search action (see "Link" in "Configuration") can be triggered by clicking on this value. Information about an action is displayed in a tooltip (shown on a mouse hover).



Appendix A: Changelog

This document lists all released versions of the DirX Corporate Directory application and describes each version's additions, modifications, and bug fixes.

A.1. Versions

- 1.0.57 (Initial release)
- 1.0.62 (Patch version - **not released yet**)

A.2. v 1.0.57 (Initial Release)

This is the initial version.There is no need to document additions, modifications, or bug fixes.

A.2.1. Requirements

- DirX Directory 9.0

A.3. v 1.0.63 (Patch Version)

First patch version.

A.3.1. Requirements

- DirX Directory 9.0 (build version >= 9.6.544)

A.3.2. Additions

- Support for anonymous access (DXD-5134)

A.3.3. Modifications

- Updated copyright note.
- Removed references to **dirx-atlassian.net** from Help page.
- Bind request uses ValueWithSpecificType component for password.
- Upgraded to the latest DirX Core framework version 1.0.0.
- Replaced brand logos in app according to current DirX style guide.

A.3.4. Bug Fixes

- Presence of authorization header in requests.
- Confirmation dialog is not shown when editing entry and user navigates to different view.

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.