EVIDEN

Identity and Access Management

DirX Directory

External Authentication

Version 9.1, Edition June 2025

Table of Contents

Preface

This manual is reference for the DirX Directory (DirX). It consists of the following sections:

- Chapter 1 describes how to perform external authentication.

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. External Authentication

Normally, the DirX Directory DSA performs the X.500 authentication procedure following an LDAP simple bind operation. The DSA reads the userPassword attribute of the entry named by the DN in the bind credentials and checks whether the value matches the one provided in the bind credentials.

If external authentication is enabled, the DSA does not follow this procedure. Instead, it forwards the DN – possibly after applying a conversion procedure – and the password to an external authentication service. After the external bind succeeds, the DSA must map the authenticated identity onto a distinguished name used as the X.500 access control identity that is applied in the access control decisions of subsequent operations.

The decision on whether a particular bind request is to be forwarded or is to be processed locally is based on configuration information. The details of the name conversion and the mapping between the external authentication ID and the X.500 DN are also configuration-based.

1.1. Enabling External Authentication

External authentication is disabled by default: every simple bind request that the DSA receives results in the execution of the X.500 bind procedure.To enable external authentication, you must create an external authentication configuration file named *install_path*/**server/conf/dirxextauth.cfg** and specify the external authentication service you want to use and all of the mandatory configuration parameters that this service requires.

The DSA searches for *install_path*/**server/conf/dirxextauth.cfg** during startup.If it finds this file, it parses the contents and enables the external authentication service specified in the file.The DSA then generates a NOTICE log entry that reports the status and the external authentication configuration in force.

If the DSA cannot open the file or if an error occurs while parsing it, external authentication is not activated and the DSA writes an ERROR log entry.

1.2. External Authentication Configuration File Format

The external authentication configuration file is an ASCII file that can contain three types of lines:

- Comment lines, which start with the '#' character.
- Empty or blank lines, for better readability
- Value assertion lines, which start with a keyword string, followed by a colon (**:**) character, followed by a value string

Each configuration parameter must occur only once.

The configuration file has mandatory and optional parameters.If a mandatory parameter is omitted, a parsing error occurs.Optional parameters have defaults; if an optional parameter is omitted, the DSA performs the default.

All keywords and values are case-insensitive, and leading and trailing blanks are ignored, except where indicated.

1.3. Generic External Authentication Configuration Parameters

This section provides information about all generic external authentication configuration parameters.

1.3.1. External Authentication Service Name

This section describes the external authentication service name parameters.

1.3.1.1. DIRX_EXT_AUTH_SERVICE

Use the **DIRX_EXT_AUTH_SERVICE** parameter to specify the external authentication service.DirX Directory supports **Windows** and **RACF**.This parameter is mandatory.

The syntax is as follows:

DIRX_EXT_AUTH_SERVICE: *ServiceName*

Example:

```
DIRX_EXT_AUTH_SERVICE: Windows
```

1.3.2. Other Generic Configuration Parameters

The names of the keywords that apply to any external authentication service have the prefix **EXT_**.

1.3.2.1. EXT_LDAP_IP_CONDITION

Use the **EXT_LDAP_IP_CONDITION** parameter to restrict the LDAP servers that can forward simple bind requests to the servers that run on the host whose IP address matches the specified value.

This parameter is optional. If it is omitted from the configuration file, the default is that there is no restriction: binds are forwarded to the external authentication service regardless of the host on which the LDAP server is running.

The syntax is as follows:

EXT_LDAP_IP_CONDITION: *IP_address*

Examples:

```
EXT_LDAP_IP_CONDITION: 123.45.67.8
```

A simple bind request is subject to forwarding if it was issued by the LDAP server running on the host with the IP address 123.45.67.8.

```
EXT_LDAP_IP_CONDITION: 123.45.67
```

A simple bind request is subject to forwarding if it was issued by an LDAP server running on the host whose IP address starts with 123.45.67

1.3.2.2. EXT_LDAP_RPCPORT_CONDITION

Use the **EXT_LDAP_RPCPORT_CONDITION** parameter to restrict the LDAP servers that can forward simple bind requests to the servers that listen for RPC requests on the port with the specified number.

This parameter is optional. If it is omitted from the configuration file, the default is that there is no restriction: binds are forwarded to the external authentication service regardless of the RPC-Port number on which the LDAP server is listening.

The syntax is as follows:

EXT_LDAP_RPCPORT_CONDITION: *rpc_port_number*

In standard installations, **dirxldapv3** uses the RPC port 6999. If multiple LDAP servers are running on one host, the value of the environment variable **DIRX_ADDITONAL_LDAP_SERVERS** contains the associated port number.

Example:

```
EXT_LDAP_RPCPORT_CONDITION: 2345
```

A simple bind request is subject to forwarding if it was issued by the LDAP server whose RPC port number is 2345.

1.3.2.3. EXT_AUTH_SEQUENCE

Use the **EXT_AUTH_SEQUENCE** parameter to control whether X.500 authentication is performed if an external authentication fails. This parameter is optional.

The syntax is as follows:

EXT_AUTH_SEQUENCE: *sequence_string*

Valid values for *sequence_string* are **EXT** and **EXT-DIRX**. The default value is **EXT**.

If **EXT** is specified and an external authentication fails, the DSA makes no attempt to authenticate the user itself using X.500 authentication.

If **EXT-DIRX** is specified and an external authentication fails (the forwarded bind request results in "invalid credentials" at the external authentication service), the DSA uses X.500 authentication: it takes the password supplied in the bind credentials and compares it to the userPassword attribute of the entry named by the DN in the bind credentials.When EXT-DIRX is specified, a user cannot distinguish whether a bind request has succeeded externally or due to internal X.500 authentication.

The fallback to X.500 does not apply to bind requests where the credentials DN consists of the external authentication ID; that is, binds that follow the Credentials-DN-Strategy for the Windows authentication service.A fallback to the X.500 bind procedure can be performed only if the entry is known and hence if the bind credentials contain its X.500 DN.

Example:

```
EXT_AUTH_SEQUENCE:EXT-DIRX
```

1.4. Windows External Authentication

This section provides information about Windows external authentication.

1.4.1. Configuration Parameters for Windows External Authentication

The names of the keywords that apply to the external authentication mechanism Windows have the prefix **WIN_**

The mapping parameters **WIN_ACCOUNT_AT**, **WIN_CONSTANT_DOMAIN** and **WIN_DOMAIN_USERNAME_SEPERATOR** control how the external authentication ID - the Windows account information - is mapped onto the X.500 Access Control ID - the entry's distinguished name.

The condition parameters **WIN_RDNTYPE_CONDITION** and **WIN_ATTRIBUTE_CONDITION** control the circumstances under which a particular bind request will be forwarded to the Windows external authentication service.

1.4.1.1. WIN_ACCOUNT_AT

The **WIN_ACCOUNT_AT** parameter is mandatory. It specifies the attribute that contains the Windows account name used to authenticate against the Windows external authentication service. The Windows account name consists of a Windows domain name and a user name. The **WIN_ACCOUNT_AT** attribute value format depends on the **WIN_CONSTANT_DOMAIN** and the **WIN_DOMAIN_USERNAME_SEPERATOR** parameters settings.

This attribute must fulfill the following conditions:

- An index must be generated for this attribute.
- The syntax of this attribute must be Directory-String

The syntax is as follows:

WIN_ACCOUNT_AT:*OID*

The attribute type *OID* must be specified in dotted OID format.

For an example, see the LDIF file **NT_StandAloneSchemaExt.ldif** in *install_path*/scripts/stand_alone/extensions*. You can load these data with **dirxmodify**. The DirX Directory abbreviation file **dirxabbr** already contains all of the required abbreviations and OIDs for this example.

1.4.1.2. WIN_DOMAIN_USERNAME_SEPERATOR

The **WIN_DOMAIN_USERNAME_SEPERATOR** parameter specifies the character used to separate the domain name from the user name in the values of the **WIN_ACCOUNT_AT** attribute.

This parameter is optional; if it is omitted from the configuration file, the default is the backslash ("****") character.

The syntax is as follows:

WIN_DOMAIN_USERNAME_SEPERATOR:*separator_char*

Example:

WIN_DOMAIN_USERNAME_SEPERATOR:-

Given the schema extension described in the **WIN_ACCOUNT_AT** parameter example, the configuration parameter specified in this example means that the values of the **WIN_ACCOUNT_AT** attribute WindowsAccount (OID = 1.3.12.1107.1.3.4.157) must contain the domain name and the user name separated by a dash (**-**) character. For example, if an entry has a WindowsAccount attribute value of **ABCdomain-user33**, the forwarded bind request sends the domain name **ABCdomain**, the username **user33** and the password sent in the bind credentials.

1.4.1.3. WIN_CONSTANT_DOMAIN

The **WIN_CONSTANT_DOMAIN** parameter specifies a constant domain name that is to be used in Windows external authentication.

This parameter is optional; if it is omitted from the configuration file, no constant domain name is specified and the **WIN_ACCOUNT_AT** attribute value must contain the Windows domain name and the Windows user name, separated by **WIN_DOMAIN_USERNAME_SEPERATOR**.

If this parameter is present, the **WIN_ACCOUNT_AT** must contain the user name.

If **WIN_CONSTANT_DOMAIN** is specified, the **WIN_DOMAIN_USERNAME_SEPERATOR** configuration parameter is ignored.

The syntax is as follows:

WIN_CONSTANT_DOMAIN:*domain_string*

Example:

```
WIN_CONSTANT_DOMAIN: XYZdomain
```

Given the schema extension described in the **WIN_ACCOUNT_AT** parameter example, the configuration parameter given in this example means that the values of the **WIN_ACCOUNT_AT** attribute WindowsAccount (OID = 1.3.12.1107.1.3.4.157) must contain the attribute user name. For example, if an entry has a WindowsAccount attribute value of **user33**, the forwarded bind request sends the domain name **XYZdomain**, the user name **user33** and the password sent in the bind credentials.

1.4.1.4. WIN_RDNTYPE_CONDITION

The **WIN_RDNTYPE_CONDITION** parameter specifies whether a Windows external authentication is to be executed if the DN in the simple bind credentials consists only of the **WIN_ACCOUNT_AT**. This parameter is optional.

The syntax is as follows:

WIN_RDNTYPE_CONDITION: *boolean_string*

Valid values for *boolean_string* are **TRUE** and **FALSE**.

The default value is **FALSE**: external authentication is not executed if the entry's DN in the bind credentials consists of the Windows account RDN only.

If the condition is set to **TRUE**, the bind request forwarding according to the credentials-DN-strategy is enabled. The credentials DN is converted into a Windows domain name and/or username (depending on the **WIN_DOMAIN_USERNAME_SEPERATOR** and **WIN_CONSTANT_DOMAIN** configuration), the Windows authentication function is called using the domain name, the user name and the password supplied in the bind credentials. If the Windows domain controller successfully authenticates the user, the DSA maps the **WIN_ACCOUNT_AT** value onto an X.500 Access Control Identity (an entry's DN): it searches the local DIT for the entry that holds the specified **WIN_ACCOUNT_AT** value. If the search delivers exactly one entry, this entry is considered to be authenticated with the authentication level **SIMPLE**.

Note that if both the **WIN_RDNTYPE_CONDITION** and the **WIN_ATTRIBUTE_CONDITION** parameters are set to **FALSE**, the external authentication is disabled.

Example:

```
WIN_RDNTYPE_CONDITION: FALSE
```

In this example, the DSA does not follow the Credentials-DN-Strategy: bind requests are not forwarded due to the bind DN. Even if the DN in the bind credentials consists only of one RDN with the type **WIN_ACCOUNT_AT**, this bind request is not subject to forwarding to Windows.

1.4.1.5. WIN_ATTRIBUTE_CONDITION

The **WIN_ATTRIBUTE_CONDITION** parameter specifies whether a Windows external authentication is to be executed if the entry named by the DN in the simple bind credentials has an attribute of the **WIN_ACCOUNT_AT** type. This parameter is optional.

The syntax is as follows:

WIN_ ATTRIBUTE_CONDITION: *boolean_string*

Valid values for *boolean_string* are **TRUE** and **FALSE**.

The default value is **FALSE:** external authentication is not executed, even if the entry named by the DN in the bind credentials has an attribute of the **WIN_ACCOUNT_AT** type.

If the condition is set to **TRUE**, the bind request forwarding according to Attribute-Entry-Strategy is enabled: the DSA reads the entry named by the DN in the bind credentials. If the entry does not have a **WIN_ACCOUNT_AT** attribute, the DSA performs the X.500 bind procedure. If the entry has a **WIN_ACCOUNT_AT** attribute, the DSA converts its value into a Windows domain name and/or user name (depending on the **WIN_DOMAIN_USERNAME_SEPERATOR** and **WIN_CONSTANT_DOMAIN** configuration) and the Windows authentication function is called using the domain name, the user name and the password supplied in the bind credentials. If the Windows domain controller successfully authenticates the user, the DSA considers the entry to be authenticated with the authentication level **SIMPLE**.

Both the **WIN_RDNTYPE_CONDITION** and the **WIN_ATTRIBUTE_CONDITION** can be set to **TRUE**.

Example:

```
WIN_RDNTYPE_CONDITION: TRUE
```

In this example, the DSA follows the Attribute-Entry-Strategy; that is, every simple bind request is examined to determine if the entry has the **WIN_ACCOUNT_AT** attribute.

1.4.2. Mapping the Windows Authentication ID onto an X.500 Entry

If a bind request is forwarded to the Windows external authentication service following the Credentials-DN-Strategy due to the **WIN_RDNTYPE_CONDITION** (the bind DN contains the

Windows domain and/or user name) the DSA maps onto an X.500 entry named by a DN that stores the value of the bind credentials name in its **WIN_ACCOUNT_AT** attribute after the external authentication has successfully completed. For the mapping purpose, the search for entries that have the value supplied as the bind DN as the **WIN_ACCOUNT_AT** attribute must deliver exactly one entry.

However, one entry may possess more than one value for the **WIN_ACCOUNT_AT** attribute; that is, the mapping of external to X.500 Entities is N:1.

If a bind request is forwarded following the Attribute-Entry-Strategy due to the **WIN_ATTRIBUTE_CONDITION**, the bind DN already names the X.500 entry. In this case, the forwarding is initiated if this entry has exactly one **WIN_ACCOUNT_AT** value naming the Windows entity.

1.4.3. Combining Conditions

The condition parameters define the circumstances in which a particular bind request is forwarded to an external authentication service. The conditions refer to the LDAP server that is initiating the bind request and to the entity performing the bind. The configuration parameters **EXT_LDAP_IP_CONDITION** and **EXT_LDAP_RPCPORT_CONDITION** control whether the LDAP condition is true or false. The LDAP condition is true for a particular bind request if the LDAP server that issued the request matches the values configured for these two parameters. (note that all LDAP servers match if the respective parameters are omitted).

The logical combination of all conditions is:

```
forward a bind to the Windows external authentication service
  if ((ldapCondition is TRUE)
         AND
     ((WIN_RDNTYPE_CONDITION is TRUE) OR (WIN_ATTYPE_CONDITION is
TRUE)))
```

1.4.4. Example of a Windows External Authentication Configuration File

The following example of a **dirxextauth.cfg** file for the Windows external authentication service illustrates the effects of the configuration parameters:

```
# External Authentication Configuration File
#
# example settings for documentation purpose

DIRX_EXT_AUTH_SERVICE: windows

# Mapping Parameters
```

```
WIN_ACCOUNT_AT: 1.3.12.2.1107.1.3.4.157
WIN_CONSTANT_DOMAIN: domainABC

# The following is ignored if constant domain is used
#WIN_DOMAIN_USERNAME_SEPERATOR: -

# Condition Parameters

# No restriction with respect to the issuing LDAP server
#EXT_LDAP_IP_CONDITION: 12.34.56.78

#EXT_LDAP_RPCPORT_CONDITION: 9990

# simple binds with DNs that consist only of one RDN with the
Attribute type # 1.3.12.1107.1.3.4.157 must be forwarded to Windows
WIN_RDNTYPE_CONDITION: TRUE

# do not follow the Attribute-Entry-Strategy
WIN_ATTRIBUTE_CONDITION: FALSE

# Do not fallback automatically after Windows bind fails
EXT_AUTH_SEQUENCE:EXT
```

1.4.5. Messages and Errors

During startup, the DSA evaluates the existence and contents of the external authentication configuration file. It then writes an event log that displays the status of the external authentication.

When the DSA parses the example configuration file given in the previous section, it writes an event entry to the Windows Event Viewer, for example:

```
External Password Authentication settings:
           Service:           "Windows"
           Status:            "Enabled"
  Mapping:
    AttributeType:        "WindowsAccount (OID 1.3.12.1107.1.3.4.157)
       Details:           "Attribute value contains UserName,
                           configured Domain is <domainABC>"
       Forward Criteria:
         Bind Front-End          "all LDAP servers"
         Bind DN:                "consists of 1 RDN with Type
WindowsAccount"
```

```
         Entry with Attribute:   "no forward due to particular
AttributeType"
       Authentication Sequence:  "Only External"
```

If the DSA detects missing, erroneous or inconsistent configuration information, it disables the Windows external authentication mechanism and writes an error log to the Event Viewer. For example:

```
External Password Authentication:
       Service:     "Windows"
       Status:      "Service Not Enabled"
       Reason:      "No Forward Condition specified"
```

1.4.6. Windows Prerequisites

This section provides information on prerequisites and privileges for external authentication on Windows platforms.

1.4.6.1. Privileges Necessary on the DirX Directory DSA Host

On Windows platforms, the DirX Directory service runs under a particular account. In order to be able to forward bind credentials to the Windows Domain Controller, this account needs to have a special privilege. The act-as-part-of-the-operating-system privilege (TCB privilege) enables the DirX Directory DSA process to call the logonUser() function.

During the installation process, the TCB privilege is automatically added to the Windows account that runs the DirX Directory service.

If the TCB privilege is removed, the DSA is no longer able to forward bind requests. All externally processed bind operations result in an “invalid credentials” return code and DSA logging (subcomponent and level set to sec.4) shows the string:

```
"Windows Logon error: 1314: A required privilege is not held by the
client."
```

To grant the “act as part of the operating system” privilege to the DirX Directory account by hand, click **Start**, type **secpol.msc**, and then press ENTER to open the Local Security Policy console. Under **Security Settings** of the console tree, select:

```
Local Policies -> User Rights Assignment
```

1.4.6.2. Privileges Necessary on Other Hosts

All other Windows hosts that are involved in the external bind procedure, need to have the “Access this computer from the network” privilege granted to all users that wish to logon. This applies also to the host of a virtual machine that runs the DirX Directory service.

If the “Access this computer from the network” privilege is not grated for the user performing the external bind, the bind operation results in an “invalid credentials” return code. In that case the DSA logging (subcomponent and level set to sec.4) shows the string:

```
"Windows Logon error: 1385: Logon failure: the user has not been
granted the requested logon type at this computer."
```

To grant the “Access this computer from the network” privilege to the respective account by hand, click **Start**, type **secpol.msc**, and then press ENTER to open the Local Security Policy console.Under **Security Settings** of the console tree, select:

```
Local Policies -> User Rights Assignment
```

and check that privilege for the user in question (for example add it to EVERYONE).

The Windows authentication service runs on a Windows domain controller.If the DSA forwards bind requests from users that are registered in domains other than the one in which the DSA is located, an inter-domain trust relationship must be established between the affected Windows domains.

1.5. RACF External Authentication

This section provides information about RACF external authentication, the method of forwarding bind requests via LDAPv3 to an external LDAP server.

1.5.1. Configuration Parameters for RACF External Authentication

The names of the keywords that apply to the external authentication mechanism RACF have the prefix **RACF_**.

The parameters **RACF_LDAP_HOSTS_*_*n*_, *RACF_LDAP_PORT** and **RACF_LDAP_TIMEOUT** provide address information about the RACF system’s LDAP server and the timeout value to be applied when connecting to that system.

The RACF-specific condition parameters **RACF_SUBTREE_CONDITION** and *RACF_ATTRIBUTE_CONDITION_HOST_**n* can be used to specify whether or not an incoming bind request is subject to forwarding to RACF.

Recall from section **How Does RACF Authentication Work?** in the *DirX Directory Administration Guide* that the DirX Directory DSA supports either:

- Dynamic DN mapping, which involves DN resolution performed by the remote RACF LDAP server based on RDN values from the incoming bind operation
- Local DN mapping based on attributes stored at the entry in the DIT of the local DSA or in the configuration file.

For static DN mapping, the administrator can also specify the following configuration parameters for RACF external authentication:

- **RACF_DYNAMIC_DN_MAPPING: FALSE** to enable static DN mapping. As an alternative the administrator can omit this configuration parameter.
- **RACF_MAP_SUBTREE_FROM** and **RACF_MAP_SUBTREE_TO** to map a specific subtree value part in the DN of the bind credentials.
- **RACF_MAP_ATTR_FROM** and **RACF_MAP_ATTR_TO** to map the most significant attribute type of RDN in the bind credentials.

The administrator must provide additional information either in the local configuration file or as attributes of the entry in the DIT of the local DSA.

For dynamic DN mapping, the administrator must also specify the following configuration parameters for RACF external authentication:

- **RACF_DYNAMIC_DN_MAPPING: TRUE** to enable dynamic DN mapping.
- **RACF_DYNAMIC_DN_MAP_ATTR** to specify the attribute type that replaces the attribute type of the most significant attribute of the RDN in the filter of the search operation that the RACF LDAP server performs to resolve the DN. This parameter is optional.
- **RACF_DYNAMIC_DN_MAP_BASE** to specify the base object for the search operation that the RACF LDAP server performs to resolve the DN. This parameter is optional.

For dynamic DN mapping, the administrator may additionally specify the credentials of the user that performs the search operation with the following configuration parameters:

- **RACF_DYNAMIC_DN_MAP_TECH_BIND_DN** to specify the distinguished name of the user performing the search operation.
- **RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE** to specify the pathname of the file that contains the password of the user performing the search operation.

If configuration parameters for static and dynamic DN mapping are specified, the DirX Directory DSA does not enable external RACF authentication and logs an error for **External Password Authentication**.

The following sections describe the configuration parameters for RACF external authentication.

1.5.1.1. RACF_LDAP_HOSTS_*n*

The **RACF_LDAP_HOSTS_*n* parameter specifies the address of the target system's LDAP server(s). The maximum number of LDAP servers that can be specified is 10; that is, the sequence number *n* runs from *1** through **10**:

RACF_LDAP_HOSTS_1

RACF_LDAP_HOSTS_2

...

RACF_LDAP_HOSTS_10

At least one LDAP server must be specified in the parameter **RACF_LDAP_HOSTS_1** with one IP address or hostname. Alternatively a list of IP addresses or hostnames or addresses can be provided for each parameter. A SPACE character must separate multiple values. The DSA uses the first IP address or hostname as the primary RACF LDAP server. In case of a connection problem the DSA proceeds with the next IP address or hostname in the list. A connection problem in this context indicates that the remote LDAP server does not respond with a successful return code or with an "invalid Credentials" or "inappropriate Authentication" error.

RACF_LDAP_PORT specifies the port number of the RACF LDAP server. You can append a specific port number to a hostname separated with a colon (**:**), for example **host1:1234**. This port number overrules the port number specified in **RACF_LDAP_PORT**.

If **RACF_LDAP_HOSTS_***n* is omitted from the configuration file, the DSA writes an exception and disables the external authentication mechanism.

The syntax is as follows:

RACF_LDAP_HOSTS_*n*:_address_string_ [*address_string* [*address_string* ...]]

where

n is a sequence number running from **1** to **10**, each specifying one LDAP server backend. The sequence numbers must be specified in ascending order without any gap.

address_string is the address of one or more LDAP servers. Valid values for *address_string* are either a dotted decimal IP address or a DNS name, optionally followed by a colon and the port number for that specific LDAP server.

Use the parameter **RACF_ATTRIBUTE_CONDITION_HOST_***n* to distribute external bind requests to different RACF backends. If more than one RACF backend is specified, there must be a **RACF_ATTRIBUTE_CONDITION_HOST_***n* parameter for each **RACF_LDAP_HOSTS_***n* parameter. The sequence number of the attribute condition is the same as for the RACF backend.

For compatibility reasons, DirX Directory supports the parameter **RACF_LDAP_HOST** instead of **RACF_LDAP_HOSTS_1**.

Examples:

```
RACF_LDAP_HOSTS_1: 123.456.78.9
```

In this example, the DSA forwards bind requests to the RACF LDAP server listening on the port specified in **RACF_LDAP_PORT** on the host with the IP address **123.456.78.9**.

```
RACF_LDAP_HOSTS_1: host1:1234 host2
RACF_LDAP_HOSTS_2: host3:5678 host4
```

In this example, the DSA uses two RACF backends to forward bind requests. For the first backend, the port **1234** on **host1** is the primary LDAP server to forward binds to. If a connection problem occurs while forwarding, the bind to the DSA fails over to use the port specified in **RACF_LDAP_PORT** on **host2** for forwarding of bind requests. As a second RACF backend, the DSA uses the port **5678** on **host3** as the primary LDAP server to forward binds to. If a connection problem occurs while forwarding the bind the DSA switches to use the port specified in **RACF_LDAP_PORT** on **host4** for forwarding of bind requests. In this example, the administrator must specify a **RACF_ATTRIBUTE_CONDITION_HOST_1** parameter for **RACF_LDAP_HOSTS_1** and a **RACF_ATTRIBUTE_CONDITION_HOST_2** parameter for **RACF_LDAP_HOSTS_2** to distribute the external bind requests.

1.5.1.2. RACF_LDAP_PORT

The **RACF_LDAP_PORT** parameter specifies the port number of the RACF LDAP server. This is an optional parameter. If this parameter is omitted from the configuration file, the DSA uses port **389**.

The syntax is as follows:

RACF_LDAP_PORT:*portnumber_string*

Example:

```
RACF_LDAP_PORT:2345
```

In this example, the DSA forwards bind requests to LDAP port 2345.

1.5.1.3. RACF_LDAP_SSL

The **RACF_LDAP_SSL** parameter specifies whether connection to the external RACF host uses SSL/TLS or not. This parameter is optional.

The syntax is as follows:

RACF_LDAP_SSL: *boolean_string*

Valid values for *boolean_string* are **TRUE** and **FALSE**.

The default value is **FALSE**: the external RACF host is connected via plain LDAP protocol.

If the condition is set to **TRUE**, the external RACF host is connected via LDAP protocol over SSL/TLS. The **RACF_LDAP_PORT** parameter must match the secure LDAP port number of

the target LDAP server. A proper certificate database file **cert8.db** for the DSA process must be provided.

The certificate database file must contain the certificate of the LDAP server that the DSA is binding to or the root certificate of the certification authority that issued the certificate of the LDAP server. The file must be located under *install_path*/server/conf*. (The value of the environment variable **DIRX_TRUSTED_CA** is ignored when locating the certificate database file for external authentication.) If the bind target is a DirX Directory V8.0 LDAP server with its default PSE files it is sufficient to copy the certificate database file from the target system *install_path*/client/conf/cert8.db* to *install_path*/server/conf/cert8.db* on the bind forwarder system.

Example:

```
RACF_LDAP_SSL: TRUE
```

In this example, the DSA forwards bind requests to LDAP using an SSL/TLS connection.

1.5.1.4. RACF_LDAP_TIMEOUT

The **RACF_LDAP_TIMEOUT** parameter specifies the number of seconds that the DSA waits for the result of the asynchronous bind operation sent to the external RACF LDAP server.

If dynamic DN mapping is enabled, the wait time is applied more than once:

- optionally, for the bind operation performed on behalf of the technical user,
- for the search operation that is sent to an external RACF LDAP server to map the DN and
- for the subsequent bind operation that is forwarded to an external RACF LDAP server to perform the authentication.

If the configured value is exceeded, the DSA returns an error to the bind initiator and writes an error message into the exception log file.

Valid values range from 1 through 120 seconds. Values outside this range result in an error during the DSA's initialization process. The DSA then disables the RACF external authentication mechanism.

If there is a list of LDAP servers for a specific backend (**RACF_LDAP_HOSTS_*n*), the DSA waits *RACF_LDAP_TIMEOUT** seconds for each LDAP server in the list.

This is an optional parameter. If this parameter omitted from the configuration file, the DSA uses 2 seconds as the default timeout value.

The syntax is as follows:

RACF_LDAP_TIMEOUT:*seconds_string*

Example:

```
RACF_LDAP_TIMEOUT: 1
```

In this example, the DSA waits 1 second for external RACF bind operations to return.

1.5.1.5. RACF_SUBTREE_CONDITION

The **RACF_SUBTREE_CONDITION** parameter specifies the distinguished name of the root of a subtree that is used to restrict the forwarding of binds to DNs that belong to the specified subtree.

This is an optional parameter. If this parameter is omitted from the configuration file, the DSA forwards all bind requests.

The syntax is as follows:

RACF_SUBTREE_CONDITION:*subtree_name* [**$** *subtree_name2* [**$** *subtree_name3* ...]]

The distinguished name of the subtree root *subtree_name* must be specified in LDAP notation. Separate multiple values with a **$** character.

The RDN-types used for the subtree names must be either one of **dc**, **c**, **o**, **ou**, **l**, or **cn** or must be the dotted OID notation of the attribute type. Attributes with syntaxes other than Directory String, IA5 String or Printable String are not supported.

The RDN values used for the subtree names must not contain characters outside the range of basic Latin (value of character > 0x7e) or special characters that require LDAPv3 escaping (for example "," or ";").

Case Ignore Matching applies to the RDN values.

The *subtree_name* must be complete: it must start with the root. For example, if *subtree_name* is ou=intern,o=pqr and the bind DN is cn=xxx,ou=intern,o=pqr,c=de the bind request is not forwarded.

Example:

```
RACF_SUBTREE_CONDITION: ou=sales,o=my-company
```

In this example, the DSA forwards bind requests if the DN in the bind credentials is located below the subtree root **ou=sales,o=my-company**.

1.5.1.6. RACF_ATTRIBUTE_CONDITION_HOST__n_

The *RACF_ATTRIBUTE_CONDITION_HOST_**n* parameter is used to restrict bind request forwarding to bind DNs represented by directory entries in the local DSA that have the specified attribute. Use this parameter to distribute external bind requests to different RACF backends.

This is an optional parameter if only one RACF backend (**RACF_LDAP_HOSTS_1**) is specified. If this parameter is not contained in the configuration file, the DSA forwards all bind requests if all other conditions match.

If more than one RACF backend (*RACF_LDAP_HOSTS_**n*) is specified, there must be a *RACF_ATTRIBUTE_CONDITION_HOST_**n* parameter for each RACF backend. The sequence number *n* of the RACF backend and the attribute condition (*RACF_ATTRIBUTE_CONDITION_HOST_**n*) must be the same for each RACF backend / attribute condition pair. All attribute conditions must use the same attribute type.

To determine the target backend, the DSA evaluates the attribute conditions in ascending order starting with **RACF_ATTRIBUTE_CONDITION_HOST_**1. The first matching condition determines the target backend.

The syntax is as follows:

RACF_ATTRIBUTE_CONDITION_HOST_*n*:**attr_type*=**attr_value* or

RACF_ATTRIBUTE_CONDITION_HOST_*n*:**attr_type*!=**attr_value* (not equal)

where

n is a sequence number from **1** through **10** specifying the RACF backend.

Specify the attribute type *attr_type* in dotted OID notation.

attr_value specifies the attribute value or a substring of the attribute value. The attribute value must not contain characters outside the range of basic Latin (value of character > 0x7e) or special characters that require escaping according to LDAPv3. Case Ignore Matching applies to all specified attribute values or value parts. Specify:

- The entire attribute value *attr_value*, or
- An initial substring of the attribute value by appending an asterisk (**) to the attribute value specified (*attr_value***), or
- A final substring of the attribute value by prefixing an asterisk (***) to the attribute value specified (**attr_value*), or
- Any substring of the attribute value by prefixing and appending an asterisk (**) to the attribute value specified (***attr_value*****), or
- Just the presence of the attribute type *attr_type* by specifying an asterisk (***) for *attr_value*.

Attributes with syntaxes other than Directory String, IA5 String, or Printable String are not supported.

For compatibility reasons, DirX Directory supports the parameter **RACF_LDAP_ATTRIBUTE_CONDITION.**

Examples:

```
RACF_ATTRIBUTE_CONDITION_HOST_1:2.5.4.13=racfUser
```

In this example, the DSA forwards bind requests to the RACF backend specified in RACF_LDAP_HOSTS_1 if the entry represented by the DN in the bind credentials has a description Attribute (OID 2.5.4.13) with the value **racfuser**.

```
RACF_ATTRIBUTE_CONDITION_HOST_1:2.5.4.4=Extern*
RACF_ATTRIBUTE_CONDITION_HOST_2:2.5.4.4=*
```

In this example, the DSA forwards bind requests to the RACF backend specified in RACF_LDAP_HOSTS_1 if the entry represented by the DN in the bind credentials has a surname attribute (OID 2.5.4.4) beginning with the string **extern**. Bind requests from users with any other surname is forwarded to the RACF backend specified in RACF_LDAP_HOSTS_2.

1.5.1.7. RACF_MAP_SUBTREE_FROM and RACF_MAP_SUBTREE_TO

The **RACF_MAP_SUBTREE_FROM** and **RACF_MAP_SUBTREE_TO** parameters control textual replacements performed to the incoming bind DN. These replacements are considered as type of static DN mapping.

These parameters are optional. However, the presence of one parameter requires the presence of the other parameter. If they are not contained in the configuration file, the DN in the bind credentials remains unchanged in the bind request forwarded to RACF unless RDN attribute mapping is applied. (See also the sections **RACF_MAP_RDN_ATTR_FROM** and **RACF_MAP_RDN_ATTR_TO**.)

If these parameters are present, the part of the DN that matches **RACF_MAP_SUBTREE_FROM** is replaced by the subtree name specified in **RACF_MAP_SUBTREE_TO**. If the incoming DN does not contain the string specified in **RACF_MAP_SUBTREE_FROM**, no replacement is performed and the DN is forwarded unchanged to RACF.

Case Ignore Matching is applies when the incoming DN is checked against **RACF_MAP_SUBTREE_FROM**.

Specify the keyword **NO-VALUE** for **RACF_MAP_SUBTREE_TO** if the resulting mapped DN should not contain the part specified by **RACF_MAP_SUBTREE_FROM**.

Specify the keyword **ANY-VALUE** for **RACF_MAP_SUBTREE_FROM** if the resulting mapped DN should contain the RDNs specified in **RACF_MAP_SUBTREE_TO** regardless of the subtree value of the original bind DN. (The most significant RDN of the mapped DN depends also on the settings in **RACF_MAP_RDN_ATTR_FROM/TO**.)

The syntax is as follows:

RACF_MAP_SUBTREE_FROM: *subtree_name*

RACF_MAP_SUBTREE_TO: *subtree_name*

subtree_name specifies a distinguished name in LDAP notation.

Examples:

```
RACF_MAP_SUBTREE_FROM:ou=sales,o=my-company
RACF_MAP_SUBTREE_TO:sysplex=ldap,profiletype=user
```

An incoming bind request with the bind credentials DN **cn=abele,ou=sales,o=my-company** is mapped to the RACF bind DN **cn=abele,sysplex=ldap,profiletype=user**.

An incoming bind request with the bind credentials DN **cn=admin,o=my-company** remains unchanged. If all other configured conditions are met for this bind request, the RACF bind is performed as the user with the DN **cn=admin,o=my-company**.

```
RACF_MAP_SUBTREE_FROM:ANY-Value
RACF_MAP_SUBTREE_TO:No-Value
```

An incoming bind request with the bind credentials DN **cn=abele,ou=sales,o=my-company** is mapped to the RACF bind DN **cn=abele**.

For the incoming bind with the DN **cn=admin,o=my-company** the mapping also results in a mapped DN that only consist of the single RDN **cn=admin** if all other configured conditions are met for both bind requests.

1.5.1.8. RACF_MAP_RDN_ATTR_FROM and RACF_MAP_RDN_ATTR_TO

The **RACF_MAP_RDN_ATTR_FROM** and **RACF_MAP_RDN_ATTR_TO** parameters provide a mapping that replaces the most significant RDN of the DN in the bind credentials. These replacements are considered as type of static DN mapping.

These parameters are optional. However, the presence of one parameter requires the presence of the other parameter. If they are not contained in the configuration file, the most significant RDN of the DN in the bind credentials remains unchanged in the bind request forwarded to RACF.

If these parameters are present, the most significant RDN of the DN in the bind credentials is replaced by the attribute type specified in **RACF_MAP_RDN_ATTR_TO** and the attribute value stored as the value of the **RACF_MAP_RDN_ATTR_FROM** attribute in the directory entry with the distinguished name provided in the bind credentials.

Specify the keyword **NO-VALUE** for **RACF_MAP_RDN_ATTR_TO** if the most significant RDN of the resulting DN should consist only of the attribute value. This may be useful when forwarding bind requests via LDAPv3 protocol to a Microsoft Active Directory Server.

If there is no attribute type specified in the **RACF_MAP_RDN_ATTR_TO** parameter stored in the directory entry or there is more than one value stored for this attribute, the RACF

authentication fails with an LDAP_INVALID_CREDENTIALS error.

The syntax is as follows:

RACF_MAP_RDN_ATTR_TO: *attrtype_string*

attrtype_oid specifies the attribute type in dotted OID notation. The value of this attribute stored with the DirX Directory entry replaces the value of the most significant RDN provided with the bind credentials before the credentials are sent to the RACF external authentication service.

attrtype_string specifies the LDAP name of the attribute that replaces the attribute type of the most significant RDN before the credentials are sent to the RACF external authentication service.

Example:

```
RACF_MAP_RDN_ATTR_FROM: 2.5.4.13
RACF_MAP_RDN_ATTR_TO: racfdescriptor
```

An incoming bind request with the bind credentials DN **cn=abele,ou=sales,o=my-company** is mapped to the RACF bind DN **racfdescriptor=user-abele,ou=sales,o=my-company** if the description attribute (OID 2.5.4.13) with the value **user-abele** is stored in the directory entry **cn=abele,ou=sales,o=my-company**.

An incoming bind request with the bind credentials DN **cn=admin,o=my-company** fails if there is no description attribute (OID 2.5.4.13) stored in the directory entry **cn=admin,o=my-company**.

1.5.1.9. RACF_DYNAMIC_DN_MAPPING

The **RACF_DYNAMIC_DN_MAPPING** parameter specifies whether the mapping onto the DN is performed by the external RACF LDAP server.

If dynamic DN mapping is enabled the DirX Directory DSA first sends a subtree search operation to the external RACF LDAP server to map the DN provided in the bind request to the DN used in the bind operation that the DirX Directory DSA subsequently forwards to the same external RACF LDAP server for authentication.

The options of the subtree search to map the DN are:

- A base object that is either the root (**/**) or the value specified in the **RACF_DYNAMIC_DN_MAP_BASE**.
- A simple equality filter that is expected to identify uniquely one entry in this external LDAP server. The filter is *RDN-type*=***RDN-value*** **where *RDN-type* is the attribute type specified in the parameter *RACF_DYNAMIC_DN_MAP_ATTR** or the attribute type of the leftmost (most significant) RDN of the original distinguished name (DN) and *RDN-value* is the value of the leftmost RDN of the original DN.

The DSA uses the DN of the entry returned by this search operation and the password provided in the original bind request in the bind operation forwarded to the external RACF LDAP server for authentication. If the search operation returns no or more than one entry the original bind request is rejected due to **invalid credentials**.

This parameter is optional.

The syntax is as follows:

RACF_DYNAMIC_DN_MAPPING: *boolean_string*

Valid values for *boolean_string* are **TRUE** and **FALSE**.

The default value is **FALSE**. (The DSA performs static DN mapping.)

If the value is set to **TRUE**, dynamic DN mapping by means of a search operation sent to the remote LDAP server is performed.

Example:

```
RACF_DYNAMIC_DN_MAPPING: TRUE
```

In this example, no further configuration parameters are specified. The DirX Directory DSA performs the default behavior for dynamic DN mapping. (See the following parameter descriptions.)

If an incoming bind is subject to external RACF authentication and the request contains the DN **sn=abele,ou=sales,o=my-company** the DSA performs the following steps:

1. The DSA sends a subtree search operation with the root (**/**) as base object and the filter **sn=abele** to the external RACF LDAP server.
2. Let us assume that the remote server returns one entry with the DN **cn=abele,sysplex=ldap,profiletype=user**. The DirX Directory DSA uses this DN and the password provided by the user in the original bind request to forward the bind operation to the external RACF LDAP server for authentication.

   (If the external RACF LDAP server does not return one unique entry, the DirX Directory DSA rejects the bind.)

1.5.1.10. RACF_DYNAMIC_DN_MAP_ATTR

The **RACF_DYNAMIC_DN_MAPPING** parameter specifies the attribute type that is used in the filter of the search operation to map the DN of the original bind operation. This attribute type replaces the attribute type of the leftmost (most significant) RDN in the original bind request.

This parameter is optional.

The syntax is as follows:

RACF_DYNAMIC_DN_MAP_ATTR: *attrtype_string*

attrtype_string specifies the LDAP name of the attribute type that is used as type in the filter of the search operation sent to the external LDAP server to map the DN for subsequent external RACF authentication.

The default behavior is to use the attribute type of the leftmost (most significant) RDN provided in the original bind request.

Example:

```
RACF_DYNAMIC_DN_MAPPING: TRUE
RACF_DYNAMIC_DN_MAP_ATTR: commonName
```

If an incoming bind is subject to external RACF authentication and the request contains the DN **sn=abele,ou=sales,o=my-company**, the DSA performs the following steps:

1. The DSA replaces the attribute type of the leftmost (most significant) RDN provided in the original bind request **sn** with the attribute type **commonName**.
2. The DSA sends a subtree search operation with the root (**/**) as base object and the filter **commonName=abele** to the external RACF LDAP server.
3. Let's assume that the remote server returns one entry with the DN **cn=abele,sysplex=ldap,profiletype=user**. The DirX Directory DSA uses this DN and the password the user provided in the original bind request to forward the bind operation to the external RACF LDAP server for authentication.

   (If the external RACF LDAP server does not return one unique entry the DirX Directory DSA rejects the bind.)

1.5.1.11. RACF_DYNAMIC_DN_MAP_BASE

The **RACF_DYNAMIC_DN_MAP_BASE** parameter specifies the search base that is used as base object in the search operation to map the DN of the original bind operation.

This parameter is optional.

The syntax is as follows:

RACF_DYNAMIC_DN_MAP_BASE: *dn_string*

dn_string specifies the LDAP name of the base object in the search operation sent to the external LDAP server to map the DN for subsequent external RACF authentication.

The default behavior is to use the root provided in the original bind request.

Example:

```
RACF_DYNAMIC_DN_MAPPING:TRUE
```

```
RACF_DYNAMIC_DN_MAP_ATTR:commonName
RACF_DYNAMIC_DN_MAP_BASE:cn=users,dc=domain2,dc=loc1,dc=comany,dc=de
```

If an incoming bind is subject to external RACF authentication and the request contains the DN **sn=abele,ou=sales,o=my-company** the DSA performs the following steps:

1. The DSA replaces the attribute type of the leftmost (most significant) RDN provided in the original bind request **sn** with the attribute type **commonName**.
2. The DSA uses the object **cn=users,dc=domain2,dc=loc1,dc=comany,dc=de** as base object.
3. The DSA sends a subtree search operation with the base object **cn=users,dc=domain2,dc=loc1,dc=comany,dc=de** and the filter **commonName=abele** to the external RACF LDAP server.
4. Let's assume that the remote server returns one entry with the DN **commonName=abele, cn=users,dc=domain2,dc=loc1,dc=company,dc=de**. The DirX Directory DSA uses this DN and the password provided by the user in the original bind request to forward the bind operation to the external RACF LDAP server for authentication.

   (If the external RACF LDAP server does not return one unique entry, the DirX Directory DSA rejects the bind.)

1.5.1.12. RACF_DYNAMIC_DN_MAP_TECH_BIND_DN

The **RACF_DYNAMIC_DN_MAP_TECH_BIND_DN** parameter specifies the DN of the account used to perform the dynamic mapping search operation. This account is referred to as the technical user.

This parameter is optional if **RACF_DYNAMIC_DN_MAPPING** is set to **TRUE**. Do not specify this parameter if **RACF_DYNAMIC_DN_MAPPING** is set to **FALSE**.

The syntax is as follows:

RACF DYNAMIC_DN_MAP_TECH_BIND_DN: *ldap_dn_string*

ldap_dn_string specifies the distinguished name in LDAP notation of the technical user performing the search operation sent to the external LDAP server to map the DN for subsequent external RACF authentications.

The default behavior in the absence of this parameter is to send the search request anonymously.

Example:

```
RACF DYNAMIC_DN_MAP_TECH_BIND_DN: cn=BindForwarder,o=my-company
```

The DSA performs a bind with the DN **cn=BindForwarder,o=my-company** and the

password retrieved from **RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE** and then uses that authentication to perform the subtree search operation for the external mapping of the user's DN.

1.5.1.13. RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE

The **RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE** parameter specifies the full pathname of the file containing the technical user that performs the dynamic mapping search operation.

The technical user's password must be entered as clear text with a text editor into the file specified by the pathname. When the DSA accesses this file for the first time, it converts the clear text password into an encrypted form and then rewrites the encrypted value into the file so that the password is no longer readable.

As a result, the account running the DirX Directory service requires read and write permission to the file specified by **RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE**.

The syntax is as follows:

RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE: *pathname*

Example:

```
RACF_DYNAMIC_DN_MAP_TECH_BIND_PWDFILE: /data/dirx/conf/techuser.pwd
```

The DSA performs a bind with the DN found in **RACF_DYNAMIC_DN_MAP_TECH_BIND_DN** and the password retrieved from **/data/dirx/conf/techuser.pwd** and then uses that authentication to perform the subtree search operation for the external mapping of the user's DN.

1.5.2. Example of an RACF External Authentication Configuration File

The following example of a **dirxextauth.cfg** file for the RACF external authentication service of a DirX Directory DSA supporting static DN mapping illustrates the effects of the configuration parameters:

```
# External Authentication Configuration File for RACF
#
# example settings for documentation purpose

# External Service
DIRX_EXT_AUTH_SERVICE: RACF

# RACF LDAP server Coordinates

# RACF_LDAP_HOSTS_1 is mandatory
```

```
# plus a second RACF backend for distributing the external bind
requests
# to different RACF backends
RACF_LDAP_HOSTS_1:auth_server.domain_a.com
RACF_LDAP_HOSTS_2:auth_server.domain_b.com
# RACF_LDAP_PORT is optional, default is 389
RACF_LDAP_PORT:389

# Use plain ldapV3 or LDAPS, default is FALSE
RACF_LDAP_SSL:FALSE

# Timeout set to 2 seconds
RACF_LDAP_TIMEOUT:  2

# Do a fallback to Local Authentication if RACF bind fails
EXT_AUTH_SEQUENCE:EXT-DIRX


# Generic Condition Parameters
# RACF forwarding is performed only for LDAP servers with IP
addresses
# starting with 123.45.67
EXT_LDAP_IP_CONDITION: 123.45.67

# RACF specific Conditions

# RACF_SUBTREE_CONDITION
# only bind requests from users situated in subtree
l=extern,dc=pqr,dc=com
# are forwarded
RACF_SUBTREE_CONDITION:l=extern,dc=pqr,dc=com

# RACF_ATTRIBUTE_CONDITION_HOST_n
# bind requests from users having an e-mail address
# with the OID 1.2.840.113549.1.9.1 attribute starting
# with the string “a” are forwarded to the RACF backend
# specified in RACF_LDAP_HOSTS_1 (auth_server.domain_a.com) and
# bind requests from users having an e-mail address
# with the OID 1.2.840.113549.1.9.1 attribute starting
# with the string “b” are forwarded to the RACF backend
# specified in RACF_LDAP_HOSTS_2 (auth_server.domain_b.com)
RACF_ATTRIBUTE_CONDITION_HOST_1:1.2.840.113549.1.9.1=a*
```

```
RACF_ATTRIBUTE_CONDITION_HOST_2:1.2.840.113549.1.9.1=b*

# Mapping Parameters

# RACF_MAP_RDN_ATTR_FROM/TO
# mapping the most significant RDN from the surname (OID 2.5.4.4) to
# the type “racfid” with the entry’s Surname value
RACF_MAP_RDN_ATTR_FROM:2.5.4.4
RACF_MAP_RDN_ATTR_TO:racfid

# RACF_MAP_SUBTREE_FROM/TO
# replace “l=extern” in incoming DN  by “ou=racf”
RACF_MAP_SUBTREE_FROM:l=extern
RACF_MAP_SUBTREE_TO:ou=racf
```

Example:

There are users in the DSA with the following values:

User abc:

Distinguished Name:	cn=abc,ou=xyz,l=extern,dc=pqr,dc=com
Common Name:	Abc
…	
Surname (OID: 2.5.4.4):	snOfabc
e-mail address (OID: 1.2.840.113549.1.9.1):	abc.snOfabc@xyz-company.com
Telephone Number	1234567

User bcd:

Distinguished Name:	cn=bcd,ou=xyz,l=extern,dc=pqr,dc=com
Common Name:	Bcd
…	
Surname (OID: 2.5.4.4):	snOfbcd
e-mail address (OID: 1.2.840.113549.1.9.1):	bcd.snOfbcd@xyz-company.com
Telephone Number	2345678

User cdf:

Distinguished Name:	cn=cdf,ou=xyz,l=extern,dc=pqr,dc=com
Common Name:	Cdf
…	
Surname (OID: 2.5.4.4):	snOfcdf
e-mail address (OID: 1.2.840.113549.1.9.1):	cdf.snOfcdf@xyz-company.com
Telephone Number	3456789

There is an LDAP server running on a host with the IP address 123.45.67.99. This LDAP server initiates the following bind request:

```
obj bind -user cn=abc,ou=xyz,l=extern,dc=pqr,dc=com \
-password xyz \
-authentication simple \
-protocol LDAPv3
```

The DirX Directory DSA processes all incoming bind requests as follows according to the parameter settings specified in the RACF external authentication configuration file shown in the previous example:

1. The DSA evaluates the LDAP server's IP address:
   The initial part of the LDAP server's IP address 123.45.67.99 matches the value of **EXT_LDAP_IP_CONDITION** 123.45.67.
2. The DSA evaluates whether the DN in the bind request is located under the subtree root specified in the **RACF_SUBTREE_CONDITION** parameter:
   The DN **cn=abc,ou=xyz,l=extern,dc=pqr,dc=com** is located under the subtree root **l=extern,dc=pqr,dc=com**.
3. The DSA evaluates whether the user's entry in the directory fulfils the attribute condition saved in the **RACF_ATTRIBUTE_CONDITION_HOST_1** parameter:
   The DSA reads the user's entry from the database. The entry's e-mail address attribute (OID: 1.2.840.113549.1.9.1) starts with **a** and therefore matches the attribute condition saved in **RACF_ATTRIBUTE_CONDITION_HOST_1** (1.2.840.113549.1.9.1=a*).
4. All conditions specified in the configuration file above are met. The DSA performs the mapping process:
   The bind credential DN is mapped from **cn=abc,ou=xyz,l=extern,dc=pqr,dc=com** to **racfid=snOfabc,ou=xyz,ou=racf,dc=pqr,dc=com** (see the **RACF_MAP_RDN_ATTR_FROM/TO** and **RACF_MAP_SUBTREE_FROM/TO** parameters specified in the mapping parameters section of the example configuration file.)
5. The DSA forwards the following bind request to the RACF system's LDAP server running on the host with the hostname **auth_server.domain_a.com** on port 389. (See the **RACF_LDAP_HOSTS_1** and **RACF_LDAP_PORT** parameters specified in the example configuration file.):

```
external bind with
    external authentication id:
racfid=sn0fabc,ou=xyz,ou=racf,dc=pqr,dc=com and
    password xyz
```

6. The DSA receives one of the following results from the RACF external authentication service:
    - The external RACF authentication succeeded and the DirX Directory service performs the bind.
    - The external RACF authentication reports an “invalid Credentials” error. The DirX Directory DSA tries to authenticate the user **cn=abc,ou=xyz,l=extern,dc=pqr,dc=com** according to the X.500 bind procedure (it compares the bind credentials password with the userPassword attribute of the entry) because the value of the **EXT_AUTH_SEQUENCE** is **EXT-DIRX**.

The DSA forwards the bind request for user **bcd** to the RACF backend **auth_server.domain_b.com** because the entry’s e-mail address attribute (OID: 1.2.840.113549.1.9.1) starts with a **b** and thus matches the attribute condition saved in **RACF_ATTRIBUTE_CONDITION_HOST_2** (1.2.840.113549.1.9.1=b*).

The DSA processes the bind request for user **cdf** locally because the entry’s e-mail address attribute (OID: 1.2.840.113549.1.9.1) starts with a **c** and so no attribute condition matches.

1.5.3. DSA Bind Procedure

The RACF external bind procedure is performed if the DSA receives a bind request and

- The credential choice is simple-unprotected (this applies in particular for all bind requests received by the DirX Directory service over LDAPv3)
- The RACF external password verification is enabled; that is, the external authentication configuration file **dirxextauth.cfg** exists and the **DIRX_EXT_AUTH_SERVICE** parameter value is **RACF**.

The normal X.500 bind procedure - check against the userPassword stored in the DSA’s database for the given entry - is performed in all other cases.

The RACF external bind procedure involves the following steps:

1. The DSA retrieves the IP address of the DUA (the LDAP server) and compares it to the **EXT_LDAP_IP_CONDITION** parameter. If there is no match, the X.500 bind procedure is performed.
2. The DSA extracts the X.500 distinguished name of the entry from the bind credentials and converts it into an LDAP DN. In the following steps, this DN is referred to as DN1.
3. The DSA checks DN1 to determine whether it completely contains the subtree name specified in the **RACF_SUBTREE_CONDITION** parameter. If not, the X.500 bind procedure is performed.

4. The DSA reads the attribute specified in the *RACF_ATTRIBUTE_CONDITION_**n* parameter from the entry DN1 and checks whether its value matches the specified conditions. If not, the X.500 bind procedure is performed.
5. The DSA reads the attribute specified in the **RACF_MAP_RDN_ATTR_FROM** parameter from the entry DN1. If DN1 has no such attribute or no value or more than one value the RACF bind request is assumed to have failed. Otherwise DN1 is converted to DN2 by replacing the most significant RDN of DN1 with the attribute type specified in **RACF_MAP_RDN_ATTR_TO** and the attribute value from the assertion read from the entry.
6. The DSA applies the subtree name mapping; that is, if the DSA finds the string specified in **RACF_MAP_SUBTREE_FROM** in DN2, it replaces it with the string specified in **RACF_MAP_SUBTREE_TO**. The result is DN3.
7. The DSA sends a bind request with DN3 and the original password to the configured RACF LDAP server.
8. The DSA examines the return code and error message received from the RACF LDAP server and maps to special DAP bind return codes.
9. If the bind procedure is successful, the DSA sets the access control identity of the bound user to DN1
10. If the bind procedure fails and the return code received from the RACF LDAP server is "invalid credentials" and the configured authentication sequence is "EXT-DIRX", the DSA retries an X.500 bind procedure with DN1.

The bind DN does not necessarily correspond to an entry in the corporate directory's DIT. If there is neither a **RACF_ATTRIBUTE_CONDITION** nor a **RACF_MAP_RDN_ATTR_FROM** specified, the DSA does not need to read an entry if RACF performs the authentication.

1.5.4. Bind Error Codes and Error Messages

During startup, the DSA evaluates the existence and contents of the external authentication configuration file. It then writes an event log that displays the status of the external authentication.

When the DSA parses the example configuration file given in the previous section, it writes an event entry to the Windows Event Viewer, for example:

```
External Password Authentication settings:
         Service:           "RACF"
         Status:            "Enabled"
         RACF LdapServer:     IP: "localhost" Port: 389  "(PLAIN-
LDAP)"
         Timeout (seconds)    30
         LDAPControl:         "(absent)" Critical: "FALSE"
       Forward Criteria:
```

```
        Bind Front-End       "all ldapservers"
        Bind DN contains:    "subtree l=extern,c=de"
        Attribute condition: "No Attribute required"
      Dynamic DN Mapping Settings:
        Search Base:         "<root>"
        Map Filter:          "o=my-company=<RDN-Value>"
      Auth-Service-Sequence: "External - DirX"
```

If the DSA detects missing, erroneous or inconsistent configuration information, it disables the RACF external authentication mechanism and writes an error log to the Event Viewer. For example:

```
External Password Authentication:
    Service:     "RACF"
    Status:      "Service Not Enabled"
    Reason:      "RACF_DYNAMIC_DN_MAPPING cannot be used together
with RACF_MAP_SUBTREE|RDN"
```

Client applications that perform the RACF bind operation via the LDAP protocol expect the RACF LDAP server to return certain LDAP result codes and LDAP error messages.

Consequently, the DirX Directory LDAP server examines the DAP bind result received from the DirX Directory DSA and re-maps the special DAP bind return codes to the “invalid credentials” LDAP error code and the error message originally returned by the RACF LDAP server.

If the RACF external authentication service is disabled, the LDAP client may receive one of the following LDAP result codes in addition to the usual bind result codes:

- "LDAP_INVALID_CREDENTIALS" and the LDAP error message "R000100 The password has expired."
- "LDAP_INVALID_CREDENTIALS" and the LDAP error message "R000101 The new password is not valid."
- "LDAP_INVALID_CREDENTIALS" and the LDAP error message "R000102 The userid is not allowed to perform the login."
- "LDAP_INVALID_CREDENTIALS" and the LDAP error message "R000103 The userid is not allowed to perform the login."
- "LDAP_INVALID_CREDENTIALS" and the LDAP error message "R000104 Credentials not correct."
- "LDAP_INVALID_CREDENTIALS" and the LDAP error message "R000105 One of the arguments is not valid."
- "LDAP_INVALID_CREDENTIALS" and the error message "Authenticated bind failed. Anonymous bind established." This will be returned when DirX Directory was the

authentication service used due to conditions that were not met.

1.5.5. External Authentication in DSA Audit

The bind record in the DSA audit always contains the original DN as the bind requestor. When evaluating the binary audit with the **dirxauddecode** or **dirxaudstatistics -v** option, additional information is added to the audit record, including the "Bind Forwarding Info" section, which contains the mapped DN used for the forwarded bind in LDAP notation and some timestamps.

The following example shows a bind record of a forwarded bind operation evaluated with the **-v** option:

```
External ############# RECORD NUMBER 023799 #############
Bind-Id:        0x30d3000a
Start Time:             Fri Apr 1 11:31:13.351000 2016
End Time:               Fri Apr 1 11:31:13.368000 2016
Concurrency:            6
BT Usage:               7 Conns, 6 Ops
Duration:               0.017000 sec
Protocol:       DAP (Responder)
OP-Name:        Con12499_Op0
Operation:      BIND
Role:           Responder
AuthMech:       Simple
Bind-Requestor:     /O=my-company/OU=extracf/CN=racftest1
IP-address:     12.34.56.789
OpResCTXSize:       32 kB
TotalCTXSize:       13 MB  (HWM: 14 MB)
Bind Forwarding Info:
 Mapped DN:              cn=mappedTo1,ou=extMappedTo,o=My-Company
 External duration:      conn/dyn map/bind 0.000 / 0.012 / 0.004  sec
Result:         Success (0 Entries, 0 Attrs, 0 Vals, 0 Bytes)
########## END RECORD NUMBER 023799 ############
```

1.5.6. Restrictions and Clarifications

The following restrictions apply to the RACF external authentication in DirX Directory.

- The access to the RACF LDAP server is not performed over SSL/TSL-protected LDAP protocol.
- The RACF LDAP server invokes a change-password-operation depending on the userPassword provided in the bind request. This is transparent for the DirX Directory DSA; such a bind will not affect userPassword attributes stored in the DSA database. To

change userPasswords in DirX Directory is to send a modifyEntry request.

- The content of the external authentication configuration file is expected to be rather static. Therefore the parser is not very elaborate and forces the administrator to use dotted OID notations for attribute types.
- For each condition type, the **dirxextauth.cfg** file parser supports only one configuration parameter; there is no AND or OR combination of multiple attribute conditions supported.
- Attribute values that are subject to subtree mapping must not contain characters that are outside the printable string range.
- Attribute types referred to by the external authentication configuration file in any context must have one of the following syntaxes:
- Printable String
- Directory String
- Ia5 String
- Attribute types referred to by the external authentication configuration file in any context must have a caseIgnoreMatching Rule for equality and substring matching.

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.