# z/VSE LDAP Query Callable Module

The following document describes the features of the z/VSE LDAP Query Callable Module and how to use it.

## *Overview*

The z/VSE LDAP Query Callable Module allows you to programmatically query an LDAP server from within your programs and retrieve attributes of an LDAP user. You can either call the z/VSE LDAP Query Callable Module directly (i.e. via an COBOL external call) , or via EXEC CICS LINK when running under CICS.

The z/VSE LDAP Query Callable Module can be used on z/VSE 4.2 or later. It does not work on earlier z/VSE releases because it uses the LDAP Client module that is only provided since z/VSE 4.2 or later as part of the z/VSE Signon Support.

The z/VSE LDAP Query Callable Module uses the same configuration as the z/VSE LDAP Signon support. That means you need to create an LDAP configuration using skeleton SKLDCFG in ICCF library 59 in order to use it. In the configuration you provide information like LDAP server addresses, encryption settings when using SSL as well as information about the authentication method used with your LDAP environment. Setting up the LDAP configuration requires some knowledge about your LDAP environment and schema. Talk to you LDAP server administrators to obtain the required information.

If you with to use the z/VSE LDAP Query Callable Module without using the z/VSE LDAP Sigon Support, you still need to create an LDAP configuration, but turn LDAP Signon Support off in the Flags field in the configuration.

## *Installation*

To install the z/VSE LDAP Query Callable Module simply send the provided catalog job IESLDGAB.BJB into the z/VSE reader in BINARY and LRECL=80. The BJB will then catalog the two programs IESLDGAB and IESLDGAC into PRD1.BASE.

If you plan to use the z/VSE LDAP Query Callable Module under CICS, you need to define the 2 programs in CICS using CEDA DEFINE PROGRAM. The Language is C, the Datalocation is ANY.

The z/VSE LDAP Query Callable Module uses the LDAP Client module IESLDAPC which performs LE/C socket calls to the TCP/IP stack. You must make sure that the TCP/IP library is in the LIBDEF when executing the program. Also make sure that you set the SYSID to point to the correct TCP/IP stack (via // OPTION SYSPARM=’nn’).

 

## *Programming Interface (API)*

You can invoke the z/VSE LDAP Query Callable Module either in batch or from CICS. In batch you perform a direct call (i.e. COBOL external call) to program IESLDGAB and pass a parameter area to the program. IESLDGAB can be called from any LE enabled language (i.e. COBOL, PL/1, C and LE conforming Assembler). IESLDGAB must be called with just one parameter which is (a pointer to) the parameter area. Under CICS, you perform an EXEC CICS LINK to program IESLDGAC and pass the same parameter area as COMMAREA. Since CICS takes care about language switching, you can LINK to IESLDGAC from any programming language supported by CICS (COBOL, PL/1, C, Assembler, REXX, etc.). The same parameter area layout is used for calls from batch or CICS.

The parameter area has a variable length format. Its length is dependent on the length and number of some fields within the area. You can choose which length and number to use for these fields, to suite the expected result data best.

Please see below the layout of the parameter area in COBOL:

```
01 LDGA-AREA.
  03 AREA-LENGTH         PIC S9(9) BINARY.  <-- In: Length of the Area in Bytes
  03 USER-ID             PIC X(64).         <-- In: LDAP user ID to get attributes for
  03 SEARCH-FILTER       PIC X(128).        <-- In: Additional Search filter or blanks
  03 RET-CODE            PIC S9(9) BINARY.  <-- Out: Return code
  03 LDAP-CODE           PIC S9(9) BINARY.  <-- Out: LDAP Return code
  03 ATTR-COUNT          PIC S9(4) BINARY.  <-- In: Number of attr entries following
  03 ATTR-ENTRY OCCURS x TIMES.
    05 ATTR-NAME         PIC X(64).         <-- In: Name of Attribute to get
    05 VALUE-LENGTH      PIC S9(4) BINARY.  <-- In: Length of ATTR-VALUE
    05 VALUE-COUNT       PIC S9(4) BINARY.  <-- In/out: Number of Values following
    05 VALUE-ENTRY OCCURS y TIMES.
      07 ATTR-VALUE      PIC X(n).          <-- Out: Attribues Values(s).
                                                Length (n) must match the VALUE-LENGTH
  03 CONFIG-NAME         PIX X(8)           <-- In: Optional: Name of the LDAP Config

01 IESLDGAB         PIC X(8) VALUE 'IESLDGAB'.

…
Fill the parameter area here
…
CALL IESLDGAB USING BY REFERENCE LDGA-AREA.
```

| Field name | Description |
|---|---|
| AREA-LENGTH | This field must be set by your program to the total length in bytes of the parameter area. Under CICS, the COMMAREA length you specify at the EXEC CICS LINK command overrides this field. Under CICS the parameter area can not exceed 32K in size due to the size limitation of the CICS COMMAREA. |
| USER-ID | This field must be set by your program to the LDAP user-ID to query attributes for. Please note that the LDAP user-id may be case sensitive. |
| SERACH-FILTER | This field can be set by your program to an addition LDAP |

 

| | |
|---|---|
| | search filter to be used for the query or all blanks if no search filter is to be used. This field has the same meaning as the search filter field in the LDAP configuration. Please note that the search filter may be case sensitive. Note: The search filter specification in the configuration is ignored. |
| RET-CODE | This field contains the return code upon return. See below for possible return codes and their meaning. |
| LDAP-CODE | This field contains the LDAP return code upon return, if the LDAP server has returned an error. It is set ir field RET-CODE contain LDGA_RC_LDAP_ERROR (7). Possible LDAP return codes can be found in member IESLDAPH.H in PRD1.BASE. |
| ATTR-COUNT | This field must be set by your program to the number of LDAP attributes to query. It also specifies the count used for the OCCURS field ATTR-ENTRY. |
| ATTR-ENTRY | This OCCURS field is a table of LDAP Attribute entries built by the following fields. The number of occurrences must match the contents of field ATTR-COUNT. |
| ATTR-NAME | This field must be set by your program to the LDAP attribute name to query. Please note that the LDAP attribute names may be case sensitive. |
| VALUE-LENGTH | This field must be set by your program to the desired length of the ATTR-VALUE field. The length should be large enough to hold the maximum expected attribute value length. |
| VALUE-COUNT | This field must be set by your program to the desired attribute value count. An LDAP attribute can have multiple values (i.e. the attribute can occur multiple times in an LDAP entry). The count should be set to the maximum number of values expected. On return, this field will contain the actual number of values returned. Please note that the number of occurrences of the ATTR-VALUE does not change if this field contains a different value as on entry. The number of occurrences will always stay as many as specified on entry. If more values are available but could not be returned due to the value specified in VALUE-COUNT, field RET-CODE will be set to LDGA_WARN_MORE_VALUES, but up to VALUE-COUNT values are returned. |
| ATTR-ENTRY | This OCCURS field is a table of LDAP attribute values. The number of occurrences must match the contents of field VALUE-COUNT. |
| ATTR-VALUE | This field will contain the attribute value on return, or blanks if it is not set. This field can occur multiple times, dependent on field ATTR-ENTRY. Please note that the attribute values may be returned in mixed case. If one or multiple values exceeded the length specified in field VALUE-LENGTH, the RET-CODE is set to LDGA_WARN_VALUE_TRUNC, but the values are returned up to the specified length (truncated). |

 

| | |
|---|---|
| CONFIF-NAME | This field is optional. If set, then field AREA-LENGTH must also include this field. Specifies the name of the LDAP configuration phase. If this field is not present or all blanks, then the default 'IESLDCFG' is used. |

## *Return codes*

The following return codes can be returned in field RET-CODE:

| **Return Code** | **Value** | **Description** |
|---|---|---|
| LDGA_RC_OK | 0 | No error |
| LDGA_RC_FETCH_FAILED | 1 | Fetch of IESLDGAB has bailed (CICS only) |
| LDGA_RC_INVALID_PARAM | 2 | Invalid parameter, i.e. parameter area is too small or not specified at all. |
| LDGA_RC_BUFFER_TOO_SMALL | 3 | The parameter area length (as specified in field AREA-LENGTH) was too small to hold the desired length and number of attributes and values. |
| LDGA_RC_CFG_ERROR | 4 | Error loading the LDAP configuration |
| LDGA_RC_INVALID_AUTH | 5 | Invalid authentication method specified in LDAP configuration. |
| LDGA_RC_SSL_INIT_FAILED | 6 | SSL initialization has failed |
| LDGA_RC_LDAP_ERROR | 7 | An LDAP operation has failed. See field LDAP-CODE for the return code passed back from the LDAP client. |
| LDGA_RC_BIND_FAILED | 8 | The LDAP Bind operation with the generic or anonymous user has failed. See field LDAP-CODE for the return code passed back from the LDAP client. |
| LDGA_RC_USER_NOT_FOUND | 9 | The user-id supplied in field USER-ID was not found by the LDAP server. |
| LDGA_WARN_MORE_VALUES | 10 | This is not an error, just a warning. For one or multiple attributes more values are available but could not be returned due to the value specified in VALUE-COUNT. Up to VALUE-COUNT values are returned. |
| LDGA_WARN_VALUE_TRUNC | 11 | This is not an error, just a warning. One or multiple values to be returned exceeded the length specified in field VALUE-LENGTH. The values are returned up to the specified length (truncated). |

 

## *Examples*

Please find two examples provided as part of this package: COBTEST.cbl and COBTEST2.cbl. Both examples are using COBOL and use a direct call to IESLDGAB. The same code can be used under CICS with just a little modification (use EXEC CICS LINK instead).

COBTEST.cbl shows how to query 2 different attributes of an LDAP user by dynamically filling the ATTR-ENTRY table using subscripting. COBTEST2.cbl performs the same query, but uses specific field names in the parameter area. Using subscripting might be more flexible to query different attributes, where using specific field names might be better readable code.

## *Remarks*

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

## *Trademarks*

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

CICS IBM Language Environment VSE/ESA

Other company, product, or service names, may be the trademarks or service marks of others.

## *Comments and Questions*

Comments or questions on this documentation are welcome. Please send your comments to:

zvse@de.ibm.com

 