Certificate Display Programmer Documentation

September 1, 2023

Introduction

This document describes a component named CERTDISP developed by Charles Mills Consulting (CMC) for NewEra. CICLIENT is intended to be called from NewEra ICEdirect Certificate Intelligence ("CI"). The intended audience for this document is an experienced Rexx language developer. CMC provides no end-user documentation for CERTDISP.

All trademarks used in this document are the exclusive property of their respective owners. No association with CMC is implied.

CERTDISP is a z/OS program written primarily in IBM XLC C++. CERTDISP is a comprehensive certificate display program.

CEERTDISP takes as input a parameter identifying a one or more certificates, and produces as output (1) a certificate display equivalent to that output by CICLIENT; and (2) certificate characteristics. Both outputs are returned to the caller in Rexx variables, similar to CICLIENT.

Input Parameters

The CI input to CERTDISP is a single character string passed as argument one of a z/OS Rexx function call. The individual CERTDISP parameters are separated in the argument string by one or more blanks. (Note that the well-known 100-character parameter limit applies only to JCL jobsteps and TSO commands and is not relevant to Rexx functions.)

The input parameter format is modeled on BPXWDYN.
https://www.ibm.com/docs/en/zos/2.3.0?topic=output-keywords

Input parameters are in one of two formats:

Keyword, for example, DEFAULT

Keyword plus argument(s) separated by commas plus one or more blanks following the comma, for example FILE(*datasetname*).

Any argument *may* be quoted; any argument that contains a blank or comma *must* be quoted. Any unquoted embedded parentheses must be properly nested: FILE(MYDSN(MEMBER)) is correct; LABEL(SOME(NAME) is not. Single quotes (apostrophes) or double quotes (standard quotation marks) may be used to quote an argument. As with BPXWDYN, doubled interior quotes are not supported. Relaxing the BPXWDYN restriction, a quote character within quotes is permitted so long as it is not immediately followed by a comma, blank or a right parenthesis: `"Paul's Cert"` and `'Paul's Cert'` are correct; ~~`'Paul''s Cert'`~~ is not (unless, of course, the value actually contains two apostrophes).

Parameters are processed from left to right, with subsequent keywords and arguments overriding earlier keywords and arguments. Keywords and arguments are case-independent: FIPS, fips and Fips are all logically equivalent. Arguments are treated as upper case, except where the context demands otherwise, such as for UNIX file names.

Some keywords have “NO” variants, such as NOVERBOSE or NOTRACE. The “NO” variants negate any preceding corresponding keyword. The “NO” variants of keywords do not take any arguments.

Comments

Comments have the format

/* any sequence of characters */

The input parameter string may contain comments anywhere that blanks are permitted, for example

FILE(MY.DATA.SET) /* Just do default */ DEFAULT

The following parameters are accepted:

Keyword	Format	Description	Default
CHAIN NOCHAIN	Keyword	May only be coded with KEYRING and with LABEL or DEFAULT. Specifies that CERTDISP is to display the certificates in the chain of trust for the indicated certificate.	Only a single certificate is displayed.
DEFAULT NODEFAULT	Keyword	Specifies that CERTDISP is to display the default certificate in the specified KEYRING. May only be coded with a real keyring, and may not be coded with a PKCS 11 token.	None. If LABEL and DEFAULT are both omitted, then all certificates in the keyring are displayed.
EXPIREWARN	Keyword plus argument	Specifies the number of days for a certificate expiration warning threshold. If any processed certificate will expire within the specified number of days then a warning message is logged. Specify a value between 1 and 180 days, or to completely disable expiration warnings, specify 0.	7 days
FILE	Keyword plus argument	Specifies the file that contains the certificate. Mutually exclusive with KEYRING, ID and STREAM. Specify a data set name, a data set name and member name, a ddname prefixed with DD:, or a zFS file name beginning with slash (/).	

Keyword	**Format**	**Description**	**Default**
FIPS NOFIPS	Keyword	Specifies FIPS mode. FIPS mode is a more restrictive TLS mode that conforms to NIST standard FIPS 140-2. FIPS mode requires a FIPS-compliant certificate database. Specifying FIPS confirms that any referenced KEYRING is FIPS compliant.	NOFIPS
ID	Keyword plus argument	ID specifies that certificates belonging to the specified userid are to be displayed. ID(*userid*) is equivalent in all respects to KEYRING(*userid*/*)	
KEYRING	Keyword plus argument	Specifies the name of the ESM keyring, PKCS 11 token or gskkyman database, in the conventional format, either userid/ringname or /complete/gsk/database/path. Agskkyman database must be accessible via a stash file following the standard naming convention. Keyring names are case-sensitive. KEYRING is mutually exclusive with FILE and STREAM.	*AUTH*/*, the CERTAUTH virtual key ring
KEYLENWARN	Keyword plus argument	Specify the key strength level for which a warning will be produced, NONE, LOW, MEDIUM or HIGH. Key strengths are rated in accordance with https://www.ibm.com/docs/en/zos/2.5.0?topic=certificates-racdcert-gencert-generate-certificate	LOW
LABEL NOLABEL	Keyword plus argument	Specifies the label of the certificate to be displayed. Label names are case-sensitive.	None. If LABEL and DEFAULT are both omitted, then all certificates in the keyring or gsk database are displayed.
STATUS	Keyword plus argument	Specifies the status of the certificates to be displayed from the specified KEYRING or gskkyman database: HIGHTRUST, NOTRUST, TRUST or ANY	ANY
STREAM	Keyword plus argument	Specifies a PEM- or DER-encoded stream representing one or more certificates, a certificate request, or a PKCS7 package. STREAM is mutually exclusive with FILE, KEYRING and ID.	STREAM("-----BEGIN ... etc.")

Keyword	Format	Description	Default
TRACE NOTRACE	Keyword plus optional arguments	Specifies that CERTDISP is to invoke the System SSL trace, format it, and return the formatted trace in the CERTDISP_TRACE. Stem. You may optionally specify two arguments, *maxlines* and *level*. *MaxIines* specifies the maximum number of lines of trace data to be returned; *level* is a decimal number between 1 and 63 specifying the sum of the values for the types of events to be traced. The values are 1 –Trace function entry 2 – Trace function exit 4 – Trace errors 8 – Include informational messages 16 – Include EBCDIC data dumps 32 – Include ASCII data dumps The values may be specified or omitted, and if specified may be specified in either order. Values of 255 or less are assumed to specify *level*; values greater than 255 are assumed to specify *maxlines*. The System SSL trace is probably not of much value in CERTDISP.	5000,15
VERBOSE NOVERBOSE	Keyword	Specifies that CERTDISP is to produce additional diagnostic messages. See Status Log below.	No additional diagnostic messages.

See Sample Invocation below for an example of a valid parameter string:

Output

The output of CERTDISP is a set of Rexx variables and a standard Rexx function return value.

The variables consist of stem variables containing textual messages intended for direct user display by CI; and a set of "Return Variables." Return Variables are documented below. CERTDISP follows the typical Rexx convention in which *stem*.0 contains the count of messages and *stem*.1, *stem*.2, etc. contain the actual messages.

Status Log

The log of CERTDISP activity is returned in a compound variable with a stem of CERTDISP_LOG. It consists of status, informational, diagnostic, warning and error messages.

Each message in CERTDISP_LOG begins with a one-character description code followed by a blank. It is intended that NewEra will choose whether to display the code or omit it from the display and instead use it to determine display characteristics (color, bold font, etc. – or omitted entirely). The codes will be as follows:

I	Informational message for the customer
D	Detailed status or diagnostic message, including Verbose output
G	Suggestion message; suggested user remediation for reported errors
W	Warning message
E	Error in protocol, certificate or cipher or similar
S	Invocation parameter or “should not occur” error from System SSL or z/OS
C	Critical error; CERTDISP abnormally terminated

It is CMC’s intention to make all messages as informative as possible. At a minimum, all error messages include the exact name of the failing function, any parameters necessary to pinpoint a specific invocation of that function, and a gsk_strerror() textual description of any error code. Additional information will be returned as agreed upon by NewEra and CMC.

See https://www.ibm.com/docs/en/zos/2.5.0?topic=reference-gsk-secure-socket-init Results for examples of the level of error detail that is returned. CERTDISP expands on the IBM error message where possible and appropriate: “Make sure of blah-blah-blah. Sometimes this error is caused by X or by Y.”

In addition the status log is written to a single data set, *userid*.NEWERA.CERTDISP.LOG. This hard data set allows for debugging in those situations where an abnormal termination of CERTDISP precludes the setting of the LOG stem variable. (The data set is re-used and overwritten on each invocation.)

Certificate Information

Certificate information is returned in a set of compound variables named CERTDISP_CERT.*symbol.n* and CERTDISP_RTN_CERT.*symbol.name*. Each *symbol.n* variable contains one line of certificate information intended for direct display by CI. Symbol is obtained from the certificate index described immediately below. The CERTDISP_RTN_CERT.*symbol.name* variables are described under Return Variables below.

The certificate information compound variables are indexed in a compound variable with a stem of CERTDISP_CERTINDEX. Each variable CERTDISP_CERTINDEX.*n* consists of a symbol name followed by one or more blanks and the name of a certificate. For example, CERTDISP_CERTINDEX.1 might contain

```
CHAIN1    GeoTrust TLS DV RSA Mixed SHA256 2020 CA-1
```

which would indicate that information for the certificate named GeoTrust TLS DV RSA Mixed SHA256 2020 CA-1 would be found in the compound variables named

CERTDISP_CERT.CHAIN1.*n*. It is intended that CI would display the list of certificate names, and in response to the user's clicking on one of the names, display the information in the variables.

The following Rexx code is intended to show the relationship among the various certificate stems. (It is not intended for direct inclusion in CI.) It would print the certificate information for all available certificates.

```
DO I = 1 TO CERTDISP_CERTINDEX.0
  PARSE VAR CERTDISP_CERTINDEX.i symbol name
  SAY "Cert info for" name
  DO c = 1 TO CERTDISP_CERT.symbol.0
    SAY CERTDISP_CERT.symbol.c
    END c
  END i
```

Trace Data

If TRACE is specified in the parameters then trace data is returned in compound variables with a stem of CERTDISP_TRACE. The returned trace data looks like

```
11/17/2022-12:02:46 Thd-1 INFO cms_validate_certificate_mode_int(): ...
11/17/2022-12:02:46 Thd-1 ENTRY check_cert_extensions_3280_and_later(): ...
11/17/2022-12:02:46 Thd-1 ENTRY gsk_decode_certificate_extension(): ...
etc.
```

Rexx Function Result

CERTDISP returns a standard Rexx function result as follows:

0	Complete success; only I, D and G messages logged
4	Success with one or more W messages logged
8	E error message logged
12	S error message logged
16	C error message logged

Return Code

CERTDISP always returns a return code of 0, except in the case of an unhandled abnormal termination (ABEND). (Non-zero return codes from external functions generally cause the calling Rexx routine to terminate.)

Return Variables

The Rexx variables named in the table below are returned to the calling program by CERTDISP. All of the named variables are always set by CERTDISP, assuming a CERTDISP function result of 0, 4, 8 or 12. That is, the Rexx program caller need not be concerned with the possibility of an undefined variable, assuming one of the above function results. The Rexx caller should make certain that the returned value is not a zero-length string before using the variable in an arithmetic expression.

Variable Names

The left-hand part of each name is CERTDISP_RTN_. In other words, CERTDISP_RTN_CIPHER is the full name of the variable listed in the table as CIPHER.

Certificate-Related Variables

The names containing CERT are certificate-related compound variables and are returned for each certificate processed. The portion of the name denoted below as *symbol* is the same value as *symbol* documented above under Certificate Information.

Right-hand Portion of Name	**Description**	**Example Value**
CERT.*symbol*.ALGORITHM_KEY	Public key encryption algorithm	rsaEncryption
CERT.*symbol*.ALGORITHM_KEY_ENUM	Public key encryption algorithm enumerator	
CERT.*symbol*.ALGORITHM_KEY_LEN	The length of the key in bits	2048
CERT.*symbol*.ALGORITHM_KEY_STRENGTH	The strength of the key	HIGH
CERT.*symbol*.ALGORITHM_SIG	Signature encryption algorithm	sha256WithRsaEncryption
CERT.*symbol*.ALGORITHM_SIG_ENUM	Signature encryption algorithm enumerator	0401
CERT.*symbol*.DEFAULT	Only for real (non-virtual) RACF keyrings. Whether the certificate is the default certificate on the ring.-	YES
CERT.symbol.HAS_PRIVATE_KEY	Indicates whether the certificate has a private key.	NO
CERT.*symbol*.ISSUER	Issuer Distinguished Name	CN=DigiCert Global Root G2, OU=www.digicert.com, O=DigiCert Inc, C=US
CERT.*symbol*.ISSUER_CN	Issuer CN	DigiCert TLS RSA SHA256 2020 CA1
CERT.*symbol*.LABEL	The certificate label, for certificates from a gskkyman or RACF database.	DigiCert Global Root CA
CERT.*symbol*.NOTAFTER	Expiration date and time for the indicated certificate in ISO 8601 format.	2023-09-26T23:59:59

Right-hand Portion of Name	Description	Example Value
CERT.*symbol*.NOTAFTER_DAYS	Number of whole and partial days of remaining validity (seconds of remaining validity divided by 86400 and rounded up to the next integer). An expired certificate will return a value of 0.	30
CERT.*symbol*.SELFSIGNED	Is certificate self-signed?	YES
CERT.*symbol*.STREAM_SIZE	Size of the certificate x.509 stream in bytes, expressed as a decimal number	673
CERT.*symbol*.SUBJECT	Subject Distinguished Name	CN=DigiCert Global Root G2, OU=www.digicert.com, O=DigiCert Inc, C=US
CERT.*symbol*.SUBJECT_CN	Subject CN	www.ibm.com
CERT.*symbol*.TRUST_STATUS	Only for RACF keyring certificates. The trust status of the certificate.	HIGHTRUST NOTRUST TRUST
CERT.*symbol*.USAGE	Only for RACF keyring certificates. The usage of the certificate.	CERTAUTH PERSONAL SITE
CERT.*symbol*.USERID	Only for RACF keyring certificates. The userid of the owner of the certificate.	PROBI1 irrcerta
FILE_SIZE	Size of the entire file x.509 stream in bytes as a decimal number.	2680
FILE_TIMESTAMP	The system timestamp of the file, if available, in ISO 8601 format.	2023-07-09T06:13:22
OKAY_FOR_ADD	Indicates whether file is suitable for RACDCERT ADD into RACF. Indicates a QSAM dataset with RECFM=VB	YES NO

Sample Invocation

```
PARM = "KEYRING(CMILL1/MYRING) DEFAULT CHAIN VERBOSE"
RET = CERTDISP(PARM)
IF RET <> 0 THEN ...
```

Sample Program

A sample calling program written in z/OS Rexx is included as part of the CERTDISP deliverables.

Appendix

Sample Demonstration FILEs

File Name	Description
/u/cmill1/gsk/cicswest.charlesmillsconsulting.com.crt	A single certificate in PEM format.
/u/cmill1/gsk/DigiCert Global Root CA.pem	A single certificate in PEM format.
/u/cmill1/gsk/serverCA.cer	A single certificate in DER format.
/u/cmill1/gsk/cicswest.charlesmillsconsulting.com.p7b	Three certificates in a PKCS#7 bundle.
CMILL1.CERTDISP.CERTS(DERTEXT)	A DER format certificate erroneously uploaded in text format.
CMILL1.CERTDISP.CERTS(PEMASCII)	A PEM format certificate erroneously uploaded in binary format.
CMILL1.CERTDISP.CERTS(PEMMULTI)	Three "stacked" PEM format certificates.
CMILL1.DIGICERT.GLOBAL.ROOT.CA	A single certificate in PEM format.
CMILL1.DIGICERT.GLOBAL.ROOT.G2	A single certificate in PEM format.

Gskkyman Key Databases

Note: The password for all gskkyman key databases is "password". These key databases should not be used to store any private keys that have any security implications.

Name	Description
/u/cmill1/gsk/kyman_primary.kdb	Primary test key database. Intended to contain most common demonstration CA root certificates.
/u/cmill1/gsk/kyman_bad.kdb	Test of "bad" cases. Contains a purported CA root for Go Daddy Secure Certificate Authority - G2 and DigiCert Global Root CA