Zowe CLI Help

Welcome to Zowe CLI!

Zowe CLI is a command line interface (CLI) that provides a simple and streamlined way to interact with IBM z/OS.

For additional Zowe CLI documentation, visit https://zowe.github.io/docs-site.

For Zowe CLI support, visit https://zowe.org.

Table of Contents

- backout
  - package
- cast
  - package
- create
  - package
- delete | del
  - element
  - package
- deny
  - package
- execute
  - package
- generate | gen
  - element
- list
  - instances
  - environments
  - stages
  - systems
  - subsystems
  - types
  - packages
  - elements
- move | mv
  - element
- print | p
  - element
  - components
- reset
  - package
- retrieve | ret
  - element
- signin | si
  - element
- submit
  - package
  - scl
- transfer | tr
  - element
- update
  - package
  - element

    - view
        - element
- file-master-plus | fmp
    - copy | c
        - data-set
    - create | cre
        - physical-sequential
        - large-format-sequential
        - partitioned-data-set
        - partitioned-data-set-extended
        - vsam-ksds
        - vsam-esds
        - vsam-lds
        - vsam-rrds
        - vsam-vrrds
        - like-model
    - delete | del
        - data-set
    - populate | pop
        - data-set
    - rename | ren
        - data-set
- ops
    - disable
        - rule
    - enable
        - rule
    - show
        - resource
        - rule
    - start
        - resource
    - stop
        - resource
- plugins
    - install
    - list
    - uninstall
    - update
    - validate
- profiles
    - create | cre
        - zosmf-profile

        - tso-profile
        - cics-profile
        - endevor-profile
        - endevor-location-profile
        - fmp-profile
        - ops-profile
        - db2-profile
    - update | upd
        - zosmf-profile
        - tso-profile
        - cics-profile
        - endevor-profile
        - endevor-location-profile
        - fmp-profile
        - ops-profile
        - db2-profile
    - delete | rm
        - zosmf-profile
        - tso-profile
        - cics-profile
        - endevor-profile
        - endevor-location-profile
        - fmp-profile
        - ops-profile
        - db2-profile
    - list | ls
        - zosmf-profiles
        - tso-profiles
        - cics-profiles
        - endevor-profiles
        - endevor-location-profiles
        - fmp-profiles
        - ops-profiles
        - db2-profiles
    - set-default | set
        - zosmf-profile
        - tso-profile
        - cics-profile
        - endevor-profile
        - endevor-location-profile
        - fmp-profile
        - ops-profile
        - db2-profile

    - validate | val
        - endevor-profile
        - fmp-profile
- provisioning | pv
    - list | ls
        - template-info
        - catalog-templates
        - instance-info
        - instance-variables
        - registry-instances
    - provision | prov
        - template
    - perform | perf
        - action
    - delete | del
        - instance
- zos-console | console
    - collect
        - sync-responses
    - issue
        - command
- zos-files | files
    - create | cre
        - data-set-sequential
        - data-set-partitioned
        - data-set-binary
        - data-set-c
        - data-set-classic
        - data-set-vsam
    - delete | del
        - data-set
        - data-set-vsam
    - invoke | call
        - ams-statements
        - ams-file
    - download | dl
        - data-set
        - all-members
        - uss-file
    - list | ls
        - all-members
        - data-set
    - upload | ul

cics

Interact with IBM CICS programs and transactions.

define | def

Define new resources (for example, programs) to CICS through IBM CMCI.

program

Define a new program to CICS.

Usage

zowe cics define program [options]

Positional Arguments

- `programName` *(string)*
  - The name of the new program to define. The maximum length of the program name is eight characters.
- `csdGroup` *(string)*
  - The CICS system definition (CSD) Group for the new program that you want to define. The maximum length of the group name is eight characters.

Options

- `--region-name` *(string)*
  - The CICS region name to which to define the new program
- `--cics-plex` *(string)*
  - The name of the CICSPlex to which to define the new program

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
  - The name of a (cics) profile to load for this command execution.

Examples

*- Define a program named PGM123 to the region name MYREGION in the CSD group MYGRP:

- `$ zowe cics define program PGM123 MYGRP --region-name MYREGION`

transaction

Define a new transaction to CICS.

Usage

zowe cics define transaction [options]

Positional Arguments

- `transactionName` *(string)*
  - The name of the new transaction to define. The maximum length of the transaction name is four characters.
- `programName` *(string)*
  - The name of the program that the transaction uses. The maximum length of the program name is eight characters.
- `csdGroup` *(string)*
  - The CICS system definition (CSD) Group for the new transaction that you want to define. The maximum length of the group name is eight characters.

Options

- `--region-name` *(string)*
  - The CICS region name to which to define the new transaction
- `--cics-plex` *(string)*
  - The name of the CICSPlex to which to define the new transaction

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
  - The name of a (cics) profile to load for this command execution.

Examples

*- Define a transaction named TRN1 for the program named PGM123 to the region named MYREGION in the CSD group MYGRP:

- `$ zowe cics define transaction TRN1 PGM123 MYGRP --region-name MYREGION`

delete | del

Delete resources (for example, programs) from CICS through IBM CMCI.

program

Delete a program from CICS.

Usage

zowe cics delete program [options]

Positional Arguments

- `programName` *(string)*
  - The name of the program to delete. The maximum length of the program name is eight characters.
- `csdGroup` *(string)*
  - The CICS system definition (CSD) Group for the program that you want to delete. The maximum length of the group name is eight characters.

Options

- `--region-name` *(string)*
  - The CICS region name from which to delete the program
- `--cics-plex` *(string)*
  - The name of the CICSPlex from which to delete the program

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
  - The name of a (cics) profile to load for this command execution.

Examples

*- Delete a program named PGM123 from the region named MYREGION:

- `$ zowe cics delete program PGM123 --region-name MYREGION`

transaction

Delete a transaction from CICS.

Usage

zowe cics delete transaction [options]

Positional Arguments

- `transactionName` *(string)*
    - The name of the transaction to delete. The maximum length of the transaction name is four characters.
- `csdGroup` *(string)*
    - The CICS system definition (CSD) Group for the new transaction that you want to delete. The maximum length of the group name is eight characters.

Options

- `--region-name` *(string)*
    - The CICS region name from which to delete the transaction
- `--cics-plex` *(string)*
    - The name of the CICSPlex from which to delete the transaction

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
    - The name of a (cics) profile to load for this command execution.

Examples

*- Delete a transaction named TRN1 from the region named MYREGION:

- `$ zowe cics delete transaction TRN1 MYGRP --region-name MYREGION`

discard | dis

Discard resources (for example, programs) from CICS through IBM CMCI.

program

Discard a program from CICS.

Usage

zowe cics discard program [options]

Positional Arguments

- `programName` *(string)*

    - The name of the program to discard. The maximum length of the program name is eight characters.

Options

- `--region-name` *(string)*
    - The CICS region name from which to discard the program
- `--cics-plex` *(string)*
    - The name of the CICSPlex from which to discard the program

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
    - The name of a (cics) profile to load for this command execution.

Examples

*- Discard a program named PGM123 from the region named MYREGION:

- `$ zowe cics discard program PGM123 --region-name MYREGION`

transaction

Discard a transaction from CICS.

Usage

zowe cics discard transaction [options]

Positional Arguments

- `transactionName` *(string)*
    - The name of the transaction to discard. The maximum length of the transaction name is four characters.

Options

- `--region-name` *(string)*
    - The CICS region name from which to discard the transaction
- `--cics-plex` *(string)*

    - The name of the CICSPlex from which to discard the transaction

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
    - The name of a (cics) profile to load for this command execution.

Examples

*- Discard a transaction named TRN1 from the region named MYREGION:

- `$ zowe cics discard transaction TRN1 --region-name MYREGION`

get

Get resources (for example, programs or transactions) from CICS through IBM CMCI.

resource

Get resources (for example, programs or transactions) from CICS.

Usage

zowe cics get resource [options]

Positional Arguments

- `resourceName` *(string)*
    - The name of the resource to get.

Options

- `--region-name` | `--rn` *(string)*
    - The CICS region name from which to get the resources
- `--cics-plex` | `--cp` *(string)*
    - The name of the CICSPlex from which to get the resources
- `--criteria` | `-c` *(string)*
    - The criteria by which to filter the resource
- `--parameter` | `-p` *(string)*

- The parameter by which to refine the resource

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
  - The name of a (cics) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- Get program resources from the region named MYREGION:

- `$ zowe cics get resource CICSProgram --region-name MYREGION`

*- Get local transaction resources from the region named MYREGION:

- `$ zowe cics get resource CICSLocalTransaction --region-name MYREGION`

  *- Get local file resources from the region named MYREGION:

- `$ zowe cics get resource CICSLocalFile --region-name MYREGION`

  *- Get program definition resources from the CSD group named GRP1 and the region named MYREGION:

- `$ zowe cics get resource CICSDefinitionProgram --region-name MYREGION --parameter "CSDGROUP(GRP1)"`

  *- Get transaction definition resources from the CSD group named GRP1 and the region named MYREGION:

- `$ zowe cics get resource CICSDefinitionTransaction --region-name MYREGION --parameter "CSDGROUP(GRP1)"`

  *- Get program resources that start with the name PRG from the region named MYREGION:

- `$ zowe cics get resource CICSProgram --region-name MYREGION --criteria "PROGRAM=PRG*"`

  *- Get a local transaction resource named TRAN from the region named MYREGION:

- `$ zowe cics get resource CICSLocalTransaction --region-name MYREGION --criteria "TRANID=TRAN"`

  *- Get program resources that start with the name MYPRG from the region named MYREGION and display various fields as a table:

- `$ zowe cics get resource CICSProgram --region-name MYREGION --criteria "PROGRAM=MYPRG*" --rft table --rfh --rff program length status`

install | ins

Install resources (for example, programs) to CICS through IBM CMCI.

program

Install a program to CICS.

Usage

zowe cics install program [options]

Positional Arguments

- `programName` *(string)*
  - The name of the program to install. The maximum length of the program name is eight characters.
- `csdGroup` *(string)*
  - The CICS system definition (CSD) Group for the program that you want to install. The maximum length of the group name is eight characters.

Options

- `--region-name` *(string)*
  - The CICS region name to which to install the program
- `--cics-plex` *(string)*
  - The name of the CICSPlex to which to install the program

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
  - The name of a (cics) profile to load for this command execution.

Examples

*- Install a program named PGM123 to the region named MYREGION in the CSD group MYGRP:

- `$ zowe cics install program PGM123 MYGRP --region-name MYREGION`

transaction

Install a transaction to CICS.

Usage

zowe cics install transaction [options]

Positional Arguments

- `transactionName` *(string)*
    - The name of the new transaction to install. The maximum length of the transaction name is four characters.
- `csdGroup` *(string)*
    - The CICS system definition (CSD) Group for the new transaction that you want to install. The maximum length of the group name is eight characters.

Options

- `--region-name` *(string)*
    - The CICS region name to which to install the transaction
- `--cics-plex` *(string)*
    - The name of the CICSPlex to which to install the transaction

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
    - The name of a (cics) profile to load for this command execution.

Examples

*- Install a transaction named TRN1 to the region named MYREGION in the CSD group MYGRP:

- `$ zowe cics install transaction TRN1 MYGRP --region-name MYREGION`

refresh | ref

Refresh a program on CICS through IBM CMCI.

program

Refresh a program on CICS.

Usage

zowe cics refresh program [options]

Positional Arguments

- `programName` *(string)*

    - The name of the program to refresh. The maximum length of the program name is eight characters.

Options

- `--region-name` *(string)*
    - The CICS region name on which you want to refresh the program
- `--cics-plex` *(string)*
    - The name of the CICSPlex on which to refresh the program

Profile Options

- `--cics-profile` | `--cics-p` *(string)*
    - The name of a (cics) profile to load for this command execution.

Examples

*- Refresh a program named PGM123 from the region named MYREGION:

- `$ zowe cics refresh program PGM123 --region-name MYREGION`

db2 (experimental)

Interact with IBM Db2 for z/OS

call (experimental)

Call a Db2 stored procedure

procedure (experimental)

Call a Db2 stored procedure. Specify the stored procedure name and optionally provide values.

Usage

zowe db2 call procedure [options]

Positional Arguments

- `routine` *(string)*
    - The name of a Db2 stored procedure

Options

- `--parameters` | `-p` *(array)*
  - Values to bind to the stored procedure parameters

Profile Options

- `--db2-profile` | `--db2-p` *(string)*
  - The name of a (db2) profile to load for this command execution.

Examples

*- Call stored procedure DEMO.SP1:

- `$ zowe db2 call procedure "DEMO.SP1"`

  *- Call a stored procedure and pass values for parameter indicators:

- `$ zowe db2 call procedure "DEMO.SP2(?, ?)" --parameters "Hello" "world!"`

execute (experimental)

Execute SQL queries against a Db2 region and retrieve the response. Enclose the query in quotes and escape any symbols that have a special meaning to the shell.

sql (experimental)

Execute one or multiple SQL statements separated by a semicolon from a command line or from a file.

Usage

zowe db2 execute sql [options]

Options

- `--query` | `-q` *(string)*
  - The SQL statement verbatim to execute
- `--file` | `-f` *(string)*
  - A local file containing the SQL statements to execute

Profile Options

- `--db2-profile` | `--db2-p` *(string)*
  - The name of a (db2) profile to load for this command execution.

Examples

*- Execute a dummy SQL query:

- `$ zowe db2 execute sql --query"SELECT 'Hello World' FROM SYSIBM.SYSDUMMY1"`

  *- Retrieve the employees table and total number of rows:

- `$ zowe db2 execute sql -q "SELECT * FROM SAMPLE.EMP; SELECT COUNT(*) AS TOTAL FROM SAMPLE.EMP"`

  *- Execute a file with SQL statements:

- `$ zowe db2 execute sql --file backup_sample_database.sql`

export (experimental)

Export data from a Db2 table

table (experimental)

Export a Db2 table to the stdout or a file.

Usage

zowe db2 export table

<

table> [options]

Positional Arguments

- `table` *(string)*
  - The name of the table to export

Options

- `--outfile` | `-o` *(string)*
    - The path to the output file

Profile Options

- `--db2-profile` | `--db2-p` *(string)*
    - The name of a (db2) profile to load for this command execution.

Examples

*- Export employees data from the table SAMPLE.EMP and save it to the file 'employees.sql':

- `$ zowe db2 export table SAMPLE.EMP --outfile employees.sql`

endevor

CA Endevor SCM plug-in for listing Endevor environment information, working with elements and packages located in specified Endevor instance.

add

Add an Element into CA Endevor SCM.

element

The add element command lets you add an Element to an Environment entry Stage in CA Endevor SCM.

Usage

zowe endevor add element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*

    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--override-signout` | `--os` *(boolean)*

    - Specify if you want to override the Signout of an Endevor element while performing this action.

- `--new-version` | `--nv` *(number)*
    - Assign a different version number to the Element.
- `--from-file` | `--ff` *(string)*
    - Use this input to provide source file.
- `--from-dataset` | `--fd` *(string)*
    - Use this input to provide source data set name.
- `--from-member` | `--fm` *(string)*
    - Use this input to provide source member name in the data set.
- `--from-path` | `--fp` *(string)*
    - Use this input to provide the path of source USS file. It must end with a slash "/".
- `--from-uss-file` | `--fuf` *(string)*
    - Use this input to provide source USS file.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Add element from local file with session profile set up:

- `$ zowe endevor add element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE --ff localfile.txt -i ENDEVOR`

approve | aprv

Approve a Package in CA Endevor SCM.

package

The approve package command approves Package in CA Endevor SCM for execution.

Usage

zowe endevor approve package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*

    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--notes` | `-n` *(string)*
    - Notes for approve/deny package.
- `--notes-from-file` | `--nff` *(string)*
    - Local file of notes for approve/deny package.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Approve package with session profile set up, specifying approval notes:

- `$ zowe endevor approve package packageName -n "notes" -i ENDEVOR`

backin

Backin a Package in CA Endevor SCM.

package

The backin package command reverses the backout action and returns the Package to a status of Executed.

Usage

zowe endevor backin package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--statement` | `--stmn` *(number)*

    - Specify the SCL statement number for the Element action that you want to back in or back out.

- `--element` | `--elm` *(string)*
    - Specify the Element name for the Element action that you want to back in or back out.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Backin package with session profile set up:

- `$ zowe endevor backin package packageName -i ENDEVOR`

backout

Backout a Package in CA Endevor SCM.

package

The backout package command restores the executable and output modules of the Package to the status they were in before execution.

Usage

zowe endevor backout package [package] [options]

Positional Arguments

- `package` *(string)*
  - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
  - Specifies the base host name.
- `--port` | `-p` *(string)*
  - Specifies the port number.
- `--instance` | `-i` *(string)*
  - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
  - Specifies the user name.
- `--password` | `--pass` *(string)*
  - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
  - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - Specify this option to have the server certificate verified against the list of supplied CAs
- `--statement` | `--stmn` *(number)*
  - Specify the SCL statement number for the Element action that you want to back in or back out.
- `--element` | `--elm` *(string)*
  - Specify the Element name for the Element action that you want to back in or back out.
- `--suppress-messages` | `--sm` *(boolean)*

    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Backout package with session profile set up:

- `$ zowe endevor backout package packageName -i ENDEVOR`

cast

Cast a Package in CA Endevor SCM.

package

The cast package command prepares the Package for review and subsequent execution. Casting a Package freezes the contents of the Package and prevents further changes to the Package.

Usage

zowe endevor cast package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*

    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--from-date-time` | `--fdt` *(string)*
    - Specify the beginning of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--to-date-time` | `--tdt` *(string)*
    - Specify the end of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--validate-components` | `--vc` *(string)*
    - Specify "yes" to enable component validation within the package, "no" to disable, and "warn" to generate a warning if component validation fails.

Allowed values: yes, no, warn

- `--no-backout` | `--nb` *(boolean)*
    - Specify this option to NOT have backout facility available for this package.

- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Cast package with session profile set up, changing the execution window of the Package:

- `$ zowe endevor cast package packageName --fdt 2018-01-01T00:00 --tdt 2018-12-31T12:00 -i ENDEVOR`

create

Create a Package in CA Endevor SCM.

package

The create package command lets you create a package in CA Endevor SCM.

Usage

zowe endevor create package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--description` | `-d` *(string)*
    - Allows you to associate a 50-character description when creating package.
- `--from-date-time` | `--fdt` *(string)*
    - Specify the beginning of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--to-date-time` | `--tdt` *(string)*
    - Specify the end of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--no-backout` | `--nb` *(boolean)*

    - Specify this option to NOT have backout facility available for this package.

- `--notes-from-file` | `--nff` *(string)*
    - Local file of notes for approve/deny package.
- `--type` | `-t` *(string)*
    - Specify the package type, where S = STANDARD and E = EMERGENCY, by default S is used.

Allowed values: S, E

- `--sharable` | `--sh` *(boolean)*
    - Specify this option if the package can be edited by more than one person when in In-edit status.
- `--append` | `-a` *(boolean)*
    - Specify this option to append the SCL you are adding to the existing package SCL. Otherwise it would be replaced.
- `--promotion` | `--pr` *(boolean)*
    - Specify this option to define the package as a promotion package.
- `--no-validate-scl` | `--nvs` *(boolean)*
    - Specify this option to NOT validate the package components while creating a package.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action
- `--from-file` | `--ff` *(string)*
    - Use this input to provide source file.
- `--from-dataset` | `--fd` *(string)*

    - Use this input to provide source data set name.

- `--from-member` | `--fm` *(string)*
    - Use this input to provide source member name in the data set.

- `--from-package` | `--fp` *(string)*
    - Directs the Create/Update action to copy the SCL from the package you specify into the package you are creating or updating.

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Create package from local file with session profile set up:

- `$ zowe endevor create package packageName -d "package description" --ff localfile.txt -i ENDEVOR`

delete | del

Delete an Element or a Package in CA Endevor SCM.

element

The delete element command deletes an Element from the specified inventory location in CA Endevor SCM.

Usage

zowe endevor delete element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*

    - The CA Endevor SCM subsystem where your project resides.

- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.

- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--proc-group` | `--pg` *(string)*
    - The CA Endevor SCM processor group you would like to use.

- `--override-signout` | `--os` *(boolean)*
    - Specify if you want to override the Signout of an Endevor element while performing this action.

- `--only-components` | `--oc` *(boolean)*
    - Applicable for CA Endevor SCM ACM users only. Indicates whether you want to delete both the Element component list and the Element, or the Element component list only. "No" is the default option

- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.

- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.

- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.

- `--suppress-messages` | `--sm` *(boolean)*

    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Delete element with session profile set up:

- `$ zowe endevor delete element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE -i ENDEVOR`

package

The delete package command lets you delete Packages of any status type in CA Endevor SCM.

Usage

zowe endevor delete package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.

- `--port` | `-p` *(string)*

    - Specifies the port number.

- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--status` | `--st` *(string)*
    - Specify the status of the packages. Valid values are [APPROVED, EXECFAILED] for execute action, and additional values [INEDIT, INAPPROVAL, INEXECUTION, EXECUTED, COMMITTED, DENIED] for list action, additional value [ALLSTATE] for delete action. It is possible to specify multiple status separated by "," during list and delete package.

Allowed values: ALLSTATE, INEDIT, INAPPROVAL, APPROVED, INEXECUTION, EXECUTED, COMMITTED, DENIED, EXECFAILED

- `--older-than` | `--ot` *(number)*
    - Specify the minimum age of the package.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*

    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Delete package with session profile set up:

- `$ zowe endevor delete package packageName -i ENDEVOR`

deny

Deny a Package in CA Endevor SCM.

package

The deny package command changes the status of a Package to Denied.

Usage

zowe endevor deny package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*

    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--notes` | `-n` *(string)*
    - Notes for approve/deny package.
- `--notes-from-file` | `--nff` *(string)*
    - Local file of notes for approve/deny package.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Deny package with session profile set up, specifying denial notes:

- `$ zowe endevor deny package packageName -n "notes" -i ENDEVOR`

execute

Execute a Package in CA Endevor SCM.

package

The execute package command executes a Package that have a status of Approved or Execfailed.

Usage

zowe endevor execute package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - Specify this option to have the server certificate verified against the list of supplied CAs
- `--from-date-time` | `--fdt` *(string)*
  - Specify the beginning of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--to-date-time` | `--tdt` *(string)*
  - Specify the end of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--status` | `--st` *(string)*
  - Specify the status of the packages. Valid values are [APPROVED, EXECFAILED] for execute action, and additional values [INEDIT, INAPPROVAL, INEXECUTION, EXECUTED, COMMITTED, DENIED] for list action, additional value [ALLSTATE] for delete action. It is possible to specify multiple status separated by "," during list and delete package.

Allowed values: ALLSTATE, INEDIT, INAPPROVAL, APPROVED, INEXECUTION, EXECUTED, COMMITTED, DENIED, EXECFAILED

- `--suppress-messages` | `--sm` *(boolean)*
  - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
  - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
  - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
  - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*

    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Execute package with session profile set up, specifying the time frame within which to execute the Package:

- `$ zowe endevor execute package packageName --fdt 2018-01-01T00:00 --tdt 2018-12-31T12:00 -i ENDEVOR`

generate | gen

Generate an Element in CA Endevor SCM.

element

The generate element command executes the generate Processor for the current level of the Element.

Usage

zowe endevor generate element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*

    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--proc-group` | `--pg` *(string)*
    - The CA Endevor SCM processor group you would like to use.
- `--search` | `--sea` *(boolean)*

    - Enables the search through the Endevor map.

- `--copy-back` | `--cb` *(boolean)*
    - Specify if you want to copy the current level of the Element back to the FROM Stage, then perform this action. Do not use with --no-source option.

- `--override-signout` | `--os` *(boolean)*
    - Specify if you want to override the Signout of an Endevor element while performing this action.

- `--no-source` | `--ns` *(boolean)*
    - Specify if you want to have source-less Element. Do not use with --copy-back option.

- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.

- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.

- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.

- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Generate an element with session profile set up, specifying option Copyback:

- `$ zowe endevor generate element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE --cb -i ENDEVOR`

list

List instances, elements, types, packages and inventory locations in CA Endevor SCM.

instances

The list instances command lists instances used by CA Endevor SCM Web Services

Usage

zowe endevor list instances [options]

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - Specify this option to have the server certificate verified against the list of supplied CAs
- `--full-output` | `--fo` *(boolean)*
  - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
  - Suppress all [INFO]/[WARN] messages from terminal output.
- `--maxrc` *(number)*
  - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
  - The name of a (endevor) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
    - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List instances with session specified:

- `$ zowe endevor list instances --host hostName --port 8080`

environments

The list environments command lists environments in CA Endevor SCM

Usage

zowe endevor list environments [environment] [options]

Positional Arguments

- `environment` *(string)*
    - Name of the CA Endevor SCM environment.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*

    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--data` | `--dat` *(string)*
    - Allows to select the type of summary data returned in the element list (defaults to all).

Default value: all Allowed values: all, bas, ele

- `--path` | `--pa` *(string)*
    - Specifies a PHYsical or LOGical path.

Allowed values: log, phy

- `--return` | `--ret` *(string)*
    - Sets mapping options for returned results: return FIRst match or ALL matching results.

Allowed values: fir, all

- `--search` | `--sea` *(boolean)*
    - Enables the search through the CA Endevor SCM map.
- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*

- The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
  - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
  - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all environments in CA Endevor SCM with session profile set up:

- `$ zowe endevor list environments -i ENDEVOR`

stages

The list stages command lists stages in CA Endevor SCM

Usage

zowe endevor list stages [stage] [options]

Positional Arguments

- `stage` *(string)*
    - Name of the CA Endevor SCM stage

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs

- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--data` | `--dat` *(string)*
    - Allows to select the type of summary data returned in the element list (defaults to all).

Default value: all Allowed values: all, bas, ele

- `--path` | `--pa` *(string)*
    - Specifies a PHYsical or LOGical path.

Allowed values: log, phy

- `--return` | `--ret` *(string)*
    - Sets mapping options for returned results: return FIRst match or ALL matching results.

Allowed values: fir, all

- `--search` | `--sea` *(boolean)*
    - Enables the search through the CA Endevor SCM map.
- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
    - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
    - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
    - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all stages in CA Endevor SCM with session profile set up:

- `$ zowe endevor list stages -i ENDEVOR`

systems

The list systems command lists system information in CA Endevor SCM

Usage

zowe endevor list systems [system] [options]

Positional Arguments

- `system` *(string)*
  - Name of the CA Endevor SCM system

Options

- `--hostname` | `--host` *(string)*
  - Specifies the base host name.
- `--port` | `-p` *(string)*
  - Specifies the port number.
- `--instance` | `-i` *(string)*
  - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
  - Specifies the user name.
- `--password` | `--pass` *(string)*
  - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
  - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
  - The CA Endevor SCM environment where your project resides.
- `--stage-number` | `--sn` *(string)*
  - The CA Endevor SCM stage number where your project resides.

Allowed values: *, 1, 2

- `--data` | `--dat` *(string)*
    - Allows to select the type of summary data returned in the element list (defaults to all).

Default value: all Allowed values: all, bas, ele

- `--path` | `--pa` *(string)*
    - Specifies a PHYsical or LOGical path.

Allowed values: log, phy

- `--return` | `--ret` *(string)*
    - Sets mapping options for returned results: return FIRst match or ALL matching results.

Allowed values: fir, all

- `--search` | `--sea` *(boolean)*
    - Enables the search through the CA Endevor SCM map.
- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all systems in CA Endevor SCM with session profile set up:

- `$ zowe endevor list systems -i ENDEVOR`

subsystems

The list subsystems command lists subsystem information in CA Endevor SCM

Usage

zowe endevor list subsystems [subsystem] [options]

Positional Arguments

- `subsystem` *(string)*
    - Name of the CA Endevor SCM subsystem

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: *, 1, 2

- `--data` | `--dat` *(string)*
    - Allows to select the type of summary data returned in the element list (defaults to all).

Default value: all Allowed values: all, bas, ele

- `--path` | `--pa` *(string)*
    - Specifies a PHYsical or LOGical path.

Allowed values: log, phy

- `--return` | `--ret` *(string)*
    - Sets mapping options for returned results: return FIRst match or ALL matching results.

Allowed values: fir, all

- `--search` | `--sea` *(boolean)*
    - Enables the search through the CA Endevor SCM map.
- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all subsystems in CA Endevor SCM with session profile set up:

- `$ zowe endevor list subsystems -i ENDEVOR`

types

The list types command lists type information in CA Endevor SCM

Usage

zowe endevor list types [type] [options]

Positional Arguments

- `type` *(string)*
    - Name of the CA Endevor SCM type

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: *, 1, 2

- `--data` | `--dat` *(string)*
    - Allows to select the type of summary data returned in the element list (defaults to all).

Default value: all Allowed values: all, bas, ele

- `--path` | `--pa` *(string)*
    - Specifies a PHYsical or LOGical path.

Allowed values: log, phy

- `--return` | `--ret` *(string)*
    - Sets mapping options for returned results: return FIRst match or ALL matching results.

Allowed values: fir, all

- `--search` | `--sea` *(boolean)*
    - Enables the search through the CA Endevor SCM map.
- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all types in CA Endevor SCM with session profile set up:

- `$ zowe endevor list types -i ENDEVOR`

packages

The list packages command lists package information in CA Endevor SCM

Usage

zowe endevor list packages [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--status` | `--st` *(string)*
    - Specify the status of the packages. Valid values are [APPROVED, EXECFAILED] for execute action, and additional values [INEDIT, INAPPROVAL, INEXECUTION, EXECUTED, COMMITTED, DENIED] for list action, additional value [ALLSTATE] for delete action. It is possible to specify multiple status separated by "," during list and delete package.

Allowed values: ALLSTATE, INEDIT, INAPPROVAL, APPROVED, INEXECUTION, EXECUTED, COMMITTED, DENIED, EXECFAILED

- `--type` | `-t` *(string)*
    - Specify the package type, where S = STANDARD and E = EMERGENCY, by default S is used.

Allowed values: S, E

- `--enterprise` | `--ent` *(string)*
    - Specify to filter the list by enterprise Package parameter. A - All, E - Enterprise, X - eXclude.

Allowed values: A, E, X

- `--promotion-status` | `--ps` *(string)*
    - Specify to filter the list by promotion Package parameter. A - All, P - Promotion, X - eXclude.

Allowed values: A, P, X

- `--prom-target-env` | `--pte` *(string)*
    - Promotion target environment. Specifies the promotion package target environment. This field only applies to promotion packages and can only be specified when the promotion package type is A or P.
- `--prom-target-stgID` | `--pts` *(string)*
    - Promotion target stage ID. Specifies the promotion package target stage ID. This field only applies to promotion packages and can only be specified when the promotion package type is A or P.
- `--approver` | `--apr` *(string)*
    - Specifies a one to eight character approver ID. Only one approver ID can be specified and name masking is not supported.
- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
    - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
    - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*

    - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all packages in CA Endevor SCM with session profile set up:

- `$ zowe endevor list packages -i ENDEVOR`

elements

The list elements command lists element information in CA Endevor SCM

Usage

zowe endevor list elements [element] [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: *, 1, 2

- `--data` | `--dat` *(string)*
    - Allows to select the type of summary data returned in the element list (defaults to all).

Default value: all Allowed values: all, bas, ele

- `--path` | `--pa` *(string)*
    - Specifies a PHYsical or LOGical path.

Allowed values: log, phy

- `--return` | `--ret` *(string)*
    - Sets mapping options for returned results: return FIRst match or ALL matching results.

Allowed values: fir, all

- `--full-output` | `--fo` *(boolean)*
    - Specify this option if you want a full output of list action.

- `--search` | `--sea` *(boolean)*
    - Enables the search through the CA Endevor SCM map.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
    - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
    - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
    - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all elements in CA Endevor SCM with session profile set up:

- `$ zowe endevor list elements -i ENDEVOR`

move | mv

Move an Element in CA Endevor SCM.

element

The move element command moves Elements between inventory locations along a map.

Usage

zowe endevor move element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*

    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--proc-group` | `--pg` *(string)*

    - The CA Endevor SCM processor group you would like to use.

- `--sync` | `-s` *(boolean)*
    - Specify if you want to synchronize source and current level of the Elements while performing this action.

- `--with-history` | `--wh` *(boolean)*
    - Specify if you want to preserve the change history of the Elements while performing this action.

- `--bypass-element-delete` | `--bed` *(boolean)*
    - Specify if you want to retain the Elements in the source Stage after successfully completing this action.

- `--retain-signout` | `--rs` *(boolean)*
    - Specify if you want to retain the source location signouts for all Elements at the target location while performing this action.

- `--signout-to` | `--st` *(string)*
    - Specify if you want to sign all Elements out to the specified user ID at the target Stage while performing this action.

- `--jump` | `-j` *(boolean)*
    - Specify if you want to move Elements across Environments even if those Elements exist at an intermediate Stage that is not on the map, while performing this action.

- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.

- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.

- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-

mask when specifying the Processor group name.

- `--suppress-messages` | `--sm` *(boolean)*
  - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
  - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
  - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
  - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
  - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Move element from specified inventory location with session profile set up:

- `$ zowe endevor move element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE -i ENDEVOR`

print | p

Print an Element or a Component in CA Endevor SCM.

element

The print element command prints selected information about Element in CA Endevor SCM.

Usage

zowe endevor print element [options]

Positional Arguments

- `element` *(string)*
  - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*

    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--level` | `--lev` *(number)*
    - Indicates the level number of the element (use along with the version option).
- `--element-version` | `--ev` *(number)*
    - Indicates the version number of the element (use along with the level option).
- `--print` *(string)*
    - Specify the type of data to print out for print element command

Default value: browse Allowed values: browse, changes, history, summary, master, listing

- `--list-string` | `--ls` *(string)*
    - Specifies the one to eight character text-string used to identify the listing data set to print.
- `--search` | `--sea` *(boolean)*
    - Enables the search through the Endevor map.
- `--no-headings` | `--nh` *(boolean)*
    - Specify to not print a header on each page.
- `--explode` | `--exp` | `--ex` *(boolean)*
    - Specify to print component info from ACMQ.
- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.
- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.
- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--to-file` | `--tf` *(string)*
    - File name in which the command output will be stored.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Print element from specified inventory location with session profile set up:

- `$ zowe endevor print element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE -i ENDEVOR`

components

The print component command prints selected component information about Element in CA Endevor SCM.

Usage

zowe endevor print components [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*

    - Name of the CA Endevor SCM element's type.

- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--level` | `--lev` *(number)*
    - Indicates the level number of the element (use along with the version option).
- `--element-version` | `--ev` *(number)*
    - Indicates the version number of the element (use along with the level option).
- `--print-comp` | `--pc` *(string)*
    - Specify the type of data to print out for print component command

Default value: browse Allowed values: browse, changes, history, summary

- `--list-string` | `--ls` *(string)*
    - Specifies the one to eight character text-string used to identify the listing data set to print.
- `--search` | `--sea` *(boolean)*
    - Enables the search through the Endevor map.
- `--no-headings` | `--nh` *(boolean)*
    - Specify to not print a header on each page.
- `--explode` | `--exp` | `--ex` *(boolean)*
    - Specify to print component info from ACMQ.
- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.
- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.

- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.
- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--to-file` | `--tf` *(string)*
    - File name in which the command output will be stored.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Print selected component information about Element with session profile set up:

- `$ zowe endevor print components elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE -i ENDEVOR`

reset

Reset a Package in CA Endevor SCM.

package

The reset package command lets you set the status of a Package back to In-edit so you can modify it.

Usage

zowe endevor reset package [package] [options]

Positional Arguments

- `package` *(string)*
  - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
  - Specifies the base host name.
- `--port` | `-p` *(string)*
  - Specifies the port number.
- `--instance` | `-i` *(string)*
  - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
  - Specifies the user name.
- `--password` | `--pass` *(string)*
  - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
  - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - Specify this option to have the server certificate verified against the list of supplied CAs
- `--suppress-messages` | `--sm` *(boolean)*
  - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*

    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Reset package with session profile set up:

- `$ zowe endevor reset package packageName -i ENDEVOR`

retrieve | ret

Retrieve an Element in CA Endevor SCM.

element

The retrieve element command retrieves an existing element in CA Endevor SCM.

Usage

zowe endevor retrieve element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*

    - Specifies the port number.

- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--element-version` | `--ev` *(number)*

    - Indicates the version number of the element (use along with the level option).
- `--level` | `--lev` *(number)*
    - Indicates the level number of the element (use along with the version option).
- `--to-path` | `--tp` *(string)*
    - Provide a USS path to a destination location.
- `--to-uss-file` | `--tuf` *(string)*
    - Provide a USS file as a destination file.
- `--to-dataset` | `--td` *(string)*
    - Provide a destination data set name.
- `--to-member` | `--tm` *(string)*
    - Provide a destination member name inside the data set.
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--override-signout` | `--os` *(boolean)*
    - Specify if you want to override the Signout of an Endevor element while performing this action.
- `--no-signout` | `--nsign` *(boolean)*
    - Specify if you want to perform the action without signing the element out.
- `--replace-member` | `--replace` | `--rm` *(boolean)*
    - Specify if you want to replace the member currently in the library with the new element contents
- `--expand-includes` | `--expand` | `--ei` *(boolean)*
    - Indicates that INCLUDE statements should be expanded in the course of the action.
- `--search` | `--sea` *(boolean)*

    - Enables the search through the Endevor map.

- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.

- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.

- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.

- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--to-file` | `--tf` *(string)*
    - File name in which the command output will be stored.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Retrieve element from specified inventory location to local file with session profile set up:

- `$ zowe endevor retrieve element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE --tf localfile.txt -i ENDEVOR`

signin | si

Signin an Element in CA Endevor SCM.

element

The signin element command signs in an existing element in CA Endevor SCM.

Usage

zowe endevor signin element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*

    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--proc-group` | `--pg` *(string)*
    - The CA Endevor SCM processor group you would like to use.
- `--search` | `--sea` *(boolean)*
    - Enables the search through the Endevor map.
- `--override-signout` | `--os` *(boolean)*

    - Specify if you want to override the Signout of an Endevor element while performing this action.

- `--signout-to` | `--st` *(string)*
    - Specify if you want to sign all Elements out to the specified user ID at the target Stage while performing this action.

- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.

- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.

- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.

- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Signin element with session profile set up:

- `$ zowe endevor signin element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE -i ENDEVOR`

submit

Submit a Package or a SCL file in CA Endevor SCM.

package

The submit package command submits a JCL job stream to execute one or more Packages.

Usage

zowe endevor submit package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*

    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--from-file` | `--ff` *(string)*
    - Use this input to provide source file.
- `--from-dataset` | `--fd` *(string)*
    - Use this input to provide source data set name.
- `--from-member` | `--fm` *(string)*
    - Use this input to provide source member name in the data set.
- `--to-CA7` | `--t7` *(boolean)*
    - Specify to send the submission of the package to CA7 scheduler.
- `--to-ddname` | `--tdd` *(string)*
    - Send the submission of the package to be processed according to a DDName specified in the starter task (STC).
- `--status` | `--st` *(string)*
    - Specify the status of the packages. Valid values are [APPROVED, EXECFAILED] for execute action, and additional values [INEDIT, INAPPROVAL, INEXECUTION, EXECUTED, COMMITTED, DENIED] for list action, additional value [ALLSTATE] for delete action. It is possible to specify multiple status separated by "," during list and delete package.

Allowed values: ALLSTATE, INEDIT, INAPPROVAL, APPROVED, INEXECUTION, EXECUTED, COMMITTED, DENIED, EXECFAILED

- `--multiple-streams` | `--ms` *(boolean)*
    - Specify to submit a separate, unique job for each package. If you do not specify this, a single job with a unique job step for each package is submitted.
- `--increment-jobname` | `--ij` *(boolean)*

    - Specify to increases the last character in the jobcard you provide.

- `--jcl-procedure` | `--jp` *(string)*
    - This option lets you to identify the name of the JCL procedure that you want to invoke. ENDEVOR is used by default if any processor is specified.

- `--CA7-dependent-job` | `--7dj` *(string)*
    - Specifies a single predecessor job which must complete while demanded job is waiting in the CA7 scheduler.

- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.

- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Submit package using jobcard from local file, with session profile set up:

- `$ zowe endevor submit package packageName --ff jobcardfile.txt -i ENDEVOR`

scl

The submit scl commands submits a SCL file to be executed.

Usage

zowe endevor submit scl [options]

Options

- `--hostname` | `--host` *(string)*
  - Specifies the base host name.
- `--port` | `-p` *(string)*
  - Specifies the port number.
- `--instance` | `-i` *(string)*
  - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
  - Specifies the user name.
- `--password` | `--pass` *(string)*
  - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
  - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - Specify this option to have the server certificate verified against the list of supplied CAs
- `--from-file` | `--ff` *(string)*
  - Use this input to provide source file.
- `--suppress-messages` | `--sm` *(boolean)*
  - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
  - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
  - The return code of a failed action

Required Options

- `--scl-file` | `--sf` *(string)*
  - The file which contains the CA Endevor SCM SCL you would like to submit.
- `--scl-type` | `--sclt` *(string)*
  - The category of CA Endevor SCM SCL.

Allowed values: list, element, package, admin, ship, addUpdRtv

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
  - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
  - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Submit a SCL of type element, with session profile set up:

- `$ zowe endevor submit scl --sf sclfile.txt --sclt element -i ENDEVOR`

transfer | tr

Transfer an Element in CA Endevor SCM.

element

The Transfer element command transfers Elements from one CA Endevor SCM location to another.

Usage

zowe endevor transfer element [options]

Positional Arguments

- `element` *(string)*
  - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*

    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--proc-group` | `--pg` *(string)*
    - The CA Endevor SCM processor group you would like to use.
- `--to-environment` | `--toenv` *(string)*
    - The target CA Endevor SCM environment.
- `--to-system` | `--tosys` *(string)*
    - The target CA Endevor SCM system.
- `--to-subsystem` | `--tosub` *(string)*
    - The target CA Endevor SCM subsystem.
- `--to-element` | `--toele` *(string)*
    - The target CA Endevor SCM element name.
- `--to-type` | `--totyp` *(string)*
    - The target CA Endevor SCM element type.
- `--to-stage-number` | `--tosn` *(string)*
    - The target CA Endevor SCM stage number.

Allowed values: 1, 2

- `--element-version` | `--ev` *(number)*
    - Indicates the version number of the element (use along with the level option).
- `--level` | `--lev` *(number)*

    - Indicates the level number of the element (use along with the version option).
- `--new-version` | `--nv` *(number)*
    - Assign a different version number to the Element.
- `--sync` | `-s` *(boolean)*
    - Specify if you want to synchronize source and current level of the Elements while performing this action.
- `--with-history` | `--wh` *(boolean)*
    - Specify if you want to preserve the change history of the Elements while performing this action.
- `--ignore-generate-failed` | `--igf` *(boolean)*
    - Process the transfer request regardless of whether the FAILED flag is set for the element or if the element was generated or moved successfully.
- `--bypass-element-delete` | `--bed` *(boolean)*
    - Specify if you want to retain the Elements in the source Stage after successfully completing this action.
- `--bypass-delete-proc` | `--bdp` *(boolean)*
    - Specity to bypasses the execution of the delete processor.
- `--bypass-generate-proc` | `--bgp` *(boolean)*
    - Specify to bypasses the execution of the generate or move processor (whichever may be chosen) upon element transfer.
- `--retain-signout` | `--rs` *(boolean)*
    - Specify if you want to retain the source location signouts for all Elements at the target location while performing this action.
- `--signout-to` | `--st` *(string)*
    - Specify if you want to sign all Elements out to the specified user ID at the target Stage while performing this action.
- `--jump` | `-j` *(boolean)*
    - Specify if you want to move Elements across Environments even if those Elements exist at an intermediate Stage that is not on the map, while performing this action.

- `--where-proc-group` | `--wpg` *(string)*
    - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.
- `--where-ccid-all` | `--wca` *(string)*
    - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.
- `--where-ccid-current` | `--wcc` *(string)*
    - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.
- `--where-ccid-retrieve` | `--wcr` *(string)*
    - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Transfer element from specified inventory location to 1 stage higher in map, with session profile set up:

- `$ zowe endevor transfer element elementName --env ENV --sn 1 --tosn 2 --sys SYS --sub SUB --typ TYPE -i ENDEVOR`

update

Update an Element or a Package in CA Endevor SCM.

package

The update package command lets you update a package in CA Endevor SCM.

Usage

zowe endevor update package [package] [options]

Positional Arguments

- `package` *(string)*
    - Name of the CA Endevor SCM package.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*

    - Specify this option to have the server certificate verified against the list of supplied CAs

- `--description` | `-d` *(string)*
    - Allows you to associate a 50-character description when creating package.
- `--from-date-time` | `--fdt` *(string)*
    - Specify the beginning of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--to-date-time` | `--tdt` *(string)*
    - Specify the end of time frame within which the package can be executed. Use yyyy-mm-ddThh:mm or see ISO 8601 standard for syntax.
- `--no-backout` | `--nb` *(boolean)*
    - Specify this option to NOT have backout facility available for this package.
- `--notes-from-file` | `--nff` *(string)*
    - Local file of notes for approve/deny package.
- `--type` | `-t` *(string)*
    - Specify the package type, where S = STANDARD and E = EMERGENCY, by default S is used.

Allowed values: S, E

- `--sharable` | `--sh` *(boolean)*
    - Specify this option if the package can be edited by more than one person when in In-edit status.
- `--append` | `-a` *(boolean)*
    - Specify this option to append the SCL you are adding to the existing package SCL. Otherwise it would be replaced.
- `--promotion` | `--pr` *(boolean)*
    - Specify this option to define the package as a promotion package.
- `--no-validate-scl` | `--nvs` *(boolean)*
    - Specify this option to NOT validate the package components while creating a package.

- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.
- `--maxrc` *(number)*
    - The return code of a failed action
- `--from-file` | `--ff` *(string)*
    - Use this input to provide source file.
- `--from-dataset` | `--fd` *(string)*
    - Use this input to provide source data set name.
- `--from-member` | `--fm` *(string)*
    - Use this input to provide source member name in the data set.
- `--from-package` | `--fp` *(string)*
    - Directs the Create/Update action to copy the SCL from the package you specify into the package you are creating or updating.

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Update package from local file with session profile set up:

- `$ zowe endevor update package packageName --ff localfile.txt -i ENDEVOR`

element

The update element command updates an Element in the entry Stage, thereby creating a new level for the Element in the entry Stage.

Usage

zowe endevor update element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.
- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*

    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--ccid` | `--cci` *(string)*
    - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
    - The comment you want to have when performing an Element action
- `--override-signout` | `--os` *(boolean)*
    - Specify if you want to override the Signout of an Endevor element while performing this action.
- `--from-file` | `--ff` *(string)*
    - Use this input to provide source file.
- `--from-dataset` | `--fd` *(string)*
    - Use this input to provide source data set name.
- `--from-member` | `--fm` *(string)*
    - Use this input to provide source member name in the data set.
- `--from-path` | `--fp` *(string)*
    - Use this input to provide the path of source USS file. It must end with a slash "/".
- `--from-uss-file` | `--fuf` *(string)*
    - Use this input to provide source USS file.
- `--suppress-messages` | `--sm` *(boolean)*
    - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
    - File name for saving output messages from CA Endevor SCM locally.

- `--maxrc` *(number)*
    - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- Update element from local file with session profile set up:

- `$ zowe endevor update element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE --ff localfile.txt -i ENDEVOR`

view

View an Element in CA Endevor SCM.

element

The view element command views an existing element in CA Endevor SCM.

Usage

zowe endevor view element [options]

Positional Arguments

- `element` *(string)*
    - Name of the CA Endevor SCM element.

Options

- `--hostname` | `--host` *(string)*
    - Specifies the base host name.
- `--port` | `-p` *(string)*
    - Specifies the port number.

- `--instance` | `-i` *(string)*
    - Specifies CA Endevor SCM Web Services dataSource name.
- `--username` | `--user` *(string)*
    - Specifies the user name.
- `--password` | `--pass` *(string)*
    - Specifies the user's password.
- `--protocol` | `--prot` *(string)*
    - Specifies the protocol used for connecting to CA Endevor® SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Specify this option to have the server certificate verified against the list of supplied CAs
- `--environment` | `--env` *(string)*
    - The CA Endevor SCM environment where your project resides.
- `--system` | `--sys` *(string)*
    - The CA Endevor SCM system where your project resides.
- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your project resides.
- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type.
- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage number where your project resides.

Allowed values: 1, 2

- `--element-version` | `--ev` *(number)*
    - Indicates the version number of the element (use along with the level option).

- `--level` | `--lev` *(number)*
  - Indicates the level number of the element (use along with the version option).
- `--ccid` | `--cci` *(string)*
  - The CCID you want to use when performing an Element action.
- `--comment` | `--com` *(string)*
  - The comment you want to have when performing an Element action
- `--search` | `--sea` *(boolean)*
  - Enables the search through the Endevor map.
- `--where-ccid-all` | `--wca` *(string)*
  - Tells Endevor to search both the Master Control File and the SOURCE DELTA levels for a specified CCIDs.
- `--where-ccid-current` | `--wcc` *(string)*
  - Tells Endevor to look through the CCID fields in the Master Control File to find a specified CCIDs.
- `--where-ccid-retrieve` | `--wcr` *(string)*
  - Tells Endevor to use the CCID in the Master Control File RETRIEVE CCID field.
- `--where-proc-group` | `--wpg` *(string)*
  - Lets you select Elements according to a specified Processor group. You can use a name-mask when specifying the Processor group name.
- `--suppress-messages` | `--sm` *(boolean)*
  - Suppress all [INFO]/[WARN] messages from terminal output.
- `--file-name` | `--fn` *(string)*
  - File name for saving output messages from CA Endevor SCM locally.
- `--to-file` | `--tf` *(string)*
  - File name in which the command output will be stored.
- `--maxrc` *(number)*
  - The return code of a failed action

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.
- `--endevor-location-profile` | `--endevor-location-p` *(string)*
    - The name of a (endevor-location) profile to load for this command execution.

Examples

*- View element from specified inventory location to local file with session profile set up:

- `$ zowe endevor view element elementName --env ENV --sn 1 --sys SYS --sub SUB --typ TYPE --tf localfile.txt -i ENDEVOR`

file-master-plus | fmp

CA File Master Plus command line interface is a file management and data manipulation tool. It speeds up file creation and manipulates virtual storage access method (VSAM), sequential and partitioned data sets. It also supports symbolic access to data via layouts and data manipulation like selection of records in data sets.

copy | c

Copy data from a data set and optionally filter or modify the copied data by using selection criteria.

data-set

Copy from a data set to another data set. If the data set does not exist, CA File Master Plus automatically creates a new data set using the attributes of data set. It supports all data set types that are supported by CA File Master Plus.

Usage

zowe file-master-plus copy data-set [options]

Positional Arguments

- `from` *(string)*
    - Specifies the name of the data set to copy from.
- `to` *(string)*
    - Specifies the name of the data set to copy to.

Options

- `--members` | `-m` *(array)*
  - Specifies the members that you want to copy from the data set. To rename the member, specify a new member name after the delimiter ','.If this parameter is not specified all the members are copied. Note: This parameter only applies to a PDS or PDSE. Example: --mem mem1,newmem1 mem2. Here 'mem2' is copied as it is, and 'mem1' is renamed as 'newmem1'.
- `--replace` | `-r` *(string)*
  - Replace resource-specific items in the target data set. Note: Used only if the target data set is a PDS or a VSAM KSDS. Example: -r n.

Default value: y Allowed values: y, n

- `--static-selection-criteria` | `--ssc` *(string)*
  - Specifies the name of pre-defined CA File Master Plus selection criteria. The name refers to a member in the defaultselection criteria data set as defined in the FMM_CLIST parameter in CA File Master Plus server. Example: --ssc testcri
- `--dynamic-selection-criteria` | `--dsc` *(string)*
  - Specifies path of the .txt file where the dynamic selection criteria exist. Ensure the format of the file is identical to the static selection criteria created by CA File Master Plus. If the selection criteria refer to field names in a Cobol or PL/I copybook, use the --layout-member and --layout-data-set parameters to name the copybook location. Example: --dsc ../selcri/testcri.txt
- `--layout-member` | `--lm` *(string)*
  - Specifies name of the Cobol or PL/I copybook. Example: -lm testlay.
- `--layout-data-set` | `--lds` *(string)*
  - Specifies name of the data set that contains the layout member. Example: -lds fmmvs.layout.dataset.

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Copying a data set:

- `$ zowe file-master-plus copy data-set fmmvs.pds1 fmmvs.pds2`

  *- Copying a data set with dynamic selection criteria:

- `$ zowe file-master-plus copy data-set fmmvs.from.ps fmmvs.to.ps --lds fmmvs.layout.dataset --lm testlay --dsc ./fmmvs/selcri/testcri`

create | cre

Create a data set.

physical-sequential

Create a physical sequential data set.

Usage

zowe file-master-plus create physical-sequential [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
  - Specifies the name of a model physical sequential data set name for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--logical-record-length` | `--lrecl` | `--lrl` *(number)*
  - Specifies the length of the logical record. Allowed values: 1-32760 Default value: 80 Example: --lrecl 80
- `--block-size` | `--blksize` | `--bs` *(number)*
  - Specifies the size of the block of records. Allowed values: 1-32760 Default value: 6160

Example: --blksize 6160

- `--record-format` | `--recfm` | `--rf` *(string)*
  - Specifies the record format. The allowed values which have the following meaning: F - Fixed-length records V - Variable-length records U - Undefined-length records B - Records are blocked A - Records contain ASCII printer control characters M - Records contain machine code control characters S - For variable-length records, records may span blocks T - Records may be written into overflow tracks Default value: FB Example: --recfm FB

Allowed values: F, FA, FM, FB, FBA, FBM, FS, FSA, FSM, FT, FTA, FTM, FBS, FBT, U, UA, UM, UT, UTA, UTM, V, VA, VM, VB, VBA, VBM, VS, VSA, VSM, VT, VTA, VTM, VBS, VBT

- `--space-units` | `--su` *(string)*
  - Specifies the space allocation unit. The allowed values which have the following meaning: TRK - Tracks CYL - Cylinders BLK - Blocksize Default value: CYL Example: --su blk

Allowed values: TRK, CYL, BLK

- `--primary-space` | `--ps` *(number)*
  - Specifies primary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ps 3
- `--secondary-space` | `--ss` *(number)*
  - Specifies secondary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ss 5
- `--volume` | `-v` *(array)*
  - Specifies a disk volume or specific tapes. Example: -v vol002
- `--unit-type` | `--ut` *(string)*
  - Specifies the DASD unit name. Example: --ut sysda
- `--expiration-date` | `--ed` *(string)*
  - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31

- `--storage-class` | `--sc` *(string)*
    - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
    - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
    - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a physical sequential data set with default option values:

- `$ zowe file-master-plus create physical-sequential fmmvs.test.dsname`

  *- Creating a physical sequential data set with options:

- `$ zowe file-master-plus create physical-sequential fmmvs.test.dsname --lrecl 180 --blksize 32720 --recfm vb --ps 5 --ss 5 -v vol005 --su trk --ed 2025-09-27`

  *- Creating a physical sequential data set like a model physical sequential data set:

- `$ zowe file-master-plus create physical-sequential fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a physical sequential data set like a model physical sequential data set and overriding the parameters with options:

- `$ zowe file-master-plus create physical-sequential fmmvs.test.dsname --model fmmvs.model.dsname --lrecl 180 --recfm VB --ps 5 --ss 5 --ed p`

large-format-sequential

Create a large format sequential data set.

Usage

zowe file-master-plus create large-format-sequential [options]

Positional Arguments

- `name` *(string)*
    - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
    - Specifies the name of a model large format sequential data set name for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--logical-record-length` | `--lrecl` | `--lrl` *(number)*
    - Specifies the length of the logical record. Allowed values: 1-32760 Default value: 80 Example: --lrecl 80
- `--block-size` | `--blksize` | `--bs` *(number)*
    - Specifies the size of the block of records. Allowed values: 1-32760 Default value: 6160 Example: --blksize 6160
- `--record-format` | `--recfm` | `--rf` *(string)*
    - Specifies the record format. The allowed values which have the following meaning: F - Fixed-length records V - Variable-length records U - Undefined-length records B - Records are blocked A - Records contain ASCII printer control characters M - Records contain machine code control characters S - For variable-length records, records may span blocks T - Records may be written into overflow tracks Default value: FB Example: --recfm FB

Allowed values: F, FA, FM, FB, FBA, FBM, FS, FSA, FSM, FT, FTA, FTM, FBS, FBT, U, UA, UM, UT, UTA, UTM, V, VA, VM, VB, VBA, VBM, VS, VSA, VSM, VT, VTA, VTM, VBS, VBT

- `--space-units` | `--su` *(string)*
    - Specifies the space allocation unit. The allowed values which have the following meaning: TRK - Tracks CYL - Cylinders BLK - Blocksize Default value: CYL Example: --su blk

Allowed values: TRK, CYL, BLK

- `--primary-space` | `--ps` *(number)*
  - Specifies primary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ps 3
- `--secondary-space` | `--ss` *(number)*
  - Specifies secondary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ss 5
- `--volume` | `-v` *(array)*
  - Specifies a disk volume or specific tapes. Example: -v vol002
- `--unit-type` | `--ut` *(string)*
  - Specifies the DASD unit name. Example: --ut sysda
- `--expiration-date` | `--ed` *(string)*
  - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
  - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
  - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
  - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a large format sequential data set with default option values:

- `$ zowe file-master-plus create large-format-sequential fmmvs.test.dsname`

  *- Creating a large format sequential data set with options:

- `$ zowe file-master-plus create large-format-sequential fmmvs.test.dsname --lrecl 180 --blksize 32720 --recfm vb --ps 5 --ss 5 -v vol005 --su trk --ed 2025-09-27`

  *- Creating a large format sequential data set like a model large format sequential data set:

- `$ zowe file-master-plus create large-format-sequential fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a large format sequential data set like a model large format sequential data set and overriding the parameters with options:

- `$ zowe file-master-plus create large-format-sequential fmmvs.test.dsname --model fmmvs.model.dsname --lrecl 180 --recfm VB --ps 5 --ss 5 --ed p`

partitioned-data-set

Create a partitioned data set.

Usage

zowe file-master-plus create partitioned-data-set [options]

Positional Arguments

- `name` *(string)*

  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*

  - Specifies the name of a model partitioned data set for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname

- `--logical-record-length` | `--lrecl` | `--lrl` *(number)*

  - Specifies the length of the logical record. Allowed values: 1-32760 Default value: 80 Example: --lrecl 80

- `--block-size` | `--blksize` | `--bs` *(number)*
  - Specifies the size of the block of records. Allowed values: 1-32760 Default value: 6160 Example: --blksize 6160
- `--record-format` | `--recfm` | `--rf` *(string)*
  - Specifies the record format. The allowed values have the following meaning: F - Fixed-length records V - Variable-length records U - Undefined-length records B - Records are blocked A - Records contain ASCII printer control characters M - Records contain machine code control characters S - For variable-length records, records may span blocks T - Records may be written into overflow tracks Default value: FB Example: --recfm FB

Allowed values: F, FA, FM, FB, FBA, FBM, FS, FSA, FSM, FT, FTA, FTM, FBS, FBT, U, UA, UM, UT, UTA, UTM, V, VA, VM, VB, VBA, VBM, VS, VSA, VSM, VT, VTA, VTM, VBS, VBT

- `--space-units` | `--su` *(string)*
  - Specifies the space allocation unit. The allowed values which have the following meaning: TRK - Tracks CYL - Cylinders BLK - Blocksize Default value: CYL Example: --su blk

Allowed values: TRK, CYL, BLK

- `--primary-space` | `--ps` *(number)*
  - Specifies the primary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ps 3
- `--secondary-space` | `--ss` *(number)*
  - Specifies the secondary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ss 5
- `--directory-blocks` | `--db` *(number)*
  - Specifies number of directory blocks. Allowed values: 1-16777215 Default value: 5 Example: --db 5
- `--volume` | `-v` *(string)*
  - Specifies a disk volume or specific tapes. Example: -v vol002
- `--unit-type` | `--ut` *(string)*

    - Specifies the DASD unit name. Example: --ut sysda

- `--expiration-date` | `--ed` *(string)*
    - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31

- `--storage-class` | `--sc` *(string)*
    - Specifies the storage class. Example: --sc scl002

- `--management-class` | `--mc` *(string)*
    - Specifies the management class. Example: --mc mcl002

- `--data-class` | `--dc` *(string)*
    - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a PDS with default option values:

- `$ zowe file-master-plus create partitioned-data-set fmmvs.test.dsname`

  *- Creating a PDS with options:

- `$ zowe file-master-plus create partitioned-data-set fmmvs.test.dsname --lrecl 180 --blksize 32720 --recfm vb --ps 5 --ss 5 --db 5 -v vol005 --su trk --ed p`

  *- Creating a PDS like a model PDS:

- `$ zowe file-master-plus create partitioned-data-set fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a PDS like a model PDS and overriding the parameters with options:

- `$ zowe file-master-plus create partitioned-data-set fmmvs.test.dsname --model`

```
fmmvs.model.dsname --lrecl 180 --recfm vb --ps 5 --ss 5 --db 5 -v vol005 --ed
2025-09-27
```

partitioned-data-set-extended

Create an extended partitioned data set.

Usage

zowe file-master-plus create partitioned-data-set-extended [options]

Positional Arguments

- `name` *(string)*
    - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
    - Specifies the name of a model extentded partitioned data set name for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--data-set-version` | `--dsv` | `--ver` *(number)*
    - Specifies the data set version. Default value depends on the system settings. Example: --ver 2

Allowed values: 1, 2

- `--generations` | `-g` *(number)*
    - Specifies the number of generations. Applicable when data set version is '2'. Default value: 0 Example: -g 10
- `--logical-record-length` | `--lrecl` | `--lrl` *(number)*
    - Specifies the length of the logical record. Allowed values: 1-32760 Default value: 80 Example: --lrecl 80
- `--block-size` | `--blksize` | `--bs` *(number)*
    - Specifies the size of the block of records. Allowed values: 1-32760 Default value: 6160 Example: --blksize 6160

- `--record-format` | `--recfm` | `--rf` *(string)*
    - Specifies the record format. The allowed values which have the following meaning: F - Fixed-length records V - Variable-length records U - Undefined-length records B - Records are blocked A - Records contain ASCII printer control characters M - Records contain machine code control characters S - For variable-length records, records may span blocks T - Records may be written into overflow tracks Default value: FB Example: --recfm FB

Allowed values: F, FA, FM, FB, FBA, FBM, FS, FSA, FSM, FT, FTA, FTM, FBS, FBT, U, UA, UM, UT, UTA, UTM, V, VA, VM, VB, VBA, VBM, VS, VSA, VSM, VT, VTA, VTM, VBS, VBT

- `--space-units` | `--su` *(string)*
    - Specifies the space allocation unit. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders BLK - Blocksize Default value: CYL Example: --su blk

Allowed values: TRK, CYL, BLK

- `--primary-space` | `--ps` *(number)*
    - Specifies primary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ps 3
- `--secondary-space` | `--ss` *(number)*
    - Specifies secondary space allocation unit. Allowed values: 1-16777215 Default value: 1 Example: --ss 5
- `--volume` | `-v` *(string)*
    - Specifies a disk volume or specific tapes. Example: -v vol002
- `--unit-type` | `--ut` *(string)*
    - Specifies the DASD unit name. Example: --ut sysda
- `--expiration-date` | `--ed` *(string)*
    - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
    - Specifies the storage class. Example: --sc scl002

- `--management-class` | `--mc` *(string)*
    - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
    - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a PDSE with default option values:

- `$ zowe file-master-plus create partitioned-data-set-extended fmmvs.test.dsname`

  *- Creating a PDSE version 2 with options:

- `$ zowe file-master-plus create partitioned-data-set-extended fmmvs.test.dsname --ver 2 -g 10 --lrecl 180 --blksize 32720 --recfm vb --ps 5 --ss 5 -v vol005 --su trk --ed 2025-09-27`

  *- Creating a PDSE data set like a model PDSE:

- `$ zowe file-master-plus create partitioned-data-set-extended fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a PDSE like a model PDSE and the parameters with options:

- `$ zowe file-master-plus create partitioned-data-set-extended fmmvs.test.dsname --model fmmvs.model.dsname --lrecl 180 --recfm vb --ps 5 --ss 5 --db 5 -v vol005 --ed p`

vsam-ksds

Create a key-sequenced data set (KSDS) Virtual Storage Access Method (VSAM) data set.

Usage

zowe file-master-plus create vsam-ksds [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
  - Specifies the name of a model KSDS (key-sequenced VSAM data set) for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--keys-position` | `--kp` *(number)*
  - Specifies the position of the key within the base cluster. This parameter is mandatory unless a model data set is specified. Example: --kp 1
- `--keys-length` | `--kl` *(number)*
  - Specifies the length of the key within the base cluster. This parameter is mandatory unless a model data set is specified. Example: --kl 7
- `--maximum-record-size` | `--mrs` *(number)*
  - Specifies the maximum length of data records. This parameter is mandatory unless a model data set is specified. Example: --mrs 180
- `--average-record-size` | `--ars` *(number)*
  - Specifies the average length of data records. If this parameter is not specified then it is set to the same as the maximum-record-size option. Example: --ars 110
- `--data-dsname` | `--ddsn` *(string)*
  - Specifies the data set name of the data VSAM component. If this parameter is not specified then by default it is set to the cluster name with '.DATA'. Example: --ddsn fmmvs.vsam.ksds.data
- `--data-control-interval-size` | `--dcis` *(number)*
  - Specifies the size of the Control Interval for the data VSAM component. Default value: 20480 Example: --dcis 3584
- `--data-space-units` | `--dsu` *(string)*

    - Specifies the space allocation unit for data vsam component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --dsu rec

Allowed values: TRK, CYL, REC, K, M

- `--data-primary-space` | `--dps` *(number)*
    - Specifies the primary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 18 Example: --dps 3
- `--data-secondary-space` | `--dss` *(number)*
    - Specifies the secondary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 3 Example: --dss 5
- `--data-volume` | `--dv` *(array)*
    - Specifies a disk volume or specific tapes for the data VSAM component. Example: -dv vol002
- `--index-dsname` | `--idsn` *(string)*
    - Specifies the data set name of the index VSAM component. If this parameter is not specified then it is set to the cluster name with '.INDEX' appended. Example: --ddsn fmmvs.vsam.ksds.index
- `--index-control-interval-size` | `--icis` *(number)*
    - Specifies the size of Control Interval for index VSAM component. Default value: 512 Example: --icis 3584
- `--index-space-units` | `--isu` *(string)*
    - Specifies the space allocation unit for index VSAM component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --isu cyl

Allowed values: TRK, CYL, REC, K, M

- `--index-primary-space` | `--ips` *(number)*
    - Specifies primary space allocation unit for index VSAM component. Allowed values: 1-16777215 Default value: 1 Example: --ips 3
- `--index-secondary-space` | `--iss` *(number)*

    - Specifies secondary space allocation unit for index VSAM component. Allowed values: 1-16777215 Default value: 1 Example: --iss 5

- `--index-volume` | `--iv` *(array)*

    - Specifies a disk volume or specific tapes for the index VSAM component. Example: -iv vol002

- `--buffer-space` | `--bs` *(number)*

    - Specifies the minimum buffer space to allocate when this VSAM file is accessed. Example: --bs 37376

- `--erase` | `-e` *(string)*

    - Indicates whether the VSAM file was allocated with the ERASE parameter, causing all components of the file to be overwritten with binary zeros, when the VSAM file is deleted from the catalog. Default value: n Example: -e y

Allowed values: y, n

- `--load-restartable` | `--lr` *(string)*

    - Specify 'Y' to request that the VSAM component definition use the 'RECOVERY' parameter (which causes the data component to be preformatted previous to the initial load). Using this option causes the initial load to take longer, but loads which do not complete successfully can be restarted. Specify 'N' to request that the VSAM component definition use the 'SPEED' parameter (which causes the data component to NOT be preformatted previous to the initial load). Default value: n Example: -lr y

Allowed values: y, n

- `--reuse` | `-r` *(string)*

    - Indicates whether the VSAM file was allocated with the REUSE parameter specifying that the cluster can be opened again and again as a reusable cluster. Default value: n Example: -r y

Allowed values: y, n

- `--spanned` | `-s` *(string)*

    - Indicates whether VSAM file was allocated with the SPANNED parameter indicating that data records larger than a control interval can span multiple control intervals. Default value: n Example: -s y

Allowed values: y, n

- `--write-check` | `--wc` *(string)*
  - Indicates whether the VSAM file was allocated with the WRITECHECK parameter requesting each write to the VSAM file to be validated by a read without data transfer. Default value: n Example: --wc y

Allowed values: y, n

- `--control-interval-freespace-percentage` | `--cifp` *(number)*
  - Specifies the percentage of empty space in each control interval when the file is initially loaded.The free space lets records be inserted or expanded within a control interval before requiring a control interval split. Example: --cifp 10
- `--control-area-freespace-percentage` | `--cafp` *(number)*
  - Specifies the percentage of control intervals to be left unused in each control area as the file is initially loaded. The use of control area free space lets some control interval splits occur before requiring a control area split. Example: --cafp 10
- `--cross-region-share-option` | `--crso` *(number)*
  - Specifies that the file can be shared among regions within the same system or within multiple systems using GRS (Global Resource Serialization). The allowed values have the following meaning: 1 - The data set can be opened for read processing by an unlimited number of users, but the data set can be accessed by only one user when that user is doing read and write processing. 2 - The data set can be opened by only one user at a time for read and write processing, but any number of users can also be accessing the data set for read processing 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 1 Example: --crso 2

Allowed values: 1, 2, 3, 4

- `--cross-system-share-option` | `--csso` *(number)*
  - Specifies how the file can be shared among systems. The allowed values have the following meaning: 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 3 Example: --csso 4

Allowed values: 3, 4

- `--expiration-date` | `--ed` *(string)*
  - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
  - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
  - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
  - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a KSDS with default option values and mandatory options:

- `$ zowe file-master-plus create vsam-ksds fmmvs.test.dsname --kp 1 --kl 7 --mrs 160`

  *- Creating a KSDS with options:

- `$ zowe file-master-plus create vsam-ksds fmmvs.test.dsname --kp 1 --kl 7 --mrs 160 --ars 120 --dcis 3584 --dsu cyl --dps 1 --dss 3 --dv vol002 --icis 512 --isu cyl --ips 1 --iss 1 --iv vol002 --bs 37376 -e y -r y -s n --ed 2025-09-27`

  *- Creating a KSDS data set like a model KSDS data set:

- `$ zowe file-master-plus create vsam-ksds fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a KSDS like a model KSDS data set and overriding the parameters with options:

- `$ zowe file-master-plus create vsam-ksds fmmvs.test.dsname --model fmmvs.model.dsname --kp 1 --kl 7 --mrs 160 --wc y --bs 37376 --crso 3 --csso 4 --ed p`

vsam-esds

Create an entry-sequenced Virtual Storage Access Method (VSAM) data set (ESDS).

Usage

zowe file-master-plus create vsam-esds [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
  - Specifies the name of a ESDS (entry-sequenced VSAM data set) for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--maximum-record-size` | `--mrs` *(number)*
  - Specifies the maximum length of data records. This parameter is mandatory unless a model data set is specified. Example: --mrs 180
- `--average-record-size` | `--ars` *(number)*
  - Specifies the average length of data records. If this parameter is not specified then it is set to the same as the maximum-record-size option. Example: --ars 110
- `--data-dsname` | `--ddsn` *(string)*
  - Specifies the data set name of the data VSAM component. If this parameter is not specified then by default it is set to the cluster name with '.DATA'. Example: --ddsn fmmvs.vsam.esds.data
- `--data-control-interval-size` | `--dcis` *(number)*
  - Specifies the size of the Control Interval for the data VSAM component. Default value: 20480 Example: --dcis 3584

- `--data-space-units` | `--dsu` *(string)*
    - Specifies the space allocation unit for data vsam component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --dsu rec

Allowed values: TRK, CYL, REC, K, M

- `--data-primary-space` | `--dps` *(number)*
    - Specifies the primary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 18 Example: --dps 3
- `--data-secondary-space` | `--dss` *(number)*
    - Specifies the secondary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 3 Example: --dss 5
- `--data-volume` | `--dv` *(array)*
    - Specifies a disk volume or specific tapes for the data VSAM component. Example: -dv vol002
- `--buffer-space` | `--bs` *(number)*
    - Specifies the minimum buffer space to allocate when this VSAM file is accessed. Example: --bs 37376
- `--erase` | `-e` *(string)*
    - Indicates whether the VSAM file was allocated with the ERASE parameter, causing all components of the file to be overwritten with binary zeros, when the VSAM file is deleted from the catalog. Default value: n Example: -e y

Allowed values: y, n

- `--load-restartable` | `--lr` *(string)*
    - Specify 'Y' to request that the VSAM component definition use the 'RECOVERY' parameter (which causes the data component to be preformatted previous to the initial load). Using this option causes the initial load to take longer, but loads which do not complete successfully can be restarted. Specify 'N' to request that the VSAM component definition use the 'SPEED' parameter (which causes the data component to NOT be preformatted previous to the initial load). Default value: n Example: -lr y

Allowed values: y, n

- `--reuse` | `-r` *(string)*
    - Indicates whether the VSAM file was allocated with the REUSE parameter specifying that the cluster can be opened again and again as a reusable cluster. Default value: n Example: -r y

Allowed values: y, n

- `--spanned` | `-s` *(string)*
    - Indicates whether VSAM file was allocated with the SPANNED parameter indicating that data records larger than a control interval can span multiple control intervals. Default value: n Example: -s y

Allowed values: y, n

- `--write-check` | `--wc` *(string)*
    - Indicates whether the VSAM file was allocated with the WRITECHECK parameter requesting each write to the VSAM file to be validated by a read without data transfer. Default value: n Example: --wc y

Allowed values: y, n

- `--control-interval-freespace-percentage` | `--cifp` *(number)*
    - Specifies the percentage of empty space in each control interval when the file is initially loaded.The free space lets records be inserted or expanded within a control interval before requiring a control interval split. Example: --cifp 10
- `--control-area-freespace-percentage` | `--cafp` *(number)*
    - Specifies the percentage of control intervals to be left unused in each control area as the file is initially loaded. The use of control area free space lets some control interval splits occur before requiring a control area split. Example: --cafp 10
- `--cross-region-share-option` | `--crso` *(number)*
    - Specifies that the file can be shared among regions within the same system or within multiple systems using GRS (Global Resource Serialization). The allowed values have the following meaning: 1 - The data set can be opened for read processing by an unlimited number of users, but the data set can be accessed by only one user when that user is doing read and write processing. 2 - The data set can be opened by only one user at a time for read and write processing, but any number of users can also be accessing the data set for read processing 3 - The data set can be fully shared by any number of users.

4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 1 Example: --crso 2

Allowed values: 1, 2, 3, 4

- `--cross-system-share-option` | `--csso` *(number)*
  - Specifies how the file can be shared among systems. The allowed values have the following meaning: 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 3 Example: --csso 4

Allowed values: 3, 4

- `--expiration-date` | `--ed` *(string)*
  - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
  - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
  - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
  - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a ESDS with default option values and mandatory options:

- `$ zowe file-master-plus create vsam-esds fmmvs.test.dsname --mrs 160`

*- Creating a ESDS with options:

- `$ zowe file-master-plus create vsam-esds fmmvs.test.dsname --mrs 160 --ars 120 --dcis 3584 --dsu cyl --dps 1 --dss 3 --dv vol002 --bs 37376 -e y -r y -s n --ed 2025-09-27`

*- Creating a ESDS data set like a model ESDS data set:

- `$ zowe file-master-plus create vsam-esds fmmvs.test.dsname --model fmmvs.model.dsname`

*- Creating a ESDS like a model ESDS data set and overriding the parameters with options:

- `$ zowe file-master-plus create vsam-esds fmmvs.test.dsname --model fmmvs.model.dsname --mrs 160 --wc y --bs 37376 --crso 3 --csso 4 --ed p`

vsam-lds

Create linear data set (LDS) Virtual Storage Access Method (VSAM) data set.

Usage

zowe file-master-plus create vsam-lds [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
  - Specifies the name of a model LDS (Linear VSAM data set) for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--data-dsname` | `--ddsn` *(string)*
  - Specifies the data set name of the data VSAM component. If this parameter is not specified then by default it is set to the cluster name with '.DATA'. Example: --ddsn fmmvs.vsam.esds.data

- `--data-control-interval-size` | `--dcis` *(number)*
  - Specifies the size of the Control Interval for the data VSAM component. Default value: 4096 Example: --dcis 3584
- `--data-space-units` | `--dsu` *(string)*
  - Specifies the space allocation unit for data vsam component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --dsu rec

Allowed values: TRK, CYL, REC, K, M

- `--data-primary-space` | `--dps` *(number)*
  - Specifies the primary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 18 Example: --dps 3
- `--data-secondary-space` | `--dss` *(number)*
  - Specifies the secondary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 3 Example: --dss 5
- `--data-volume` | `--dv` *(array)*
  - Specifies a disk volume or specific tapes for the data VSAM component. Example: -dv vol002
- `--buffer-space` | `--bs` *(number)*
  - Specifies the minimum buffer space to allocate when this VSAM file is accessed. Example: --bs 37376
- `--erase` | `-e` *(string)*
  - Indicates whether the VSAM file was allocated with the ERASE parameter, causing all components of the file to be overwritten with binary zeros, when the VSAM file is deleted from the catalog. Default value: n Example: -e y

Allowed values: y, n

- `--load-restartable` | `--lr` *(string)*
  - Specify 'Y' to request that the VSAM component definition use the 'RECOVERY' parameter (which causes the data component to be preformatted previous to the initial load). Using this option causes the initial load to take longer, but loads which do not

complete successfully can be restarted. Specify 'N' to request that the VSAM component definition use the 'SPEED' parameter (which causes the data component to NOT be preformatted previous to the initial load). Default value: n Example: -lr y

Allowed values: y, n

- `--reuse` | `-r` *(string)*
  - Indicates whether the VSAM file was allocated with the REUSE parameter specifying that the cluster can be opened again and again as a reusable cluster. Default value: n Example: -r y

Allowed values: y, n

- `--write-check` | `--wc` *(string)*
  - Indicates whether the VSAM file was allocated with the WRITECHECK parameter requesting each write to the VSAM file to be validated by a read without data transfer. Default value: n Example: --wc y

Allowed values: y, n

- `--control-interval-freespace-percentage` | `--cifp` *(number)*
  - Specifies the percentage of empty space in each control interval when the file is initially loaded.The free space lets records be inserted or expanded within a control interval before requiring a control interval split. Example: --cifp 10
- `--control-area-freespace-percentage` | `--cafp` *(number)*
  - Specifies the percentage of control intervals to be left unused in each control area as the file is initially loaded. The use of control area free space lets some control interval splits occur before requiring a control area split. Example: --cafp 10
- `--cross-region-share-option` | `--crso` *(number)*
  - Specifies that the file can be shared among regions within the same system or within multiple systems using GRS (Global Resource Serialization). The alloweThe allowed values have the following meaning: 1 - The data set can be opened for read processing by an unlimited number of users, but the data set can be accessed by only one user when that user is doing read and write processing. 2 - The data set can be opened by only one user at a time for read and write processing, but any number of users can also be accessing the data set for read processing 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs.

Default value: 1 Example: --crso 2

Allowed values: 1, 2, 3, 4

- `--cross-system-share-option` | `--csso` *(number)*
  - Specifies how the file can be shared among systems. The allowed values have the following meaning: 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 3 Example: --csso 4

Allowed values: 3, 4

- `--expiration-date` | `--ed` *(string)*
  - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
  - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
  - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
  - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a LDS with default option values:

- `$ zowe file-master-plus create vsam-lds fmmvs.test.dsname`

*- Creating a LDS with options:

- `$ zowe file-master-plus create vsam-lds fmmvs.test.dsname --dcis 3584 --dsu cyl --dps 1 --dss 3 --dv vol002 --bs 37376 -e y -r y -s n --ed p`

  *- Creating a LDS data set like a model LDS data set:

- `$ zowe file-master-plus create vsam-lds fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a LDS like a model LDS data set and overriding the parameters with options:

- `$ zowe file-master-plus create vsam-lds fmmvs.test.dsname --model fmmvs.model.dsname --wc y --bs 37376 --crso 3 --csso 4 --ed 2025-09-27`

vsam-rrds

Create a relative-record data set (RRDS) Virtual Storage Access Method (VSAM) data set.

Usage

zowe file-master-plus create vsam-rrds [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
  - Specifies the name of a model RRDS (relative-record VSAM data set) for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--maximum-record-size` | `--mrs` *(number)*
  - Specifies the maximum length of data records. This parameter is mandatory unless a model data set is specified. Example: --mrs 180
- `--data-dsname` | `--ddsn` *(string)*
  - Specifies the data set name of the data VSAM component. If this parameter is not specified then by default it is set to the cluster name with '.DATA'. Example: --ddsn fmmvs.vsam.rrds.data

- `--data-control-interval-size` | `--dcis` *(number)*
  - Specifies the size of the Control Interval for the data VSAM component. Default value: 20480 Example: --dcis 3584
- `--data-space-units` | `--dsu` *(string)*
  - Specifies the space allocation unit for data vsam component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --dsu rec

Allowed values: TRK, CYL, REC, K, M

- `--data-primary-space` | `--dps` *(number)*
  - Specifies the primary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 18 Example: --dps 3
- `--data-secondary-space` | `--dss` *(number)*
  - Specifies the secondary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 3 Example: --dss 5
- `--data-volume` | `--dv` *(array)*
  - Specifies a disk volume or specific tapes for the data VSAM component. Example: -dv vol002
- `--buffer-space` | `--bs` *(number)*
  - Specifies the minimum buffer space to allocate when this VSAM file is accessed. Example: --bs 37376
- `--erase` | `-e` *(string)*
  - Indicates whether the VSAM file was allocated with the ERASE parameter, causing all components of the file to be overwritten with binary zeros, when the VSAM file is deleted from the catalog. Default value: n Example: -e y

Allowed values: y, n

- `--load-restartable` | `--lr` *(string)*
  - Specify 'Y' to request that the VSAM component definition use the 'RECOVERY' parameter (which causes the data component to be preformatted previous to the initial load). Using this option causes the initial load to take longer, but loads which do not

complete successfully can be restarted. Specify 'N' to request that the VSAM component definition use the 'SPEED' parameter (which causes the data component to NOT be preformatted previous to the initial load). Default value: n Example: -lr y

Allowed values: y, n

- `--reuse` | `-r` *(string)*
  - Indicates whether the VSAM file was allocated with the REUSE parameter specifying that the cluster can be opened again and again as a reusable cluster. Default value: n Example: -r y

Allowed values: y, n

- `--write-check` | `--wc` *(string)*
  - Indicates whether the VSAM file was allocated with the WRITECHECK parameter requesting each write to the VSAM file to be validated by a read without data transfer. Default value: n Example: --wc y

Allowed values: y, n

- `--control-interval-freespace-percentage` | `--cifp` *(number)*
  - Specifies the percentage of empty space in each control interval when the file is initially loaded.The free space lets records be inserted or expanded within a control interval before requiring a control interval split. Example: --cifp 10
- `--control-area-freespace-percentage` | `--cafp` *(number)*
  - Specifies the percentage of control intervals to be left unused in each control area as the file is initially loaded. The use of control area free space lets some control interval splits occur before requiring a control area split. Example: --cafp 10
- `--cross-region-share-option` | `--crso` *(number)*
  - Specifies that the file can be shared among regions within the same system or within multiple systems using GRS (Global Resource Serialization). The allowed values have the following meaning: 1 - The data set can be opened for read processing by an unlimited number of users, but the data set can be accessed by only one user when that user is doing read and write processing. 2 - The data set can be opened by only one user at a time for read and write processing, but any number of users can also be accessing the data set for read processing 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 1 Example:

--crso 2

Allowed values: 1, 2, 3, 4

- `--cross-system-share-option` | `--csso` *(number)*
  - Specifies how the file can be shared among systems. The allowed values have the following meaning: 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 3 Example: --csso 4

Allowed values: 3, 4

- `--expiration-date` | `--ed` *(string)*
  - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
  - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
  - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
  - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a RRDS with default option values and mandatory options:

- `$ zowe file-master-plus create vsam-rrds fmmvs.test.dsname --mrs 160`

*- Creating a RRDS with options:

- `$ zowe file-master-plus create vsam-rrds fmmvs.test.dsname --mrs 160 --dcis 3584 --dsu cyl --dps 1 --dss 3 --dv vol002 --bs 37376 -e y -r y -s n --ed p`

  *- Creating a RRDS data set like a model RRDS data set:

- `$ zowe file-master-plus create vsam-rrds fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a RRDS like a model RRDS data set and overriding the parameters with options:

- `$ zowe file-master-plus create vsam-rrds fmmvs.test.dsname --model fmmvs.model.dsname --mrs 160 --wc y --bs 37376 --crso 3 --csso 4 --ed 2025-09-27`

vsam-vrrds

Create a variable-length relative-record data set (VRRDS) Virtual Storage Access Method (VSAM) data set.

Usage

zowe file-master-plus create vsam-vrrds [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.

Options

- `--model` | `-m` *(string)*
  - Specifies the name of a model VRRDS (variable-length relative-record VSAM data set) for allocating parameters. The parameters of the model data set override all defaults. Example: -m fmmvs.model.dsname
- `--maximum-record-size` | `--mrs` *(number)*
  - Specifies the maximum length of data records. This parameter is mandatory unless a model data set is specified and it should be greater than the average record size. Example: --mrs 180
- `--average-record-size` | `--ars` *(number)*
  - Specifies the average length of data records. This parameter is mandatory unless a

model data set is specified and it should be less than the maximum record size. Example: --ars 110

- `--data-dsname` | `--ddsn` *(string)*
  - Specifies the data set name of the data VSAM component. If this parameter is not specified then by default it is set to the cluster name with '.DATA'. Example: --ddsn fmmvs.vsam.ksds.data
- `--data-control-interval-size` | `--dcis` *(number)*
  - Specifies the size of the Control Interval for the data VSAM component. Default value: 20480 Example: --dcis 3584
- `--data-space-units` | `--dsu` *(string)*
  - Specifies the space allocation unit for data vsam component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --dsu rec

Allowed values: TRK, CYL, REC, K, M

- `--data-primary-space` | `--dps` *(number)*
  - Specifies the primary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 18 Example: --dps 3
- `--data-secondary-space` | `--dss` *(number)*
  - Specifies the secondary space allocation unit for the data VSAM component. Allowed values: 1-16777215 Default value: 3 Example: --dss 5
- `--data-volume` | `--dv` *(array)*
  - Specifies a disk volume or specific tapes for the data VSAM component. Example: -dv vol002
- `--index-dsname` | `--idsn` *(string)*
  - Specifies the data set name of the index VSAM component. If this parameter is not specified then it is set to the cluster name with '.INDEX' appended. Example: --ddsn fmmvs.vsam.ksds.index
- `--index-control-interval-size` | `--icis` *(number)*
  - Specifies the size of Control Interval for index VSAM component. Default value: 512 Example: --icis 3584

- `--index-space-units` | `--isu` *(string)*
    - Specifies the space allocation unit for index VSAM component. The allowed values have the following meaning: TRK - Tracks CYL - Cylinders REC - Records K - Kilobytes M - Megabytes Default value: TRK Example: --isu cyl

Allowed values: TRK, CYL, REC, K, M

- `--index-primary-space` | `--ips` *(number)*
    - Specifies primary space allocation unit for index VSAM component. Allowed values: 1-16777215 Default value: 1 Example: --ips 3
- `--index-secondary-space` | `--iss` *(number)*
    - Specifies secondary space allocation unit for index VSAM component. Allowed values: 1-16777215 Default value: 1 Example: --iss 5
- `--index-volume` | `--iv` *(array)*
    - Specifies a disk volume or specific tapes for the data VSAM component. Example: -iv vol002
- `--buffer-space` | `--bs` *(number)*
    - Specifies the minimum buffer space to allocate when this VSAM file is accessed. Example: --bs 37376
- `--erase` | `-e` *(string)*
    - Indicates whether the VSAM file was allocated with the ERASE parameter, causing all components of the file to be overwritten with binary zeros, when the VSAM file is deleted from the catalog. Default value: n Example: -e y

Allowed values: y, n

- `--load-restartable` | `--lr` *(string)*
    - Specify 'Y' to request that the VSAM component definition use the 'RECOVERY' parameter (which causes the data component to be preformatted previous to the initial load). Using this option causes the initial load to take longer, but loads which do not complete successfully can be restarted. Specify 'N' to request that the VSAM component definition use the 'SPEED' parameter (which causes the data component to NOT be preformatted previous to the initial load). Default value: n Example: -lr y

Allowed values: y, n

- `--reuse` | `-r` *(string)*
  - Indicates whether the VSAM file was allocated with the REUSE parameter specifying that the cluster can be opened again and again as a reusable cluster. Default value: n Example: -r y

Allowed values: y, n

- `--write-check` | `--wc` *(string)*
  - Indicates whether the VSAM file was allocated with the WRITECHECK parameter requesting each write to the VSAM file to be validated by a read without data transfer. Default value: n Example: --wc y

Allowed values: y, n

- `--control-interval-freespace-percentage` | `--cifp` *(number)*
  - Specifies the percentage of empty space in each control interval when the file is initially loaded.The free space lets records be inserted or expanded within a control interval before requiring a control interval split. Example: --cifp 10
- `--control-area-freespace-percentage` | `--cafp` *(number)*
  - Specifies the percentage of control intervals to be left unused in each control area as the file is initially loaded. The use of control area free space lets some control interval splits occur before requiring a control area split. Example: --cafp 10
- `--cross-region-share-option` | `--crso` *(number)*
  - Specifies that the file can be shared among regions within the same system or within multiple systems using GRS (Global Resource Serialization). The allowed values have the following meaning: 1 - The data set can be opened for read processing by an unlimited number of users, but the data set can be accessed by only one user when that user is doing read and write processing. 2 - The data set can be opened by only one user at a time for read and write processing, but any number of users can also be accessing the data set for read processing 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 1 Example: --crso 2

Allowed values: 1, 2, 3, 4

- `--cross-system-share-option` | `--csso` *(number)*

    - Specifies how the file can be shared among systems. The allowed values have the following meaning: 3 - The data set can be fully shared by any number of users. 4 - The data set can be fully shared by any number of users. VSAM immediately updates the data set for PUTs and refreshes all input buffers for GETs. Default value: 3 Example: --csso 4

Allowed values: 3, 4

- `--expiration-date` | `--ed` *(string)*
    - Specifies the expiration date after which the data set can be deleted. Specify 'P' or 'p' to make the data set permanent, or an expiration date in YYYY-MM-DD format. Example: --ed 2032-07-31
- `--storage-class` | `--sc` *(string)*
    - Specifies the storage class. Example: --sc scl002
- `--management-class` | `--mc` *(string)*
    - Specifies the management class. Example: --mc mcl002
- `--data-class` | `--dc` *(string)*
    - Specifies the data class. Example: --dc dcl002

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a VRRDS with default option values and mandatory options:

- `$ zowe file-master-plus create vsam-vrrds fmmvs.test.dsname --mrs 160`

  *- Creating a VRRDS with options:

- `$ zowe file-master-plus create vsam-vrrds fmmvs.test.dsname --mrs 160 --ars 120 --dcis 3584 --dsu cyl --dps 1 --dss 3 --dv vol002 --icis 512 --isu cyl --ips 1 --iss 1 --iv vol002 --bs 37376 -e y -r y --ed 2025-09-27`

  *- Creating a VRRDS data set like a model VRRDS data set:

- `$ zowe file-master-plus create vsam-vrrds fmmvs.test.dsname --model fmmvs.model.dsname`

  *- Creating a VRRDS like a model VRRDS data set and overriding the parameters with options:

- `$ zowe file-master-plus create vsam-vrrds fmmvs.test.dsname --model fmmvs.model.dsname --mrs 160 --wc y --bs 37376 --crso 3 --csso 4 --ed p`

like-model

Create a data set by allocating parameters from a model data set.

Usage

zowe file-master-plus create like-model [options]

Positional Arguments

- `name` *(string)*
  - Specifies the name of the data set to create.
- `model` *(string)*
  - Specifies the name of the model data set.

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Creating a data set like a model data set:

- `$ zowe file-master-plus create like-model fmmvs.create.dsname fmmvs.model.dsname`

delete | del

Permanently deletes a data set.

data-set

Permanently deletes a data set. It supports all data set types that are supported by CA File Master Plus.

Usage

zowe file-master-plus delete data-set [options]

Positional Arguments

- `name` *(string)*
    - Specifies the name of the data set that you want to delete.

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

Examples

*- Deleting a data set:

- `$ zowe file-master-plus delete data-set fmmvs.dsname`

populate | pop

Populate the specified data set with records. It supports all data set types that are supported by CA File Master Plus.

data-set

Populate a specific data set with records. The layout of the records to add are described by a Cobol or PL/I copybook.

Usage

zowe file-master-plus populate data-set [options]

Positional Arguments

- `name` *(string)*
    - Specifies the name of the data set to populate.

Options

- `--member` | `-m` *(string)*
    - Specifies name of the member. Note: Used only if the specified data set is a PDS or

PDSE. Example: -m member1.

Required Options

- `--layout-member` | `--lm` *(string)*
  - Specifies the name of the Cobol or PL/I copybook. Example: --lm testlay.
- `--layout-data-set` | `--lds` *(string)*
  - Specifies the name of the data set that contains the layout member. Example: --lds fmmvs.layout.dataset.
- `--data` | `-d` *(string)*
  - Specifies path of the .txt/.json file of the data stream. Ensure the data is in JSON format represented by a layout data set and its member. Example: -d ../instream/data1.txt

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
  - The name of a (fmp) profile to load for this command execution.

Examples

*- Populating a data set:

- `$ zowe file-master-plus populate data-set fmmvs.dsname --lds fmmvs.layout.dataset --lm testlay --data ../instream/data1.txt`

rename | ren

Rename the specified data set.

data-set

Rename a data set. It supports all data set types that are supported by CA File Master Plus.

Usage

zowe file-master-plus rename data-set [options]

Positional Arguments

- `old` *(string)*
  - Specifies the name of the data set that you want to rename.

- new *(string)*
    - Specifies the new name of the data set.

Options

- `--vsam-component` | `--vsamc` *(string)*
    - Rename the data and index components of a VSAM cluster if they share the cluster name. Note: Used only if the target data set is a VSAM cluster. Example: --vsamc y.

Default value: n Allowed values: y, n

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

Examples

*- Renaming a data set:

- `$ zowe file-master-plus rename data-set fmmvs.old.dsname fmmvs.new.dsname`

  *- Renaming a VSAM data set along with its components:

- `$ zowe file-master-plus rename data-set fmmvs.oldvsam.dsname fmmvs.newvsam.dsname --vsamc y`

ops

Control OPS/MVS from Brightside.

disable

Sets the new status of the SSM rule to 'DISABLE'This will cause SSM to take the necessary action to disable that rule.

rule

Does not take into account preregs

Usage

zowe ops disable rule [options]

Positional Arguments

- `ruleset` *(string)*
  - The rule set in which the rule is located
- `rule` *(string)*
  - The name of the desired rule

Options

- `--subsystem` *(string)*
  - OPS subsystem to route the command to. Must be up on the LPAR selected in the profile

Profile Options

- `--ops-profile` | `--ops-p` *(string)*
  - The name of a (ops) profile to load for this command execution.

enable

Sets the new status of the SSM rule to 'ENABLE'This will cause SSM to take the necessary action to enable that rule.

rule

Does not take into account preregs

Usage

zowe ops enable rule [options]

Positional Arguments

- `ruleset` *(string)*
  - The rule set in which the rule is located
- `rule` *(string)*

    - The name of the desired rule

Options

- `--subsystem` *(string)*
    - OPS subsystem to route the command to. Must be up on the LPAR selected in the profile

Profile Options

- `--ops-profile` | `--ops-p` *(string)*
    - The name of a (ops) profile to load for this command execution.

show

Displays various data relevant to or contained within OPS/MVSFor example, rule enabled/disabled status or SSM resource state.

resource

Currently, only rule state is displayed. More detailed information forthcoming in later plugin updates, depending on popular demand

Usage

zowe ops show resource [options]

Positional Arguments

- `resourcename` *(string)*
    - The name of the desired resource

Options

- `--tablename` *(string)*
    - The name of the table in which the resource is located, if known
- `--subsystem` *(string)*
    - OPS subsystem to route the command to. Must be up on the LPAR selected in the profile

Profile Options

- `--ops-profile` | `--ops-p` *(string)*
    - The name of a (ops) profile to load for this command execution.

rule

In the future, this command will return detailed information about the selected rule.

Currently, only rule status is returned.

Usage

zowe ops show rule [options]

Positional Arguments

- `ruleset` *(string)*
    - The rule set in which the rule is located
- `rule` *(string)*
    - The name of the desired rule

Profile Options

- `--ops-profile` | `--ops-p` *(string)*
    - The name of a (ops) profile to load for this command execution.

start

Sets the desired state of the SSM resource to 'UP'This will cause SSM to take the necessary action to bring that resource up.

resource

Does not take into account prereqs.

Usage

zowe ops start resource [options]

Positional Arguments

- `resourcename` *(string)*
    - The name of the desired resource

Options

- `--tablename` *(string)*
  - The name of the table in which the resource is located, if known
- `--subsystem` *(string)*
  - OPS subsystem to route the command to. Must be up on the LPAR selected in the profile

Profile Options

- `--ops-profile` | `--ops-p` *(string)*
  - The name of a (ops) profile to load for this command execution.

stop

Sets the desired state of the SSM resource to 'DOWN'This will cause SSM to take the necessary action to bring that resource down.

resource

Does not take into account prereqs.

Usage

zowe ops stop resource [options]

Positional Arguments

- `resourcename` *(string)*
  - The name of the desired resource

Options

- `--tablename` *(string)*
  - The name of the table in which the resource is located, if known
- `--subsystem` *(string)*
  - OPS subsystem to route the command to. Must be up on the LPAR selected in the profile

Profile Options

- `--ops-profile` | `--ops-p` *(string)*
  - The name of a (ops) profile to load for this command execution.

plugins

Install and manage plug-ins

install

Install plug-ins to an application.

Usage

zowe plugins install [plugin...] [options]

Positional Arguments

- `plugin...` *(string)*
  - A space-separated list of plug-ins to install. A plug-in can be any format that is accepted by the `npm install` command (local directory, TAR file, git URL, public package, private package, etc...).

To use a relative local directory, at least one '/' or '\' must exist in the plug-in path. For example, you have a local plug-in in a folder called 'test-plugin' that you want to install. Specify the relative local directory by issuing the following command:

zowe plugins install ./test-plugin

If you omit the './', then the install command looks for 'test-plugin' in an npm registry.

If the plugin argument is omitted, the plugins.json file will determine which plug-ins are installed. For more information on the plugins.json file, see the --file option.

Options

- `--file` *(local file path)*
  - Specifies the location of a plugins.json file that contains the plug-ins you want to install.

All plug-ins specified in plugins.json will be installed to the base CLI and the contents will be placed into C:\Users\RIJFE01.zowe\plugins\plugins.json.

If you do not specify a plugins.json file and do not specify a plug-in, the default plugin.json file (C:\Users\RIJFE01.zowe\plugins\plugins.json) will be used. This provides a way to install plug-ins that were lost or corrupted after reinstalling or updating Zowe CLI.

- `--registry` *(string)*
    - The npm registry that is used when installing remote packages. When this value is omitted, the value returned by `npm config get registry` is used.

For more information about npm registries, see: https://docs.npmjs.com/misc/registry

Examples

*- Install plug-ins saved in C:\Users\RIJFE01.zowe\plugins\plugins.json:

- `$ zowe plugins install`

  *- Install plug-ins saved in a properly formatted config file:

- `$ zowe plugins install --file /some/file/path/file_name.json`

  *- Install a remote plug-in:

- `$ zowe plugins install my-plugin`

  *- Install a remote plug-in using semver:

- `$ zowe plugins install my-plugin@"^1.2.3"`

  *- Install a remote plug-in from the specified registry:

- `$ zowe plugins install my-plugin --registry https://registry.npmjs.org/`

  *- Install a local folder, local TAR file, and a git URL:

- `$ zowe plugins install ./local-file /root/tar/some-tar.tgz git://github.com/project/repository.git#v1.0.0`

list

List all plug-ins installed.

Usage

zowe plugins list [options]

uninstall

Uninstall plug-ins.

Usage

zowe plugins uninstall [plugin...] [options]

Positional Arguments

- `plugin...` *(string)*
    - The name of the plug-in to uninstall.

If the plug-in argument is omitted, no action is taken.

Examples

*- Uninstall a plug-in:

- `$ zowe plugins uninstall my-plugin`

update

Update plug-ins.

Usage

zowe plugins update [plugin...] [options]

Positional Arguments

- `plugin...` *(string)*
    - The name of the plug-in to update.

If the plug-in argument is omitted, no action is taken.

Options

- `--registry` *(string)*
    - The npm registry that is used when installing remote packages. When this value is omitted, the value returned by `npm config get registry` is used.

For more information about npm registries, see: https://docs.npmjs.com/misc/registry

Examples

*- Update a plug-in:

- `$ zowe plugins update my-plugin`

validate

Validate a plug-in that has been installed.

Usage

zowe plugins validate [plugin] [options]

Positional Arguments

- `plugin` *(string)*
    - The name of the plug-in to validate. Validation issues identified for this plug-in are displayed.

If the plug-in argument is omitted, all installed plug-ins are validated.

Examples

*- Validate a plug-in named my-plugin:

- `$ zowe plugins validate my-plugin`

  *- Validate all installed plug-ins:

- `$ zowe plugins validate`

profiles

Create and manage configuration profiles

create | cre

Create new configuration profiles.

zosmf-profile

z/OSMF Profile

Usage

zowe profiles create zosmf-profile [options]

Positional Arguments

- `profileName` *(string)*
  - Specifies the name of the new zosmf profile. You can load this profile by using the name on commands that support the "--zosmf-profile" option.

Required Options

- `--host` | `-H` *(string)*
  - The z/OSMF server host name.

Options

- `--port` | `-P` *(number)*
  - The z/OSMF server port.

Default value: 443

- `--user` | `-u` *(string)*
  - Mainframe (z/OSMF) user name, which can be the same as your TSO login.
- `--password` | `-p` *(string)*
  - Mainframe (z/OSMF) password, which can be the same as your TSO password.
- `--reject-unauthorized` | `--ru` *(boolean)*
  - Reject self-signed certificates.

Default value: true

- `--overwrite` | `--ow` *(boolean)*
  - Overwrite the zosmf profile when a profile of the same name exists.

Examples

*- Create a zosmf profile called 'zos123' to connect to z/OSMF at host zos123 and port 1443:

- `$ zowe profiles create zosmf-profile zos123 --host zos123 --port 1443 --user ibmuser --password myp4ss`

  *- Create a zosmf profile called 'zos124' to connect to z/OSMF at the host zos124 (default port - 443) and allow self-signed certificates:

- `$ zowe profiles create zosmf-profile zos124 --host zos124 --user ibmuser --password myp4ss --reject-unauthorized false`

tso-profile

z/OS TSO/E User Profile

Usage

zowe profiles create tso-profile [options]

Positional Arguments

- `profileName` *(string)*
  - Specifies the name of the new tso profile. You can load this profile by using the name on commands that support the "--tso-profile" option.

Required Options

- `--account` | `-a` *(string)*
  - Your z/OS TSO/E accounting information.

Options

- `--character-set` | `--cs` *(string)*
  - Character set for address space to convert messages and responses from UTF-8 to EBCDIC.

Default value: 697

- `--code-page` | `--cp` *(string)*

    - Codepage value for TSO/E address space to convert messages and responses from UTF-8 to EBCDIC.

Default value: 1047

- `--columns` | `--cols` *(number)*
    - The number of columns on a screen.

Default value: 80

- `--logon-procedure` | `-l` *(string)*
    - The logon procedure to use when creating TSO procedures on your behalf.

Default value: IZUFPROC

- `--region-size` | `--rs` *(number)*
    - Region size for the TSO/E address space.

Default value: 4096

- `--rows` *(number)*
    - The number of rows on a screen.

Default value: 24

- `--overwrite` | `--ow` *(boolean)*
    - Overwrite the tso profile when a profile of the same name exists.

Examples

*- Create a tso profile called 'myprof' with default settings and JES accounting information of 'IZUACCT':

- `$ zowe profiles create tso-profile myprof -a IZUACCT`

  *- Create a tso profile called 'largeregion' with a region size of 8192, a logon procedure of MYPROC, and JES accounting information of '1234':

- `$ zowe profiles create tso-profile largeregion -a 1234 --rs 8192`

cics-profile

A cics profile is required to issue commands in the cics command group that interact with CICS regions. The cics profile contains your host, port, user name, and password for the IBM CICS management client interface (CMCI) server of your choice.

Usage

zowe profiles create cics-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new cics profile. You can load this profile by using the name on commands that support the "--cics-profile" option.

Required Options

- `--host` | `-H` *(string)*
    - The CMCI server host name
- `--user` | `-u` *(string)*
    - Your username to connect to CICS
- `--password` | `-p` *(string)*
    - Your password to connect to CICS

Options

- `--port` | `-P` *(number)*
    - The CMCI server port

Default value: 1490

- `--region-name` *(string)*
    - The name of the CICS region name to interact with
- `--cics-plex` *(string)*
    - The name of the CICSPlex to interact with

- `--overwrite` | `--ow` *(boolean)*
    - Overwrite the cics profile when a profile of the same name exists.

Examples

*- Create a cics profile named 'cics123' to connect to CICS at host zos123 and port 1490:

- `$ zowe profiles create cics-profile cics123 --host zos123 --port 1490 --user ibmuser --password myp4ss`

endevor-profile

The CA Endevor SCM session profile schema, where you specify your session information and credentials

Usage

zowe profiles create endevor-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new endevor profile. You can load this profile by using the name on commands that support the "--endevor-profile" option.

Options

- `--hostname` | `--host` *(string)*
    - The hostname of the session
- `--port` | `-p` *(number)*
    - The port number of the session
- `--username` | `--user` *(string)*
    - The username of the session
- `--password` | `--pass` *(string)*
    - The password of the user
- `--protocol` | `--prot` *(string)*
    - The protocol used for connecting to CA Endevor SCM Rest API

Default value: http Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - If set, the server certificate is verified against the list of supplied CAs
- `--overwrite` | `--ow` *(boolean)*
  - Overwrite the endevor profile when a profile of the same name exists.

Examples

*- Create an endevor profile called 'ndvrSample' to connect to CA Endevor SCM web services at host ndvr123 and port 8080, using http protocol, allowing self-signed certificates:

- `$ zowe profiles create endevor-profile ndvrSample --host ndvr123 --port 8080 --user ibmuser --password myp4ss --prot http --reject-unauthorized false`

endevor-location-profile

The CA Endevor SCM element location, where you specify your working environment, system and subsystem

Usage

zowe profiles create endevor-location-profile [options]

Positional Arguments

- `profileName` *(string)*
  - Specifies the name of the new endevor-location profile. You can load this profile by using the name on commands that support the "--endevor-location-profile" option.

Options

- `--instance` | `-i` *(string)*
  - The STC/datasource of the session
- `--environment` | `--env` *(string)*
  - The CA Endevor SCM environment where your project resides

Default value: DEV

- `--system` | `--sys` *(string)*

    - The CA Endevor SCM system where the element resides

- `--subsystem` | `--sub` *(string)*
    - The CA Endevor SCM subsystem where your element resides

- `--type` | `--typ` *(string)*
    - Name of the CA Endevor SCM element's type

- `--stage-number` | `--sn` *(string)*
    - The CA Endevor SCM stage where your project resides

Allowed values: 1, 2

- `--comment` | `--com` *(string)*
    - The CA Endevor SCM comment you want to use when performing an action

- `--ccid` | `--cci` *(string)*
    - The CA Endevor SCM CCID you want to use when performing an action

- `--maxrc` *(number)*
    - The return code of CA Endevor SCM that defines a failed action

Default value: 12

- `--overwrite` | `--ow` *(boolean)*
    - Overwrite the endevor-location profile when a profile of the same name exists.

Examples

*- Create a location profile called 'ndvrLoc' to work at CA Endevor SCM location ENV/1/SYS/SUBSYS, with elements of type COBOL, using CA Endevor SCM web services configuration ENDEVOR:

- `$ zowe profiles create endevor-location-profile ndvrLoc --env ENV --sys SYS --sub SUBSYS --typ COBOL --sn 1 -i ENDEVOR`

  *- Create a location profile called 'ndvrLoc2' to work at CA Endevor SCM location ENV/1/SYS/SUBSYS, using CCID 'CCID' and comment 'sample comment':

- `$ zowe profiles create endevor-location-profile ndvrLoc2 --env ENV --sys SYS --sub SUBSYS --sn 1 --com 'sample comment' --cci 'CCID'`

fmp-profile

CA File Master Plus profile schema.

Usage

zowe profiles create fmp-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new fmp profile. You can load this profile by using the name on commands that support the "--fmp-profile" option.

Required Options

- `--host` | `-H` *(string)*
    - Specifies CA File Master Plus server host name.
- `--port` | `-P` *(string)*
    - Specifies CA File Master Plus server port.
- `--user` | `-u` *(string)*
    - Specifies Mainframe user name. May be the same as TSO login.
- `--pass` | `-p` *(string)*
    - Specifies Mainframe password. May be the same as TSO password.

Options

- `--protocol` | `-o` *(string)*
    - Specifies CA File Master Plus REST API protocol.

Default value: https Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*

    - Reject self-signed certificates.

Default value: true

- `--overwrite` | `--ow` *(boolean)*
    - Overwrite the fmp profile when a profile of the same name exists.

Examples

*- Create a fmp profile with http protocol:

- `$ zowe profiles create fmp-profile fmp123 --host fmphost --port 19853 --user mfuser --pass m4pass --protocol http`

  *- Create a fmp profile with https protocol and allow self-signed certificates:

- `$ zowe profiles create fmp-profile fmp234 --host fmphost --port 19854 --user mfuser --pass m4pass --protocol https --reject-unauthorized false`

ops-profile

The OPS Web Services session profile schema, where you specify your session information and credentials

Usage

zowe profiles create ops-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new ops profile. You can load this profile by using the name on commands that support the "--ops-profile" option.

Options

- `--hostname` | `--host` *(string)*
    - The hostname of the session
- `--port` | `-p` *(number)*
    - The port number of the session

- `--username` | `--user` *(string)*
    - The username of the session
- `--password` | `--pass` *(string)*
    - The password of the user
- `--protocol` | `--prot` *(string)*
    - The protocol used for connecting to OPS Web Services

Default value: https Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - If set, the server certificate is verified against the list of supplied CAs
- `--overwrite` | `--ow` *(boolean)*
    - Overwrite the ops profile when a profile of the same name exists.

Examples

*- Create an OPS profile called 'myLPAR' to connect to OPS Web Services at host lpar123 and port 8080, using http protocol, allowing self-signed certificates:

- `$ zowe profiles create ops-profile myLPAR --host lpar123 --port 8080 --user ibmuser --password !@#$^ --prot http --reject-unauthorized false`

db2-profile

A profile for interaction with Db2 for the z/OS region

Usage

zowe profiles create db2-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new db2 profile. You can load this profile by using the name on commands that support the "--db2-profile" option.

Options

- `--hostname` | `-H` *(string)*

    - The Db2 server host name

- `--port` | `-P` *(number)*
    - The Db2 server port number

- `--username` | `-u` *(string)*
    - The Db2 user ID (may be the same as the TSO login)

- `--password` | `-p` *(string)*
    - The Db2 password (may be the same as the TSO password)

- `--database` | `-d` *(string)*
    - The name of the database

- `--ssl-file` | `-s` *(string)*
    - Path to an SSL Certificate file

- `--overwrite` | `--ow` *(boolean)*
    - Overwrite the db2 profile when a profile of the same name exists.

update | upd

Update a profile.You can update any property present within the profile configuration. The updated profile will be printed so that you can review the result of the updates.

zosmf-profile

z/OSMF Profile

Usage

zowe profiles update zosmf-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new zosmf profile. You can load this profile by using the name on commands that support the "--zosmf-profile" option.

Options

- `--host` | `-H` *(string)*
    - The z/OSMF server host name.
- `--port` | `-P` *(number)*
    - The z/OSMF server port.
- `--user` | `-u` *(string)*
    - Mainframe (z/OSMF) user name, which can be the same as your TSO login.
- `--password` | `-p` *(string)*
    - Mainframe (z/OSMF) password, which can be the same as your TSO password.
- `--reject-unauthorized` | `--ru` *(boolean)*
    - Reject self-signed certificates.

tso-profile

z/OS TSO/E User Profile

Usage

zowe profiles update tso-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new tso profile. You can load this profile by using the name on commands that support the "--tso-profile" option.

Options

- `--account` | `-a` *(string)*
    - Your z/OS TSO/E accounting information.
- `--character-set` | `--cs` *(string)*
    - Character set for address space to convert messages and responses from UTF-8 to EBCDIC.
- `--code-page` | `--cp` *(string)*
    - Codepage value for TSO/E address space to convert messages and responses from UTF-8

to EBCDIC.

- `--columns` | `--cols` *(number)*
  - The number of columns on a screen.
- `--logon-procedure` | `-l` *(string)*
  - The logon procedure to use when creating TSO procedures on your behalf.
- `--region-size` | `--rs` *(number)*
  - Region size for the TSO/E address space.
- `--rows` *(number)*
  - The number of rows on a screen.

cics-profile

A cics profile is required to issue commands in the cics command group that interact with CICS regions. The cics profile contains your host, port, user name, and password for the IBM CICS management client interface (CMCI) server of your choice.

Usage

zowe profiles update cics-profile [options]

Positional Arguments

- `profileName` *(string)*
  - Specifies the name of the new cics profile. You can load this profile by using the name on commands that support the "--cics-profile" option.

Options

- `--host` | `-H` *(string)*
  - The CMCI server host name
- `--port` | `-P` *(number)*
  - The CMCI server port
- `--user` | `-u` *(string)*
  - Your username to connect to CICS

- `--password` | `-p` *(string)*
    - Your password to connect to CICS
- `--region-name` *(string)*
    - The name of the CICS region name to interact with
- `--cics-plex` *(string)*
    - The name of the CICSPlex to interact with

endevor-profile

The CA Endevor SCM session profile schema, where you specify your session information and credentials

Usage

zowe profiles update endevor-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new endevor profile. You can load this profile by using the name on commands that support the "--endevor-profile" option.

Options

- `--hostname` | `--host` *(string)*
    - The hostname of the session
- `--port` | `-p` *(number)*
    - The port number of the session
- `--username` | `--user` *(string)*
    - The username of the session
- `--password` | `--pass` *(string)*
    - The password of the user
- `--protocol` | `--prot` *(string)*
    - The protocol used for connecting to CA Endevor SCM Rest API

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
  - If set, the server certificate is verified against the list of supplied CAs

endevor-location-profile

The CA Endevor SCM element location, where you specify your working environment, system and subsystem

Usage

zowe profiles update endevor-location-profile [options]

Positional Arguments

- `profileName` *(string)*
  - Specifies the name of the new endevor-location profile. You can load this profile by using the name on commands that support the "--endevor-location-profile" option.

Options

- `--instance` | `-i` *(string)*
  - The STC/datasource of the session
- `--environment` | `--env` *(string)*
  - The CA Endevor SCM environment where your project resides
- `--system` | `--sys` *(string)*
  - The CA Endevor SCM system where the element resides
- `--subsystem` | `--sub` *(string)*
  - The CA Endevor SCM subsystem where your element resides
- `--type` | `--typ` *(string)*
  - Name of the CA Endevor SCM element's type
- `--stage-number` | `--sn` *(string)*
  - The CA Endevor SCM stage where your project resides

Allowed values: 1, 2

- `--comment` | `--com` *(string)*
  - The CA Endevor SCM comment you want to use when performing an action
- `--ccid` | `--cci` *(string)*
  - The CA Endevor SCM CCID you want to use when performing an action
- `--maxrc` *(number)*
  - The return code of CA Endevor SCM that defines a failed action

fmp-profile

CA File Master Plus profile schema.

Usage

zowe profiles update fmp-profile [options]

Positional Arguments

- `profileName` *(string)*
  - Specifies the name of the new fmp profile. You can load this profile by using the name on commands that support the "--fmp-profile" option.

Options

- `--host` | `-H` *(string)*
  - Specifies CA File Master Plus server host name.
- `--port` | `-P` *(string)*
  - Specifies CA File Master Plus server port.
- `--user` | `-u` *(string)*
  - Specifies Mainframe user name. May be the same as TSO login.
- `--pass` | `-p` *(string)*
  - Specifies Mainframe password. May be the same as TSO password.
- `--protocol` | `-o` *(string)*

    - Specifies CA File Master Plus REST API protocol.

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - Reject self-signed certificates.

ops-profile

The OPS Web Services session profile schema, where you specify your session information and credentials

Usage

zowe profiles update ops-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new ops profile. You can load this profile by using the name on commands that support the "--ops-profile" option.

Options

- `--hostname` | `--host` *(string)*
    - The hostname of the session
- `--port` | `-p` *(number)*
    - The port number of the session
- `--username` | `--user` *(string)*
    - The username of the session
- `--password` | `--pass` *(string)*
    - The password of the user
- `--protocol` | `--prot` *(string)*
    - The protocol used for connecting to OPS Web Services

Allowed values: http, https

- `--reject-unauthorized` | `--ru` *(boolean)*
    - If set, the server certificate is verified against the list of supplied CAs

db2-profile

A profile for interaction with Db2 for the z/OS region

Usage

zowe profiles update db2-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the new db2 profile. You can load this profile by using the name on commands that support the "--db2-profile" option.

Options

- `--hostname` | `-H` *(string)*
    - The Db2 server host name
- `--port` | `-P` *(number)*
    - The Db2 server port number
- `--username` | `-u` *(string)*
    - The Db2 user ID (may be the same as the TSO login)
- `--password` | `-p` *(string)*
    - The Db2 password (may be the same as the TSO password)
- `--database` | `-d` *(string)*
    - The name of the database
- `--ssl-file` | `-s` *(string)*
    - Path to an SSL Certificate file

delete | rm

Delete existing profiles.

zosmf-profile

Delete a zosmf profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete zosmf-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the zosmf profile to be deleted. You can also load this profile by using the name on commands that support the "--zosmf-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

tso-profile

Delete a tso profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete tso-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the tso profile to be deleted. You can also load this profile by using the name on commands that support the "--tso-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

cics-profile

Delete a cics profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete cics-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the cics profile to be deleted. You can also load this profile by using the name on commands that support the "--cics-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

endevor-profile

Delete a endevor profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete endevor-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the endevor profile to be deleted. You can also load this profile by using the name on commands that support the "--endevor-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

endevor-location-profile

Delete a endevor-location profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete endevor-location-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the endevor-location profile to be deleted. You can also load this profile by using the name on commands that support the "--endevor-location-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

fmp-profile

Delete a fmp profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete fmp-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the fmp profile to be deleted. You can also load this profile by using the name on commands that support the "--fmp-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be

displayed before deletion occurs.

ops-profile

Delete a ops profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete ops-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the ops profile to be deleted. You can also load this profile by using the name on commands that support the "--ops-profile" option.

Options

- `--force` *(boolean)*
    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

db2-profile

Delete a db2 profile. You must specify a profile name to be deleted. To find a list of available profiles for deletion, issue the profiles list command. By default, you will be prompted to confirm the profile removal.

Usage

zowe profiles delete db2-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the db2 profile to be deleted. You can also load this profile by using the name on commands that support the "--db2-profile" option.

Options

- `--force` *(boolean)*

    - Force deletion of profile, and dependent profiles if specified. No prompt will be displayed before deletion occurs.

list | ls

List profiles of the type

zosmf-profiles

z/OSMF Profile

Usage

zowe profiles list zosmf-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
    - List zosmf profiles and their contents. All profile details will be printed as part of command output.

tso-profiles

z/OS TSO/E User Profile

Usage

zowe profiles list tso-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
    - List tso profiles and their contents. All profile details will be printed as part of command output.

cics-profiles

A cics profile is required to issue commands in the cics command group that interact with CICS regions. The cics profile contains your host, port, user name, and password for the IBM CICS management client interface (CMCI) server of your choice.

Usage

zowe profiles list cics-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
    - List cics profiles and their contents. All profile details will be printed as part of command output.

endevor-profiles

The CA Endevor SCM session profile schema, where you specify your session information and credentials

Usage

zowe profiles list endevor-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
    - List endevor profiles and their contents. All profile details will be printed as part of command output.

endevor-location-profiles

The CA Endevor SCM element location, where you specify your working environment, system and subsystem

Usage

zowe profiles list endevor-location-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
    - List endevor-location profiles and their contents. All profile details will be printed as part of command output.

fmp-profiles

CA File Master Plus profile schema.

Usage

zowe profiles list fmp-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
  - List fmp profiles and their contents. All profile details will be printed as part of command output.

ops-profiles

The OPS Web Services session profile schema, where you specify your session information and credentials

Usage

zowe profiles list ops-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
  - List ops profiles and their contents. All profile details will be printed as part of command output.

db2-profiles

A profile for interaction with Db2 for the z/OS region

Usage

zowe profiles list db2-profiles [options]

Options

- `--show-contents` | `--sc` *(boolean)*
  - List db2 profiles and their contents. All profile details will be printed as part of command output.

set-default | set

Set which profiles are loaded by default.

zosmf-profile

The zosmf set default-profiles command allows you to set the default profiles for this command group. When a zosmf command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands

profile requirements.

Usage

zowe profiles set-default zosmf-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specify a profile for default usage within the zosmf group. When you issue commands within the zosmf group without a profile specified as part of the command, the default will be loaded instead.

tso-profile

The tso set default-profiles command allows you to set the default profiles for this command group. When a tso command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile requirements.

Usage

zowe profiles set-default tso-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specify a profile for default usage within the tso group. When you issue commands within the tso group without a profile specified as part of the command, the default will be loaded instead.

cics-profile

The cics set default-profiles command allows you to set the default profiles for this command group. When a cics command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile requirements.

Usage

zowe profiles set-default cics-profile [options]

Positional Arguments

- `profileName` *(string)*

    - Specify a profile for default usage within the cics group. When you issue commands within the cics group without a profile specified as part of the command, the default will be loaded instead.

endevor-profile

The endevor set default-profiles command allows you to set the default profiles for this command group. When a endevor command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile requirements.

Usage

zowe profiles set-default endevor-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specify a profile for default usage within the endevor group. When you issue commands within the endevor group without a profile specified as part of the command, the default will be loaded instead.

endevor-location-profile

The endevor-location set default-profiles command allows you to set the default profiles for this command group. When a endevor-location command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile requirements.

Usage

zowe profiles set-default endevor-location-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specify a profile for default usage within the endevor-location group. When you issue commands within the endevor-location group without a profile specified as part of the command, the default will be loaded instead.

fmp-profile

The fmp set default-profiles command allows you to set the default profiles for this command group. When a fmp command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile

requirements.

Usage

zowe profiles set-default fmp-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specify a profile for default usage within the fmp group. When you issue commands within the fmp group without a profile specified as part of the command, the default will be loaded instead.

ops-profile

The ops set default-profiles command allows you to set the default profiles for this command group. When a ops command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile requirements.

Usage

zowe profiles set-default ops-profile [options]

Positional Arguments

- `profileName` *(string)*
    - Specify a profile for default usage within the ops group. When you issue commands within the ops group without a profile specified as part of the command, the default will be loaded instead.

db2-profile

The db2 set default-profiles command allows you to set the default profiles for this command group. When a db2 command is issued and no profile override options are specified, the default profiles for the command group are automatically loaded for the command based on the commands profile requirements.

Usage

zowe profiles set-default db2-profile [options]

Positional Arguments

- `profileName` *(string)*

    - Specify a profile for default usage within the db2 group. When you issue commands within the db2 group without a profile specified as part of the command, the default will be loaded instead.

validate | val

Test the validity of your profiles.

endevor-profile

Test the validity of a endevor profile.

Usage

zowe profiles validate endevor-profile [profileName] [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the endevor profile to be deleted. You can also load this profile by using the name on commands that support the "--endevor-profile" option.

Options

- `--print-plan-only` | `--plan` | `-p` *(boolean)*
    - Instead of validating your profile, print out a table of the tasks used for validation. This will explain the different services and functionality that will be tested during profile validation.

Profile Options

- `--endevor-profile` | `--endevor-p` *(string)*
    - The name of a (endevor) profile to load for this command execution.

fmp-profile

Test the validity of a fmp profile.

Usage

zowe profiles validate fmp-profile [profileName] [options]

Positional Arguments

- `profileName` *(string)*
    - Specifies the name of the fmp profile to be deleted. You can also load this profile by using the name on commands that support the "--fmp-profile" option.

Options

- `--print-plan-only` | `--plan` | `-p` *(boolean)*
    - Instead of validating your profile, print out a table of the tasks used for validation. This will explain the different services and functionality that will be tested during profile validation.

Profile Options

- `--fmp-profile` | `--fmp-p` *(string)*
    - The name of a (fmp) profile to load for this command execution.

provisioning | pv

Perform z/OSMF provisioning tasks on Published Templates in the Service Catalog and Provisioned Instances in the Service Registry.

list | ls

Lists z/OSMF provisioning information such as the provisioned instances from the registry, the provisioned instance details, the available provisioning templates and provisioning template details.

template-info

List details about a template published with z/OSMF Cloud Provisioning.

Usage

zowe provisioning list template-info [options]

Positional Arguments

- `name` *(string)*
    - The name of a z/OSMF cloud provisioning template.

Options

- `--all-info` | `--ai` *(boolean)*
    - Display detailed information about published z/OSMF service catalog template (summary information is printed by default).

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- List summary information for template "template1":

- `$ zowe provisioning list template-info template1`

catalog-templates

Lists the z/OSMF service catalog published templates.

Usage

zowe provisioning list catalog-templates [options]

Options

- `--all-info` | `--ai` *(boolean)*
    - Display information about published z/OSMF service catalog templates (summary information is printed by default).

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- List all published templates in the z/OSMF service catalog (with full detail):

- `$ zowe provisioning list catalog-templates --all-info`

instance-info

List details about an instance provisioned with z/OSMF.

Usage

zowe provisioning list instance-info [options]

Positional Arguments

- `name` *(string)*
  - Provisioned Instance Name

Options

- `--display` *(string)*
  - Level of information to display for the provisioned instance. Possible values:

summary - summary information, no actions or variables actions - (default) summary with actions, no variables vars - summary information with variables, no actions extended - extended information with actions full - all available information

Allowed values: extended, summary, vars, actions, full

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- List summary information with a list of actions for an instance with the name "instance1":

- `$ zowe provisioning list instance-info instance1`

  *- Show extended general information with actions for a provisioned instance with the name "instance1":

- `$ zowe provisioning list instance-info instance1 --display extended`

instance-variables

List a set of variables and their values for a given name.

Usage

zowe provisioning list instance-variables [options]

Positional Arguments

- `name` *(string)*
  - Provisioned Instance Name

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List instance variables of "instance1":

- `$ zowe provisioning list instance-variables instance1`

registry-instances

List the provisioned instances from the z/OSMF software registry.

Usage

zowe provisioning list registry-instances [options]

Options

- `--all-info` | `--ai` *(boolean)*
  - Display all available information about provisioned instances (summary by default).
- `--filter-by-type` | `--fbt` *(string)*
  - Filter the list of provisioned instances by type (e.g. DB2 or CICS).
- `--filter-by-external-name` | `--fben` *(string)*
  - Filter the list of provisioned instances by External Name.
- `--types` | `-t` *(boolean)*
  - Display a list of all types for provisioned instances (e.g. DB2 or CICS).

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- List all provisioned instances (with full detail):

- `$ zowe provisioning list registry-instances --all-info`

provision | prov

Using z/OSMF cloud provisioning services provision available templates.

template

Using z/OSMF cloud provisioning services, provision available templates. You can view available templates using the zowe provisioning list catalog-templates command.

Usage

zowe provisioning provision template [options]

Positional Arguments

- `name` *(string)*
  - The name of a z/OSMF cloud provisioning template.

Options

- `--properties` | `-p` *(string)*
  - A sequence of string enclosed "name=value" pairs of prompt variables. e.g: "CSQ_MQ_SSID=ZCT1,CSQ_CMD_PFX=!ZCT1".
- `--properties-file` | `--pf` *(string)*
  - Path to .yml file containing properties.
- `--domain-name` | `--dn` *(string)*
  - Required if the user has consumer authorization to more than one domain with this template name.
- `--tenant-name` | `--tn` *(string)*
  - Required if the user has consumer authorization to more than one tenant in the same domain that contains this template name.
- `--user-data-id` | `--udi` *(string)*
  - ID for the user data specified with user-data. Passed into the software services registry.
- `--user-data` | `--ud` *(string)*
  - User data that is passed into the software services registry. Can be specified only if user-data-id is provided.
- `--account-info` | `--ai` *(string)*
  - Account information to use in the JCL JOB statement. The default is the account information that is associated with the resource pool for the tenant.
- `--system-nick-names` | `--snn` *(string)*
  - Each string is the nickname of the system upon which to provision the software service

defined by the template. The field is required if the resource pool associated with the tenant used for this operation is not set up to automatically select a system. Only one nickname is allowed.If the field is provided it is validated. e.g: "SYSNAME1,SYSNAME2".

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Provision a published software service template.:

- `$ zowe provisioning provision template template1`

perform | perf

Perform actions against instances provisioned with z/OSMF.

action

Perform actions on instances previously provisioned with z/OSMF cloud provisioning services. To view the list of provisioned instances, use the "zowe provisioning list registry-instances" command. Once you have obtained an instance name you can use the "zowe provisioning list instance-info " command to view the available instance actions.

Usage

zowe provisioning perform action [options]

Positional Arguments

- `name` *(string)*
  - Provisioned Instance name.
- `actionname` *(string)*
  - The action name. Use the "zowe provisioning list instance-info " command to view available instance actions.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Perform the "start" action on the provisioned instance "instance1":

- `$ zowe provisioning perform action instance1 start`

delete | del

Deletes instance previously provisioned with z/OSMF cloud provisioning services.

instance

Deletes selected deprovisioned instance.

Usage

zowe provisioning delete instance [options]

Positional Arguments

- `name` *(string)*
    - Deprovisioned Instance name.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Delete deprovisioned instance "instance1":

- `$ zowe provisioning delete instance instance1`

zos-console | console

Interact with z/OSMF console services. Issue z/OS console commands and collect responses. z/OS console services establishes extended MCS (EMCS) consoles on behalf of the user, which are used to issue the commands and collect responses.

Important! Before you use commands in the zos-console command group, ensure that you understand the implications of issuing z/OS console commands in your environment.

collect

z/OSMF console services provides a command response key upon successful issue of a console command. You can use this key to collect additional console message responses.

sync-responses

The z/OSMF console REST APIs return a "solicited response key" after successfully issuing a synchronous console command that produces solicited responses. You can use the "solicited response key"on the "sync-responses" command to collect any additional outstanding solicited responses from the console the command was issued.

In general, when issuing a z/OS console command, z/OS applications route responses to the originating console. The command response messages are referred to as "solicited command responses" (i.e. direct responses to the command issued). When issuing a z/OS console command using Zowe CLI, collection of all solicited command responses is attempted by default. However, there is no z/OS mechanism that indicates the total number of response messages that may be produced from a given command. Therefore, the Zowe CLI console APIs return a "solicited response key" that can be used to "follow-up" and collect any additional solicited command responses.

Usage

zowe zos-console collect sync-responses [options]

Positional Arguments

- `responsekey` *(string)*
    - The "solicited response key" provided in response to a previously issued console command. Used by the z/OSMF console API to collect any additional outstanding solicited responses from a previously issued console command. Must match regular expression: `^[a-zA-Z0-9]+$`

Options

- `--console-name` | `--cn` | `-c` *(string)*
    - The name of the z/OS extended MCS console to direct the command. You must have the required authority to access the console specified. You may also specify an arbitrary name, if your installation allows dynamic creation of consoles with arbitrary names.

Allowed values: ^[a-zA-Z0-9]+$

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Collect any outstanding additional solicited response messages:

- `$ zowe zos-console collect sync-responses C4866969`

issue

Issue z/OS console commands and optionally collect responses.

command

Issue a z/OS console command and print command responses (known as "solicited command responses").

In general, when issuing a z/OS console command, z/OS applications route responses to the originating console. The command response messages are referred to as "solicited command responses" (i.e. direct responses to the command issued). When issuing a z/OS console command using Zowe CLI, collection of all solicited command responses is attempted by default. However, there is no z/OS mechanism that indicates the total number of response messages that may be produced from a given command. Therefore, the Zowe CLI console APIs return a "solicited response key" that can be used to "follow-up" and collect any additional solicited command responses.

Zowe CLI will issue "follow-up" API requests by default to collect any additional outstanding solicited command responses until a request returns no additional responses. At that time, Zowe CLI will attempt a final collection attempt. If no messages are present, the command is complete. If additional messages are present, the process is repeated. However, this does not guarantee that all messages produced in direct response (i.e. solicited) have been collected. The z/OS application may produce additional messages in direct response to your command at some point in the future. You can manually collect additional responses using the "command response key" OR specify additional processing options to, for example, delay collection attempts by a specified interval.

Usage

zowe zos-console issue command [options]

Positional Arguments

- `commandtext` *(string)*
    - The z/OS console command to issue

Options

- `--console-name` | `--cn` | `-c` *(string)*
    - The name of the z/OS extended MCS console to direct the command. You must have the required authority to access the console specified. You may also specify an arbitrary name, if your installation allows dynamic creation of consoles with arbitrary names.

Allowed values: ^[a-zA-Z0-9]+$

- `--include-details` | `--id` | `-i` *(boolean)*
    - Include additional details at the end of the Zowe CLI command response, such as the "command response key" and the z/OSMF command response URL.
- `--key-only` | `--ko` | `-k` *(boolean)*
    - Displays only the "command response key" returned from the z/OSMF console API. You can collect additional messages using the command key with 'zowe zos-console collect sync-responses '. Note that when using this option, you will not be presented with the "first set" of command response messages (if present in the API response). However, you can view them by using the --response-format-json option.
- `--return-first` | `--rf` | `-r` *(boolean)*
    - Indicates that Zowe CLI should return immediately with the response message set returned in the first z/OSMF API request (even if no responses are present). Using this option may result in partial or no response, but quicker Zowe CLI command response time. The z/OSMF console API has an implicit wait when collecting the first set of console command responses, i.e you will normally receive at least one set of response messages.
- `--solicited-keyword` | `--sk` | `-s` *(string)*
    - For solicited responses (direct command responses) the response is considered complete if the keyword specified is present. If the keyword is detected, the command will immediately return, meaning the full command response may not be provided. The key only applies to the first request issued, follow up requests do not support searching for the keyword.
- `--sysplex-system` | `--ss` | `--sys` *(string)*
    - Specifies the z/OS system (LPAR) in the current SYSPLEX (where your target z/OSMF resides) to route the z/OS console command.
- `--wait-to-collect` | `--wtc` | `-w` *(number)*

    - Indicates that Zowe CLI wait at least the specified number of seconds before attempting to collect additional solicited response messages. If additional messages are collected on "follow-up" requests, the timer is reset until an attempt is made that results in no additional response messages.

- `--follow-up-attempts` | `--fua` | `-a` *(number)*

    - Number of request attempts if no response returned

Default value: 1

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*

    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Issue a z/OS console command to display the IPL information for the system:

- `$ zowe zos-console issue command "D IPLINFO"`

  *- Issue a z/OS console command to display the local and coordinated universal time and date:

- `$ zowe zos-console issue command "D T"`

zos-files | files

Manage z/OS data sets, create data sets, and more

create | cre

Create data sets

data-set-sequential

Create physical sequential data sets (PS)

Usage

zowe zos-files create data-set-sequential [options]

Positional Arguments

- `dataSetName` *(string)*
    - The name of the data set that you want to create

Options

- `--block-size` | `--bs` *(number)*
    - The block size for the data set (for example, 6160)

Default value: 6160

- `--data-class` | `--dc` *(string)*
    - The SMS data class to use for the allocation
- `--device-type` | `--dt` *(string)*
    - The device type, also known as 'unit'
- `--directory-blocks` | `--db` *(number)*
    - The number of directory blocks (for example, 25)
- `--management-class` | `--mc` *(string)*
    - The SMS management class to use for the allocation
- `--record-format` | `--rf` *(string)*
    - The record format for the data set (for example, FB for "Fixed Block")

Default value: FB

- `--record-length` | `--rl` *(number)*
    - The logical record length. Analogous to the length of a line (for example, 80)

Default value: 80

- `--secondary-space` | `--ss` *(number)*
    - The secondary space allocation (for example, 1)

Default value: 1

- `--show-attributes` | `--pa` *(boolean)*
    - Show the full allocation attributes
- `--size` | `--sz` *(string)*
    - The size of the data set (specified as nCYL or nTRK - where n is the number of cylinders or tracks). Sets the primary allocation (the secondary allocation becomes ~10% of the primary).

Default value: 1CYL

- `--storage-class` | `--sc` *(string)*
    - The SMS storage class to use for the allocation
- `--volume-serial` | `--vs` *(string)*
    - The volume serial (VOLSER) on which you want the data set to be placed. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Create an empty physical sequential data set with default parameters:

- `$ zowe zos-files create data-set-sequential NEW.PS.DATASET`

data-set-partitioned

Create partitioned data sets (PDS)

Usage

zowe zos-files create data-set-partitioned [options]

Positional Arguments

- `dataSetName` *(string)*
    - The name of the data set that you want to create

Options

- `--block-size` | `--bs` *(number)*
    - The block size for the data set (for example, 6160)

Default value: 6160

- `--data-class` | `--dc` *(string)*
    - The SMS data class to use for the allocation
- `--data-set-type` | `--dst` *(string)*
    - The data set type
- `--device-type` | `--dt` *(string)*
    - The device type, also known as 'unit'
- `--directory-blocks` | `--db` *(number)*
    - The number of directory blocks (for example, 25)

Default value: 5

- `--management-class` | `--mc` *(string)*
    - The SMS management class to use for the allocation
- `--record-format` | `--rf` *(string)*
    - The record format for the data set (for example, FB for "Fixed Block")

Default value: FB

- `--record-length` | `--rl` *(number)*
    - The logical record length. Analogous to the length of a line (for example, 80)

Default value: 80

- `--secondary-space` | `--ss` *(number)*
    - The secondary space allocation (for example, 1)

Default value: 1

- `--show-attributes` | `--pa` *(boolean)*
  - Show the full allocation attributes
- `--size` | `--sz` *(string)*
  - The size of the data set (specified as nCYL or nTRK - where n is the number of cylinders or tracks). Sets the primary allocation (the secondary allocation becomes ~10% of the primary).

Default value: 1CYL

- `--storage-class` | `--sc` *(string)*
  - The SMS storage class to use for the allocation
- `--volume-serial` | `--vs` *(string)*
  - The volume serial (VOLSER) on which you want the data set to be placed. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Create an empty PDS with default parameters:

- `$ zowe zos-files create data-set-partitioned NEW.PDS.DATASET`

data-set-binary

Create executable data sets

Usage

zowe zos-files create data-set-binary [options]

Positional Arguments

- `dataSetName` *(string)*

    - The name of the data set that you want to create

Options

- `--block-size` | `--bs` *(number)*
    - The block size for the data set (for example, 6160)

Default value: 27998

- `--data-class` | `--dc` *(string)*
    - The SMS data class to use for the allocation
- `--data-set-type` | `--dst` *(string)*
    - The data set type
- `--device-type` | `--dt` *(string)*
    - The device type, also known as 'unit'
- `--directory-blocks` | `--db` *(number)*
    - The number of directory blocks (for example, 25)

Default value: 25

- `--management-class` | `--mc` *(string)*
    - The SMS management class to use for the allocation
- `--record-format` | `--rf` *(string)*
    - The record format for the data set (for example, FB for "Fixed Block")

Default value: U

- `--record-length` | `--rl` *(number)*
    - The logical record length. Analogous to the length of a line (for example, 80)

Default value: 27998

- `--secondary-space` | `--ss` *(number)*

    - The secondary space allocation (for example, 1)

Default value: 10

- `--show-attributes` | `--pa` *(boolean)*
    - Show the full allocation attributes
- `--size` | `--sz` *(string)*
    - The size of the data set (specified as nCYL or nTRK - where n is the number of cylinders or tracks). Sets the primary allocation (the secondary allocation becomes ~10% of the primary).

Default value: 10CYL

- `--storage-class` | `--sc` *(string)*
    - The SMS storage class to use for the allocation
- `--volume-serial` | `--vs` *(string)*
    - The volume serial (VOLSER) on which you want the data set to be placed. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Create an empty binary partitioned data set (PDS) with default parameters:

- `$ zowe zos-files create data-set-binary NEW.BINARY.DATASET`

data-set-c

Create data sets for C code programming

Usage

zowe zos-files create data-set-c [options]

Positional Arguments

- `dataSetName` *(string)*
    - The name of the data set that you want to create

Options

- `--block-size` | `--bs` *(number)*
    - The block size for the data set (for example, 6160)

Default value: 32760

- `--data-class` | `--dc` *(string)*
    - The SMS data class to use for the allocation
- `--data-set-type` | `--dst` *(string)*
    - The data set type
- `--device-type` | `--dt` *(string)*
    - The device type, also known as 'unit'
- `--directory-blocks` | `--db` *(number)*
    - The number of directory blocks (for example, 25)

Default value: 25

- `--management-class` | `--mc` *(string)*
    - The SMS management class to use for the allocation
- `--record-format` | `--rf` *(string)*
    - The record format for the data set (for example, FB for "Fixed Block")

Default value: VB

- `--record-length` | `--rl` *(number)*
    - The logical record length. Analogous to the length of a line (for example, 80)

Default value: 260

- `--secondary-space` | `--ss` *(number)*
    - The secondary space allocation (for example, 1)

Default value: 1

- `--show-attributes` | `--pa` *(boolean)*
    - Show the full allocation attributes
- `--size` | `--sz` *(string)*
    - The size of the data set (specified as nCYL or nTRK - where n is the number of cylinders or tracks). Sets the primary allocation (the secondary allocation becomes ~10% of the primary).

Default value: 1CYL

- `--storage-class` | `--sc` *(string)*
    - The SMS storage class to use for the allocation
- `--volume-serial` | `--vs` *(string)*
    - The volume serial (VOLSER) on which you want the data set to be placed. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Create an empty C code PDS with default parameters:

- `$ zowe zos-files create data-set-c NEW.CCODE.DATASET`

data-set-classic

Create classic data sets (JCL, HLASM, CBL, etc...)

Usage

zowe zos-files create data-set-classic [options]

Positional Arguments

- `dataSetName` *(string)*
  - The name of the data set that you want to create

Options

- `--block-size` | `--bs` *(number)*
  - The block size for the data set (for example, 6160)

Default value: 6160

- `--data-class` | `--dc` *(string)*
  - The SMS data class to use for the allocation
- `--data-set-type` | `--dst` *(string)*
  - The data set type
- `--device-type` | `--dt` *(string)*
  - The device type, also known as 'unit'
- `--directory-blocks` | `--db` *(number)*
  - The number of directory blocks (for example, 25)

Default value: 25

- `--management-class` | `--mc` *(string)*
  - The SMS management class to use for the allocation
- `--record-format` | `--rf` *(string)*
  - The record format for the data set (for example, FB for "Fixed Block")

Default value: FB

- `--record-length` | `--rl` *(number)*
  - The logical record length. Analogous to the length of a line (for example, 80)

Default value: 80

- `--secondary-space` | `--ss` *(number)*
  - The secondary space allocation (for example, 1)

Default value: 1

- `--show-attributes` | `--pa` *(boolean)*
  - Show the full allocation attributes
- `--size` | `--sz` *(string)*
  - The size of the data set (specified as nCYL or nTRK - where n is the number of cylinders or tracks). Sets the primary allocation (the secondary allocation becomes ~10% of the primary).

Default value: 1CYL

- `--storage-class` | `--sc` *(string)*
  - The SMS storage class to use for the allocation
- `--volume-serial` | `--vs` *(string)*
  - The volume serial (VOLSER) on which you want the data set to be placed. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Create an empty z/OS 'classic' PDS with default parameters:

- `$ zowe zos-files create data-set-classic NEW.CLASSIC.DATASET`

data-set-vsam

Create a VSAM cluster

Usage

zowe zos-files create data-set-vsam [options]

Positional Arguments

- `dataSetName` *(string)*
    - The name of the dataset in which to create a VSAM cluster

Options

- `--data-class` | `--dc` *(string)*
    - The SMS data class to use for the allocation
- `--data-set-organization` | `--dso` *(string)*
    - The data set organization.

Default value: INDEXED Allowed values: INDEXED, IXD, LINEAR, LIN, NONINDEXED, NIXD, NUMBERED, NUMD, ZFS

- `--management-class` | `--mc` *(string)*
    - The SMS management class to use for the allocation
- `--retain-for` | `--rf` *(number)*
    - The number of days that the VSAM cluster will be retained on the system. You can delete the cluster at any time when neither retain-for nor retain-to is specified.
- `--retain-to` | `--rt` *(string)*
    - The earliest date that a command without the PURGE parameter can delete an entry. Specify the expiration date in the form yyyyddd, where yyyy is a four-digit year (maximum value: 2155) and ddd is the three-digit day of the year from 001 through 365 (for non-leap years) or 366 (for leap years). You can delete the cluster at any time when neither retain-for nor retain-to is used. You cannot specify both the 'retain-to' and 'retain-for' options.
- `--secondary-space` | `--ss` *(number)*
    - The number of items for the secondary space allocation (for example, 840). The type of item allocated is the same as the type used for the '--size' option. If you do not specify a secondary allocation, a value of ~10% of the primary allocation is used.

- `--show-attributes` | `--pa` *(boolean)*
    - Show the full allocation attributes
- `--size` | `--sz` *(string)*
    - The primary size to allocate for the VSAM cluster. Specify size as the number of items to allocate (nItems). You specify the type of item by keyword.

Default value: 840KB

- `--storage-class` | `--sc` *(string)*
    - The SMS storage class to use for the allocation
- `--volumes` | `-v` *(string)*
    - The storage volumes on which to allocate a VSAM cluster. Specify a single volume by its volume serial (VOLSER). To specify more than one volume, enclose the option in double-quotes and separate each VOLSER with a space. You must specify the volumes option when your cluster is not SMS-managed.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Create a VSAM data set named "SOME.DATA.SET.NAME" using default values of INDEXED, 840 KB primary storage and 84 KB secondary storage:

- `$ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME`

  *- Create a 5 MB LINEAR VSAM data set named "SOME.DATA.SET.NAME" with 1 MB of secondary space. Show the properties of the data set when it is created:

- `$ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME --data-set-organization LINEAR --size 5MB --secondary-space 1MB --show-attributes`

  *- Create a VSAM data set named "SOME.DATA.SET.NAME", which is retained for 100 days:

- `$ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME --retain-for 100`

delete | del

Delete a data set

data-set

Delete a data set permanently

Usage

zowe zos-files delete data-set [options]

Positional Arguments

- `dataSetName` *(string)*
    - The name of the data set that you want to delete

Required Options

- `--for-sure` | `-f` *(boolean)*
    - Specify this option to confirm that you want to delete the data set permanently.

Options

- `--volume` | `--vol` *(string)*
    - The volume serial (VOLSER) where the data set resides. The option is required only when the data set is not catalogued on the system.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Delete the data set named 'ibmuser.cntl':

- `$ zowe zos-files delete data-set "ibmuser.cntl" -f`

data-set-vsam

Delete a VSAM cluster permanently

Usage

zowe zos-files delete data-set-vsam [options]

Positional Arguments

- `dataSetName` *(string)*
  - The name of the VSAM cluster that you want to delete

Options

- `--erase` | `-e` *(boolean)*
  - Specify this option to overwrite the data component for the cluster with binary zeros. This option is ignored if the NOERASE attribute was specified when the cluster was defined or altered.
- `--purge` | `-p` *(boolean)*
  - Specify this option to delete the VSAM cluster regardless of its retention period or date.

Required Options

- `--for-sure` | `-f` *(boolean)*
  - Specify this option to confirm that you want to delete the VSAM cluster permanently.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Delete the VSAM data set named 'ibmuser.cntl.vsam':

- `$ zowe zos-files delete data-set-vsam "ibmuser.cntl.vsam" -f`

  *- Delete all expired VSAM data sets that match 'ibmuser.AAA.**.FFF':

- `$ zowe zos-files delete data-set-vsam "ibmuser.AAA.**.FFF" -f`

  *- Delete a non-expired VSAM data set named 'ibmuser.cntl.vsam':

- `$ zowe zos-files delete data-set-vsam "ibmuser.cntl.vsam" -f --purge`

  *- Delete an expired VSAM data set named 'ibmuser.cntl.vsam' by overwriting the components with zeros:

- `$ zowe zos-files delete data-set-vsam "ibmuser.cntl.vsam" -f --erase`

invoke | call

Invoke z/OS utilities such as Access Method Services (AMS)

ams-statements

Submit control statements for execution by Access Method Services (IDCAMS). You can use IDCAMS to create VSAM data sets (CSI, ZFS, etc...), delete data sets, and more. You must format the control statements exactly as the IDCAMS utility expects. For more information about control statements, see the IBM publication 'z/OS DFSMS Access Method Services Commands'.

Usage

zowe zos-files invoke ams-statements [options]

Positional Arguments

- `controlStatements` *(string)*

  - The IDCAMS control statement that you want to submit. Zowe CLI attempts to split the inline control statement at 255 characters.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*

  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Defines a cluster named 'DUMMY.VSAM.CLUSTER':

- `$ zowe zos-files invoke ams-statements "DEFINE CLUSTER ( NAME (DUMMY.VSAM.CLUSTER) CYL(1 1))"`

  *- Deletes a cluster named 'DUMMY.VSAM.CLUSTER':

- `$ zowe zos-files invoke ams-statements "DELETE DUMMY.VSAM.CLUSTER CLUSTER"`

ams-file

Submit control statements for execution by Access Method Services (IDCAMS). You can use IDCAMS to create VSAM data sets (CSI, ZFS, etc...), delete data sets, and more. You must format the control statements exactly as the IDCAMS utility expects. For more information about control statements, see the IBM publication 'z/OS DFSMS Access Method Services Commands'.

Usage

zowe zos-files invoke ams-file [options]

Positional Arguments

- `controlStatementsFile` *(string)*
    - The path to a file that contains IDCAMS control statements. Ensure that your file does not contain statements that are longer than 255 characters (maximum allowed length).

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Reads the specified file and submits the controls statements:

- `$ zowe zos-files invoke ams-file "./path/to/file/MyControlStatements.idcams"`

download | dl

Download content from z/OS data sets and USS files to your PC

data-set

Download content from a z/OS data set to a local file

Usage

zowe zos-files download data-set [options]

Positional Arguments

- `dataSetName` *(string)*

    - The name of the data set that you want to download

Options

- `--binary` | `-b` *(boolean)*
    - Download the file content in binary mode, which means that no data conversion is performed. The data transfer process returns each line as-is, without translation. No delimiters are added between records.
- `--extension` | `-e` *(string)*
    - Save the local files with a specified file extension. For example, .txt.
- `--file` | `-f` *(string)*
    - The path to the local file where you want to download the content. When you omit the option, the command generates a file name automatically for you.
- `--volume-serial` | `--vs` *(string)*
    - The volume serial (VOLSER) where the data set resides. You can use this option at any time. However, the VOLSER is required only when the data set is not cataloged on the system. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Download the data set "ibmuser.loadlib(main)" in binary mode to the local file "main.obj":

- `$ zowe zos-files download data-set "ibmuser.loadlib(main)" -b -f main.obj`

all-members

Download all members from a partitioned data set to a local folder

Usage

zowe zos-files download all-members [options]

Positional Arguments

- `dataSetName` *(string)*

- The name of the data set from which you want to download members

Options

- `--binary` | `-b` *(boolean)*
  - Download the file content in binary mode, which means that no data conversion is performed. The data transfer process returns each line as-is, without translation. No delimiters are added between records.
- `--directory` | `-d` *(string)*
  - The directory to where you want to save the members. The command creates the directory for you when it does not already exist. By default, the command creates a folder structure based on the data set qualifiers. For example, the data set ibmuser.new.cntl's members are downloaded to ibmuser/new/cntl).
- `--extension` | `-e` *(string)*
  - Save the local files with a specified file extension. For example, .txt.
- `--max-concurrent-requests` | `--mcr` *(number)*
  - Specifies the maximum number of concurrent z/OSMF REST API requests to download members. Increasing the value results in faster downloads. However, increasing the value increases resource consumption on z/OS and can be prone to errors caused by making too many concurrent requests. If the download process encounters an error, the following message displays: The maximum number of TSO address spaces were created. When you specify 0, Zowe CLI attempts to download all members at once without a maximum number of concurrent requests.

Default value: 1

- `--volume-serial` | `--vs` *(string)*
  - The volume serial (VOLSER) where the data set resides. You can use this option at any time. However, the VOLSER is required only when the data set is not cataloged on the system. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Download the members of the data set "ibmuser.loadlib" in binary mode to the directory "loadlib/":

- `$ zowe zos-files download all-members "ibmuser.loadlib" -b -d loadlib`

  *- Download the members of the data set "ibmuser.cntl" in text mode to the directory "jcl/":

- `$ zowe zos-files download all-members "ibmuser.cntl" -d jcl`

uss-file

Download content from a USS file to a local file on your PC

Usage

zowe zos-files download uss-file [options]

Positional Arguments

- `ussFileName` *(string)*
  - The name of the USS file you want to download

Options

- `--binary` | `-b` *(boolean)*
  - Download the file content in binary mode, which means that no data conversion is performed. The data transfer process returns each line as-is, without translation. No delimiters are added between records.
- `--file` | `-f` *(string)*
  - The path to the local file where you want to download the content. When you omit the option, the command generates a file name automatically for you.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Download the file "/a/ibmuser/my_text.txt" to ./my_text.txt:

- `$ zowe zos-files download uss-file "/a/ibmuser/my_text.txt" -f ./my_text.txt`

  *- Download the file "/a/ibmuser/MyJava.class" to "java/MyJava.class" in binary mode:

- `$ zowe zos-files download uss-file "/a/ibmuser/MyJava.class" -b -f "java/MyJava.class"`

list | ls

List data sets and data set members. Optionally, you can list their details and attributes.

all-members

List all members of a partitioned data set. To view additional information about each member, use the --attributes option under the Options section of this help text.

Usage

zowe zos-files list all-members [options]

Positional Arguments

- `dataSetName` *(string)*
  - The name of the data set for which you want to list the members

Options

- `--attributes` | `-a` *(boolean)*
  - Display more information about each member. Data sets with an undefined record format display information related to executable modules. Variable and fixed block data sets display information about when the members were created and modified.
- `--max-length` | `--max` *(number)*
  - The option --max-length specifies the maximum number of items to return. Skip this parameter to return all items. If you specify an incorrect value, the parameter returns up to 1000 items.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Show members of the data set "ibmuser.asm":

- `$ zowe zos-files list all-members "ibmuser.asm"`

  *- Show attributes of members of the data set "ibmuser.cntl":

- `$ zowe zos-files list all-members "ibmuser.cntl" -a`

  *- Show the first 5 members of the data set "ibmuser.cntl":

- `$ zowe zos-files list all-members "ibmuser.cntl" --max 5`

data-set

List data sets that match a pattern in the data set name

Usage

zowe zos-files list data-set [options]

Positional Arguments

- `dataSetName` *(string)*
  - The name or pattern of the data set that you want to list

Options

- `--attributes` | `-a` *(boolean)*
  - Display more information about each member. Data sets with an undefined record format display information related to executable modules. Variable and fixed block data sets display information about when the members were created and modified.
- `--max-length` | `--max` *(number)*
  - The option --max-length specifies the maximum number of items to return. Skip this parameter to return all items. If you specify an incorrect value, the parameter returns up to 1000 items.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Show the data set "ibmuser.asm":

- `$ zowe zos-files list data-set "ibmuser.asm"`

  *- Show attributes of the data set "ibmuser.cntl":

- `$ zowe zos-files list data-set "ibmuser.cntl" -a`

  *- Show all data sets of the user "ibmuser":

- `$ zowe zos-files list data-set "ibmuser.*"`

  *- Show attributes of all data sets of the user "ibmuser":

- `$ zowe zos-files list data-set "ibmuser.*" -a`

  *- Show the first 5 data sets of the user "ibmuser":

- `$ zowe zos-files list data-set "ibmuser.cntl" --max 5`

upload | ul

Upload the contents of a file to z/OS data sets

file-to-data-set

Upload the contents of a file to a z/OS data set

Usage

zowe zos-files upload file-to-data-set [options]

Positional Arguments

- `inputfile` *(string)*
    - The local file that you want to upload to a data set

- `dataSetName` *(string)*
    - The name of the data set to which you want to upload the file

Options

- `--binary` | `-b` *(boolean)*
    - Data content in binary mode, which means that no data conversion is performed. The data transfer process returns each record as-is, without translation. No delimiters are added between records.
- `--migrated-recall` | `--mr` *(string)*
    - The method by which migrated data set is handled. By default, a migrated data set is recalled synchronously. You can specify the following values: wait, nowait, error

Default value: nowait

- `--volume-serial` | `--vs` *(string)*
    - The volume serial (VOLSER) where the data set resides. You can use this option at any time. However, the VOLSER is required only when the data set is not cataloged on the system. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Upload file contents to a sequential data set named "ibmuser.ps":

- `$ zowe zos-files upload file-to-data-set "file.txt" "ibmuser.ps"`

  *- Upload file contents to a PDS member named "ibmuser.pds(mem)":

- `$ zowe zos-files upload file-to-data-set "file.txt" "ibmuser.pds(mem)"`

  *- Upload file contents to a migrated data set and wait for it to be recalled:

- `$ zowe zos-files upload file-to-data-set "file.txt" "ibmuser.ps" --mr wait`

stdin-to-data-set

Upload the content of a stdin to a z/OS data set

Usage

zowe zos-files upload stdin-to-data-set [options]

Positional Arguments

- `dataSetName` *(string)*
    - The name of the data set to which you want to upload data

Options

- `--binary` | `-b` *(boolean)*
    - Data content in binary mode, which means that no data conversion is performed. The data transfer process returns each record as-is, without translation. No delimiters are added between records.
- `--migrated-recall` | `--mr` *(string)*
    - The method by which migrated data set is handled. By default, a migrated data set is recalled synchronously. You can specify the following values: wait, nowait, error

Default value: nowait

- `--volume-serial` | `--vs` *(string)*
    - The volume serial (VOLSER) where the data set resides. You can use this option at any time. However, the VOLSER is required only when the data set is not cataloged on the system. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Stream content from stdin to a sequential data set named "ibmuser.ps" from a Windows console:

- `$ zowe zos-files upload stdin-to-data-set "ibmuser.ps" < echo "hello world"`

*- Stream content from stdin to a partition data set member named "ibmuser.pds(mem)" from a Windows console:

- `$ zowe zos-files upload stdin-to-data-set "ibmuser.pds(mem)" < echo "hello world"`

*- Stream content from stdin to a migrated data set and wait for it to be recalled from a Windows console:

- `$ zowe zos-files upload stdin-to-data-set "ibmuser.ps" --mr wait < echo "hello world"`

dir-to-pds

Upload files from a directory to a partioned data set (PDS)

Usage

zowe zos-files upload dir-to-pds [options]

Positional Arguments

- `inputdir` *(string)*
  - The path for a local directory that you want to upload to a PDS
- `dataSetName` *(string)*
  - The name of the partitioned data set to which you want to upload the files

Options

- `--binary` | `-b` *(boolean)*
  - Data content in binary mode, which means that no data conversion is performed. The data transfer process returns each record as-is, without translation. No delimiters are added between records.
- `--migrated-recall` | `--mr` *(string)*
  - The method by which migrated data set is handled. By default, a migrated data set is recalled synchronously. You can specify the following values: wait, nowait, error

Default value: nowait

- `--volume-serial` | `--vs` *(string)*

- The volume serial (VOLSER) where the data set resides. You can use this option at any time. However, the VOLSER is required only when the data set is not cataloged on the system. A VOLSER is analogous to a drive name on a PC.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Upload a directory named "src" to a PDS named "ibmuser.src":

- `$ zowe zos-files upload dir-to-pds "src" "ibmuser.src"`

  *- Upload a directory named "src" to a migrated PDS named "ibmuser.src" and wait for it to be recalled:

- `$ zowe zos-files upload dir-to-pds "src" "ibmuser.src" --mr wait`

file-to-uss

Upload content to a USS file from local file

Usage

zowe zos-files upload file-to-uss [options]

Positional Arguments

- `inputfile` *(string)*
  - The local file that you want to upload to a USS file
- `USSFileName` *(string)*
  - The name of the USS file to which you want to upload the file

Options

- `--binary` | `-b` *(boolean)*
  - Data content in binary mode, which means that no data conversion is performed. The data transfer process returns each record as-is, without translation. No delimiters are

added between records.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Upload to the USS file "/a/ibmuser/my_text.txt" from the file "file.txt":

- `$ zowe zos-files upload file-to-uss "file.txt" "/a/ibmuser/my_text.txt"`

zos-jobs | jobs

Manage z/OS jobs.

submit | sub

Submit jobs (JCL) contained in data sets.

data-set

Submit a job (JCL) contained in a data set. The data set may be of type physical sequential or a PDS member. The command does not pre-validate the data set name. The command presents errors verbatim from the z/OSMF Jobs REST endpoints. For more information about z/OSMF Jobs API errors, see the z/OSMF Jobs API REST documentation.

Usage

zowe zos-jobs submit data-set [options]

Positional Arguments

- `dataset` *(string)*
  - The z/OS data set containing the JCL to submit. You can specify a physical sequential data set (for example, "DATA.SET") or a partitioned data set qualified by a member (for example, "DATA.SET(MEMBER)").

Options

- `--volume` | `--vol` *(string)*
  - The volume serial (VOLSER) where the data set resides. The option is required only

when the data set is not catalogued on the system.

- `--view-all-spool-content` | `--vasc` *(boolean)*
  - Print all spool output. If you use this option you will wait the job to complete.
- `--directory` | `-d` *(string)*
  - The local directory you would like to download the output of the job. Creates a subdirectory using the jobID as the name and files are titled based on DD names. If you use this option you will wait the job to complete.
- `--extension` | `-e` *(string)*
  - A file extension to save the job output with. Default is '.txt'.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- Submit the JCL in the data set "ibmuser.cntl(deploy)":

- `$ zowe zos-jobs submit data-set "ibmuser.cntl(deploy)"`

  *- Submit the JCL in the data set "ibmuser.cntl(deploy)", wait for the job to complete and print all output from the job:

- `$ zowe zos-jobs submit data-set "ibmuser.cntl(deploy)" --vasc`

local-file

Submit a job (JCL) contained in a local file. The command presents errors verbatim from the z/OSMF Jobs REST endpoints. For more information about z/OSMF Jobs API errors, see the z/OSMF Jobs API REST documentation.

Usage

zowe zos-jobs submit local-file [options]

Positional Arguments

- `localFile` *(string)*
  - The local file containing the JCL to submit.

Options

- `--view-all-spool-content` | `--vasc` *(boolean)*
  - View all spool content for specified job ID
- `--directory` | `-d` *(string)*
  - The local directory you would like to download the output for the job to. Creates a subdirectory using the jobID as the name and files are titled based on DD names.
- `--extension` | `-e` *(string)*

    - A file extension to save the job output with

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
    - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
    - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
    - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- Submit the JCL in the file "iefbr14.txt":

- `$ zowe zos-jobs submit local-file "iefbr14.txt"`

download | dl

Download the output of a job as separate files.

output

Download all job output to a local directory. Each spool DD will be downloaded to its own file in the directory.

Usage

zowe zos-jobs download output [options]

Positional Arguments

- `jobid` *(string)*
    - The z/OS JOBID of the job containing the spool files you want to view. No pre-validation of the JOBID is performed.

Options

- `--directory` | `-d` | `--dir` *(string)*
    - The local directory you would like to download the output for the job to.
- `--extension` | `-e` *(string)*
    - A file extension to save the job output with. Defaults to '.txt'.
- `--omit-jobid-directory` | `--ojd` *(boolean)*
    - If specified, job output will be saved directly to the specified directory rather than creating a subdirectory named after the ID of the job.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Download all the output of the job with job ID JOB00234 to an automatically generated directory.:

- `$ zowe zos-jobs download output JOB00234`

view | vw

View details of z/OS jobs on spool/JES queues.

job-status-by-jobid

View status details of a single z/OS job on spool/JES queues. The command does not prevalidate the JOBID. The command presents errors verbatim from the z/OSMF Jobs REST endpoints (expect for "no jobs found").

Usage

zowe zos-jobs view job-status-by-jobid [options]

Positional Arguments

- `jobid` *(string)*
    - The z/OS JOBID of the job you want to view. No prevalidation of the JOBID is performed.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
    - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
    - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of

"table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- View status and other details of the job with the job ID JOB00123:

- `$ zowe zos-jobs view job-status-by-jobid j123`

  *- Print only the status (for example, "OUTPUT" or "ACTIVE") of the job with the job ID JOB00123:

- `$ zowe zos-jobs view job-status-by-jobid j123 --rff status --rft string`

spool-file-by-id

View the contents of a spool file from a z/OS job on spool/JES queues. The command does not pre-validate the JOBID or spool ID. The command presents errors verbatim from the z/OSMF Jobs REST endpoints.

Usage

zowe zos-jobs view spool-file-by-id [options]

Positional Arguments

- `jobid` *(string)*
  - The z/OS JOBID of the job containing the spool file you want to view. No pre-validation of the JOBID is performed.
- `spoolfileid` *(number)*
  - The spool file ID number for the spool file to view. Use the "zowe zos-jobs list spool-files-by-jobid" command to obtain spool ID numbers.No pre-validation of the ID is performed.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- View the spool file with ID 4 for the job with job ID JOB00123:

- `$ zowe zos-jobs view spool-file-by-id JOB00123 4`

list | ls

List z/OS jobs and list the spool files (DDs) for a z/OS job on the JES/spool queues.

spool-files-by-jobid

Given a z/OS job JOBID, list the spool files (DDs) for a z/OS job on the JES/spool queues. The command does not pre-validate the JOBID. The command presents errors verbatim from the z/OSMF Jobs REST endpoints.

Usage

zowe zos-jobs list spool-files-by-jobid [options]

Positional Arguments

- `jobid` *(string)*
    - The z/OS JOBID of the job with the spool files you want to list. No pre-validation of the JOBID is performed.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
    - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.

- `--response-format-type` | `--rft` *(string)*
    - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
    - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List the spool files of the job with JOBID JOB00123:

- `$ zowe zos-jobs list spool-files-by-jobid job00123`

jobs

List jobs on JES spool/queues. By default, the command lists jobs owned (owner) by the user specified in your z/OSMF profile. The default for prefix is "*". The command does not prevalidate your user ID. The command surfaces errors verbatim from the z/OSMF Jobs REST endpoints.

Usage

zowe zos-jobs list jobs [options]

Options

- `--owner` | `-o` *(string)*
    - Specify the owner of the jobs you want to list. The owner is the individual/user who submitted the job OR the user ID assigned to the job. The command does not prevalidate the owner. You can specify a wildcard according to the z/OSMF Jobs REST endpoint documentation, which is usually in the form "USER*".
- `--prefix` | `-p` *(string)*

- Specify the job name prefix of the jobs you want to list. The command does not prevalidate the owner. You can specify a wildcard according to the z/OSMF Jobs REST endpoint documentation, which is usually in the form "JOB*".

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

response format options

- `--response-format-filter` | `--rff` *(array)*
  - Filter (include) fields in the response. Accepts an array of field/property names to include in the output response. You can filter JSON objects properties OR table columns/fields. In addition, you can use this option in conjunction with '--response-format-type' to reduce the output of a command to a single field/property or a list of a single field/property.
- `--response-format-type` | `--rft` *(string)*
  - The command response output format type. Must be one of the following:

table: Formats output data as a table. Use this option when the output data is an array of homogeneous JSON objects. Each property of the object will become a column in the table.

list: Formats output data as a list of strings. Can be used on any data type (JSON objects/arrays) are stringified and a new line is added after each entry in an array.

object: Formats output data as a list of prettified objects (or single object). Can be used in place of "table" to change from tabular output to a list of prettified objects.

string: Formats output data as a string. JSON objects/arrays are stringified.

Allowed values: table, list, object, string

- `--response-format-header` | `--rfh` *(boolean)*
  - If "--response-format-type table" is specified, include the column headers in the output.

Examples

*- List all jobs with default settings. The command returns jobs owned by your user ID with any job name:

- `$ zowe zos-jobs list jobs`

  *- List all jobs owned by user IDs starting with 'ibmu' and job names starting with 'myjo':

- `$ zowe zos-jobs list jobs -o "ibmu*" -p "myjo*"`

  *- List all jobs with default owner and prefix settings, displaying only the job ID of each job:

- `$ zowe zos-jobs list jobs --rff jobid --rft table`

delete | del

Delete a single job by job ID or delete multiple jobs in OUTPUT status. This cancels the job if it is running and purges its output from the system

job

Delete a single job by job ID

Usage

zowe zos-jobs delete job [options]

Positional Arguments

- `jobid` *(string)*

  - The job ID (e.g. JOB00123) of the job. Job ID is a unique identifier for z/OS batch jobs -- no two jobs on one system can have the same ID. Note: z/OS allows you to abbreviate the job ID if desired. You can use, for example "J123".

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*

  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Delete job with job ID JOB03456.:

- `$ zowe zos-jobs delete job JOB03456`

zos-tso | tso

Issue TSO commands and interact with TSO address spaces

send

Send data to TSO and collect responses until the prompt is reached

address-space

Send data to the TSO address space, from which you previously started and received a token (a.k.a 'servlet-key').

Usage

zowe zos-tso send address-space [options]

Positional Arguments

- `servletKey` *(string)*
    - The servlet key from a previously started TSO address space.

Required Options

- `--data` *(string)*
    - The data to which we want to send to the TSO address space represented by the servlet key.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- "Send the TIME TSO command to the TSO address space identified by IBMUSER-329-aafkaaoc":

- `$ zowe zos-tso send address-space IBMUSER-329-aafkaaoc --data "TIME"`

start | st

Start TSO/E address space

address-space

Start a TSO address space, from which you will receive a token (a.k.a 'servelet-key') for further address space interaction (e.g. termination).

Usage

zowe zos-tso start address-space [options]

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.
- `--tso-profile` | `--tso-p` *(string)*
    - The name of a (tso) profile to load for this command execution.

Examples

*- Start TSO/E address space:

- `$ zowe zos-tso start address-space`

  *- Start TSO/E address space, and receive response in JSON format:

- `$ zowe zos-tso start address-space --rfj`

ping

Ping a TSO address space, from which you previously started and received a token (a.k.a 'servelet-key').

address-space

Ping a TSO address space, from which you previously started and received a token (a.k.a 'servlet-key').

Usage

zowe zos-tso ping address-space [options]

Positional Arguments

- `servletKey` *(string)*

    - The servlet key from a previously started TSO address space.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.
- `--tso-profile` | `--tso-p` *(string)*
    - The name of a (tso) profile to load for this command execution.

Examples

*- Ping the TSO address space identified by IBMUSER-329-aafkaaoc:

- `$ zowe zos-tso ping address-space IBMUSER-329-aafkaaoc`

stop | sp

Stop TSO/E address space

address-space

Stop a TSO address space, from which you previously started and received a token (a.k.a 'servlet-key').

Usage

zowe zos-tso stop address-space [options]

Positional Arguments

- `servletkey` *(string)*
    - The servlet key from a previously started TSO address space.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.

Examples

*- Stop the TSO address space identified by IBMUSER-329-aafkaaoc:

- `$ zowe zos-tso stop address-space IBMUSER-329-aafkaaoc`

issue

Issue TSO commands

command

Creates a TSO address space, issues a TSO command through the newly created address space, waits for the READY prompt to print the response, and terminates the TSO address space. All response data are returned to the user up to (but not including) the TSO 'READY' prompt.

Usage

zowe zos-tso issue command [options]

Positional Arguments

- `commandText` *(string)*
    - The TSO command to issue.

Options

- `--suppress-startup-messages` | `--ssm` *(boolean)*
    - Suppress console messages from start of address space.

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
    - The name of a (zosmf) profile to load for this command execution.
- `--tso-profile` | `--tso-p` *(string)*
    - The name of a (tso) profile to load for this command execution.

Examples

*- Issue the TSO command "status" to display information about jobs for your user ID.:

- `$ zowe zos-tso issue command "status"`

zosmf

Retrieve and show the properties of a z/OSMF web server

check

Confirm that z/OSMF is running on a specified system and gather information about the z/OSMF server for diagnostic purposes.

status

Confirm that z/OSMF is running on a system specified in your profile and gather information about the z/OSMF server for diagnostic purposes. The command outputs properties of the z/OSMF server such as version, hostname, and installed plug-ins.

Usage

zowe zosmf check status [options]

Profile Options

- `--zosmf-profile` | `--zosmf-p` *(string)*
  - The name of a (zosmf) profile to load for this command execution.

Examples

*- Report the status of the z/OSMF server that you specified in your default z/OSMF profile:

- `$ zowe zosmf check status`

  *- Report the status of the z/OSMF server that you specified in a supplied z/OSMF profile:

- `$ zowe zosmf check status --zosmf-profile SomeZosmfProfileName`