Hewlett Packard
Enterprise

HPE Operations Orchestration

Software Version: 10.60
Windows and Linux Operating Systems

OO Shell for Authoring (OOSHA) User Guide

Document Release Date: May 2016
Software Release Date: May 2016

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

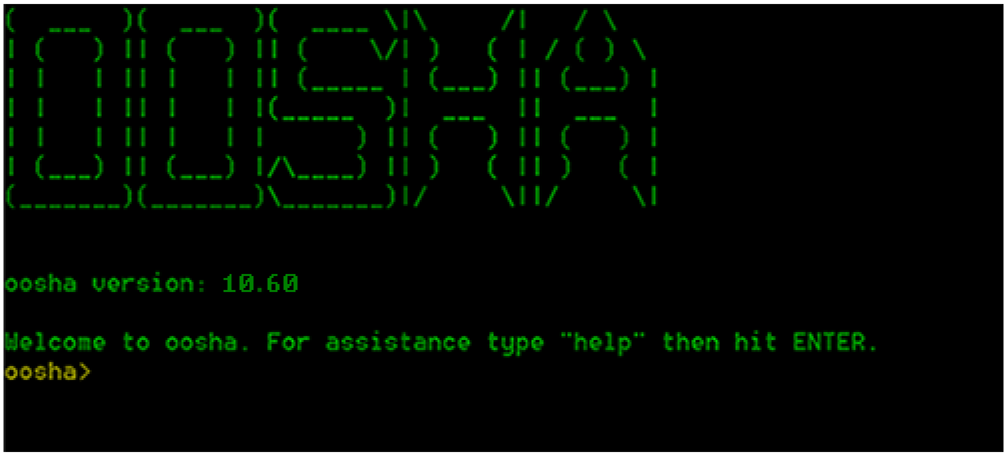
To download the most recent edition of a document, go to https://softwaresupport.hp.com.

Contents

Running HPE OO Authoring Commands from the Command Line

The Operations Orchestration Shell Authoring (OOSHA) utility lets you run HPE OO commands from the command line, instead of from the Studio 10.x interface. This enables you to create scripts to run commands, to automate some of your HPE OO authoring processes.

> **Note:** OOSHA is supported on both Linux and Windows platforms and no special permissions are required.



```
(  ___  )(  ___  )(  ____ \|\     /|   / \
| (   ) || (   ) || (    \/| )   ( | / ( ) \
| |   | || |   | || (_____ | (___) || (___) |
| |   | || |   | |(_____  )|  ___  ||  ___  |
| |   | || |   | |      ) || (   ) || (   ) |
| (___) || (___) |/\____) || )   ( || )   ( |
(_______)(_______)\_______)|/     \||/     \|


oosha version: 10.60

Welcome to oosha. For assistance type "help" then hit ENTER.
oosha>
```

You can run OOSHA in two ways:

- As a standalone package
- Within an HPE OO installation, version 10.60 and later

Install OOSHA

In an existing installation of HPE 10.60 or later:

- OOSHA is installed when Studio is selected in the OO installer, and is uninstalled when Studio is uninstalled.
- The OOSHA utility is located under **<oo_installation_folder>/studio/tools**.

To install OOSHA as a standalone utility:

1. Download the **OOSHA.zip** file.
2. Unzip the **OOSHA.zip** file to the desired location.

Run Commands in OOSHA

You can run commands within the OOSHA utility or from a normal prompt/shell.

1. Start the OOSHA shell:
   - In an installation of HPE OO, navigate to **<oo_installation_folder>/studio/tools**, and run the **oosha.bat** executable file for the Windows operating system or the **oosha.sh** executable file for the Linux operating system.

- In a standalone OOSHA utility, navigate to the **bin** folder, and run the **oosha.bat** executable file for the Windows operating system or the **oosha.sh** executable file for the Linux operating system.
- To run commands from a normal prompt/shell, type: `<OO installation path>/studio/tools oosha.bat <command>`.

2. Type the command in the following format:

   `<commandName> --<parameterName1> <parameterValue1>`

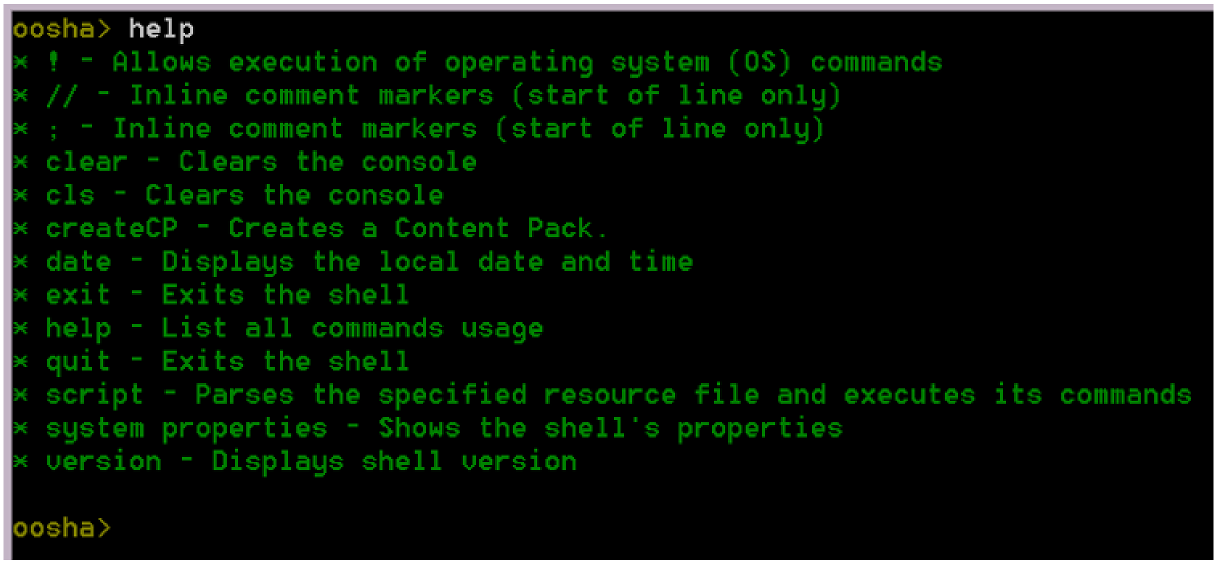
   `--<parameterName2>  <parameterValue2>`

   Some of the available commands are:

   - `createCP` - enables you to package a content pack. For details, see "Creating Content Packs via OOSHA" on page 8
   - `help` - displays a list and description of all available commands
   - `exit, quit` - exits OOSHA
   - `clr, clear` - clears the console
   - `system properties` - displays the OOSHA system properties
   - `date` - displays the current date
   - `version` - displays the current version of OOSHA

Display Help in OOSHA

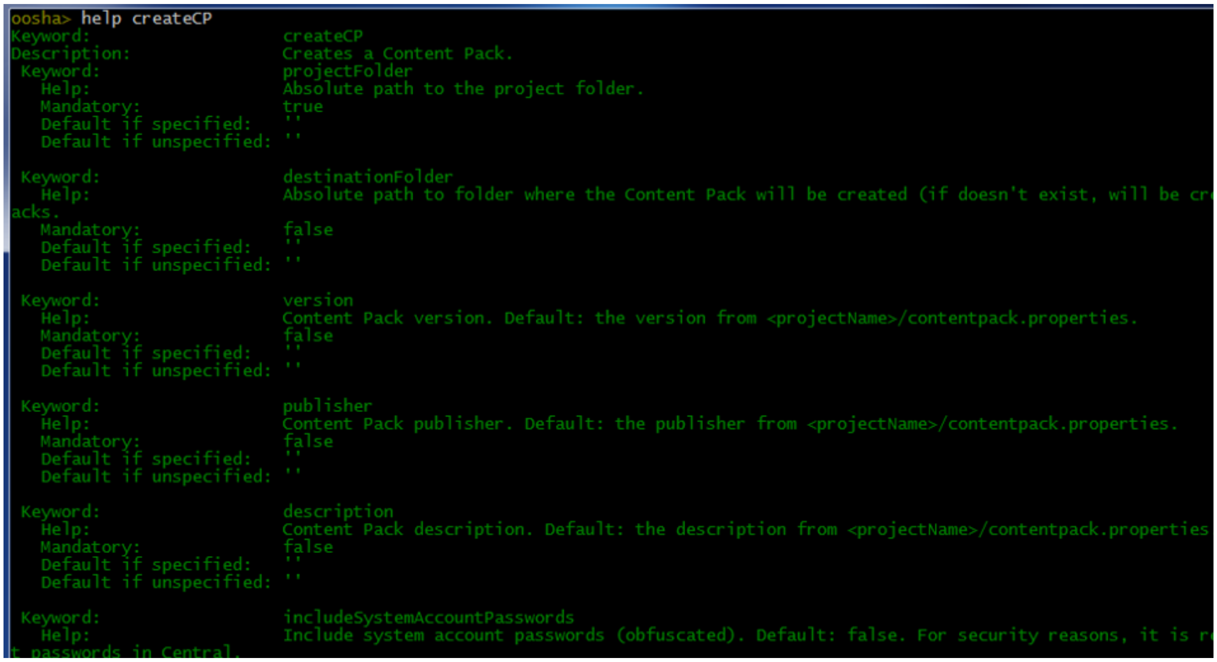
- To display a list and description of all available commands in the OOSH utility, type: `help`



```
oosha> help
* ! - Allows execution of operating system (OS) commands
* // - Inline comment markers (start of line only)
* ; - Inline comment markers (start of line only)
* clear - Clears the console
* cls - Clears the console
* createCP - Creates a Content Pack.
* date - Displays the local date and time
* exit - Exits the shell
* help - List all commands usage
* quit - Exits the shell
* script - Parses the specified resource file and executes its commands
* system properties - Shows the shell's properties
* version - Displays shell version

oosha>
```

- To display help about a particular command, type: `help <command>`

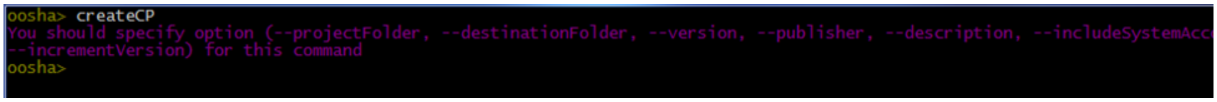
  For example: `help createCP`



```
oosha> help createCP
Keyword:                   createCP
Description:               Creates a Content Pack.
 Keyword:                  projectFolder
   Help:                   Absolute path to the project folder.
   Mandatory:              true
   Default if specified:   ''
   Default if unspecified: ''

 Keyword:                  destinationFolder
   Help:                   Absolute path to folder where the Content Pack will be created (if doesn't exist, will be cr
acks.
   Mandatory:              false
   Default if specified:   ''
   Default if unspecified: ''

 Keyword:                  version
   Help:                   Content Pack version. Default: the version from <projectName>/contentpack.properties.
   Mandatory:              false
   Default if specified:   ''
   Default if unspecified: ''

 Keyword:                  publisher
   Help:                   Content Pack publisher. Default: the publisher from <projectName>/contentpack.properties.
   Mandatory:              false
   Default if specified:   ''
   Default if unspecified: ''

 Keyword:                  description
   Help:                   Content Pack description. Default: the description from <projectName>/contentpack.properties
   Mandatory:              false
   Default if specified:   ''
   Default if unspecified: ''

 Keyword:                  includeSystemAccountPasswords
   Help:                   Include system account passwords (obfuscated). Default: false. For security reasons, it is r
t passwords in Central.
```

- To display a list of arguments for a valid command that has at least one argument, type: `<command>`

  For example: `createCP`



```
oosha> createCP
You should specify option (--projectFolder, --destinationFolder, --version, --publisher, --description, --includeSystemAcc
--incrementVersion) for this command
oosha>
```

Navigate the Command History

Use the **Up** arrow key or **Down** arrow key on your keyboard to navigate inside the command history for OOSHA.

- The **Up** arrow key history navigates one item back in the history of commands.
- The **Down** arrow key navigates one item forward in the history of commands.

View the OOSHA Log

For a standalone OOSHA installation, the OOSHA application logs are stored inside the **<OOSHA_zip_extraction_location>/logs** folder.

For OOSHA inside an OO installation, the OOSHA application logs are stored inside the **<oo_installation_folder>/studio/tools/logs** folder.

Change the OOSHA Log Location

1. Locate the **log4j.properties** file.
   - For a standalone installation: **<OOSHA_zip_extraction_location>/conf/log4j.properties**.
   - For OOSHA inside an OO installation: **<oo_installation_folder>/studio/tools/conf/log4j.properties**.
2. Locate the following line inside the **log4j.properties** file:

   ```
   log4j.appender.FILE=org.apache.log4j.RollingFileAppender
   ```

3. Below that line, add the following line:

   ```
   log4j.appender.FILE.File=<absolute path of the oosha.log file>
   ```

   For example: `log4j.appender.FILE.File=C:/logs/oosha.log`

4. Save the **log4.properties** file and start the OOSHA shell.

Configure the OOSHA encryption

By default, OOSHA uses the same encryption options as a default Studio installation. To configure OOSHA to use the encryption of another Studio installation, set the value of the **oosha.encryptor.dir** property inside the **oosha.properties** file to point to the desired encryption folder.

1. Locate the **oosha.properties** file.

   - For a standalone installation: **<OOSHA_zip_extraction_location>/conf/oosha.properties**.

   - For OOSHA inside an OO installation: **<oo_installation_folder>/studio/tools/conf/oosha.properties**.

2. Edit the **oosha.properties** file by setting the **oosha.encryptor.dir** property with the value of the absolute path of the Studio installation security folder.

   For example:

   ```
   oosha.encryptor.dir=<oo_installation_folder>/studio/var/security
   ```

3. Save the **oosha.properties** file and start the OOSHA shell.

Creating Content Packs via OOSHA

Using the `createCP` command, you can create a content pack from a project developed in Studio or in CloudSlang.

While a content pack is being created via OOSHA, the project undergoes a series of validation checks, to check for invalid flows or operations (for example, those containing duplicate UUIDs), which will not be included in the content pack.

A content pack created via OOSHA undergoes the same validation checks as one exported from Studio. For more information, see "Exporting a Content Pack" in the *Studio Authoring Guide*.

Running Commands

Create a Content Pack

The `createCP` command packages a project into a content pack. The content pack will contain all the objects in the project.

> **Note:** The invalid flows will also be included.

For a complete list of all the options you can use with the `createCP` command, see the Reference section or type `help createCP` in the command line.

1. Double-click the **oosha.bat** batch file, to start the OOSHA utility.
2. In the command line, type the following command, replacing the highlighted values with your own values:

   ```
   createCP --projectFolder <projectFolder> --destinationFolder <CPDestinationFolder> -
   -version <CPversion> --publisher <publisher> --description <description> --
   includeSystemAccountPasswords <shouldIncludeSAPasswords>
   ```

   Windows example:

   ```
   oosha.bat createCP --projectFolder C:\Users\User1\.oo\Workspace\project1 --version 7
   ```

   This command creates a content pack in the default location (**<user home>/.oo/Content Packs**) based on the project at the location **C:\Users\User1\.oo\Worspace\project1** using version 7, using the publisher and description taken from project and not including the password for system accounts.

   Linux example:

   ```
   ./oosha.sh createCP --projectFolder '/home/user1/.oo/Workspace/my project' --
   destinationFolder  /tmp/cps --description 'test content pack' --publisher 'the
   publisher' --version 1.9
   ```

3. Press the **Enter** key.

   The content pack is created in the location that was defined in the `--destinationFolder` argument.

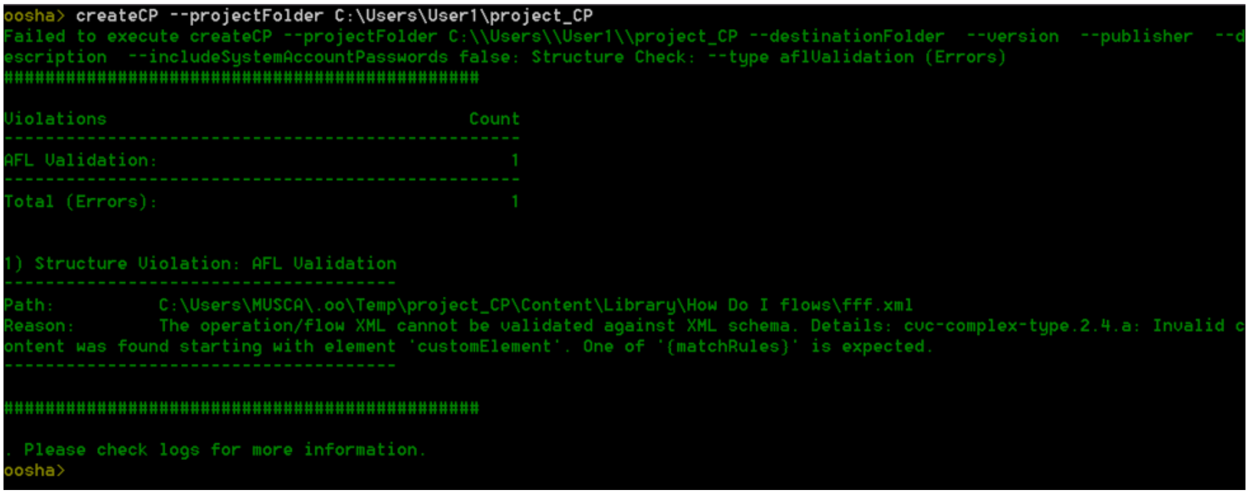
   The name of the content pack is taken from the **pom.xml** file. If not defined there, it is taken from the **contentpack.properties** file. If not defined there, it is taken from the project folder.

Validate a Content Pack

When you run the `createCP` command from OOSHA, the structure of the project is validated in the same way as in Studio.

The `createCP` command will fail if the project includes corrupt XML files or XML files that do not respect the HPE OO validation schema.

For example:



```
oosha> createCP --projectFolder C:\Users\User1\project_CP
Failed to execute createCP --projectFolder C:\\Users\\User1\\project_CP --destinationFolder  --version  --publisher  --d
escription  --includeSystemAccountPasswords false: Structure Check: --type aflValidation (Errors)
###############################################

Violations                                    Count
---------------------------------------------------
AFL Validation:                                   1
---------------------------------------------------
Total (Errors):                                   1


1) Structure Violation: AFL Validation
--------------------------------------
Path:          C:\Users\MUSCA\.oo\Temp\project_CP\Content\Library\How Do I flows\fff.xml
Reason:        The operation/flow XML cannot be validated against XML schema. Details: cvc-complex-type.2.4.a: Invalid c
ontent was found starting with element 'customElement'. One of '{matchRules}' is expected.
--------------------------------------

###############################################

. Please check logs for more information.
oosha>
```

Note: In this version, OOSHA validates the following:

- Start steps
- Connected transitions
- Lanes
- Operation responses
- Folder and file structure
- No missing references to other flows, operations or configuration items inside this project, as well as other projects and content packs

OOSHA does not exclude invalid items (flows, operations, configuration items) from the resulting content pack.

Include Hard Copies Inside the Content Pack

In order for OOSHA to pack operation dependencies (operations' plugins and their dependencies) for hard copy operations, you must configure the **oosha.maven.home** property inside the **oosha.properties** file by following the steps below.

1. Locate the **oosha.properties** file.
   - For a standalone installation: **<OOSHA_zip_extraction_location>/conf/oosha.properties**.
   - For OOSHA inside an OO installation: **<oo_installation_folder>/studio/tools/conf/oosha.properties**.
2. Edit the **oosha.properties** file by setting the **oosha.maven.home** property with the value of the absolute path of the repository containing the operation dependencies.

The value of this property is typically **<user_home>/.oo/data/maven**. This is the absolute path of the Studio maven repository folder that was used to create the project, **<studio_workspace>/.oo/data/maven**.

If this property is not specified, the default maven home is **<user_home>/.m2/repository**.

3. Save the **oosha.properties** file and start the OOSHA shell.

Configure System Accounts

For a standalone OOSHA installation, in some cases, you may want to a package a content pack from a project that contains system accounts without including the system account passwords.

For example, when you want to run the `createCP` command using a false value for the `--includeSystemAccountPasswords` argument or when the value for the `--includeSystemAccountPasswords` argument is not provided.

In such a case, you must first configure the **oosha.encryptor.dir** property inside the **oosha.properties** file by following the steps below.

1. Locate the **oosha.properties** file.
   - For a standalone installation: **<OOSHA_zip_extraction_location>/conf/oosha.properties**.
   - For OOSHA inside an OO installation: **<oo_installation_folder>/studio/tools/conf/oosha.properties**.
2. Edit the **oosha.properties** file by setting the **oosha.encryptor.dir** property with the value of the absolute path of the Studio installation security folder **<oo_installation_folder>/studio/var/security**.
3. Save the **oosha.properties** file and start the OOSHA shell.

Reference Material

Available Arguments for the createCP Command

Argument	Description	Default Value/Behavior	Required
`--projectFolder`	The location of the project to be packaged. This must be the absolute path of the folder that represents a project in the Studio format.		Yes
`--destinationFolder`	The path to a local folder where the project will be packaged into a content pack. The destination folder and all parent folders up to the destination folder will be created, if they do not exist	• The default location is where the content pack was created the previous time. • The first time the content pack is created, the default path is: C:\Users\<user_	No

Argument	Description	Default Value/Behavior	Required
	on the file system.	name>\.oo\Content Packs	
`--version`	Lets you specify the version of the content pack.	1.0.0 If the version is not specified, the default version number is taken from the project, with the "-SNAPSHOT" suffix removed. For example, if the project version is **1.7.0-SNAPSHOT**, the content pack version will be **1.7.0**.	No
`--incrementVersion`	The content pack version is taken from the project version with the "-SNAPSHOT" suffix removed, and the project version is increased by 0.0.1. For example, assuming that you have not specified a version number with `--version`: If the project version is **1.7.0-SNAPSHOT**, the content pack version will be **1.7.0** and the project version will be moved up to **1.7.1-SNAPSHOT**.	False The default is that the project version is not incremented.	No
`--publisher`	Lets you specify the publisher of the content pack.	"" (empty)	No
`--description`	Lets you specify a description of the content pack.	If this is not specified, the description is taken from the **contentpack.properties** file.	No
`--includeSystemAccountPasswords`	Lets you specify whether the content pack should include system account passwords.	False	No

Argument	Description	Default Value/Behavior	Required
	See "Exporting a Content Pack" in the *Studio Authoring Guide*. For security reasons, it is recommended to configure the system account passwords in Central.		
`-- overrideDestination`	Lets you specify whether to override an existing file with the same name as the content pack in the destination folder. If the destination file that the content pack represents exists on the file system, and `overrideDestination` is set to false, the content pack will not be created. If the destination file that the content pack represents exists on the file system, and `overrideDestination` is set to true, the content pack will override the original destination file.	False	No

Hewlett Packard
Enterprise

Java™
COMPATIBLE