CA Release Automation - 5.0.2

Actions and Custom Actions Development

Date: 28-Aug-2014

Table of Contents

Actions and Custom Actions Development

Actions are the building blocks that perform operations in your deployment environments. Design processes that perform actions in sequences, iterations, and branching workflows. An action becomes part of an automated process when it is defined in a flow. A flow is a group of actions with a defined sequence. CA Release Automation provides action templates that are used to design the processes that deploy applications to your environments.

The action template packages make up the inventory of actions from the following sources:

- Standard action template packages. These packages are built into the installation. Standard actions can be upgraded and down graded but cannot be deleted.
- Technology-specific action packs. These Action Packs are not provided with installation and must be downloaded from the Actions Management functionality in the Release Operations Center. Actions can be upgraded, downgraded, and deleted.
- Customized action template packages. These Custom actions must be imported into Automation Studio. See Manage Actions in Automation Studio for more information. See the Custom Actions SDK for information about how to create custom actions.

Actions Help

To view a listing of input and output descriptions for deployed actions, refer to the embedded actions help available in the installation at:

`http://<yourdeployment>/nolio-app/actionshelp.jsp.`

Note: When working with action packs, preliminary configuration tasks may be required before using the actions. These preliminary tasks are described in the documentation for the relevant Action Pack available with the download.

Manage Actions in Automation Studio

The inventory of action templates is managed in Automation Studio from Action Management in the Administration tab. These action templates are organized in packages and categories. Action packages include standard actions, technology-specific actions, and customized actions. Packages are collections of category groups. Each category represents a group of action templates that have a common usage.

Note: Managing action packs require the administrative user to have Superuser or General Administrator access.

Action Management provides the following viewing and management options:

- **Package View**
  This tab alphabetically displays the actions that are installed or have been imported. In this view you can:
  - Filter package display.

    Note: By default, deprecated actions do not appear in the actions filter and search options.
  - View the number of packages installed. Number includes standard action packages, customized action packages, and technology-specific action packs.
  - View the number of actions for all the packages installed.
  - Display third-party dependencies.

    Note: Select this check box when importing to include third-party dependancies.
  - Import action packages. Use the Import Action Packages icon to import customized actions into Automation Studio.

    Note: Technology-specific action packs can be downloaded using the Actions Management functionality in Release Operations Center.
  - Remove action packs. Use this icon to delete action packs in Automation Studio.
    Note: Action packs are synchronized to a separate repository for actions. Action packs are uploaded to the repository when they are imported. An action pack deleted in Automation Studio is also removed from the repository.
- **Category View**
  This tab displays the actions according to category and sub-category. Action categories are predefined and cannot be edited. In this view you can:
  - Filter category display.
- **Console View**
  This tab displays the recent history of package imports and removals. In this view you can:
  - Export log to a text file.

Note: If Automation Studio is open when you download an action pack using Release Operations Center, open the Action Management page in Automation Studio and click the Refresh button to update the list of available actions.

Manage Action Packs in Release Operations Center

The inventory of action templates is managed and viewed from Action Management in the Administration tab. Action packages include standard actions, technology-specific actions, and customized actions. Packages are collections of category groups. Each category represents a group of action templates that have a common usage.

Note: Managing action packs require the user to have a Superuser authorization.

Action Management provides the following viewing and management tabs:

- **Available Packs**
  This tab displays the action packs that are available for download to your system. As a convenience, your system synchronizes with the CA server and lists newly published action packs in the Available Packs listing. These packages are denoted with new icons and are displayed above the alphabetically sorted packages.
  To download a package, click the check box for the package, click Download, and follow the prompts to download and install automatically the package to your system.

- **Installed Packs**
  This tab displays the action packs that are installed or have been imported to your system. From this tab you can:

  - View the action packs currently installed on your system.

  - View action lists. To display all actions within a package, click the action pack name link to display the documentation.

  - Import action packs from a local server. If, for example, your Release Automation server is not connected to the Internet for security protocol, you can import the package from another server. Import all customized action packs. Click Import and navigate to the local file, such as the FTP site or support.ca.com. When importing from the CA FTP site, ensure that the 8083 default port is open. Optionally, configure an alternative port through the <product>\UpdateService\URL.ini file. The imported packages are often .jar files. Manually download and install the package.
    Note: If the Download Manager FTP functionality is not available due to the network configuration, then manually download the packs. Then, upload the action packs through Automation Studio.

  - Delete action packs. To remove action packs from your system, click the check box for the package and click Delete. Note: An action pack deleted in the Release Operations Center is also removed from the repository.

  - Update action packs. Installed packages with newer versions are identified though an update icon. Click the check box for the package, click Update, and follow the prompts to replace your package with the latest published version.

Note: If Automation Studio is open when you download an action pack using Release Operations Center, open the Action Management page in Automation Studio and click the Refresh button to update the list of available actions.

Standard Actions

This reference contains a complete list of standard actions that are built into CA Release Automation. These actions are installed in the actionsLib directory during the installation. Actions are predefined operations that you use to design processes and flows. You can expand the list of available actions with action packs.

Actions are organized into the following categories in Automation Studio:

Comparison Actions

Contents

Arrays - Compare Two Arrays

Compares two arrays of any object as if they are Strings. All non string elements of the arrays are converted to strings before comparison.

Input Values

Name	Type	Description
*** First Array**	Serializable[]	The first array.
*** Second Array**	Serializable[]	The second array.
Expect To Be Equal	Boolean	Whether to expect the two arrays to be equal. True for equal or False for not.
Ignore Case	Boolean	Indicates whether the comparison ignores case. True for ignore or False for not.

Compare Two Booleans

Checks if two Boolean arguments are equal.

Input Values

Name	Type	Description
*** First Value**	Boolean	The first value.
*** Second Value**	Boolean	The second value.

Compare Two Dates

Compares two dates in the same date format.

Example: Format = yyyy/MM/dd HH:mm:ss Date = 2007/06/28 20:30:55

Input Values

Name	Type	Description
*** First Value**	String	The first date value. The date format is "yyyy/MM/dd HH:mm:ss"
*** Second Value**	String	The second date value. The date format is "yyyy/MM/dd HH:mm:ss"
Format	String	The date format. **Example:** yyyy/MM/dd HH:mm:ss
Operator	ComparisonOp	The expected comparison operator.

Compare Two File Checksums

Compares the checksum of two files.

Input Values

Name	Type	Description
*** First File Path**	String	Specifies the first file path.
*** Second File Path**	String	Specifies the second file path.
Should The Checksums Match	Boolean	Indicates whether the checksums should match. **Note:** Effects the result of the step.

Output Values

Name	Type	Description
First File Checksum	String	The first file checksum.
Result Of Comparison	Boolean	The result of the comparison.
Second File Checksum	String	The second file checksum.

Compare Two Folders

Compares the contents of two folders according to Comparison Strategy.

Input Values

Name	Type	Description
*** First Folder Path**	String	Specifies the first folder path.
*** Second Folder Path**	String	Specifies the second folder path.
Comparison Report Path	String	Specifies the comparison report path.
Exclude Filters	String	Specifies Folder Exclude filters.
Filters Separator	String	Specifies the character to use as the filter separator.
Folders Comparison Strategy	FoldersComparisonStrategy	Indicates the comparison strategy to use. FULL_COMPARISON = Return true only in case of a full match. IGNORE_EXTRA_FILES_IN_FIRST /SECOND = Ignore the extra files in the first/second directory
Include Filters	String	Specifies Folder Include filters.
Include Ok Status In Report	Boolean	Indicates whether to include OK status in the report.
Should The Folders Match	Boolean	Indicates whether the folders should match. **Note:** Effects the result of the step.

Output Values

Name	Type	Description
Diff Files In First Folder	String[]	The result of the comparison, the diff file paths in the first folder.
Diff Files In Second Folder	String[]	The result of the comparison, the diff file paths in the second folder.

Name	Type	Description
Missing Files In First Folder	String[]	The result of the comparison, the missing file paths in the first folder.
Missing Files In Second Folder	String[]	The result of the comparison, the missing file paths in the second folder.
Result Of Comparison	String	The result of the comparison.

Compare Two Numbers

Compares two numbers.

Input Values

Name	Type	Description
*** First Value**	Number	Specifies the first number value.
*** Second Value**	Number	Specifies the second number value.
Operator	ComparisonOp	Specifies the expected comparison operator.

Compare Two Strings

Checks whether two strings are equal.

Input Values

Name	Type	Description
*** First String**	String	Specifies the first string value.
*** Second String**	String	Specifies the second string value.
Ignore Case	boolean	Indicates whether to ignore case.
Should Be Equal	boolean	Indicates the expected result of the comparison.

Compare Two Values

Compares two values. For example, use this action to check if two parameters are equal.

Input Values

Name	Type	Description

Name	Type	Description
*** First Value**	Serializable	Specifies the first value to compare. Set this value from Parameter.
*** Second Value**	Serializable	Specifies the second value to compare. Set this value from Parameter.

Control Actions

Contents

Delay

Creates a delay for the specified amount of time.

Input Values

Name	Type	Description
Expected Result	Boolean	The result of the step after the time has elapsed. Use True for pass or False for fail.
Result Description	String	Description of the result that is reported for this action.
Time For Delay	Long	The amount of time to delay.
Time Unit	TimeUnit	The time unit to use: MILLISECONDS, SECONDS, and so on.

Run Command Line

Executes the command line like it was executed from the command line shell. You can execute the command line in the background and not wait for it to finish the execution by setting the Wait For Process To Finish input to False.

Input Values

Name	Type	Description
*** Command Line String**	String	The command line to run.
Environment Variables Names	String[]	A list of environment variable names to set as the environment variables for the created process.
Environment Variables Values	String[]	A list of environment variable values to set as the environment variables for the created process. The values must match the names order in the Environment Variables Names input.
Expected Return Value	Integer	Expected return value of the finished process. If the actual return value is specified and not equal, the step fails.
Expected Return Values	Integer[]	Expected return values of the finished process. Only used if more than one possible return codes is expected.
Overwrite Files	Boolean	Overwrite or append the std out/err files. By default it overwrites the files if it already exist.
Std Err File Path	String	The file path to use to write the error stream of the process. It can be set to be the same as stdout
Std Out File Path	String	The file path to use to write the output stream of the process.
Success Return List	Boolean	True, the expected return value list is used to determine if the command executed successfully. False, to determine if the command failed.
Time Out Duration	Long	Time to wait for the process in seconds. Relevant only if Wait For Process To Finish is set to True.
Wait For Process To Finish	Boolean	Whether to wait for the process to finish its execution or not. True for wait, False not to wait. On non-windows systems the step may fail if the input is set to False.
Work Directory	String	The working directory to run from.

Output Values

Name	Type	Description
Exceed Time Out	Boolean	Does command execution timeout occur?
Return Value	Integer	Exit code of the finished process.
Std Err Output	String	Standard error of the process. Valid only if Wait for process to finish is set. A maximum of 256 characters will be returned..
Std Out Output	String	Standard output of the process. Valid only if Wait for process to finish is set. A maximum of 256 characters will be returned.
Success Return List	Boolean	True, the expected return value list is used to determine if the command executed successfully. False, to determine if the command failed.

User Input - Stop for Manual Operation

The manual operation instructions appears to the user describing the manual operations to be performed. Once complete, the user approves a successful or unsuccessful completion of the specified operation.

Input Values

Name	Type	Description
*** Operation Description**	String	The description of the operation to be displayed. By default the field is displayed in an HTML container.
Description Is HTML	Boolean	Whether the description of the operation to be displayed is in HTML format.
Failure Label	String	The label that indicates operation failure. The label is used as the Failure button text.
Failure Message	String	The description of the step result that indicates the failure of the manual operation.
Success Label	String	The label that indicates the operation success. The label is used as the Success button text.
Success Message	String	The description of the step result that indicates the success of the manual operation.

Wait for Manual Restart

Places the agent that it runs on into a safe state so the agent service or the host can be restarted.

Note: To continue the process after the step finishes, restart the agent service.

Database Actions

Contents

Note: Any database action other than MS SQL Server and Oracle, requires manual installation of the corresponding jdbc drivers (jar files) to the CA ReleaseAutomation Server (actionslib directory) before executing the Action. For more information, see Manage Action Packs in Release Operations Center

Execute SQL Query on IBM DB2 Server

Executes any type of SQL on IBM DB2 server.

Input Values

Name	Type	Description
* **Database Name**	String	Specifies the database name.

Name	Type	Description
*** SQL Query To Execute**	String	Specifies the SQL Query to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Batch Mode	Boolean	Indicates whether to handle the SQL query supplied in batch mode. **Note:** Batch mode is available in UPDATE mode only.
Batch Mode Separator	String	Specifies the separator to use in batch mode. The separator splits the SQL Query, and resulting strings run in batch mode.
Connection String	String	Specifies the database driver connection string. **Note:** Specify the value if you specify Driver Class Name.
Csv File Encoding	String	Specifies the encoding to use for the CSV file. For example: ISO-8859-1, US-ASCII, UTF-8, and UTF-16. **Note:** For more information, see http://www.iana.org/assignments/character-sets.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Expected Number Of Affected Lines	Integer	Specifies the expected number of lines that the SQL query affects. Use the input to verify the success of the query, by comparing expected number of affected lines to the actual affected lines. If the affected number of lines is different than expected the step fails. **Note:** Leave the value blank to skip this verification.

Name	Type	Description
Ignore Errors During Batch Mode	Boolean	Indicates whether to ignore errors during the execution of commands in batch mode.
Is Update	Boolean	Specifies whether the SQL query is an update. True for CREATE TABLE, UPDATE, INSERT or DELETE queries. False for all other queries.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Results To Csv Full Path	String	Specifies the full path of the file to save results to. The file is in CSV format. The input is relevant only if the operation is select query.
Server Port	int	Specifies the database server port.

Output Values

Name	Type	Description
Last Batch Command Number Of Actual Affected Lines	int	The number of lines that the last SQL Query command affected in batch mode. **Note:** If not in batch mode, the value is the same as Number Of Actual Affected Lines.
Number Of Actual Affected Lines	int	The number of lines that the SQL Query affected.
Result Is Empty	Boolean	Indicates whether result set was empty.
Result Set	Object[][]	The results in a 2 dimensions array (Object[][]). The first dimension is the columns and the second dimension is the rows.

Execute SQL Query on Microsoft(c) SQL Server

Executes any type of SQL on Microsoft(c) SQL Server.

Input Values

Name	Type	Description

Name	Type	Description
*** Database Name**	String	Specifies the database.
*** SQL Query To Execute**	String	Specifies the SQL Query to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Batch Mode	Boolean	Indicates whether to handle the SQL query supplied in batch mode. **Note:** Batch mode is available in UPDATE mode only.
Batch Mode Separator	String	Specifies the separator to use in batch mode. The separator splits the SQL Query, and resulting strings run in batch mode.
Connection String	String	Specifies the database driver connection string. **Note:** Specify this value if you specify Driver Class Name.
Csv File Encoding	String	Specifies the encoding to use for the CSV file. For example ISO-8859-1, US-ASCII, UTF-8, and UTF-16. **Note:** For more information, see http://www.iana.org/assignments/character-sets.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Expected Number Of Affected Lines	Integer	Specifies the expected number of lines that the SQL query affects. Use the input to verify the success of the query, by comparing expected number of affected lines to the actual affected lines. If the affected number of lines is different than expected the step fails. **Note:** Leave the value blank to skip the verification.

Name	Type	Description
Ignore Errors During Batch Mode	Boolean	Indicates whether to ignore errors during the execution of commands in batch mode.
Is Update	Boolean	Specifies whether the SQL query is an update. True for CREATE TABLE, UPDATE, INSERT or DELETE queries. False for all other queries.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Results To Csv Full Path	String	Specifies the full path of the file to save results to. The file is in CSV format. The input is relevant only if the operation is select query.
Server Port	int	Specifies the database server port.
Use Windows Authentication	Boolean	Indicates whether to use Windows authentication to log to the server.

Output Values

Name	Type	Description
Last Batch Command Number Of Actual Affected Lines	int	The number of lines that the last SQL Query command affected in batch mode. **Note:** If not in batch mode, the value is the same as Number Of Actual Affected Lines.
Number Of Actual Affected Lines	int	The number of lines that the SQL Query affected.
Result Is Empty	Boolean	Indicates whether result set is empty.
Result Set	Object[][]	The results in a 2 dimensions array (Object[][]). The first dimension is the columns and the second dimension is the rows.

Execute SQL Query on MySQL

Executes any type of SQL on the MySQL database server.

Input Values

Name	Type	Description
*** Database Name**	String	Specifies the database name.
*** SQL Query To Execute**	String	Specifies the SQL Query to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Batch Mode	Boolean	Indicates whether to handle the SQL query supplied in batch mode. **Note:** Batch mode is available in UPDATE mode only.
Batch Mode Separator	String	Specifies the separator to use in batch mode. The separator splits the SQL Query, and resulting strings run in batch mode.
Connection String	String	Specifies the database driver connection string. **Note:** Specify the value if you specify Driver Class Name.
Csv File Encoding	String	Specifies the encoding to use for the CSV file. For example ISO-8859-1, US-ASCII, UTF-8, or UTF-16. **Note:** For more information, see http://www.iana.org/assignments/character-sets.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.

Name	Type	Description
Expected Number Of Affected Lines	Integer	Specifies the expected number of lines that the SQL query affects. Use the input to verify the success of the query, by comparing expected number of affected lines to the actual affected lines. If the affected number of lines is different than expected the step fails. **Note:** Leave the value blank to skip the verification.
Ignore Errors During Batch Mode	Boolean	Indicates whether to ignore errors during the execution of commands in batch mode.
Is Update	Boolean	Specifies whether the SQL query is an update. True for CREATE TABLE, UPDATE, INSERT or DELETE queries. False for all other queries.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Results To Csv Full Path	String	Specifies the full path of the file to save results to. The file is in CSV format. The input is relevant only if the operation is select query.
Server Port	int	Specifies the database server port.

Output Values

Name	Type	Description
Last Batch Command Number Of Actual Affected Lines	int	The number of lines that the last SQL Query command affected in batch mode. **Note:** If not in batch mode, this value is the same as Number Of Actual Affected Lines.
Number Of Actual Affected Lines	int	The number of lines that the SQL Query affected.
Result Is Empty	boolean	Indicates whether result set was empty.

Name	Type	Description
Result Set	Object[][]	The results in a 2 dimensions array (Object[][]). The first dimension is the columns and the second dimension is the rows.

Execute SQL Query on Oracle(c) Database Server

Executes any type of SQL on Oracle(c) Database Server.

Input Values

Name	Type	Description
*** SQL Query To Execute**	String	Specifies the SQL Query to execute.
*** Server Name**	String	Specifies the database server name.
*** Sid**	String	Specifies the SID of database.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Batch Mode	Boolean	Indicates whether to handle the SQL query supplied in batch mode. **Note:** Batch mode is available in UPDATE mode only.
Batch Mode Separator	String	Specifies the separator to use in batch mode. The separator splits the SQL Query, and resulting strings run in batch mode.
Connection String	String	Specifies the database driver connection string. **Note:** Specify the value if you specify Driver Class Name.
Csv File Encoding	String	Specifies the encoding to use for the CSV file. For example ISO-8859-1, US-ASCII, UTF-8, and UTF-16. **Note:** For more information, see http://www.iana.org/assignments/character-sets.

Name	Type	Description
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Expected Number Of Affected Lines	Integer	Specifies the expected number of lines that the SQL query affects. Use this input to verify the success of the query, by comparing expected number of affected lines to the actual affected lines. If the affected number of lines is different than expected the step fails. **Note:** Leave the value blank to skip the verification.
Ignore Errors During Batch Mode	Boolean	Indicates whether to ignore errors during the execution of commands in batch mode.
Is Update	Boolean	Specifies whether the SQL query is an update. True for CREATE TABLE, UPDATE, INSERT or DELETE queries. False for all other queries.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Results To Csv Full Path	String	Specifies the full path of the file to save results to. The file is in CSV format. The input is relevant only if the operation is select query.
Server Port	int	Specifies the database server port.

Output Values

Name	Type	Description
Last Batch Command Number Of Actual Affected Lines	int	The number of lines that the last SQL Query command affected in batch mode. **Note:** If not in batch mode, this value is the same as Number Of Actual Affected Lines.
Number Of Actual Affected Lines	int	The number of lines that the SQL Query affected.

Name	Type	Description
Result Is Empty	boolean	Indicates whether result set was empty.
Result Set	Object[][]	The results in a 2 dimensions array (Object[][]). The first dimension is the columns and the second dimension is the rows.

Execute SQL Query on Sybase Database Server

Executes any type of SQL on Sybase Database Server.

Input Values

Name	Type	Description
*** Database Name**	String	Specifies the database name.
*** SQL Query To Execute**	String	Specifies the SQL Query to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Batch Mode	Boolean	Indicates whether to handle the SQL query supplied in batch mode. **Note:** Batch mode is available in UPDATE mode only.
Batch Mode Separator	String	Specifies the separator to use in batch mode. The separator splits the SQL Query, and resulting strings run in batch mode.
Connection String	String	Specifies the database driver connection string. **Note:** Specify the value if you specify the Driver Class Name.
Csv File Encoding	String	Specifies the encoding to use for the CSV file. For example ISO-8859-1, US-ASCII, UTF-8, and UTF-16. **Note:** For more information, see http://www.iana.org/assignments/character-sets.

Name	Type	Description
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Expected Number Of Affected Lines	Integer	Specifies the expected number of lines that the SQL query affects. Use the input to verify the success of the query, by comparing the expected number of affected lines to the actual affected lines. If the affected number of lines is different than expected the step fails. **Note:** Leave the value blank to skip the verification.
Ignore Errors During Batch Mode	Boolean	Indicates whether to ignore errors during the execution of commands in batch mode.
Is Update	Boolean	Specifies whether the SQL query is an update. True for CREATE TABLE, UPDATE, INSERT or DELETE queries. False for all other queries.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Results To Csv Full Path	String	Specifies the full path of the file to save results to. The file is in CSV format. This input is relevant only if the operation is select query.
Server Port	int	Specifies the database server port.

Output Values

Name	Type	Description
Last Batch Command Number Of Actual Affected Lines	int	The number of lines that the last SQL Query command affected in batch mode. **Note:** If not in batch mode, this value is the same as Number Of Actual Affected Lines.

Name	Type	Description
Number Of Actual Affected Lines	int	The number of lines that the SQL Query affected.
Result Is Empty	boolean	Indicates whether result set was empty.
Result Set	Object[][]	The results in a 2 dimensions array (Object[][]). The first dimension is the columns and the second dimension is the rows.

Run SQL File on IBM DB2 Server

Executes any type of *.sql file on IBM DB2 Server.

Input Values

Name	Type	Description
*** Database Name**	String	Specifies the database name.
*** SQL File To Execute**	String	Specifies the files path of the SQL File to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Auto Commit	Boolean	Specifies whether each statement is in its own transaction that commits immediately.
Connection String	String	Specifies the database driver connection string. **Note:** Specify the value if you specify Driver Class Name.
Delimiters	String[]	Specifies the delimiters that separate the queries in the SQL file.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)

Name	Type	Description
Server Port	int	Specifies the database server port.
Stop On Error	Boolean	Indicates whether the action stops executing on an error. True to stop on error, False to continue on error.

Run SQL File on Microsoft(c) SQL Server

Executes any type of *.sql file on Microsoft(c) SQL Server.

Input Values

Name	Type	Description
*** Database Name**	String	Specifies the database name.
*** SQL File To Execute**	String	Specifies the files path of the SQL File to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Auto Commit	Boolean	Specifies whether each statement is in its own transaction that commits immediately.
Connection String	String	Specifies the database driver connection string. **Note:** Specify the value if you specify Driver Class Name.
Delimiters	String[]	Specifies the delimiters that separate the queries in the SQL file.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Server Port	int	Specifies the database server port.

Name	Type	Description
Stop On Error	Boolean	Indicates whether the action stops executing on an error. True to stop on error, False to continue on error.
Use Windows Authentication	Boolean	Indicates whether to use Windows authentication to log to the server.

Run SQL File on MySQL Server

Executes any type of *.sql file on MySQL Server.

Input Values

Name	Type	Description
*** Database Name**	String	Specifies the database name.
*** SQL File To Execute**	String	Specifies the files path of the SQL File to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Auto Commit	Boolean	Specifies whether each statement is in its own transaction that commits immediately.
Connection String	String	Specifies the database driver connection string. **Note:** Specify this value if you specify Driver Class Name.
Delimiters	String[]	Specifies the delimiters that separate the queries in the SQL file.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify the value.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Server Port	int	Specifies the database server port.

Name	Type	Description
Stop On Error	Boolean	Indicates whether the action stops executing on an error. True to stop on error, False to continue on error.

Run SQL File on Oracle(c) Database Server

Executes any type of *.sql file on Oracle(c) Database Server.

Input Values

Name	Type	Description
*** SQL File To Execute**	String	Specifies the files path of the SQL File to execute.
*** Server Name**	String	Specifies the database server name.
*** Sid**	String	Specifies the SID of database.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Auto Commit	Boolean	Specifies whether each statement is in its own transaction that commits immediately.
Connection String	String	Specifies the database driver connection string. **Note:** Specify this value if you specify Driver Class Name.
Delimiters	String[]	Specifies the delimiters that separate the queries in the SQL file.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify this value.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Server Port	int	Specifies the database server port.
Stop On Error	Boolean	Indicates whether the action stops executing on an error. True to stop on error, False to continue on error.

Run SQL File on Sybase Database Server

Executes any type of *.sql file on Sybase Database Server.

Input Values

Name	Type	Description
*** Database Name**	String	Specifies the database name.
*** SQL File To Execute**	String	Specifies the files path of the SQL File to execute.
*** Server Name**	String	Specifies the database server name.
*** User Name**	String	Specifies the database user name.
*** User Password**	Password	Specifies the database user password.
Auto Commit	Boolean	Specifies whether each statement is in its own transaction that commits immediately.
Connection String	String	Specifies the database driver connection string. **Note:** Specify this value if you specify Driver Class Name.
Delimiters	String[]	Specifies the delimiters that separate the queries in the SQL file.
Driver Class Name	String	Specifies the database driver class name. **Note:** Specify the Connection String if you specify this value.
Properties	String[]	Specifies additional properties to pass to the database driver. Use an array of key=value pairs. ('=' is required.)
Server Port	int	Specifies the database server port.
Stop On Error	Boolean	Indicates whether the action stops executing on an error. True to stop on error, False to continue on error.

Test DSN Connection

Tests a predefined DSN connection.

Supported OS: Windows

Note: If the username and password are not specified, NT authentication is used based on the identity of the agent that executes the step.

Input Values

Name	Type	Description
*** DS Nname**	String	Specifies the DSN to test.
User Name	String	Specifies the username.
User Pass	String	Specifies the password.

Development Tool Actions

Contents

Run Ant Script

Runs an Ant script.

Input Values

Name	Type	Description
*** Build File**	String	The build file path / working directory.
Ant Home	String	Ant Home path (default is ANT_HOME environment setting).
Ant Properties	String	Ant Properties file to pass to ant (-lib, -keep-going, etc...).
Build Targets	String	The build targets.
JAVA_HOME	String	JAVA_HOME environment variable. Only required if not already defined in system.
Log File Path	String	Path to the created log file.
Properties	String[]	Properties to pass to ant (no need for -D).
Properties File	String	Properties file to pass to ant.

Output Values

Name	Type	Description
Exit Code	Integer	Exit code
Result String	String	Ant result string (Build Failed/Build Success)

Run Maven Script

Runs a Maven script.

Input Values

Name	Type	Description
*** Phases and Goals**	String	The phases and goals.
*** Working Directory / Pom file**	String	The working directory or the pom file path.
JAVA_HOME	String	JAVA_HOME environment variable. Required if not already defined in system.
Log File Path	String	Path to the created log file.
Maven Home	String	Maven home path (default is M2_HOME environment setting).
Maven Properties	String	Maven Properties to pass to maven (-N, -o, -fn, etc..).
Properties	String[]	System Properties to pass to Maven (no need for -D).

Output Values

Name	Type	Description
Exit Code	Integer	Exit code.
Result String	String	Maven result string (Build Failed/Build Success).

Run Selenium Script

Runs Selenium scripts that are written in Java.

Input Values

Name	Type	Description

Name	Type	Description
*** Script Path**	String	The full path to the Selenium script file. **Note:** To transform HTML scripts to Java, use the Selenium IDE. Save the script using the Java / JUnit4 / Remote Control option.

Output Values

Name	Type	Description
Error Report	String	Selenium detailed error report.

Email Actions

Contents

Send Email

Sends an email according to the specified properties.

Input Values

Name	Type	Description
*** Smtp Mail Server**	String	The outgoing mail server address (e.g. smtp.somewhere.com)
Attachments Paths	String[]	The list of the full paths to the locations of the attachments to use.
Bcc	String[]	The list of the BCC targets of the email (the BCC field).
Body	String	The body of the mail.
Body Type	String	The body content type, either text/html or text/plain.
Cc	String[]	The list of the CC targets of the email (the CC field).
From	String	The from address to use.

Name	Type	Description
Server requires Authentication	Boolean	Indicates whether the server requires authentication.
Smtp Port	int	The outgoing mail server port.
Smtp User Name	String	The outgoing mail server user name to use.
Smtp User Password	Password	The outgoing mail server user password to use.
Subject	String	The subject of the mail.
To	String[]	The list of the targets of the email (the To field).

Output Values

Name	Type	Description
Failed to send	String[]	The list of the emails that failed to be sent.

File/Folder Actions

Contents

- Create A Symbolic Link
- Create Empty File
- Create Folder
- Create ZIP File
- Delete File or Folder (for legacy OS)
- Delete Files or Folders
- Download File (HTTP)
- Download File From Network Drive
- Extract Text from a File
- Extract ZIP
- Find Files or Folders
- Find Text in File
- FTP - Create Directory
- FTP - Delete Directory
- FTP - Delete File
- FTP - Get Directory
- FTP - Get File
- FTP - List Files
- FTP - Upload File
- Get File Checksum
- Get File Details
- Get File or Folder from Remote Agent
- Get File or Folder from Remote Agent (for Legacy OS)
- Get File Properties
- Log to File on Center Machine
- Move File(s) or Folder(s) (for Legacy OS)
- Move Files or Folders
- Put File or Folder in Remote Agent

- Put File or Folder in Remote Agent (for Legacy OS)
- Read String from Text File
- Replace Text in File
- Replace Text in Multiple Files
- Synchronize Two Folders
- Upload File (HTTP)
- ZIP Many Files

Add Text to a File

Inserts text into a file. Specify the text that is to be searched in the file and add the Text To Add input after it.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file where the text is added.
*** Text To Add**	String	The text to be added. The input also accepts one of the String Parameters except for the String Array Parameter.
Add After This Text	String	A text to search for in the file. The specified Text To Add input is added after.
Add Text At The Beginning	Boolean	Whether to add the text at the beginning, True for beginning, False for end. (Relevant only if Add After This Text is empty).
Added new line character at the end of the text added	Boolean	Whether to add a new line character at the end of the text added.
Added text starts with new line.	Boolean	Whether to start the added text with a new line character.
Create File If Not Exist	Boolean	Create the file if it does not exist.
File Encoding	String	The encoding of the specified file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Name	Type	Description
Is Regex	Boolean	Whether to treat Add After This Text as a regular expression.

Arrays - Read String Array from Text File

Reads a text file and puts every line into a new array element.

Input Values

Name	Type	Description
*** Filename**	String	The file name to read.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Output Values

Name	Type	Description
Result Array	String[]	The result array.

Arrays - Write Array to Text File

Writes an array to a file.

Input Values

Name	Type	Description
*** Array**	Object[]	The array to write.
*** Filename**	String	The file name to write to.
Append	Boolean	True to append the file or False to rewrite the file.
Create File If Not Exists	Boolean	Indicates whether to create the file if it does not exist.
File Encoding	String	The Encoding of the specified file. For example ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Change Mode of File or Folder

Performs a change mode operation on the file or folder.
Note: Supports Unix/Linux systems only.

Input Values

Name	Type	Description
*** File Or Folder Path**	String	The path to the file or folder to change the mode on. The input also accepts a File parameters.
*** Mode**	Int	The mode number to change the file or folder to. Refer to the Unix or Linux 'chmod' command for more details.
Change Recursively	Boolean	Indicates whether to apply the changed mode recursively on a folder and the included files and sub-folders.
Fail If Not Exist	Boolean	Indicates whether to return false if the path is invalid.

Change Owner of File or Folder

Changes the owner of the file or folder.

Note: Supports Unix/Linux only.

Input Values

Name	Type	Description
*** File Name**	String	The path to the file or folder to change the owner.
*** New Owner**	String	The new owner user name.
Change Recursively	Boolean	If the resource is a folder, indicates whether to change the owenrship recursively.
Fail If Not Exists	Boolean	What to do if the file or folder does not exist, True to fail, False to ignore.

Change Read Only Mode of File or Folder

Sets the Read Only flag on a file or folder.

Input Values

Name	Type	Description
*** File Name**	String	The path to the file or folder to set the read-only flag on.
Change Recursively	Boolean	If the resource is a folder, indicates whether to change the read-only flag recursively.
Fail If Not Exists	Boolean	What to do if the file or folder does not exist, True to fail, False to ignore.
Read Only Mode	Boolean	Set or unset the read only flag of the file or folder, True to set, False to unset.

Check if File or Folder Exists

Checks if a file or folder exists in the file system.

Input Values

Name	Type	Description
*** File Or Folder Path**	String	The path to the file or folder to verify if it exists. The input also accepts File parameters. Wildcards * can be used.
Should Exist	Boolean	Indicates whether to expect the file or folder to exist or not, True for exist, False for does not exist.

Check if Specific User has Permissions on File or Folder

Checks if the specified user credentials has permissions to the file or folder.

Input Values

Name	Type	Description
*** File Or Folder Path**	String	The path to the file or folder to check. The input also accepts File parameters.
*** Password**	Password	The user password to verify the permissions on. Refer to the step help for more details.
*** User Name**	String	The user name to verify the permissions on. Refer to the step help for more details.

Name	Type	Description
		Note: On Window systems that use Domain, specify the domain name with the user name. Example: *DOMAIN\Username*. On non-windows systems that use domain, do not specify the domain name.
Check Execute Permissions	Boolean	Indicates to check if the specified user has execution permissions for the file or folder
Check Read Permissions	Boolean	Indicates to check if the specified user has read permissions for the file or folder
Check Write Permissions	Boolean	Indicates to check if the specified user has write permissions for the file or folder

Compare Two File Checksums

Compares the checksum of two files.

Input Values

Name	Type	Description
*** First File Path**	String	Specifies the first file path.
*** Second File Path**	String	Specifies the second file path.
Should The Checksums Match	Boolean	Indicates whether the checksums should match. **Note:** Effects the result of the step.

Output Values

Name	Type	Description
First File Checksum	String	The first file checksum.
Result Of Comparison	Boolean	The result of the comparison.
Second File Checksum	String	The second file checksum.

Compare Two Folders

Compares the contents of two folders according to Comparison Strategy.

Input Values

Name	Type	Description
*** First Folder Path**	String	Specifies the first folder path.
*** Second Folder Path**	String	Specifies the second folder path.
Comparison Report Path	String	Specifies the comparison report path.
Exclude Filters	String	Specifies Folder Exclude filters.
Filters Separator	String	Specifies the character to use as the filter separator.
Folders Comparison Strategy	FoldersComparisonStrategy	Indicates the comparison strategy to use. FULL_COMPARISON = Return true only in case of a full match. IGNORE_EXTRA_FILES_IN_FIRST /SECOND = Ignore the extra files in the first/second directory
Include Filters	String	Specifies Folder Include filters.
Include Ok Status In Report	Boolean	Indicates whether to include OK status in the report.
Should The Folders Match	Boolean	Indicates whether the folders should match. **Note:** Effects the result of the step.

Output Values

Name	Type	Description
Diff Files In First Folder	String[]	The result of the comparison, the diff file paths in the first folder.
Diff Files In Second Folder	String[]	The result of the comparison, the diff file paths in the second folder.
Missing Files In First Folder	String[]	The result of the comparison, the missing file paths in the first folder.
Missing Files In Second Folder	String[]	The result of the comparison, the missing file paths in the second folder.
Result Of Comparison	String	The result of the comparison.

Copy File or Folder with Backup

Copies a file or folder from the source path to the target path. All files that are overwritten in target path are saved in the folder specified by the backup path input.

Note: If the folder does not exist, one is created.

Input Values

Name	Type	Description
*** Backup Path**	String	The path of the backup files. The input also accepts File parameters.
*** Source Path**	String	The path of the source file or folder. The input also accepts File parameters.
*** Target Path**	String	The path of the target file or folder.

Copy File(s) or Folder(s) (for Legacy OS)

Copies the file(s) or folder(s) from the source path into the target folder.

Input Values

Name	Type	Description
*** Is Regular Expression**	Boolean	Checks the expression that the user entered is a regular expression (java Regex).
*** Overwrite**	Boolean	Indicates whether to overwrite a file if it already exists in the target location
*** Source Path or expression**	String	The path of the source file or folder
*** Target Parent Folder**	String	Path of the folder where the file or folder is copied.
Preserve Permissions	Boolean	Indicates whether to preserve the file permissions and owner information or not

Output Values

Name	Type	Description
Failed File Names	String[]	Array that contains names of files that could not be copied. (Might be a partial list if many files could not be copied).

Copy Files or Folders

Copy a file, folder, or the folder content to a specified destination. With the exception of a wildcard expression appearing in the source path, filters are applied recursively on the underlying file tree of the source path.

Example: If the exclude pattern is '*.txt' then none of the files matching the pattern will be copied, whether they lay directly under the source path or not.

Input Values

Name	Type	Description
*** Source Path**	String	The path where to copy from. Wildcard expression is permitted for the last path element only.
*** Target Path**	String	The path where to copy to. Must point to a specific file or folder.
Exclude Pattern	String	The wildcard or regular expression that defines which files or folders are excluded. Use either wildcard or regular expression, according to the chosen pattern type. Place several patterns using the ';' delimiter - files or folders that match at least one of the patterns are excluded.
Include Hidden Files	Boolean	Hidden files or folders as well as files under hidden folders are not copied if the parameter is false.
Include Pattern	String	The wildcard or regular expression that defines which files or folders are included. Use either wildcard or regular expression, according to the chosen pattern type. Place several patterns using the ';' delimiter - files or folders that match at least one of the patterns are included.
Include/Exclude Pattern Type	PatternType	Specifies whether the include and exclude patterns are regular expressions or wildcard expressions.
Overwrite	Boolean	If True, the agent overwrites the existing files with the copied files.

Output Values

Name	Type	Description
Failed to Copy	String[]	The file/folder paths of files or folders which were not successfully copied.
Was Copied	boolean	

Name	Type	Description
		True if at least one file or folder was copied.

Create A Symbolic Link

Creates a symbolic link on Unix/Linux computers.

Input Values

Name	Type	Description
*** Symbolic Link Path**	String	Full path of the symbolic link to create
*** Target Path**	String	Full path of the target (path of entity the symbolic link is refer to)

Create Empty File

Create an empty file.

Input Values

Name	Type	Description
*** Full Path**	String	The full path of the file to create.
Create Parent Path	Boolean	Set to False if you do not want to create a parent folder if it does not exist.
Fail If Exist	Boolean	By default the step fails if the file exist. Set to False to override the behavior.

Create Folder

Creates a folder in the file system. The full path specified is created.

Input Values

Name	Type	Description
*** New Folder Path**	String	The full path of the folder to create.
Fail If Exist	Boolean	By default the step does not fail if the folder exist. Set to True to override the behavior.

Create ZIP File

Creates a zip file.

Input Values

Name	Type	Description
*** Source Path**	String	File or directory path for the content to be zipped.
*** Zip File Path**	String	Fully qualified file name for the new zip file.
Directory Exclude Filter	String	Directory Exclude filter (e.g., "tmp" - directories that match the pattern are not included in the new zip file. **Note:** Filter tests against the path relative to "Source Path"
Files Exclude Filter	String	Exclude files filter (e.g., "*.dbg" - when the filter is set, files that match the filter are excluded from the new zip file.
Files Filter	String	Include files filter (e.g. "*.dll") - when the filter is set, only the files that match the filter are included in the new zip file.
Include Root Folder	Boolean	Whether to include the root folder as part of the new zip file.
Include Sub Folders	Boolean	Whether to traverse sub folders when creating the new zip file. Empty folders are included as well.
Regular Expression Filter	Boolean	Whether the filters are specified in RegEx syntax (typically only useful for more complex filtering).

Delete File or Folder (for legacy OS)

Deletes a file or folder. Choose whether to delete recursively or to delete folders.

Input Values

Name	Type	Description
*** Delete Folders**	Boolean	Whether to delete folders or not. If set to False, only the files are deleted and the folders remain empty.
*** Delete Recursive**	Boolean	

Name	Type	Description
		Delete files recursively, including files in subfolders.
*** Fail If Not Empty**	Boolean	Whether to fail if the folder is not empty or to delete its content. True to fail or False to delete its content.
*** Fail If Not Exist**	Boolean	Whether to fail if the path does not exist, True to fail or False to ignore.
*** File/Folder Path**	String	Full Path of the file or folder that you want to delete.
*** Is Regular Expression**	Boolean	Whether the expression that the user has entered is a regular expression (java Regex).

Delete Files or Folders

Deletes files or folders that corresponds to a specified path and according to various filters.

Input Values

Name	Type	Description
*** Path**	String	The path to the file or folder to be deleted. If a folder, only the folder is deleted if no files under it are filtered out. Only the last path element can contain a wildcard expression.
Deletion Strategy	DeletionStrategy	Select whether to delete only ordinary files or both files and folders.
Exclude Pattern	String	The wildcard or regular expression that defines which files or folders are excluded. Use either wildcard or regular expression according to the chosen pattern type. Place several patterns using the ';' delimiter - files or folders that match at least one of the patterns are excluded.
Fail If Not Exists	Boolean	If true, the action fails if the specified path, (or the base path if it contains a wildcard), does not lead to an existing file or folder. Also, If symbolic links are followed, the action fails if the file exists but is a broken symbolic link.

Name	Type	Description
Follow Symbolic Links	Boolean	If True, soft symbolic links are resolved before deletion, that is, both the symbolic link and its final target are deleted. Otherwise, only the symbolic link is deleted.
Include Hidden Files	Boolean	Hidden files or folders, as well as files under hidden folders are not deleted if the parameter is False.
Include Pattern	String	The wildcard or regular expression that defines which files or folders are included. Use either wildcard or regular expression according to the chosen pattern type. You can place several patterns using the ';' delimiter. Files or folders that match at least one of the patterns are included.
Include/Exclude Pattern Type	PatternType	Specifies whether the include and exclude patterns are regular expressions or wildcard expressions.

Output Values

Name	Type	Description
Failed to Delete	String[]	The file paths of files or folders that were not successfully deleted.
Was Deleted	Boolean	True if at least one file is deleted.

Download File (HTTP)

Downloads a file from a url using HTTP/S.

Input Values

Name	Type	Description
*** Destination File Path**	String	The destination path for the downloaded file.
*** Url**	String	HTTP or HTTPS URL to access.
Authentication Method	AuthMethod	

Name	Type	Description
		Set to one of the following methods: Basic, Digest or NTLM. Are used to authenticate with http servers or proxies. "NONE" means no authentication is required.
Connection Timeout (seconds)	Integer	Determines the seconds in seconds until a connection is established. Default value is 1 minute. A value of zero means the timeout is not used.
Domain	String	The servers domain. (Relevant if the Authentication Method is not "NONE")
Expected Code	Integer	Expected HTTP return code. The action fails if the expected code is not returned. If left empty accepts any code in the range 200-299.
Password	Password	The users password. (Relevant if the Authentication Method is not "NONE")
Proxy Server Name	String	The name of proxy server.
Proxy Server Port	Integer	The port of proxy server.
Read Timeout (seconds)	Integer	Defines the default socket timeout in seconds which is the timeout for awaiting data. A timeout value of zero is interpreted as an infinite timeout.
Realm	String	The realm that authenticates the user. (Relevant if the Authentication Method is "BASIC" or "DIGEST")
Use Preemptive Authentication	Boolean	Whether to use Preemptive Authentication when using a user name to login.
User Name	String	The user name for authentication. (Relevant if the Authentication Method is not "NONE")

Output Values

Name	Type	Description
HTTP Response Code	int	HTTP return code. - 1 is returned if no HTTP connection could be made.

Download File From Network Drive

Downloads a file from a network drive to a specified location. Please use HTTP download files or FTP files.

Input Values

Name	Type	Description
*** Destination Path**	String	The full path, including file name, to the location to download the file to.
*** Source URL**	String	The URL of the remote file to get.
Domain	String	The domain of the user
Fail If Not Exist	Boolean	Whether to fail if the path does not exist, True to fail or False to ignore.
Password	Password	The users password
User Name	String	The user account name to use for the connection.

Extract Text from a File

Extracts text from a file by regular expression and stores the matched groups as defined by the result template input.

The extraction result is stored to a String Parameter.

Example: if a regular expression is "url=http://(server\d)/(path\d)", the result template is "$2 on $1" and file contains the line "url=http://server8/path5", the extraction result is "path5 on server8". Refer to http://www.regular-expressions.info/ for more details.

Input Values

Name	Type	Description
*** Filename**	String	The file from where the text is to be extracted.
*** Regular Expression**	String	The text to find in the file. Refer to http://www.regular-expressions.info/ for more details about regular expressions syntax.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets
Result Template	String	

Name	Type	Description
		The template for the extraction result. Use $n for group n in regular expression input. Refer to step help for an example. To get character $ type \$.

Output Values

Name	Type	Description
Extraction Count	int	The number of extraction results in the file.
Extraction Results Array	String[]	An array of the extraction results in the file.
First Extraction Result	String	The result of the extraction.

Extract ZIP

Extracts a ZIP file to a destination path. The action fails if the path does not exist..

Input Values

Name	Type	Description
*** Destination Path**	String	The path to the destination where the contents of the ZIP file is extracted to.
*** Zip File Path**	String	The path to the ZIP file to extract.

Find Files or Folders

Find files and folders in the file system using various matching patterns and filters.

Input Values

Name	Type	Description
*** Path**	String	The base path for the files or folders lookup. Wildcard expression is permitted for the last path element only.
Exclude Pattern	String	The wildcard or regular expression that defines which files or folders are excluded. Use wildcard or regular expression according to the chosen pattern type. Place

Name	Type	Description
		several patterns using the ';' delimiter. Files or folders that match at least one of the patterns are excluded.
Fail If Not Exists	Boolean	If True, the action fails if the path (or base path if it contains wildcard) does not lead to an existing file or folder. If symbolic links are followed, the action fails if the file exists but is a broken symbolic link.
Include Empty Folders	Boolean	Empty folders are excluded from the output if the input is false.
Include Hidden Files	Boolean	Hidden files or folders, as well as files under hidden folders, are excluded from the output if the parameter is False.
Include Pattern	String	The wildcard or regular expression that defines which files or folders are included. Use wildcard or regular expression according to the chosen pattern type. Place several patterns using the ';' delimiter. Files or folders that match at least one of the patterns are included.
Include/Exclude Pattern Type	PatternType	Specifies whether the include and exclude patterns are regular expressions or wildcard expressions.
Included File Types	IncludedFileTypes	Select whether to include files, folders, or both in the output.
Last Modified After	String	Include only files or folders which were modified after this date. Date format is yyyy-MM-dd HH:mm:ss
Last Modified Before	String	Include only files or folders which were modified before this date. Date format is yyyy-MM-dd HH:mm:ss
Order Files By	OrderFilesBy	Determines the order by which the file or folder paths in the output array appears. **Note:** The order also depend on the chosen output format (example when ordering by name).
Order Type	OrderType	Order files and folders by ascending or descending.

Name	Type	Description
Output Format	FilePathFormat	Determines the format of the file or folder paths in the output.
Recursive	Boolean	Determines whether or not to recursively search the whole file tree under the path.

Output Values

Name	Type	Description
First Result	String	The first result in the output array of file/folder paths.
Inaccessible Files/Folders	String[]	An array of directories which were not readable or files which were inaccessible and undistinguished from directories. The existence of entries in the array does not imply the action failed.
Number of Results	int	The number of matching files/folders included in the output array.
Result File Paths	String[]	The file/folder paths for all the matching files.
Was Found	Boolean	True if at least one file/folder was found.

Find Text in File

Finds a word or a string in a text file.

Important! For regular expression searches, the supported files are limited to 10 MB.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file to search in.
*** Text To Find**	String	The text to find in the file. To search using Regular Expressions set the Is Regular Expression input to True. Refer to http://www.regular-expressions.info/ for more details.
Case Sensitive	Boolean	To consider the character case. True for case-sensitive or False for case-insensitive.
File Character Set	String	

Name	Type	Description
		The file character set if it is known. **Example:** Unicode or Windows-1255. If the input is left blank, the step starts with the systems default character set and if the text is not found another try is done with Unicode.
Regular Expression	Boolean	Whether to search using Regular Expressions. If True, the Text To Find input is considered as a Regular Expression.
Text Should Exist	Boolean	Whether to expect the text to be found or not. True, to be found or False, not to be found.
Timeout	int	Timeout to wait before fail (action retries until success or timeout), the units are in seconds.

FTP - Create Directory

Creates a directory on an FTP server.

Input Values

Name	Type	Description
*** New Folder Name**	String	The folder to create.
*** Server**	String	FTP Server hostname or IP
*** Username**	String	The user account name used for the connection. For an anonymous connection the value is anonymous.
Base Directory	String	Parent directory where to create the folder. Leave empty to use the root directory.
Password	Password	The users password. For an anonymous connection use a valid email address.
Port	int	The FTP Server port
Secured Connection	FtpConnection	Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS).
Timeout	int	Connection timeout in seconds.

FTP - Delete Directory

Deletes a directory from an FTP server.

Input Values

Name	Type	Description
*** Directory To Delete**	String	Full path of the directory to delete.
*** Server**	String	The FTP Server hostname or IP.
*** Username**	String	The user account name that is used for the connection. If using an anonymous connection the value is "anonymous".
Delete Content	Boolean	If True, all the content of the directory is deleted. If False, the action passes if the directory is empty.
Password	Password	The users password. For an anonymous connection use a valid email address.
Port	int	The FTP Server port
Secured Connection	FtpConnection	Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS).
Timeout	int	Connection timeout in seconds.

FTP - Delete File

Deletes a file from an FTP server.

Input Values

Name	Type	Description
*** File To Delete**	String	Full path of the file to delete.
*** Server**	String	The FTP Server hostname or IP.
*** Username**	String	The user account name that is used for connection. For an anonymous connection the value is anonymous.
Password	Password	The users password. For an anonymous connection use a valid email address.
Port	int	The FTP Server port
Secured Connection	FtpConnection	

Name	Type	Description
		Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS).
Timeout	int	Connection timeout in seconds.

FTP - Get Directory

Retrieves a directory from a remote FTP server.

Input Values

Name	Type	Description
*** Destination Directory**	String	Name of the directory in the destination. The full path is required.
*** Server**	String	The FTP Server hostname or IP.
*** Source Directory**	String	The Directory to retrieve.
*** Username**	String	The user account name that is used for connection. For an anonymous connection the value is anonymous.
Password	Password	The users password. For an anonymous connection use a valid email address.
Port	int	The FTP Server's port.
Secured Connection	FtpConnection	Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS).
Timeout	int	Connection timeout in seconds.
Transfer Type	DataTransferType	Using a binary transfer, the file is treated as a binary stream and is stored by the target computer as it is received. Using a textual data transfer treats the transferred file as a character stream, performing charset transformation.

FTP - Get File

Retrieves a file from a remote FTP server.

Input Values

Name	Type	Description
*** Destination File**	String	

Name	Type	Description
		Name of the file in the destination. The full path is required.
*** Server**	String	FTP Server's hostname or IP.
*** Source File**	String	File to retrieve.
*** Username**	String	User account name to use for the connection. In case of anonymous connection, the value should be anonymous.
Create Parent Path	Boolean	Set to false if you do not want to create a parent folder. If it does not exist.
Password	Password	The users password. For anonymous connection use a valid email address.
Port	int	FTP Server's port
Secured Connection	FtpConnection	Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS).
Timeout	int	Connection timeout in seconds.
Transfer Type	DataTransferType	Using a binary transfer, the file is treated as a binary stream and is stored by the target computer as it is received. A textual data transfer treats the transferred file as a character stream, performing charset transformation

FTP - List Files

Lists a remote directory content (first level only) over FTP using the provided parameters. The list is sorted by the modified date (newest first).

Input Values

Name	Type	Description
*** Server**	String	The FTP Server hostname or IP.
*** Username**	String	The user account name that is used for the connection. For an anonymous connection, the value is anonymous.
Directory	String	The directory to list the content.
Include Directories	Boolean	Include the directories in the list of files. If False only the files are listed.

Name	Type	Description
Password	Password	The users password. For an anonymous connection use a valid email address.
Port	int	The FTP Server port.
Return Full Path	Boolean	Use True returns the full path of the file or directory, relative to the root directory of the FTP server, (Example: directory input + the file name). Use false, only the files or directories names will be returned.
Secured Connection	FtpConnection	Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS).
Timeout	int	The connection timeout in seconds.

Output Values

Name	Type	Description
First Name Found	String	First name found.
List Output	String[]	The list of files and folders.
Number of Entries	int	Number of entries.
Was Found	Boolean	Indicating whether or not the directory had entries.

FTP - Upload File

Uploads a file to an FTP server.

Input Values

Name	Type	Description
*** File To Upload**	String	File path to update to the FTP site. You can not create a directory. For setting the directories hierarchy please use the FTP - Create Directory action,
*** Server**	String	the FTP Server hostname or IP.
*** Username**	String	The user account name that is used for connection. For an anonymous connection, the value is anonymous.
Password	Password	

Name	Type	Description
		The users password. For an anonymous connection use a valid email address.
Port	int	The FTP Server port
Secured Connection	FtpConnection	Specify the connection settings: FTP or FTPS (implicit or explicit over SSL/TLS)
Target Directory	String	The target directory in the FTP server, (the default is the base directory).
Timeout	int	The connection timeout in seconds.
Transfer Type	DataTransferType	Using a binary transfer, the file is treated as a binary stream and is stored by the target computer as its received. Using a textual data transfer treats the transferred file as a character stream, performing charset transformation

Get File Checksum

Generates a checksum of the file.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file that the checksum is generated for.

Output Values

Name	Type	Description
File Checksum	String	File checksum.

Get File Details

Receives a file name and checks the details on it

Input Values

Name	Type	Description
*** Full File Name**	String	The file name to check

Output Values

Name	Type	Description
Exist	Boolean	True if the file exist, False if not
File Name	String	The file name
Is Directory	Boolean	True if the file is directory, False if not
Parent Name	String	The file location

Get File or Folder from Remote Agent

Copies a file or folder from a path on a remote agent to a local agent.
The destination is the new location and the name of the file or folder. If the Destination path does not exist it is created with any missing parent folders. If the Destination path exists the action reacts as follows:

If both the source and the destination are files, the destination is replaced with the source.

If both the source and the destination are folders, the source and the destination are merged with the source taking precedence.

If a file is attempted to be replaced with a folder or vice versa the action fails.

Input Values

Name	Type	Description
*** Destination Path**	String	The path of the destination file or folder on the local computer. CA technologies recommends to use '//' as a file separator.
*** Remote Agent**	String	The IP address of the remote agent or Agent ID in the CA Communication Network
*** Source Path**	String	The path of the source file or folder on the remote computer. CA Technologies recommends to use '//' as a file separator.
Exclude Pattern	String	The wildcard or regular expression that defines which files or folders are excluded. Use wildcard or regular expression according to the specified pattern type. Place several patterns using the ';' delimiter. Files or folders that match at least one of the patterns are excluded.
File Owner	String	Sets the new Unix file owner and group for the destination. Relevant only if Preserve Unix Mode Permissions is

Name	Type	Description
		set). Usages: - only owner (for example - 'user1') - only group (for example - ':group1') - owner & group(for example - 'user1:mygroup') If left empty an attempt is made to preserve the owner from the source file or folder.
Include Hidden Files	Boolean	Hidden files or folders, as well as files under the hidden folders, are not copied if the parameter is False.
Include Pattern	String	The wildcard or regular expression that defines which files or folders are included. Use wildcard or regular expression according to the specified pattern type. Place several patterns using the ';' delimiter. Files or folders that match at least one of the patterns are included.
Include/Exclude Pattern Type	PatternType	Specifies whether the include and exclude patterns are regular expressions or wildcard expressions.
Overwrite	Boolean	Whether to overwrite a file or a folder if it exists in the destination path. If a file or a folder exists and the flag is set to False, the action fails.
Preserve Unix Mode Permissions	Boolean	Preserve the Unix Mode Permissions of the destination

Get File or Folder from Remote Agent (for Legacy OS)

Copies a file or a folder from a path on the remote agent to the local agent.
The destination is the new location and name of the file or folder.

If the Destination path does not exist it is created with any missing parent folders.

If the Destination path exists the action reacts as follows:

If both source and destination are files, the destination is replaced with the source.

If both source and destination are folders, the source and the destination are merged with the source taking precedence. If a file is attempted to be replaced with a folder or vice versa, the action fails.

Input Values

Name	Type	Description
*** Destination Path**	String	The path of the destination file or folder on the local computer. CA Technologies recommends to use '//' as a file separator.
*** Remote Agent**	String	The IP address of the remote agent or the Agent ID in CA Communication Network.
*** Source Path**	String	The path of the source file or folder on the remote computer. CA Technologies recommends to use '//' as file a separator.
File Owner	String	Sets the new Unix file owner and group for the destination. Relevant if 'Preserve Unix Mode Permissions' is set. Usages: - only owner (for example - 'user1') - only group (for example - ':group1') - owner & group(for example - 'user1:mygroup') Left empty an attempt is made to preserve the owner from the source file or folder.
Overwrite	Boolean	Whether to overwrite file or folder if it exists in the destination path. If file or folder exists and the flag is set to False, the action fails.
Preserve Unix Mode Permissions	Boolean	Preserve Unix Mode Permissions of the destination.

Get File Properties

Extracts the properties of a file.

Input Values

Name	Type	Description
*** File Path**	String	Path to file or folder to extract properties from.
Date Format	String	The string format to use. Default is yyyy/MM/dd HH:mm:ss .

Output Values

Name	Type	Description
File Size	long	File size in bytes. For folders use 0.

Name	Type	Description
Last Modified Date As Long	long	Last modified date as number of milliseconds from 00:00:00 GMT, January 1, 1970.
Last Modified Date As String	String	Last modified date as string. String format is specified by Date Format input.

Log to File on Center Machine

Allows the user to write a message to a CSV or a Text file on the center computer.

The CSV file format is: [time stamp],[agent],[message].

The Text file format is: [time stamp]tab[agent]tab[message].

Input Values

Name	Type	Description
*** File Path**	String	The path of a file. If the file does not exist it is created.
File Type	FileType	Select the log file type. Selecting TXT the values delimiter is tab. Selecting CSV the values delimiter is "," and each of the values is wrapped with ".
Message	String	The message to be logged.
Messages array	String[]	Array of messages. The array values are separated by the file delimiter.

Move File(s) or Folder(s) (for Legacy OS)

Moves a file or folder, or files and folders that match the regular expression, from the source path to the target path. Folders that exist at both the source and destination are merged. The Overwrite and the Fail If Exists flags refers to files and not directories, including files in the merged directories. If a file is skipped during the move the execution does not halt even if the Fail If Exists flag is True. The action fails after trying to move all the files.

Input Values

Name	Type	Description
*** Is Regular Expression**	Boolean	Whether the expression that the user types is a regular expression (java Regex).
*** Source Path or Expression**	String	The path of the source file or folder.
*** Target of File/Folder Path**	String	

Name	Type	Description
		The target location for the file or folder that you want to move.
Fail If Exists	Boolean	Whether to fail the action if one of the files to move exists in the target location.
Overwrite	Boolean	Whether to overwrite a file that exists in the target location.

Output Values

Name	Type	Description
Skipped Files	String[]	The array that contains the paths of the files that exist in the destination and are not overwritten.

Move Files or Folders

Move a file, folder, or the folder content to a specified destination. With the exception of a wildcard expression appearing in the source path, all specified filters are applied recursively on the underlying file tree of the source path.

Example: If the exclude pattern is '*.txt', none of the files the match the pattern are moved whether directly under the source path or not.

Input Values

Name	Type	Description
*** Source Path**	String	The path where to move the files from. If it is a folder then the folder itself is moved only if no files under it are filtered out. Wildcard expression is permitted for the last path element only.
*** Target Path**	String	The path where to move the files to. Point to a specific file or folder.
Exclude Pattern	String	The wildcard or regular expression that defines the files or folders that are excluded. Use wildcard or regular expression according to the specified pattern type. Place several patterns using the ';' delimiter and files or folders that match at least one of the patterns are excluded.
Include Hidden Files	Boolean	

Name	Type	Description
		Hidden files or folders as well as files under hidden folders are not moved if the parameter is False.
Include Pattern	String	The wildcard or regular expression that defines the files or folders that are included. Use wildcard or regular expression according to the specified pattern type. Place several patterns using the ';' delimiter and files or folders that match at least one of the patterns are included.
Include/Exclude Pattern Type	PatternType	Specified whether the include and exclude patterns are regular expressions or wildcard expressions.
Overwrite	Boolean	Select True, the agent overwrites the existing files with the copied files.

Output Values

Name	Type	Description
Failed to Move	String[]	The file/folder paths of files or folders that are not successfully moved.
Was Moved	Boolean	True if at least one file or folder was moved.

Put File or Folder in Remote Agent

Copies a file or folder from the local agent to a path on the specified remote agent.
The destination is the new location and name of the file or folder. If the destination path does not exist it is created with any missing parent folders.

If the Destination path exists then the action reacts as follows:

If the source and the destination are files, the destination is replaced with the source.

If the source and the destination are folders, the source and destination are merged with the source taking precedence.

If a file is attempted to be replaced with a folder or vice versa the action fails.

Input Values

Name	Type	Description
*** Agent Ip**	String	

Name	Type	Description
		The IP address of the remote agent.
*** Destination Path**	String	The path of the destination file or folder on the remote computer. CA Technologies recommends to use '//' as a file separator.
*** Source Path**	String	The path of the source file or folder on the local computer. The last path element may contain a wildcard expression.
Exclude Pattern	String	The wildcard or regular expression that defines the files or folders that are excluded. Use wildcard or regular expression according to the selected pattern type. Place several patterns using the ';' delimiter and the files or folders that match at least one of the patterns are excluded.
File Owner	String	Sets a new Unix file owner and group for the destination. Relevant if the Preserve Unix Mode Permissions is set. Usages: - only owner (for example - 'user1') - only group (for example - ':group1') - owner & group(for example - 'user1:mygroup') Left empty an attempt is made to preserve the owner from the source file or folder.
Include Hidden Files	Boolean	Hidden files or folders as well as files under hidden folders are not copied if the parameter is False.
Include Pattern	String	The wildcard or regular expression that defines the files or folders that are included. Use wildcard or regular expression according to the selected pattern type. You can place several patterns using the ';' delimiter - files or folders matching at least one of the patterns will be included.
Include/Exclude Pattern Type	PatternType	Specifies whether the include and exclude patterns are regular expressions or wildcard expressions.

Name	Type	Description
Overwrite	Boolean	Whether to overwrite the file or folder if it exists in the destination path. If file or folder exists and the flag is set to False, the action fails.
Preserve Unix Mode Permissions	Boolean	Preserve Unix Mode Permissions of the destination

Put File or Folder in Remote Agent (for Legacy OS)

Copies a file or folder from the local agent to a path on the specified remote agent.
The destination is the new location and name of the file or folder. If the Destination path does not exist it is created with any missing parent folders. If the Destination path exists the action responds as follows:

If source and destination are files the destination is replaced with the source.

If source and destination are folders the source and destination are merged with the source taking precedence.

If a file is attempted to be replaced with a folder or vice versa the action fails.

Input Values

Name	Type	Description
*** Agent Ip**	String	The IP address of the remote agent.
*** Destination Path**	String	The path of the destination file or folder on the remote computer. CA Technologies recommends to use '//' as a file separator.
*** Source Path**	String	The path of the source file or folder on the local computer. CA Technologies recommends to use '//' as a file separator.
File Owner	String	Sets new Unix file owner and group for the destination. Relevant if the Preserve Unix Mode Permissions is set. Usages: - only owner (for example - 'user1') - only group (for example - ':group1') - owner & group(for example - 'user1:mygroup') Left empty an attempt is made to preserve the owner from the source file or folder.
Overwrite	Boolean	

Name	Type	Description
		Whether to overwrite the file or folder if it exists in the destination path. If file or folder exists and the flag is set to False, the action fails.
Preserve Unix Mode Permissions	Boolean	Preserve Unix Mode Permissions of the destination

Read String from Text File

Reads a specified text file. Used to read the file contents in a String Parameter. The file size should not exceed 100 KB.

Input Values

Name	Type	Description
*** Filename**	String	The file name to read.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Output Values

Name	Type	Description
Result String	String	The result string.

Replace Text in File

Replaces a string in a text file. The step fails if either the file does not exist or the text is not found. Files that are less then 10 MB are supported.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file to replace in.
*** Text To Find**	String	The text to find in the file. To search using Regular Expressions set the Is Regular Expression input to True. Refer to http://www.regular-expressions.info/ for more details.
*** Text To Replace**	String	The text to replace the occurrences of Text To Find input in the file.

Name	Type	Description
Escape Replacement	Boolean	Set to False if the replacement text is already escaped in case of regular expression.
Fail If Text Not Found	Boolean	Whether to fail the action if the text is not found.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://docs.oracle.com/javase/6/docs/technotes/guides/intl/encoding.doc.html
Is Regular Expression	Boolean	Whether to search using Regular Expressions. if True, the Text To Find input is considered as a Regular Expression.

Replace Text in Multiple Files

Replaces a string in a many text files. The step fails if one of the files does not exist. Files that are less then 10 MB are supported.

Input Values

Name	Type	Description
*** Files Paths**	String[]	The paths to the files.
*** Text To Find**	String	The text to find in the every file. To search using Regular Expressions set the Is Regular Expression input to True. Refer to http://www.regular-expressions.info/ for more details.
*** Text To Replace**	String	The text to replace the occurrences of Text To Find input in the files.
Escape Replacement	Boolean	Set to False if the replacement text is already escaped in case of regular expression.
File Encoding	String	The encoding of the file.
Is Regular Expression	Boolean	Indicates whether to search using Regular Expressions, if true the Text To Find input will be considered as a Regular Expression.
May Not Contain The Text	Boolean	Whether step fails if text is not found. Set to True if some of the files do not contain the text to replace.

Synchronize Two Folders

Synchronizes two folders. The synchronization is done by copying files that are needed from the first folder to the second. The needed files are decided according to the selected strategy.

Important! FULL_COMPARISON and IGNORE_EXTRA_FILES_IN_FIRST causes extra files or folders in second folder to be deleted.

Strategies: FULL_COMPARISON = Fully synchronized the two folder and at the end they are identical. IGNORE_EXTRA_FILES_IN_FIRST = New files or folders in first folder are not copied and extra files in second file are deleted. IGNORE_EXTRA_FILES_IN_SECOND = New files or folders in first are copied and extra files in second are not deleted. IGNORE_EXTRA_FILES_IN_BOTH = New files or folders in first are not copied and extra files in second are not deleted.

Input Values

Name	Type	Description
*** First Folder Path**	String	First folder path.
*** Second Folder Path**	String	Second folder path.
Copy Different Files	Boolean	If set to True, the files that differ between the two folders are copied from the first folder to the second. If set to False, that files that exist in both folders are not copied even if they differ
Exclude Filters	String	Folder Exclude filters.
Filters Separator	String	Indicates the character to use as filters separator.
Folders Comparison Strategy	FoldersComparisonStrategy	Indicates the comparison strategy to select. Whether to consider extra files on the compared folders or not, Important! FULL_COMPARISON and IGNORE_EXTRA_FILES_IN_FIRST causes extra files or folders in second folder to be deleted.
Include Filters	String	Folder include filters.

Output Values

Name	Type	Description
Changed Files List	String[]	The changed files list from the first folder.

Upload File (HTTP)

Uploads a file to a specified url using HTTP/S.

Input Values

Name	Type	Description
*** File Path To Upload**	String	The path of the file that is uploaded.
*** Url**	String	the HTTP or HTTPS URL to access.
Authentication Method	AuthMethod	Set to one of the following methods: Basic, Digest or NTLM. These are used to authenticate with http servers or proxies. NONE means no authentication is required.
Connection Timeout (seconds)	Integer	Determines the seconds in seconds until a connection is established. the default value is 1 minute. A value of zero means the timeout is not used.
Domain	String	The servers domain. Relevant if the authentication method is not NONE.
Expected Code	Integer	The expected HTTP return code. The action fails if an expected code is not returned. Left empty accepts any code in the range 200-299
Method	HttpPutPostMethodTypes	The method for the HTTP Request. Post or Put.
Password	Password	The users password. Relevant if the authentication method is not NONE.
Proxy Server Name	String	The name of the proxy server.
Proxy Server Port	Integer	The port of the proxy server.
Read Timeout (seconds)	Integer	Defines the default socket timeout in seconds which is the timeout awaiting data. A timeout value of zero is interpreted as an infinite timeout.
Realm	String	The realm that authenticates the user. Relevant if the authentication method is BASIC or DIGEST.
Use Preemptive Authentication	Boolean	Whether to use Preemptive Authentication when using the user name to login.
User Name	String	The user name for authentication. Relevant if the Authentication Method is not NONE.

Output Values

Name	Type	Description
HTTP Response Code	int	HTTP return code. -1 is returned if no HTTP connection is made.

ZIP Many Files

Creates a ZIP file from the source paths. The action fails if the source paths do not exist.

Input Values

Name	Type	Description
*** Source Paths**	String[]	The path to the source where the contents of the ZIP file is created from.
*** Zip File Path**	String	The path to the ZIP file to create.
Files Filter	String	The filter to use to filter the files.
Include Root Folder	Boolean	Whether to include the root folder as part of the new Zip file
Include Sub Folders	Boolean	Whether to include sub folders. Empty folders are included as well.
Regular Expression Filter	Boolean	Whether the files filter is a regular expression.

Installation Actions

Contents

- Run MSI Installer
- Uninstall RPM Package

Check if RPM Package is Installed

Checks whether the specified package is listed in the RPM packages database.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** Package Name**	String	Specifies the name package name to check.
Rpm Executable Path	String	Specifies the path to the RPM executable.

Output Values

Name	Type	Description
Was Installed	Boolean	Indicates whether the RPM was already installed. True for installed, False for not installed.

Install RPM Package

Installs an RPM package.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** RPM Package Path**	String	Specifies the path to the RPM package.
Rpm Executable Path	String	Specifies the path to the RPM executable

Output Values

Name	Type	Description
RPM Package Name	String	The RPM package name of the package contained in the specified RPM file.

Name	Type	Description
		If the package installation is successful, the value of this output is the name of the installed package.

Install Windows Service

Installs and configures a Windows Service.

Note: The action does not start the service.

Input Values

Name	Type	Description
*** Executable Path**	String	Specifies the full path of the executable to install as a service. **Note:** The action does not validate that the executablePath exists.
*** Service Display Name**	String	Specifies the service display name of service.
*** Service Name**	String	Specifies the service name of service..
Password	Password	Specifies the password for the start account.
Start Account	String	Specifies the start account for the service. Provided the value in the following form of DomainName\UserName. **Note:** For local users use: .\LocalUser and for local system account leave blank.
Startup Type	ServiceStartupType	Specifies the service startup type.

Retrieve Installed RPM Package Name by RPM File

Retrieves the RPM package name from an RPM file.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** RPM File Path**	String	Specifies the path to RPM file.

Name	Type	Description
Rpm Executable Path	String	Specifies the path to the RPM executable

Output Values

Name	Type	Description
RPM Package Name	String	The package name of the RPM file.

Run InstallAnywhere

Runs an InstallAnywhere installer.

Use List Keys and List Values to set the properties of the installer. For example, a key would be PORT_NUMBER and a matching value would be 8080. The Log File Directory and Log File Name inputs require valid values.

Important! The installer must support silent installation.

Note: Some InstallAnywhere installers display a Progress Dialog. Do not close this dialog.

Input Values

Name	Type	Description
*** Installer Path**	String	Specifies the path to the InstallAnywhere setup file. The input also accepts File parameters.
*** Log File Directory**	String	Specifies the directory for the log file.
*** Log File Name**	String	Specifies the name of the log file.
Is XML Log Format	Boolean	Indicates whether to generate an XML Log file or Text Log file. True for XML, False for text. **Note:** The type of the Log file that the installer generates depends on the version of InstallAnywhere that was used to build the installer. For installers that were built with InstallAnywhere Studio version 8 and above, use XML Log. For lower versions of InstallAnywhere use Text Log. If

Name	Type	Description
		the Is XML Log Format input is set to incorrect value then the installer runs successfully but the step may fail.
List Keys	String[]	Specifies the list of property keys. Leave the value empty if no properties are necessary.
List Values	String[]	Specifies the list of property values. Leave the value empty if no properties are necessary.

Output Values

Name	Type	Description
Generated Log File Full Path	String	The full path to the generated log file.

Run InstallAnywhere with Predefined Answer File

Runs an InstallAnywhere installer with a predefined answer file.

By default, the installer looks for the answer file in the same directory as the installer. The answer file name must be installer.properties. The properties in the installer.properties file must be in key=value format, for example, PORT_NUMBER=8080. The answer file requires the property INSTALLER_UI=silent to force the installer to run in silent mode.

Important! The installer must support silent installation.

Note: Some InstallAnywhere installers display a Progress Dialog. Do not close this dialog.

Input Values

Name	Type	Description
*** Installer Path**	String	Specifies the path to the InstallAnywhere setup file. The input also accepts File parameters..
*** Log File Directory**	String	Specifies the directory for the log file.
*** Log File Name**	String	Specifies the name of the log file.
Is XML Log Format	Boolean	Indicates whether to generate an XML Log file or Text Log file. True for XML, False for text. **Note:** The type of the Log file that the installer generates depends on the version of InstallAnywhere that was used

Name	Type	Description
		to build the installer. For installers that were built with InstallAnywhere Studio version 8 and above, use XML Log. For lower versions of InstallAnywhere use Text Log. If the Is XML Log Format input is set to incorrect value then the installer runs successfully but the step may fail.

Output Values

Name	Type	Description
Generated Log File Full Path	String	The full path to the generated log file.

Run InstallShield with Answer File

Runs an InstallShield installer with an answer file.

Note: For details about how to create the required answer file, refer to the InstallShield documentation.

Input Values

Name	Type	Description
*** Answer File Path**	String	Specifies the path to the answer file.
*** Installer Path**	String	Specifies the path to the InstallShield setup file.

Output Values

Name	Type	Description
Generated Log File Path	String	The path to the log file that installer creates.

Run MSI Installer

Runs an MSI installer with the specified property keys and property value.

Note: The MSI installer is run in two execution modes, Install or Uninstall.

Input Values

Name	Type	Description
*** Installer Path**	String	Specifies the path of the MSI Installer file.
List Keys	String[]	Specifies the list of property keys. Leave the value empty if no properties are necessary.
List Values	String[]	Specifies the list of property values. Leave the value empty if no properties are necessary.
Log File Details Level	String	Specifies the log details level. Options are /L[i][w][e][a][r][u][c][m][p][v][+][!][*]. If the value is empty the installer does not create a Log file .
Log File Path	String	Specifies the path to the log file that the installer creates.
MSI Execution Mode	RunMSIInstall$Mode	Specifies the MSI Installer execution mode, Install or Uninstall.

Output Values

Name	Type	Description
Exit Code	int	The installation exit code.

Uninstall RPM Package

Uninstalls an RPM package.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** RPM Package Name**	String	Specifies the name of the RPM Package.
Rpm Executable Path	String	Specifies the path of the RPM executable

JMX Actions

Contents

- Activate JMX

Activate JMX

Invokes an operation on an existing MBean with or without objects.

If the remote connector uses SSL, set Is SSL to true and specify a valid Trust Store Path and Keystore Path.

Input Values

Name	Type	Description
*** JMX Name**	String	Specifies the JMX MBean to activate. A lookup for the MBean name is performed before the MBean is activated.
*** Operation Name**	String	Specifies the name of the method (Operation) in the specified MBean to activate.
*** Server URL**	String	Specifies the JMX server host name. Add the character '/' in front of JNDI references, for example: /jndi/rmi://DOMAIN
Base Dir Path	String	Specifies the JMX base directory path to use in the JMX server connection string. For example, in the connection string: service:jmx:rmi://myjmxhost:8080/jmxrmi
Is SSL	boolean	Indicates if the connection to the JMX server should use SSL.
Keystore Password	Password	Specifies the key store password to use for the JMX server connection. The Keystore Password applies only if Is SSL is set to true.
Keystore Path	String	Specifies the path to the key store to use for the JMX server connection. The Keystore Path applies only if Is SSL is set to true.
Operation Parameter Types	String[]	Specifies an array of the types of values specified in the Operation Parameters input.

Name	Type	Description
		Specify the types in the same order as the Operation Parameters.
Operation Parameters	Serializable[]	Specifies an array of values to pass to the method specified in the Operation Name input. To pass an object array, use an Object Array Parameter.
Password	Password	Specifies the user password to login to the JMX Server. If no login is required, leave the Password blank.
Port	Integer	Specifies the JMX server port number to use for the JMX server connection.
Trust Store Password	Password	Specifies the trust store password to use for the JMX server connection. The Trust Store Password applies only if Is SSL is set to true.
Trust Store Path	String	Specifies the path to the trust store to use for the JMX server connection. The Trust Store Path applies only if Is SSL is set to true.
Username	String	A user name to use to login to the JMX Server. If no login is required, leave the Username blank.

Output Values

Name	Type	Description
Return Object	Object	The return value of the method (Operation) specified after invocation. If the action fails the output returns a null. Use this output with an Object Parameter.

JSON Actions

Contents

JSON - Extract Properties from a JSON Object

Extracts properties from a JSON object. This action can extract up to 3 strings and up to 3 arrays from the top level of a JSON object. The extracted strings or arrays can be used to capture child JSON objects.

Input Values

Name	Type	Description
*** JSON Object**	String	Specifies the JSON object.
Array Property 1	String	Specifies the name of a top level property which contains an array of values. The result appears in Array Property Value 1 in an array format.
Array Property 2	String	Specifies the name of a top level property which contains an array of values. The result appears in Array Property Value 2 in an array format.
Array Property 3	String	Specifies the name of a top level property which contains an array of values. The result appears in Array Property Value 3 in an array format.
Property 1	String	Specifies the name of a top level property which contains a string value. The result appears in Property Value 1 in a String format.
Property 2	String	Specifies the name of a top level property which contains a string value. The result) appears in Property Value 2 in a String format.
Property 3	String	Specifies the name of a top level property which contains a string value. The result appears in Property Value 3 in a String format.

Output Values

Name	Type	Description
Array Property Value 1	String[]	The first output of an Array property.
Array Property Value 2	String[]	The second output of an Array property.
Array Property Value 3	String[]	The third output of an Array property.
Property Value 1	String	The first output of a String property.
Property Value 2	String	The second output of a String property.
Property Value 3	String	The third output of a String property.

Math Actions

Contents

Calculate Two Numbers

Calculates an operation with two number.

Input Values

Name	Type	Description
*** First Number**	Number	Specifies the first number.
*** Operation**	ArithmeticOperation	Specifies the arithmetic operation.
*** Second Number**	Number	Specifies the second number.

Output Values

Name	Type	Description
Result As Double	Double	The result as Double.
Result As Float	Float	The result as Float.
Result As Integer	Integer	The result as Integer.

Name	Type	Description
Result As Long	Long	The result as Long.

Create Random Number

Creates a Random number according to specified inputs.

Input Values

Name	Type	Description
Int Between Zero To This	int	Specifies the limit for the random number. If the result is an int, the output value is between zero and the value. Leave empty to return any integer.

Output Values

Name	Type	Description
Random Boolean	Boolean	A random boolean.
Random Double	Double	A random double.
Random Float	Float	A random float.
Random Gaussian	Double	A random Gaussian (double).
Random Int	Int	A random integer.
Random Long	Long	A random double.

Network Actions

Contents

Check Free Ports

Checks if ports are free.

If any of ports are not free, the step result fails.

The Array Of Ports list are specified by array of integers or a comma-separated list.

The contents of both inputs (Comma-Separated Ports input and Array Of Ports input) are checked, and at least one of the inputs must be filled.

Input Values

Name	Type	Description
Array Of Ports	Integer[]	Array of ports, can be used with an Integer Array Parameter. The input can be empty if the Comma-Separated Ports is used
Check As UDP Ports	Boolean	Indicates whether the checked ports are UDP (True) or TCP (False).
Comma Separated Ports	String	Comma-separated list of ports to check, for example: 8080,8081,8082. The input can be empty an string if the Array Of Ports is used.
Expect Ports To Be Free	Boolean	Indicates whether the step should expect the ports to be free or taken. True for free or False for taken.

Disconnect Network Drive

Disconnects a network mapped drive. The action is for Windows only.

Input Values

Name	Type	Description
*** Local Drive Letter**	String	The local drive letter to be disconnected

Get IPs of Localhost

Retrieves the IP assigned to the current computer.

Input Values

Name	Type	Description
Starts With	String	

Name	Type	Description
		Selects the IP address that starts with the string (e.g. 192.168.0), leave empty for any.

Output Values

Name	Type	Description
Ip	String	Contains the first IP address returned of the current computer or the one that matches the look up string.
Ips	String[]	Contains the list of IPs found for the current computer.

Map Network Drive

Maps a local drive to a network path. The action is for Windows only. The drive is mapped inside the CA environment. CA Technologies recommends to disconnect the drive at the end of flow.

Input Values

Name	Type	Description
*** Local Drive Letter**	String	The local drive letter to be mapped.
*** Network Path**	String	The network full path to be mapped.
Password	Password	The remote computer password.
User Name	String	The remote computer username. For domain users please use Domain\username

Ping a Device

Pings a device (Host, Network Switch, Storage Unit, etc).

Input Values

Name	Type	Description
*** Device name**	String	The target server name or IP
Time out	Integer	The maximum time in seconds to wait for a response

Output Values

Name	Type	Description
Reply time	Long	Round-trip time in milliseconds. Is null if the ping fails

Telnet Session

Provides the ability to create a Telnet session to authenticate and send telnet commands to the remote Telnet server.

Input Values

Name	Type	Description
*** Telnet Commands**	String[]	The telnet Commands to execute once connected.
*** Telnet Host**	String	The Telnet Host name.
Login Prompt	String	The prompt to expect as the log in once connected.
Login Successfull Prompt	String	The prompt to expect as successful log in once connected.
Password Prompt	String	The prompt to expect as the password once connected.
Prompt	String	The prompt to expect before each command is sent.
Telnet Port	Integer	The Telnet port to use.
User Name	String	The user name to log in with to the Telnet host.
User Password	Password	The user password to use with the log in user name.

Output Values

Name	Type	Description
Telnet output	String	The output of the Telnet commands.

OS Actions

Contents

- Get Java System Property
- Linux / Unix Actions
  - Check if RPM Package is Installed
  - Check if User has Root Credentials
  - Get Unix / Linux OS Details
  - Install RPM Package
  - Retrieve Installed RPM Package Name by RPM File
  - Uninstall RPM Package
- Windows Actions
  - Change Credentials for Windows Service
  - Check if Service Exists
  - Check if Windows Registry Folder or Key Exists
  - Check Service Status
  - Create a Windows Registry Key
  - Delet a Windows Service
  - Delete Windows Registry Folder
  - Delete Windows Registry Key
  - Get Microsoft Windows OS Details
  - Get Windows Registry Key Value
  - Get Windows Service Properties
  - Install Windows Service
  - Modify Windows Service Properties
  - Restart Windows Service
  - Run MSI Installer
  - Run Windows Process
  - Start Windows Service
  - Stop Windows Service
  - Update Environment Variable
  - Update Windows Registry Key

- INI File Actions
  - Get Value from an INI File
  - Manipulate INI File

Check OS Type

Checks for the running OS Type and matches it to the expected OS Type value.

Input Values

Name	Type	Description
*** Expected OS Type**	OSType	OS Type value to compare against the running OS.

Output Values

Name	Type	Description
Actual OS Type	OSType	The actual OS Type.
Actual OS Type As String	String	String representation of OS Type.

Get Environment Variable Value

Retrieves the environment variable value. If the environment variable with the specified name is not defined, the step fails.

Input Values

Name	Type	Description
*** Variable Name**	String	The name of the environment variable.

Output Values

Name	Type	Description
Variable Value	String	The value of the environment variable.
Was Found	Boolean	Indicates whether the variable was found.

Get Java System Property

Used to set a java system property to a parameter.

Input Values

Name	Type	Description
*** Property Name**	String	The property name.
Fail If Not Exist	Boolean	Whether to fail if the property does not exist.

Output Values

Name	Type	Description
Property Value	String	The property value.

Linux / Unix Actions

Check if RPM Package is Installed

Checks whether the specified package is listed in the RPM packages database.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** Package Name**	String	Specifies the name package name to check.
Rpm Executable Path	String	Specifies the path to the RPM executable.

Output Values

Name	Type	Description
Was Installed	Boolean	Indicates whether the RPM was already installed. True for installed, False for not installed.

Check if User has Root Credentials

Checks if the user credentials belongs to the root group. Supports only Unix/Linux systems.

Input Values

Name	Type	Description
*** Password**	Password	The user password to verify the root rights.
*** User Name**	String	The user name to verify the root rights.

Get Unix / Linux OS Details

Retrieves the various parameters of the underlying Unix/Linux and similar OS.The action relies on the "uname" system utility.

Output Values

Name	Type	Description
Is 32bit	Boolean	Returns True if the underlying platform is 32bit, False if it's 64bit
OS Name	String	Underlying OS name as reported by uname -s
Os Platform	String	Underlying OS platform as reported by uname -m
Os Version	String	Underlying OS release (version) as reported by uname -r

Install RPM Package

Installs an RPM package.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** RPM Package Path**	String	Specifies the path to the RPM package.
Rpm Executable Path	String	Specifies the path to the RPM executable

Output Values

Name	Type	Description
RPM Package Name	String	The RPM package name of the package contained in the specified RPM file. If the package installation is successful, the value of this output is the name of the installed package.

Retrieve Installed RPM Package Name by RPM File

Retrieves the RPM package name from an RPM file.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** RPM File Path**	String	Specifies the path to RPM file.
Rpm Executable Path	String	Specifies the path to the RPM executable

Output Values

Name	Type	Description
RPM Package Name	String	The package name of the RPM file.

Uninstall RPM Package

Uninstalls an RPM package.

Note: The action supports only systems that support RPM packages.

Input Values

Name	Type	Description
*** RPM Package Name**	String	Specifies the name of the RPM Package.
Rpm Executable Path	String	Specifies the path of the RPM executable

Windows Actions

Change Credentials for Windows Service

Changes the start account and password of the Windows service.

Input Values

Name	Type	Description
*** Service Name**	String	The service name or display name of service to start. The name specified is case-insensitive but must be the exact service name or the exact display name.
*** Start Account**	String	The start account for the service. Provided in the form of DomainName\UserName. For a local users use: .\LocalUser and for a local system account use: LocalSystem.
Password	Password	The password for the start account.

Name	Type	Description
Startup Type	ServiceStartupType	The service startup type.

Check if Service Exists

Checks if a certain service exists. Supported only on Windows.

Input Values

Name	Type	Description
*** Service Name**	String	The name of the service you want to query.
Fail If Not Exist	Boolean	Set to "True" if you want the action to fail if the service does not exist. Set to "False" otherwise.

Check if Windows Registry Folder or Key Exists

Checks if a specific folder or key exists in the Windows registry.

Input Values

Name	Type	Description
*** Folder Path**	String	The path to the folder in the Windows registry. Specify the path without the root hive name. For example : "SOFTWARE\Test" .
Key Name	String	The key name in Windows registry.
Root Hive Name	RootRegistryHive	The registry root hive name.
Should Exist	Boolean	Whether to expect the folder or key to exist or not, True for exist or False for not exist.

Check Service Status

Checks whether a certain service is in a certain status. Supported only on Windows.

Input Values

Name	Type	Description
*** Expected Status**	ServiceStatus	The Expected status of the service.
*** Service Name**	String	The name of the service you want to query.

Create a Windows Registry Key

Creates a Windows Registry Key. Supported only on Windows.

Input Values

Name	Type	Description
*** New Key Name**	String	The new registry key name to create.
*** New Key Value**	String	The value to assign to the new registry key.
*** Parent Folder Path**	String	The path to the folder that the target registry key is created in.
If Exists Action Mode	CreateRegistryKey$AMode	Indicates what action to take if the key already exists.
New Key Type	String	The type of registry key to create. Can be either: 'SZ', 'BINARY' or 'DWORD'.
Root Hive Name	CreateRegistryKey$RootRegistryHive	The registry root hive name.
Should Create Parent Folder Path	Boolean	Whether to create the full path if it does not exist.

Delet a Windows Service

Deletes a windows service. If the service is not stopped it is marked for deletion and is deleted once it is stopped, Supported only on Windows.

If the service is not stopped or another process has an active reference to it, it is marked for deletion and deleted only once it stops and no other process has an active reference to it.

Input Values

Name	Type	Description
*** Service Name**	String	The service name of the service to be deleted.
Fail If Not Immediately Deleted	Boolean	Indicates if the action should fail if the deletion is not immediate.

Delete Windows Registry Folder

Deletes a folder in the Windows registry.

Input Values

Name	Type	Description
*** Folder Path**	String	The path to the folder to delete.
Root Hive Name		The registry root hive name.

Name	Type	Description
	CreateRegistryKey$RootRegistryHive	

Delete Windows Registry Key

Deletes a Windows Registry folder.

Input Values

Name	Type	Description
*** Folder Path**	String	The path of the folder that contains the key to delete.
*** Key Name**	String	The registry key name to delete.
Fail If Not Exist	Boolean	Whether to fail if the key does not exist. True to fail or False to ignore.
Key Type	String	Type of registry key to delete. The valid types are SZ, BINARY and DWORD.
Root Hive Name	CreateRegistryKey$RootRegistryHive	The registry root hive name.

Get Microsoft Windows OS Details

Retrieves various parameters of the underlying Microsoft Windows OS.

Output Values

Name	Type	Description
Is 32bit	Boolean	Returns True if the underlying platform is 32bit, false if it is 64bit.
Is Windows 2000	Boolean	Returns True if underlying OS is Microsoft Windows 2000
Is Windows 2003	Boolean	Returns True if underlying OS is Microsoft Windows 2003
Is Windows 2008	Boolean	Returns True if underlying OS is Microsoft Windows 2008
Is Windows 7	Boolean	Returns True if underlying OS is Microsoft Windows Vista
Is Windows Vista	Boolean	Returns True if underlying OS is Microsoft Windows Vista
Is Windows XP	Boolean	Returns True if underlying OS is Microsoft Windows XP
Service Pack	String	Returns the service pack information.
Service Pack Number	int	

Name	Type	Description
		Returns the service pack number if present, -1 if not present

Get Windows Registry Key Value

Locates and retrieves the Registry Key value.

Input Values

Name	Type	Description
*** Key Name**	String	The registry key value that needs to be retrieved.
*** Parent Folder Path**	String	The path to the folder that the target registry key is in.
*** Root Hive Name**	CreateRegistryKey$RootRegistryHive	The registry root hive name.
Expected Value	String	A value to expect in the key, the input can be used to either expect a certain value to exist or to expect a certain value not to exist.
Is Value Expected	Boolean	Whether the registry key should exist and its value matches the expected value, or the key should not exist or its value should not match the expected value, True to exist, False does not exist.
Key Type	String	Type of registry key to retrieve. Valid values are : SZ, BINARY and DWORD.

Output Values

Name	Type	Description
Retrieved Value	String	The value of the registry key.

Get Windows Service Properties

Gets an existing Windows service properties: display name, description, and such.

Input Values

Name	Type	Description
*** Service Name**	String	The service name to modify.

Output Values

Name	Type	Description
Binary Path	String	The service binary path, leave blank for unchanged value.
Interactive	Boolean	Whether to set the interactive flag, Relevant only if the LocalSystem account is used, leave blank for unchanged value.
Service Description	String	The service description, leave blank for unchanged value.
Service Display Name	String	The service display name, leave blank for unchanged value.
Service Startup Type	ServiceStartupType	The service startup mode, leave blank for unchanged value.
User Name	String	The service log on user name. For domain users use the DOMAIN\USERNAME syntax or use the .\USERNAME syntax, leave blank for unchanged value.

Install Windows Service

Installs and configures a Windows Service.

Note: The action does not start the service.

Input Values

Name	Type	Description
*** Executable Path**	String	Specifies the full path of the executable to install as a service. **Note:** The action does not validate that the executablePath exists.
*** Service Display Name**	String	Specifies the service display name of service.
*** Service Name**	String	Specifies the service name of service..
Password	Password	Specifies the password for the start account.
Start Account	String	Specifies the start account for the service. Provided the value in the following form of DomainName\UserName.

Name	Type	Description
		Note: For local users use: .\LocalUser and for local system account leave blank.
Startup Type	ServiceStartupType	Specifies the service startup type.

Modify Windows Service Properties

Enabled to modify an existing windows service's properties: display name, description and so on. Supported OS: Windows.

Input Values

Name	Type	Description
*** Service Name**	String	The service to modify.
Binary Path	String	The service binary path, leave blank for unchanged value.
Command To Run	String	The command to run when performing the Run Command Action, leave blank for unchanged value.
First Failure Action	ServiceFailureAction	The service First Failure Action, leave blank for unchanged value.
Interactive	Boolean	Indicates whether to set the interactive flag, this field is relevant only if the LocalSystem account is used, leave blank for unchanged value.
Local System Account	Boolean	Indicates whether to use the LocalSystem account as the logon account, leave blank for unchanged value.
Password	Password	The service logon user password, leave blank for unchanged value.
Reset Period Days	Integer	The time in days to wait before reseting the failure counter, leave blank for unchanged value.
Restart Computer Delay Minutes	Integer	The time in minutes to wait before performing the Restart Computer Action, leave blank for unchanged value.
Restart Computer Message	String	The message to send to the network before performing the Restart Computer Action, leave blank for unchanged value.

Name	Type	Description
Restart Service Delay Minutes	Integer	The time in minutes to wait before performing the Restart Service Action, leave blank for unchanged value.
Second Failure Action	ServiceFailureAction	The service Second Failure Action, leave blank for unchanged value.
Service Description	String	The service description, leave blank for unchanged value.
Service Display Name	String	The service display name, leave blank for unchanged value.
Service Startup Type	ServiceStartupType	The service startup mode, leave blank for unchanged value.
Subsequent Failure Action	ServiceFailureAction	The service Subsequent Failure Action, leave blank for unchanged value.
User Name	String	The service logon user name, for domain users use the DOMAIN\USERNAME syntax otherwise use the .\USERNAME syntax, leave blank for unchanged value.

Restart Windows Service

Restarts a Windows service by its service name or display name.

Input Values

Name	Type	Description
*** Service Name**	String	The service name or display name of the service to stop. The name is case-insensitive but must be the exact service name or exact display name.

Run MSI Installer

Runs an MSI installer with the specified property keys and property value.

Note: The MSI installer is run in two execution modes, Install or Uninstall.

Input Values

Name	Type	Description
*** Installer Path**	String	Specifies the path of the MSI Installer file.
List Keys	String[]	Specifies the list of property keys. Leave the value empty if no properties are necessary.

Name	Type	Description
List Values	String[]	Specifies the list of property values. Leave the value empty if no properties are necessary.
Log File Details Level	String	Specifies the log details level. Options are /L[i][w][e][a][r][u][c][m][p][v][+][!][*]. If the value is empty the installer does not create a Log file .
Log File Path	String	Specifies the path to the log file that the installer creates.
MSI Execution Mode	RunMSIInstall$Mode	Specifies the MSI Installer execution mode, Install or Uninstall.

Output Values

Name	Type	Description
Exit Code	int	The installation exit code.

Run Windows Process

Executes the specified executable. To start the process in the background and not wait for it to finish its execution, set the Wait For Process To Finish input to False.

Input Values

Name	Type	Description
*** Executable Path**	String	The path to the executable to run.
*** Work Directory**	String	The working directory for the executable.
Environment Variables Names	String[]	A list of environment variables names to be set as the environment variables for the created process.
Environment Variables Values	String[]	A list of environment variables values to be set as the environment variables for the created process, The values should match the name order in the environment Variables Names input.
Executable Arguments	String[]	The list of arguments to pass to the executable.
Std Err File Path	String	The file path to use to write the error stream of the process.
Std Out File Path	String	The file path to use to write the output stream of the process.

Name	Type	Description
Wait For Process To Finish	Boolean	Whether to wait for the process to finish its execution or not. True to wait or False to not wait. On non-windows systems the step will fail if this input is set to False.
Wrap Arguments	Boolean	Whether to wrap the process arguments as one argument, True for wrap or False to leave as is. Used for avoiding where one of the arguments contains quotation mark(s) '"'.

Output Values

Name	Type	Description
Return Value	long	Return value of the finished process. Long Parameter can be used with this output.

Start Windows Service

Starts a Windows service by its service name or display name.

Input Values

Name	Type	Description
*** Service Name**	String	The service name or display name of service to start, The name is case-insensitive but must be the exact service name or exact display name.
Fail If Not Found	Boolean	Whether the action should fail if the service is not found.
Time To Wait For Service	Long	The time in seconds to wait to check the service status after the signal was sent.

Stop Windows Service

Stops a windows service by its service name or display name.

Input Values

Name	Type	Description
*** Service Name**	String	The service name or display name of service to start. The name is case-insensitive but must be the exact service name or exact display name.

Name	Type	Description
Fail If Not Found	Boolean	Whether the action should fail if the service is not found.
Time To Wait For Service	Long	The time in seconds to wait to check the service status after the signal was sent.

Update Environment Variable

Updates the environment variables. In POSIX system updates only the process environment. In Windows it also updates the system environment.

Input Values

Name	Type	Description
*** Variable Name**	String	The name of the environment variable to update.
*** Variable Value**	String	The new value for the environment variable.
Create If Not Exists	Boolean	Whether to create the environment variable if it does not exist.

Update Windows Registry Key

Updates the registry key value.

Input Values

Name	Type	Description
*** Key Name**	String	The registry key whose value needs to be updated.
*** New Key Value**	String	The value to update the specified registry key with.
*** Parent Folder Path**	String	The path to the folder that the target registry key is in.
*** Root Hive Name**	CreateRegistryKey$RootRegistryHive	The registry root hive name.
Create If Not Exist	Boolean	Whether to create the specified registry key if it does not exist.
Key Type	String	Type of registry key to retrieve. Valid values are: SZ, BINARY and DWORD.

Output Values

Name	Type	Description
New Key Value	String	The value to update the specified registry key with.

INI File Actions

Get Value from an INI File

Retrieves a value for a specific key in a section of an INI file. If the ini section is left empty, and there are other attributes under no section, the attributes are added to the head of the file. Otherwise the attributes are added to the last section. The separator between the keys is: "="

Input Values

Name	Type	Description
*** INI File Path**	String	The path to the INI file.
*** INI Key Name**	String	The name of the key in the file.
INI Section	String	The name of the INI section that contains the keys.

Output Values

Name	Type	Description
INI Key Value	String	The retrieved value of the key

Manipulate INI File

Adds or updates an INI file. You can update/add keys that belong to a specific section, if the ini section is left empty. If there are other attributes under no section the attributes are added to the head of the file, otherwise the attributes are added to the last section. The separator between the keys is: "="

Input Values

Name	Type	Description
*** INI File Path**	String	The path to the INI file.
*** Key Value Array**	String[]	An array of strings, each string is in the format of 'key=value'.
INI Section	String	The name of the INI section that contains the keys.

Parameter Actions

Contents

- Arrays - Get Array Length
- Arrays - Get Index of a Value in an Array
- Arrays - Initialize Array
- Arrays - Remove Value from Array
- Arrays - Update Vaue in Array (String)
- Convert String to Number
- Extract Column From ResultSet
- Extract Multi Values From ResultSet
- Extract Value From ResultSet
- Set Parameter Value - Boolean
- Set Parameter Value - Integer
- Set Parameter Value - String
- User Input - Ask For Parameter Value
- User Input - Choose from Available Options

Arrays - Add a Value to an Array

Adds a value to an array at a specified index (starting from 1).

Input Values

Name	Type	Description
*** Input Array**	Object[]	The array to add the value to. Must be the same type as the output array.
*** Value**	Object	The value to add. Must be the same type as the input array's element type.
Index	Integer	The index to insert the value to in the output array. The first element in the array is at index 1. Left blank the value is appended to the end of the array.

Output Values

Name	Type	Description
Output Array	Object[]	

Name	Type	Description
		The array with the added value. Must be the same type as the input array.

Arrays - Extract Value From an Array

Extracts a value from an array by the specified array index.

Important! The index is 1-based index, so for the first value in an array you should specify index .1. If the array does not have the index specified the action fails. If the value is null an empty string is returned.

Input Values

Name	Type	Description
*** Array**	Object[]	The array to perform the operation on.
*** Value Index**	Integer	Index of the value to extract.

Output Values

Name	Type	Description
Extraction Result	String	The result of the extraction.

Arrays - Get Array Length

Returns the length of an array.

Input Values

Name	Type	Description
*** Array**	Serializable[]	The array

Output Values

Name	Type	Description
Length	int	The array length

Arrays - Get Index of a Value in an Array

Searches for a value in an array and returns the first index of the value if it is found. If the value is not found, -1 is returned.

Note: The first element of an array is at index 1.

Input Values

Name	Type	Description
*** Array**	Object[]	The array to perform operation on.
*** Value**	Object	Value to search for.

Output Values

Name	Type	Description
Value Index	Integer	The index of the value, or -1 if value is not present in the array.

Arrays - Initialize Array

Initialize an array to a clear empty array.

Output Values

Name	Type	Description
Array	Object[]	The array

Arrays - Remove Value from Array

Removes value from array at the specified index (starting from 1).

Input Values

Name	Type	Description
*** Index**	Integer	The index to remove the value from.
*** Input Array**	Object[]	The array to remove the value from. Must be the same type as the output array.

Output Values

Name	Type	Description
Output Array	Object[]	The array with the removed value. Must be the same type as the input array.

Arrays - Update Vaue in Array (String)

Updates a value in an array by the specified array index.

Note: The first element of an array is at index 1. If the array does not have the index specified the action fails.

Input Values

Name	Type	Description
*** Array**	String[]	The array to perform the operation on.
*** Index**	Integer	Index of the value to update.
*** New Value**	String	New value to set.

Output Values

Name	Type	Description
Array	String[]	The array to perform the operation on.

Convert String to Number

Convert a String to a Number. If the String can not be casted the action fails.

Input Values

Name	Type	Description
*** Number As String**	String	The number to convert. If the String does not represent a number the action fails.

Output Values

Name	Type	Description
Result As Double	Double	The result as Double.
Result As Float	Float	The result as Float.
Result As Integer	Integer	The result as Integer.
Result As Long	Long	The result as Long.

Extract Column From ResultSet

Extracts an Array from a ResultSet by column indexes.

Input Values

Name	Type	Description
*** Array**	Object[][]	The ResultSet to perform operation on.
*** Column**	Integer	Column of the value to extract. **Note:** Arrays have a one-based index so for first index is 1.

Output Values

Name	Type	Description
Extracted Column	Object[]	The specified column.

Extract Multi Values From ResultSet

Extracts values from a ResultSet (two dimensional array) by specifying several columns and row indexes. ResultSets use one-based index, so for the first cell specify column 1 and row 1. If the ResultSet does not have the column or row specified the action fails.

Input Values

Name	Type	Description
*** Columns**	Integer[]	Columns of the values to extract. ResultSets use one-based index, so the first index of the ResultSet is 1.
*** Result Set**	Object[][]	The ResultSet to perform operation on.
*** Row**	Integer	Row of the value to extract. ResultSets use one-based index, so the first index of the ResultSet is 1.
Fail On Null	Boolean	If True, the action fails if one of the extracted values is null or empty.

Output Values

Name	Type	Description
Value [1]	String	The result of the extraction (#1).
Value [2]	String	The result of the extraction (#2).
Value [3]	String	The result of the extraction (#3).
Value [4]	String	The result of the extraction (#4).
Value [5]	String	The result of the extraction (#5).
Value [6]	String	The result of the extraction (#6).
Value [7]	String	The result of the extraction (#7).
Value [8]	String	The result of the extraction (#8).
Value [9]	String	

Name	Type	Description
		The result of the extraction (#9).

Extract Value From ResultSet

Extracts a value from the ResultSet (two dimensional array) by specifying column and row indexes. Arrays have a one-based index so for first cell specify column 1 and row 1. If the array does not have the column or row specified the action fails.

Input Values

Name	Type	Description
*** Column**	Integer	Column of the value to extract. ResultSets use one-based index, so for first index of the ResultSet is 1
*** Result Set**	Object[][]	The array to perform the operation on.
*** Row**	Integer	Row of the value to extract. ResultSets use one-based index, so for first index of the ResultSet is 1
Fail On Null Value	Boolean	Whether the step fails if the extracted value is null.

Output Values

Name	Type	Description
Extraction Result	String	The result of the extraction.

Set Parameter Value - Boolean

Sets the value of a Boolean parameter to a predefined one during an execution.

Note: The step always succeeds.

Input Values

Name	Type	Description
Value	Boolean	Use the input to hold the needed value and export it to a parameter where it is needed.

Output Values

Name	Type	Description
Value	Boolean	

Name	Type	Description
		Use the input to hold the needed value and export it to a parameter where it is needed.

Set Parameter Value - Integer

Sets the value of a Integer parameter to a predefined one during execution.
Note: The step always succeeds.

Input Values

Name	Type	Description
Value	Integer	Use the input to hold the needed value and export it to the parameter where it is needed.

Output Values

Name	Type	Description
Value	Integer	Use the input to hold the needed value and export it to the parameter where it is needed.

Set Parameter Value - String

Sets the value of a String parameter to a predefined one during execution.

Note: The step always succeeds.

Input Values

Name	Type	Description
Value	String	Use the input to hold the needed value and export it to the parameter where it is needed.

Output Values

Name	Type	Description
Value	String	Use the input to hold the needed value and export it to parameter where it is needed.

User Input - Ask For Parameter Value

Enables the user to enter free text and then store it to a parameter.

Input Values

Name	Type	Description
*** Operation Description**	String	The description of the operation to be displayed, the field displays string in HTML container.
Fail On Empty String	Boolean	The input controls weather the step fails if the user typed an empty string.

Output Values

Name	Type	Description
Entered Text	String	The entered value.

User Input - Choose from Available Options

Allows you to select single or multiple values from a list of available options. The selected value/s can be set to a parameter for future use.

Input Values

Name	Type	Description
*** Description**	String	The description to be displayed, the field displays string in HTML container.
*** Options**	String[]	List of available options to select from.
Allow Multiple Selection	Boolean	Allows multiple selection. Select True to allow multiple select or False for single selection. Selecting Single selection, the selected value is saved in both outputs parameters. Selecting multiple selection, only the first selection is saved in the string output and the selected values will be saved in the string array output.

Output Values

Name	Type	Description
Chosen Option	String	The outcome as indicated by the user.
Chosen Options	String[]	The outcome as indicated by the user as an array.

Process Actions

Contents

Check if Process is Running

Checks if a process is running.

Input Values

Name	Type	Description
*** Process Name**	String	Specifies the process name. Use the exact process name, or use regular expressions. **Windows:** specify the process name according to the executable file name. **Non-Windows:** specify the process name as it appear in the file /proc/PID/status.
Case Sensitive	Boolean	Indicates whether the search is case-sensitive. **Default:** False
Is Regular Expression	Boolean	

Name	Type	Description
		Indicates whether the value of Process Name is a regular expression (java Regex). **Default:** False
Match Full Command	Boolean	Indicates whether the process name is matched against the process command line. **Windows only. Default**: False **Note:** The value is referred as true in Linux.
Should Be Running	Boolean	Indicates whether to expect the process to be running. True for running, False for not running.

Execute JavaScript Code

Evaluates and executes a JavaScript code.

Input Values

Name	Type	Description
*** Script**	String	Specifies the JavaScript code.
Array Input1	String[]	Specifies an Array input variable for the script. The name in the script is arrInput1.
Array Input2	String[]	Specifies an Array input variable for the script. The name in the script is arrInput2.
Array Input3	String[]	Specifies an Array input variable for the script. The name in the script is arrInput3.
Input1	String	Specifies a String input variable for the script. The name in the script is input1.
Input2	String	Specifies a String input variable for the script. The name in the script is input2.
Input3	String	Specifies a String input variable for the script. The name in the script is input3.

Output Values

Name	Type	Description
Array Output1	String[]	The Array output for variable arrOutput1.
Array Output2	String[]	The Array output for variable arrOutput2.

Name	Type	Description
Array Output3	String[]	The Array output for variable arrOutput3.
Array Result Output	String[]	The output of the script execution. The output is valid only if the output is an array.
Output1	String	The String output for variable output1.
Output2	String	The String output for variable output2.
Output3	String	The String output for variable output3.
Result Output	String	The output of the script execution. The output is valid only if the output is not an array.

Kill Process

Kills a process.

Valid for Windows

Input Values

Name	Type	Description
*** Process Name**	String	Specifies the process name. Use the exact process name, or use regular expressions.
Case Sensitive	Boolean	Indicates whether the search is case-sensitive. **Default:** False
Fail If No Matches Are Found	Boolean	Indicates whether the action fail if no processes matches the search. **Default:** False
Is Regular Expression	Boolean	Indicates whether the value of Process Name is a regular expression (java Regex). **Default:** False
Match Full Command	Boolean	Indicates whether the process name is matched against the process command line. **Windows only. Default**: False **N ote:** The value is referred as true in Linux.
Timeout	Long	Specifies the number of seconds to wait for the process to shutdown.
User Name	String	Specifies the name of the user that started the process.

Kill Process by PID

Kills a process specifies by the process ID.

Input Values

Name	Type	Description
*** Pid**	String	Specifies the process ID.
Fail If Not Exist	Boolean	Indicates whether the action fails if no processes matches the search. **Default:** False
Timeout	Long	Specifies the number of seconds to wait for the process to shutdown.

Run Process

Executes an executable.

Input Values

Name	Type	Description
*** Executable Path**	String	Specifies the path of the executable.
*** Work Directory**	String	Specifies the working directory.
Environment Variables Names	String[]	Specifies a list of environment variables names to set as the environment variables for the created process.
Environment Variables Values	String[]	Specifies a list of environment variables values to set as the environment variables for the created process. **Note:** The values must match the order of the names in the Environment Variables Names input.
Executable Arguments	String[]	Specifies the list of arguments to pass to the executable. **Note:** If the parameter is "param0", enter the parameter in one line. If the parameter is

Name	Type	Description
		"param1 param2", enter the values in two separated lines to identify the separate parameters.
Expected Return Value	Long	Specifies the expected return value of the finished process .Use the input if you expect only one return code. **Note:** If you specify the value, and actual return value is not equal to the expected value, the step fails.
Expected Return Values	Long[]	Specifies the expected return values of the finished process. Use the input if executables are with more than one possible return code.
Std Err File Path	String	Specifies the file path to use to write the error stream of the process. **Default:** If no value is set, the standard error is redirected to the standard output.
Std Out File Path	String	Specifies the file path to use to write the output stream of the process.
Success Return List	Boolean	Indicates whether the expected return value list causes the action to succeed or fail. True for succeed, False for fail.
Time Out Duration	long	Specifies the time to wait for the process in seconds. **Note:** The input is relevant only if Wait For Process To Finish is True.
Wait For Process To Finish	Boolean	Indicates whether to wait for the process to finish its execution. To start the process in the background and not wait for it to finish its execution on Windows systems, set this value to False. On non-windows systems the step may fail if the input is set to False.

Output Values

Name	Type	Description
Return Value	Long	The return value of the finished process. Long Parameter can be used with this output.
Std Err Output	String	The standard error of the process. **Note:** This output is relevant only if Wait For Process To Finish is True. **Limit:** 256 character
Std Out Output	String	The standard output of the process. **Note:** This output is relevant only if Wait For Process To Finish is True. **Limit:** 256 character
Success Return List	Boolean	Identifies whether the expected return value list determines if the command executed successfully, or if the commend failed. True for successful, False for fail.

Run Windows Process

Executes the specified executable. To start the process in the background and not wait for it to finish its execution, set the Wait For Process To Finish input to False.

Input Values

Name	Type	Description
*** Executable Path**	String	The path to the executable to run.
*** Work Directory**	String	The working directory for the executable.
Environment Variables Names	String[]	A list of environment variables names to be set as the environment variables for the created process.
Environment Variables Values	String[]	A list of environment variables values to be set as the environment variables for the created process, The values

Name	Type	Description
		should match the name order in the environment Variables Names input.
Executable Arguments	String[]	The list of arguments to pass to the executable.
Std Err File Path	String	The file path to use to write the error stream of the process.
Std Out File Path	String	The file path to use to write the output stream of the process.
Wait For Process To Finish	Boolean	Whether to wait for the process to finish its execution or not. True to wait or False to not wait. On non-windows systems the step will fail if this input is set to False.
Wrap Arguments	Boolean	Whether to wrap the process arguments as one argument, True for wrap or False to leave as is. Used for avoiding where one of the arguments contains quotation mark(s) '"'.

Output Values

Name	Type	Description
Return Value	long	Return value of the finished process. Long Parameter can be used with this output.

Telnet Session

Provides the ability to create a Telnet session to authenticate and send telnet commands to the remote Telnet server.

Input Values

Name	Type	Description
*** Telnet Commands**	String[]	The telnet Commands to execute once connected.
*** Telnet Host**	String	The Telnet Host name.
Login Prompt	String	The prompt to expect as the log in once connected.
Login Successfull Prompt	String	The prompt to expect as successful log in once connected.
Password Prompt	String	The prompt to expect as the password once connected.

Name	Type	Description
Prompt	String	The prompt to expect before each command is sent.
Telnet Port	Integer	The Telnet port to use.
User Name	String	The user name to log in with to the Telnet host.
User Password	Password	The user password to use with the log in user name.

Output Values

Name	Type	Description
Telnet output	String	The output of the Telnet commands.

Wait for Processes

Checks for a list of process names, and retries until the results meet condition of the Process Lookup Policy.

Input Values

Name	Type	Description
* **Process Names**	String[]	Specifies a list of process names. The process name is not case-sensitive. On Windows systems, specify the process name according to the executable file name. On non-Windows systems, specify the process name as it appear in the file /proc/PID/status.
Processes Lookup Policy	WaitForProcess$ProcessesLookupPolicy	Indicates whether to expect all the processes in the list, or at least one process from the list.
Should Be Running	Boolean	Indicates whether to expect the process to be running. True for running, False for not running.
Time To Wait Between Retries	Long	Specifies how long to wait between retries.
Time Unit For The Total Time To Wait For The Process	TimeUnit	Specifies the time unit to use for Total Time To Wait For The Process.

Name	Type	Description
		Examples: MILLISECONDS, SECONDS
Time Unit To Wait Between Retries	TimeUnit	Specifies the time unit to use for Time To Wait Between Retries.**Examples:** MILLISECONDS, SECONDS
Total Time To Wait For The Process	Long	Specifies the total time to wait before the action fails.

Release Operations Center Actions

Contents

- ROC - Create TFS Artifact
- ROC - Get Artifact
- ROC - Get Artifact Package XML
- ROC - Get Artifact Retrieval Source Property
- ROC - Report Artifact Deployment
- ROC - Report Artifact Deployment By File Parameter
- ROC - Upload Artifact To Local Repository

- Deployment Actions
  - ROC - Assign Multiple Servers to Server Types
  - ROC - Create Deployment from Deployment Plan
  - ROC - Create Deployment from Template Category
  - ROC - Get Deployment Status
  - ROC - Run Deployment
- Deployment Property Actions
  - ROC - Get Deployment Info
  - ROC - Get Deployment Property Value
  - ROC - Update Deployment Info
  - ROC - Update Deployment Property Value
- Deployment Step Actions
  - ROC - Add Step Dependencies
  - ROC - Configure Step Rollback Impact
  - ROC - Create Deployment Plan Step
  - ROC - Create Step
  - ROC - Delete Step
  - ROC - Fail Deployment Step
  - ROC - Get Deployment Steps
  - ROC - Get Process Tags
  - ROC - Get Step Dependencies
  - ROC - Get Step Rollback Impact

- ROC - Get Step's Server Types
- ROC - Set Process Tag in Step
- ROC - Set Step Version

- Manifest Actions
  - ROC - Assign Manifest To Deployment Plan
- Parameter Actions
  - ROC - Clear Parameter Value
  - ROC - Get All Parameters
  - ROC - Get Environment Parameter
  - ROC - Get Parameter Value
  - ROC - Update Array Environment Parameter in Deployment
  - ROC - Update Environment Parameter
  - ROC - Update Environment Parameter in Deployment
  - ROC - Update Password Environment Parameter in Deployment
  - ROC - Update Password Parameter
  - ROC - Update Password Release Parameter
  - ROC - Update Property to Release Parameter
  - ROC - Update String Array Parameter
  - ROC - Update String Array Release Parameter
  - ROC - Update String Parameter
  - ROC - Update String Release Parameter
- Rollback Actions
  - ROC - Configure Step Rollback Impact
  - ROC - Get Step Rollback Impact
  - ROC - Set Rollback Definition for Deployment Plan
  - ROC - Set Rollback Plan - Use a Duplication of a Successful Deployment
  - ROC - Set Rollback Plan - Use an Existing Deployment
  - ROC - Set Rollback Plan - Use Latest Successful Deployment Duplicate

- Server Assignment Actions
  - ROC - Assign All Servers From Environment
  - ROC - Assign Multiple Servers to Server Types
  - ROC - Assign Servers
  - ROC - Assign Specific Server Dependency
  - ROC - Get Assigned Servers
  - ROC - Unassign Servers

Approval Gate Actions

ROC - Update Step Approval Gate Status

Updates the approval status for the step approval gate for the current deployment. The action can only run from within the approval step of the deployment.

Input Values

Name	Type	Description
*** Primary Approval Status**	PrimaryApprovalStatus	The primary approval status. Pending - deployment is not approved yet. Approved - deployment is approved. Rejected - deployment is not approved and will be cancelled.
Secondary Approval Status	String	The secondary approval status is meant for the status text given by the external change management system or some other approval system that is used to grant the approval. The status text will be displayed in the Release Operation Center but has no other functional significance.

ROC - Updates ServiceNow Change Request Number

Updates the ServiceNow change request number for the Deployment Approval Gate.

The deployment must have a ServiceNow approval gate.

Input Values

Name	Type	Description
*** Change Request Number**	String	

Name	Type	Description
		ServiceNow Change Request Number to set in deployment approval gate

Artifact Actions

ROC - Add Version to Artifact Package

Adds the specified ROC artifact (set by the artifact type Name, Artifact definition name, and Version) to the artifact package.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Package Name**	String	The name of the artifact package.
*** Artifact Type Name**	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL will default to "Legacy Artifact".
*** Artifact Version**	String	The artifacts version as it appears in the application artifacts list in ROC (Case sensitive)
Create Package If Not Exist	Boolean	Whether or not to create a new artifact package if the package with specified name does not exist

ROC - Assign Artifact Package To Deployment Plan

Assign an artifact package to a deployment plan.

Input Values

Name	Type	Description
Artifact Package Name	String	The name of the artifact package.
Artifact Package XML	String	The XML for the artifact package.
Artifact Package XML File Path	String	The path to the file containing the XML for the artifact package.

ROC - Assign Artifact To File Parameter

Assign an artifact to a file parameter of a server type in a deployment step.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** File Parameter Path**	String	The path to the file parameter to be populated with the artifact.
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".

ROC - Check if Artifact Exists

Checks whether an artifact with the specified name and version already exists.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Version**	String	The version of the artifact. (case sensitive)
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL will default to "Legacy Artifact".

Output Values

Name	Type	Description
Artifact Enabled	Boolean	If the artifact exists, indicates whether the artifact is enabled or not.
Artifact Exists	Boolean	Indicates whether the artifact exists or not.

ROC - Check if Artifact Exists - HTTP

Checks whether an artifact with the specified name and version already exists.

Input Values

Name	Type	Description
*** Artifact Name**	String	The name of the artifact. (case sensitive)
*** Artifact Repository Url**	String	Nexus repository url (for example 'http://localhost:8080/nexus')
*** Artifact Version**	String	The version of the artifact. (case sensitive)

Output Values

Name	Type	Description
Artifact Enabled	Boolean	If the artifact exists, indicates whether the artifact is enabled or not.
Artifact Exists	Boolean	Indicates whether the artifact exists or not.
Artifacts Resource URI	String[]	An array of the resource Uris of the artifacts with the suitable GAV parameters from the input
Extension	String	The extension of the first artifact with the suitable GAV parameters from the input
Packaging	String	The packaging of the first artifact with the suitable GAV parameters from the input.
Resource URI	String	The name of the first artifact with the suitable GAV parameters from the input

ROC - Check if Artifact Package Exists

Checks whether an artifact package already exists.

Input Values

Name	Type	Description
*** Artifact Package Name**	String	The name of the artifact package.

Output Values

Name	Type	Description
Artifact Package Exists	Boolean	Indicates whether the artifact package exists or not.

ROC - Create Artifact Definition

Create an Artifact Definition.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Type Name**	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".
Attributes	String[]	Artifact Definitions Attributes: Line in array should be in the form of an attribute:value
Fail if definition already exists	Boolean	The action fails or not if a definition with the same name already exists. True for failing or False for not. Default is False
Server Types	String[]	List of Server Types to associate the definition to (if no value - the definition is mapped to all server types)

ROC - Create FTP Artifact

Define an FTP artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** Full File Path**	String	The full path to the file.
*** Host Name**	String	Host name
*** User Name**	String	User account name to use for connection. In case of anonymous connection, the value should be anonymous.

Name	Type	Description
Allow Artifact Modifications	Boolean	Set to true to allow modifications to the artifact's content between deployments.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL will default to Legacy Artifact.
Description	String	Description for the created artifact.
FTP Port	Integer	FTP Port.
Password	Password	The users password. For anonymous connection use a valid email address
Secured Connection	String	Specify the connection settings: FTP(Default), FTPS(FTPS_Implicit) or FTPS_Explicit. If left empty FTP is used.
Store Now	Boolean	If set to TRUE, the artifact is uploaded to CA's default repository during the execution of the action. Setting true in the field overrides the value set on Store on next execution.
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to CA's default repository on the first time the artifact is used in a deployment.
Timeout	Integer	Connection timeout in seconds.

ROC - Create HTTP Artifact

Define an http artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** File Alias**	String	Name given to the retrieved file in destination path.

Name	Type	Description
*** Url**	String	The Artifact url.
Allow Artifact Modifications	Boolean	Set to True if you want to allow modifications to the artifact content between deployments.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".
Description	String	Description for the created artifact.
Password	Password	Password
Store Now	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting True in the field overrides the value set on 'Store on next execution'.
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository on the first time the artifact is used in a deployment.
User Name	String	User Name

ROC - Create Local File Artifact

Define a local file artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** Full File Path**	String	The full path to the file.
Allow Artifact Modifications	Boolean	Set to True if you want to allow modifications to the artifact content between deployments.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".

Name	Type	Description
Description	String	Description for the created artifact.
Store Now	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting True in the field overrides the value set on 'Store on next execution'.
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository on the first time the artifact is used in a deployment.

ROC - Create Remote File Artifact

Define a Remote File artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** Full File Path**	String	The full path to the file. Usage example: Windows - file://hostname/c$/folder/file.txt Unix - file://hostname/smb_folder/folder/file.txt.
Allow Artifact Modifications	Boolean	Set to true if you want to allow modifications to the artifact content between deployments.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".
Description	String	Description for the created artifact.
Password	Password	Password
Store Now	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting True in the field overrides the value set on 'Store on next execution'.

Name	Type	Description
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository the first time the artifact is used in a deployment.
User Name	String	User Name

ROC - Create Remote Repository Artifact

Define a Remote Repository artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** Group Id**	String	Artifact group id
*** Packaging**	String	The artifact type or file extension (JAR or WAR for example)
*** Repository Url**	String	Artifact repository url **Example:** http://192.168.0.1:8081/artifactory/libs-release-local
Allow Artifact Modifications	Boolean	Set to True if you want to allow modifications to the artifact content between deployments.
Artifact Classifier	String	Optional artifact classifier (dev or prod for example)
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".
Description	String	Description for the created artifact.
Password	Password	The user password for authentication.
Repository artifact id	String	The artifact id in the repository (if left empty 'Artifact name' will be used)
Repository artifact version	String	The artifact version in the repository (if left empty 'Artifact version' will be used)

Name	Type	Description
Store Now	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting true in the field overrides the value set on 'Store on next execution'.
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository the first time the artifact is used in a deployment.
Username	String	The user name for authentication.

ROC - Create SSH Artifact

Define an SSH file artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** Full File Path**	String	The full path to the file.
*** Host Name**	String	Host name
*** Password**	Password	Password
*** User Name**	String	User Name
Allow Artifact Modifications	Boolean	Set to True if you want to allow modifications to the artifact content between deployments.
Artifact Type Name	String	The artifact`s type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving this as NULL will default to "Legacy Artifact".
Description	String	Description for the created artifact.
Keystore File	String	Keystore file path
Keystore Password	Password	Keystore file password
SSH Port	Integer	SSH Port.
Store Now	Boolean	

Name	Type	Description
		If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting True in the field overrides the value set on 'Store on next execution'.
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository the first time the artifact is used in a deployment.

ROC - Create SVN Artifact

Define an SVN file artifact.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** File Alias**	String	Name given to the retrieved file in destination path.
*** Password**	Password	Password
*** Revision Number**	String	SVN Revision Number
*** Server Url**	String	The Artifact server url.
*** User Name**	String	User Name
Allow Artifact Modifications	Boolean	Set to True if you want to allow modifications to the artifact content between deployments.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".
Description	String	Description for the created artifact.
Store Now	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting True in the field overrides the value set on 'Store on next execution'.
Store on First Use	Boolean	

Name	Type	Description
		If set to TRUE, the artifact is uploaded to the CA default repository the first time the artifact is used in a deployment.

ROC - Create TFS Artifact

Define a TFS file artifact.

Input Values

Name	Type	Description
*** 'tf.exe' command path**	String	The installation path of the 'tf' command line interface. **Example:** 'C:/Program Files/Microsoft Visual Studio 10.0/Common7/IDE/tf.exe', or 'C:/Program Files/Microsoft Visual Studio 9.0/Common7/IDE/tf.exe'.
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Retrieval Identifier**	String	An Agent name/IP/node id, or Artifact Retrieval Agent Group name
*** Artifact Version**	String	The version of the artifact. (case sensitive)
*** Collection Name**	String	The Team Project Collection.
*** File Alias**	String	Name given to the retrieved file in destination path.
*** Password**	Password	Password
*** TFS Server URL**	String	The url of the tfs server
*** User Name**	String	User Name
Allow Artifact Modifications	Boolean	Set to true if you want to allow modifications to the artifact content between deployments.
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving this as NULL will default to "Legacy Artifact".
Description	String	Description for the created artifact.
Item Version	String	Optional version specification. The versionspec is the version of the item you want to retrieve. You can specify a version by: Date/time

Name	Type	Description
		(D10/20/2005), Changeset version (C1256), Label (Lmylabel), Latest version (T), Workspace version (Wworkspacename;owner). If no version is provided, Team Foundation Server retrieves the latest server version.
Store Now	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository during the execution of the action. Setting True in the field overrides the value set on 'Store on next execution'.
Store on First Use	Boolean	If set to TRUE, the artifact is uploaded to the CA default repository the first time the artifact is used in a deployment.
Target Item	String	Specifies target file or folder

ROC - Get Artifact

Downloads the selected ROC artifact (set by Name and Version) to the executing agent.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Version**	String	The artifacts version as it appears in the application artifacts list in ROC (Case sensitive)
*** Target Folder**	String	The folder on the executing agent to which the artifact is downloaded to. (Do not include file name)
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving this as NULL will default to "Legacy Artifact".
Overwrite	Boolean	If set to True, the downloaded artifact overwrites any existing artifact with same name.
Report As Deployed	Boolean	If set to True, the artifact is reported as deployed if download finishes successfully.

ROC - Get Artifact Package XML

Export an artifact package as XML.

Input Values

Name	Type	Description
*** Artifact Package Name**	String	The name of the artifact package.
*** Artifact Package XML File Path**	String	Full path to the file to save the artifact package XML.
Overwrite Existing File	Boolean	Indicates whether or not to overwrite the xml file if it exists

ROC - Get Artifact Retrieval Source Property

Retrieves the correlated value of the source property used as an input. The artifact is set by the artifact type Name, artifact definition name, and Version.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Property Name**	String	The name of the property getter in the retreival source of the artifact.
*** Artifact Type Name**	String	The artifact type name (Administration - Artifacts management). Mandatory. Due to legacy concerns leaving as NULL defaults to Legacy Artifact.
*** Artifact Version**	String	The artifact version as it appears in the application artifacts list in ROC (Case sensitive)
Fail If Not Exist	Boolean	Whether to fail the action if the artifact or the property exist or not

Output Values

Name	Type	Description
Artifact Property Value	String	The value of the property getter name from the input

ROC - Report Artifact Deployment

Reports the selected artifact as deployed on the Environment, Server-Type, and Server that it is executed on. The action can also be executed outside of the Release operations Center.

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Version**	String	The version of the artifact to be reported as deployed. The version must be a valid artifact version from the artifacts list of the application in ROC. (Case sensitive)
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".

ROC - Report Artifact Deployment By File Parameter

Reports as deployed the artifact that is assigned to the selected file parameter in a step artifacts list in the ROC. The artifact is reported as deployed to the environment, server type, and server that run the action. The action can be executed in a deployment step that is not an initialization step

Input Values

Name	Type	Description
*** File Parameter**	String	The artifact that is assigned to the selected file parameter is reported as deployed. Input for the field is a file parameter from the parameters list.

ROC - Upload Artifact To Local Repository

Uploads the target ROC artifact to the local repository. (If it is not set to be stored in the repository it is after the action is executed)

Input Values

Name	Type	Description
*** Artifact Definition Name**	String	The artifact definition name.
*** Artifact Version**	String	The artifacts version as it appears in the application artifacts list in ROC (Case sensitive)
Artifact Type Name	String	The artifact type name (Administration - Artifacts management) . Mandatory field. Due to legacy concerns leaving as NULL defaults to "Legacy Artifact".

Deployment Actions

ROC - Assign Multiple Servers to Server Types

Assign multiple servers across different server types and steps.

The JSON object does not have to be in array shape, it can be in a regular JSON object shape.

If its a JSON object, and not a file, make sure it does not contains new lines in it.

The JSON format is an array example:

[{"Step Name":"example of step", "Server Type Name":"server type 1", "Server Identifiers":["192.168.0.56","server-name-example"], "Overwrite Existing Servers":"true"},

{"Step Name":"example of step", "Server Type Name":"server type 2", "Server Identifiers":["192.168.0.196","localhost"], "Overwrite Existing Servers":"true"}].

Input Values

Name	Type	Description
Failuare Condition	Boolean	Whether to fail the action if one or more of the assignees tasks fails
JSON File Path	String	The path to the file that contains the JSON object representing the assignees tasks. Either the JSON obect or the JSON file path must be specified.
JSON Object	String	The JSON object representing the assignees tasks - Either the JSON obect or the JSON file path must be specified.

Output Values

Name	Type	Description
Failed Assignees Tasks	String	A list of all failed assignees tasks - if exists

ROC - Create Deployment from Deployment Plan

This action creates a new deployment from an existing deployment plan. You mau use:

1. Deployment plan ID.

2. Deployment plan name, project name and version.

If both were entered it will use the deployment plan ID.

Input Values

Name	Type	Description
*** Deployment Name**	String	The new deployment name
*** Environment Name**	String	The Environment name to run the deployment on.
Application Name	String	The name of the deployment application.
Build Version	String	The (optional) version (Build) name to create a deployment from.
Deployment Plan ID	Long	The (optional) Deployment plan ID to create a deployment from.
Deployment Plan Name	String	The (optional) Deployment plan name to create a deployment from
Deployment Stage	DeploymentStageMode	Execute The stage after deployment has been created. All The stages preceding will be executed.
Password	Password	The Release Automation password. Must be set if and only if the username is set.
Project Name	String	The (optional) project name to create a deployment in
User name	String	The (optional) Release Automation username. If not given, the process runner's credentials will be used.

Output Values

Name	Type	Description
Deployment ID	long	The new Deployment ID.

ROC - Create Deployment from Template Category

Creates a new deployment from an existing template category. You can create the deployment and run init step if exists (Execution Mode parameter). The output is the created deployment id which is used as input parameter to Run Deployment action.

Input Values

Name	Type	Description
*** Application Name**	String	The name of the deployment application.
*** Deployment Name**	String	The new name for the deployment.
*** Deployment Version**	String	

Name	Type	Description
		The new version for the deployment.
*** Environment Name**	String	The name of the deployment environment.
*** Template Category Name**	String	The name of the template category.
Deployment Description	String	The new description for the deployment.
Deployment Type	ReleaseType	The new type for the deployment.
Execution Mode	ExecutionMode	Set Execution Mode - Create deployment and run init step, or create deployment and does not run init step
Password	Password	The Release Automation password. Must be set if and only if the username is set.
Username	String	The (optional) Release Automation username. If not used, the process runner credentials is used.

Output Values

Name	Type	Description
Deployment ID	Long	The ID of the deployment.

ROC - Get Deployment Status

Gets the deployment status. The input parameter is the deployment id (the same as the deployment id output in Create Deployment From Template Category action). The output are Deployment Status, Current Stage, and Current Stage Status.

Input Values

Name	Type	Description
*** Deployment ID**	Long	The ID of the deployment.

Output Values

Name	Type	Description
Current Stage	String	The deployment current stage - Initialization, Approval-Gate, Deployment, Post-Deployment
Current Stage Status	String	The deployment current stage status - Pending, Running, Paused, Running-With-Errors, Succeeded, Failed, Canceled.

Name	Type	Description
Deployment Status	String	The status of the deployment - Active, Succeeded, Failed, Canceled.

ROC - Run Deployment

Runs an existing deployment. The input parameter is the deployment id (the same as the deployment id output in Create Deployment From Template Category action).

Input Values

Name	Type	Description
*** Deployment ID**	long	The ID of the deployment.
Password	Password	The Release Automation password. Must be set if and only if the username is set.
Username	String	The (optional) Release Automation username. If not used the process runner credentials will be used.

Deployment Property Actions

ROC - Get Deployment Info

Retrieves basic deployment information such as name, version, description, type, application name, and environment.

Output Values

Name	Type	Description
Deployment Application	String	The name of the deployment application.
Deployment Description	String	The current deployment description.
Deployment Environment	String	The name of the deployment environment.
Deployment Name	String	The current deployment name.
Deployment Type	String	The current deployment type.
Deployment Version	String	The current deployment version.

ROC - Get Deployment Property Value

Retrieves the value of a specified deployment property. The supported properties are the user defined properties and the following predefined properties: "Name", "Version", "Application", and "Environment".

Input Values

Name	Type	Description
*** Property Name**	String	The name of the property

Output Values

Name	Type	Description
Property value	String	The value of the property

ROC - Update Deployment Info

Modifies built-in properties of the deployment and retrieves updated values. It is not required to update a property in order to read its value.

Note: Modifying properties that are visible via built-in application parameters (e.g. "Deployment Name") does not affect the parameter values - use the action output for subsequent usages.

Input Values

Name	Type	Description
Deployment Description	String	The new description for the deployment.
Deployment Name	String	The new name for the deployment.
Deployment Type	ReleaseType	The new type for the deployment.
Deployment Version	String	The new version for the deployment.

Output Values

Name	Type	Description
Deployment Application	String	The name of the deployment application.
Deployment Description	String	The new description for the deployment.
Deployment Environment	String	The name of the deployment environment.
Deployment Name	String	The new name for the deployment.
Deployment Type	String	The new type for the deployment.
Deployment Version	String	The new version for the deployment.

ROC - Update Deployment Property Value

Updates the value of a specified deployment property. The supported properties are the user defined properties and the following predefined properties: "Name" and "Version".

Input Values

Name	Type	Description
*** Property Name**	String	The name of the property
Property value	String	The value of the property
Should fail when property name does not exist	Boolean	If True the action fails if the property name does not exist.

Deployment Step Actions

ROC - Add Step Dependencies

Set the deployment steps that are to be completed before the step starts.

Input Values

Name	Type	Description
*** Preceding Step Names**	String[]	The deployment steps that are completed before the step starts.
*** Step Name**	String	The name of the step.
Overwrite Existing Dependencies	Boolean	Indicates whether or not to overwrite the existing dependencies of the deployment step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Configure Step Rollback Impact

Configure how a step impacts the rollback plan. Trigger_Rollback_on_Failure - If the step fails then the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Rollback Impact	RollbackImpact	Determines how the step affects the rollback plan. Trigger_Rollback_on_Failure - If

Name	Type	Description
		the step fails the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Create Deployment Plan Step

Create a deployment step and add it to the current deployment plan.

Input Values

Name	Type	Description
*** Process Full Path Name**	String	The category path name of the deployment process. You may select not to specify a full path only if there are not two processes with the same name assigned to the environment, e.g. if your environment contains a single process named 'P1' under the 'Processes' category you may enter either 'Processes/P1' or simply 'P1', but, if the environment is assigned with another process named 'P1' under 'DB Processes' you must enter the full path for each process.
*** Step Name**	String	The name of the step.
Process Tag Name	String	Optional Process tag name. If not specified, latest published process is used.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Create Step

Create a deployment step and add it to the current deployment.

Input Values

Name	Type	Description
*** Process Full Path Name**	String	The category path name of the deployment process. You may select not to specify a full path if there are not two processes with the same name assigned to

Name	Type	Description
		the environment, e.g. if your environment contains a single process named 'P1' under the 'Processes' category you may enter either 'Processes/P1' or simply 'P1', but, if the environment is assigned with another process named 'P1' under 'DB Processes' you must enter the full path for each process.
*** Step Name**	String	The name of the step.
Process Tag Name	String	Optional Process tag name. If not specified, latest published process will be used.
Step Version	String	The step version.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Delete Step

Remove a step from the deployment.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Fail Deployment Step

Fail the deployment step that runs the action.

ROC - Get Deployment Steps

Retrieves the steps for the deployment.

Input Values

Name	Type	Description
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Step Names	String[]	The deployment step names.

ROC - Get Process Tags

Retrieves the tags for the process.

Input Values

Name	Type	Description
*** Process name**	String	The name of the process that the list of tags will returned
Filter String	String	If used, all results that do not contain the string are filtered

Output Values

Name	Type	Description
Process Tags	String[]	The list of process tags for the target process

ROC - Get Step Dependencies

Get the dependencies list of a deployment step. The dependencies list contains deployment steps that must be completed before the step starts.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Preceding Step Names	String[]	The deployment steps that must be completed before the step starts.

ROC - Get Step Rollback Impact

Get how a step impacts the rollback plan. Trigger_Rollback_on_Failure - If the step fails then the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Rollback Impact	String	Determines how the step affects the rollback plan. Trigger_Rollback_on_Failure - If the step fails then the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.

ROC - Get Step's Server Types

Retrieves a specified step server types.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Server Type Names	String[]	The step server types.

ROC - Set Process Tag in Step

Set the process tag for a step.
The process tag must be of the process that is already set in the step.

Input Values

Name	Type	Description
*** Process Tag Name**	String	Process tag name to be used by the step. The process tag must be of the process that is already set in the step.
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	

Name	Type	Description
		The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Set Step Version

Set the version property for a step.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
*** Step Version**	String	The step version.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Manifest Actions

ROC - Assign Manifest To Deployment Plan

Assign a manifest to a deployment plan.

Input Values

Name	Type	Description
Manifest XML	String	XML for the manifest.
Manifest XML File Path	String	Path to the file containing the XML for the manifest.

Parameter Actions

ROC - Clear Parameter Value

Clears the values of specified parameter(s).

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (use the "Full Path" attribute of the parameter as it appears in the ROC).
*** Step Name**	String	The name of the step.
Server Identifiers	String[]	An array of host names, IP addresses, or Node ids identifying the servers.

Server Type Name	String	The name of the server type
Should fail when a parameter name or step name doesn't exist	Boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Get All Parameters

Allows you to get global / server type parameters. To get the global parameters type in the step name input field alone. To get the server type parameters type in both the step name and server type name input fields.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Server Type Name	String	The name of the server type
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Parameter Names	String[]	The names of the parameters

ROC - Get Environment Parameter

Allows you to get the value of the parameter. If the parameter is a simple value, the output appears in the simple value output. If the parameter is a simple value, the output appears in the array value output.

Input Values

Name	Type	Description
*** Parameter Path**	String	The path of the parameter (use the "Full Path" attribute of the parameter as it appears in the ASAP)
Application Name	String	The Name of the application, the default is taken from the built in parameters
Environment Name	String	The Name of the environment, the default is taken from the built in parameters
Server Type	String	

Name	Type	Description
		The Name of the server type, if enterd - the output contains the values of the server type

Output Values

Name	Type	Description
Array Value	String[]	Array value - STRING [], INT [] etc..
Simple Value	String	Simple value - STRING, INT, BOOLEAN etc..

ROC - Get Parameter Value

Allows you to get the value of a parameter. To get a global parameter type the step name input field. To get the server type parameter fill both the step name and server type name input fields. To get a server parameter fill the step name, server type name, and the server identifier.

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (use the "Full Path" attribute of the parameter as it appears in the ROC).
*** Step Name**	String	The name of the step.
Server Identifier	String	The server's IP, host name, or node id.
Server Type Name	String	The name of the server type

Output Values

Name	Type	Description
Parameter value as Password	Password	The value of the parameter
Parameter value as String	String	The value of the parameter
Parameter value as String Array	String[]	The value of the parameter

ROC - Update Array Environment Parameter in Deployment

Update the array environment parameter in a deployment

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter with the folder preceding.

Name	Type	Description
		Example: Application Parameters/param or Default Component/param.
*** Parameter value**	Serializable[]	The value of the parameter.
Should Fail	Boolean	Whether the action fails in case of an logical error or not. **Example:** Parameter not found.

ROC - Update Environment Parameter

Allows you to update the value of the parameter.

Input Values

Name	Type	Description
*** Parameter Path**	String	The path of the parameter (use the "Full Path" attribute of the parameter as it appears in the ASAP)
Application Name	String	The Name of the application, the default is taken from the built in parameters
Array Value	String[]	Array value - STRING [], INT [] etc..
Environment Name	String	The Name of the environment, the default is taken from the built in parameters
Server Type	String	By adding a name here, the values entered updates the values per the server type
Simple Value	String	Simple value - STRING, INT, BOOLEAN etc..

ROC - Update Environment Parameter in Deployment

Update the environment parameter in a deployment

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter with the folder preceding. **Example:** Application Parameters/param or Default Component/param.
*** Parameter value**	String	The value of the parameter.

Name	Type	Description
Should Fail	Boolean	Whether the action fails in case of an logical error or not. **Example:** Parameter not found.

ROC - Update Password Environment Parameter in Deployment

Update the password environment parameter in a deployment

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter with the folder preceding. **Example:** Application Parameters/param or Default Component/param.
*** Parameter value**	Password	The value of the parameter.
Should Fail	Boolean	Whether the action fails in case of an logical error or not. **Example:** Parameter not found.

ROC - Update Password Parameter

Updates the Password Parameter. To get update a application parameter type the step name input. To update the server type parameter type both the step name and server type name inputs. To update the server parameters type the step name, server type name, and the server identifiers.

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (use the "Full Path" attribute of the parameter as it appears in the ROC).
*** Parameter value**	Password	The value of the parameter
*** Step Name**	String	The name of the step.
Server Identifiers	String[]	An array of host names, IP addresses, or Node ids identifying the servers.
Server Type Name	String	The name of the server type
Should fail when a parameter name or step name doesn't exist	Boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	

Name	Type	Description
		The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Update Password Release Parameter

Update the Password Release Parameter

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (Parameter with SERVER_TYPE value scope needs to be prefixed with Server Type name and '/').
*** Parameter value**	Password	The value of the parameter
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Should fail when a parameter name or step name doesn't exist	Boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Update Property to Release Parameter

Update the release scoped parameter with a property name.

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (Parameter with SERVER_TYPE value scope needs to be prefixed with Server Type name and '/').
*** Property Name**	String	The name of the property
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Should fail when a parameter name or step name doesn't exist	Boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Update String Array Parameter

Update the String Array Parameter. To update a application parameter type the step name input. To update the server type parameter type both the step name and server type name inputs. To update server parameters type the step name, server type name, and the server identifiers.

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (use the "Full Path" attribute of the parameter as it appears in the ROC).
*** Parameter value**	String[]	The value of the parameter
*** Step Name**	String	The name of the step.
Server Identifiers	String[]	An array of host names, IP addresses or Node ids identifying the servers.
Server Type Name	String	The name of the server type
Should fail when a parameter name or step name doesn't exist	Boolean	If it's true the action will fail if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Update String Array Release Parameter

Update the String Array Release Parameter

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (Parameter with SERVER_TYPE value scope needs to be prefixed with Server Type name and '/').
*** Parameter value**	String[]	The value of the parameter.
*** Server Type Name**	String	The name of the server type.
*** Step Name**	String	The name of the step.
Should fail when a parameter name or step name doesn't exist	Boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step, can be Deployment or Post_Deployment.

ROC - Update String Parameter

Update the String Parameter. To update a application parameter type the step name input. To update the server type parameter type both the step name and server type name inputs. To update server parameters type the step name, server type name, and the server identifiers.

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (use the "Full Path" attribute of the parameter as it appears in the ROC).
*** Parameter value**	String	The value of the parameter
*** Step Name**	String	The name of the step.
Server Identifiers	String[]	An array of host names, IP addresses or Node ids identifying the servers.
Server Type Name	String	The name of the server type
Should fail when a parameter name or step name doesn't exist	boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Update String Release Parameter

Update the String Release Parameter

Input Values

Name	Type	Description
*** Parameter Name**	String	The name of the parameter (Parameter with SERVER_TYPE value scope needs to be prefixed with Server Type name and '/').
*** Parameter value**	String	The value of the parameter
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Should fail when a parameter name or step name doesn't exist	Boolean	If True the action fails if the parameter name or step name does not exist.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Rollback Actions

ROC - Configure Step Rollback Impact

Configure how a step impacts the rollback plan. Trigger_Rollback_on_Failure - If the step fails then the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Rollback Impact	RollbackImpact	Determines how the step affects the rollback plan. Trigger_Rollback_on_Failure - If the step fails the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Get Step Rollback Impact

Get how a step impacts the rollback plan. Trigger_Rollback_on_Failure - If the step fails then the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.

Input Values

Name	Type	Description
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Rollback Impact	String	Determines how the step affects the rollback plan. Trigger_Rollback_on_Failure - If the step fails then the rollback plan is triggered. No_Impact - The step does not affect the rollback plan.

ROC - Set Rollback Definition for Deployment Plan

Configure the rollback definitions of a deployment plan

Input Values

Name	Type	Description
*** Rollback name**	String	Rollback deployment name.
Deployment Plan name	String	The deployment plan name that is used in case of a rollback
Invocation method	Boolean	Whether to strat the rollback automatically in case of a failure
Project name	String	The deployment plan project's name that is used in case of a rollback
Version	String	The deployment plan version's name that is used in case of a rollback

ROC - Set Rollback Plan - Use a Duplication of a Successful Deployment

Set the rollback plan for the deployment. The rollback deployment is a duplicate of a successful deployment with the name and version. The successful deployment can be of the same application and environment of the current deployment. The rollback deployment is created with the given name and version which must be unique per application and environment.

Input Values

Name	Type	Description
*** Rollback Deployment Name**	String	The name of the rollback deployment to be created.
*** Rollback Deployment Version**	String	The version of the rollback deployment to be created.
*** Successful Deployment Name**	String	The name of the successful deployment to be duplicated.
*** Successful Deployment Version**	String	The version of the successful deployment to be duplicated.
Invocation Method	InvocationMethod	The invocation method of the rollback deployment. Automatic - rollback is invoked automatically once triggered. Manual - rollback is invoked manually. Automatic_If_Allowed - If the current environment/template category rollback settings relevant for the current deployment allow automatic rollback then the invocation method is automatic, otherwise it is manual.

ROC - Set Rollback Plan - Use an Existing Deployment

Set the rollback plan for the deployment. The rollback deployment is the one with the name and version - an existing deployment that has not yet been executed. The rollback deployment must be of the same application and environment of the current deployment.

Input Values

Name	Type	Description
*** Rollback Deployment Name**	String	The name of the existing open deployment.
*** Rollback Deployment Version**	String	The version of the existing open deployment.
Disposal Method	DisposalMethod	The disposal method for the rollback deployment. Once the current deployment finishes without the use of the rollback deployment, the rollback deployment is either retained for future use or removed from the system completely. Keep_Release - the rollback deployment is not deleted and is available for future use. Remove_Release - the rollback deployment is deleted completely once the current deployment finishes without requiring a rollback.
Invocation Method	InvocationMethod	The invocation method of the rollback deployment. Automatic - rollback is invoked automatically once triggered. Manual - rollback is invoked manually. Automatic_If_Allowed - If the current environment/template category rollback settings relevant for the current deployment allow automatic rollback then the invocation method is automatic, otherwise it is manual.

ROC - Set Rollback Plan - Use Latest Successful Deployment Duplicate

Set the rollback plan for the deployment. The rollback deployment is a duplicate of the latest successful deployment of the same application, on the same environment that was created from the same template category as the current deployment. If the current deployment was not created from a template category then the action fails. The rollback deployment is created with the name and version that must be unique per application and environment.

Input Values

Name	Type	Description
*** Rollback Deployment Name**	String	The name of the rollback deployment to be created.
Invocation Method	InvocationMethod	The invocation method of the rollback deployment. Automatic - rollback is invoked automatically once triggered. Manual - rollback must be invoked manually. Automatic_If_Allowed - If the current environment/template category rollback settings relevant for the current deployment allow automatic rollback then the invocation method is automatic, otherwise it is manual.
Rollback Deployment Version	String	The version of the rollback deployment to be created. If no version is specified then the same version as the duplicated deployment is used.

Server Assignment Actions

ROC - Assign All Servers From Environment

Assign servers mapped to a server type in the deployment environment to the server type in the step.

Input Values

Name	Type	Description
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Assign Multiple Servers to Server Types

Assign multiple servers across different server types and steps.

The JSON object does not have to be in array shape, it can be in a regular JSON object shape.

If its a JSON object, and not a file, make sure it does not contains new lines in it.

The JSON format is an array example:

[{"Step Name":"example of step", "Server Type Name":"server type 1", "Server Identifiers":["192.168.0.56","server-name-example"], "Overwrite Existing Servers":"true"},

{"Step Name":"example of step", "Server Type Name":"server type 2", "Server Identifiers":["192.168.0.196","localhost"], "Overwrite Existing Servers":"true"}].

Input Values

Name	Type	Description
Failuare Condition	Boolean	Whether to fail the action if one or more of the assignees tasks fails
JSON File Path	String	The path to the file that contains the JSON object representing the assignees tasks. Either the JSON obect or the JSON file path must be specified.
JSON Object	String	The JSON object representing the assignees tasks - Either the JSON obect or the JSON file path must be specified.

Output Values

Name	Type	Description
Failed Assignees Tasks	String	A list of all failed assignees tasks - if exists

ROC - Assign Servers

Assign servers to a server type in a step.

Input Values

Name	Type	Description
*** Server Identifiers**	String[]	An array of host names, IP addresses, or Node ids identifying the servers.
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Overwrite Existing Servers	Boolean	Indicates whether or not to overwrite the existing servers that are assigned to the server type.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Unidentified Servers	String[]	

Name	Type	Description
		An array of the given server identifiers that did not match any server in the environment.

ROC - Assign Specific Server Dependency

Assign dependency between two specific servers of distinguished types.

Input Values

Name	Type	Description
*** Source Server Identifier**	String	The source server's IP, host name, or node id.
*** Source Server Type Name**	String	The name of the source server type
*** Step Name**	String	The name of the step.
*** Target Server Identifier**	String	The target server's IP, host name, or node id.
*** Target Server Type Name**	String	The name of the target server type.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

ROC - Get Assigned Servers

Retrieve assigned servers of a server type in a step.

Input Values

Name	Type	Description
*** Server Type Name**	String	The name of the server type
*** Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Output Values

Name	Type	Description
Server IPs	String[]	An array of IP addresses identifying the servers.
Server Names	String[]	An array of host names identifying the servers.
Server Node Ids	String[]	An array of node ids uniquely identifying the servers.

ROC - Unassign Servers

Unassign servers from a server type in a step.

Input Values

Name	Type	Description
* **Server Identifiers**	String[]	An array of host names, IP addresses or Node ids identifying the servers.
* **Server Type Name**	String	The name of the server type
* **Step Name**	String	The name of the step.
Step's stage	SingleModuleStageDescriptor$ RCActionStageType	The stage of step (can be 'Deployment' or 'Post_Deployment')

Repository Actions

Contents

Download Repository Artifact

Downloads an artifact from a repository manager. Supports Nexus, Artifactory, or any repository that complies with Maven repository layout definition.

Input Values

Name	Type	Description
* **Artifact Id**	String	Specifies the Artifact id.
* **Artifact Version**	String	Specifies the Artifact version.
* **Group Id**	String	Specifies the Artifact group id.
* **Packaging**	String	Specifies the artifact type or file extension. **Examples:** JAR, WAR
* **Repository Url**	String	Specifies the Artifact repository url. **Example:** http://192.168.0.1:8081/artifactory/libs-release-local
* **Target Path**	String	

Name	Type	Description
		Specifies the folder path on the executing agent to download the artifact to.
Artifact Classifier	String	Specifies an optional artifact classifier. **Example:** dev, prod
Password	Password	Specifies the password for authentication.
Username	String	Specifies the user name for authentication.

Security Actions

Contents

Authenticate User

Verifies the validity of a user name and password.

Input Values

Name	Type	Description
*** Password**	Password	Specifies the user password.
*** User Name**	String	Specifies the user name. **Note:** On Windows systems that use Domain, specify the domain name with the user name.

Name	Type	Description
		Example: *DOMAIN\Username*. On non-windows systems that use domain, do not specify the domain name.

Output Values

Name	Type	Description
User Authenticated	Boolean	Indicates whether the credentials are valid.

Change Credentials for Windows Service

Changes the start account and password of the Windows service.

Input Values

Name	Type	Description
*** Service Name**	String	The service name or display name of service to start. The name specified is case-insensitive but must be the exact service name or the exact display name.
*** Start Account**	String	The start account for the service. Provided in the form of DomainName\UserName. For a local users use: .\LocalUser and for a local system account use: LocalSystem.
Password	Password	The password for the start account.
Startup Type	ServiceStartupType	The service startup type.

Change Owner of File or Folder

Changes the owner of the file or folder.

Note: Supports Unix/Linux only.

Input Values

Name	Type	Description
*** File Name**	String	The path to the file or folder to change the owner.
*** New Owner**	String	The new owner user name.

Name	Type	Description
Change Recursively	Boolean	If the resource is a folder, indicates whether to change the owenrship recursively.
Fail If Not Exists	Boolean	What to do if the file or folder does not exist, True to fail, False to ignore.

Check if Specific User has Administrative Rights

Verifies whether the user name is listed in the Administrators group.

Windows Only

Input Values

Name	Type	Description
*** Password**	Password	Specifies the user password.
*** User Name**	String	Specifies the user name. **Note:** To check a domain user, specify the domain name with the user name. **Example:** *DOMAIN**Username*.

Check if Specific User has Permission on File or Folder

Checks if the specified user credentials has permissions to the file or folder.

Input Values

Name	Type	Description
*** File Or Folder Path**	String	The path to the file or folder to check. The input also accepts File parameters.
*** Password**	Password	The user password to verify the permissions on. Refer to the step help for more details.
*** User Name**	String	The user name to verify the permissions on. Refer to the step help for more details. **Note:** On Window systems that use Domain, specify the domain name with the user name. Example: *DOMAIN**Username*.

Name	Type	Description
		On non-windows systems that use domain, do not specify the domain name.
Check Execute Permissions	Boolean	Indicates to check if the specified user has execution permissions for the file or folder
Check Read Permissions	Boolean	Indicates to check if the specified user has read permissions for the file or folder
Check Write Permissions	Boolean	Indicates to check if the specified user has write permissions for the file or folder

Check if User has Root Credentials

Checks if the user credentials belongs to the root group. Supports only Unix/Linux systems.

Input Values

Name	Type	Description
* **Password**	Password	The user password to verify the root rights.
* **User Name**	String	The user name to verify the root rights.

Decrypt String

Decrypts a string with a given password and security method

Input Values

Name	Type	Description
* **Password**	String	Specifies the password. **Note:** The input must be 16-byte for AES, and 8-byte for DES.
* **Sec Algorithm**	SecAlgorithm	Specifies the Security algorithm to use, AES, DES, Triple DES
* **String To Decrypt**	String	Specifies the string to decrypt. The result is placed in the 'Decrypted value' output
Salt	String	Specifies a random, 8-byte only string to use with the the password.

Output Values

Name	Type	Description
Decrypted Value	String	The decrypted value.

Encrypt String

Encrypst a string with a given password and security method.

Input Values

Name	Type	Description
*** Password**	String	Specifies the password. **Note:** The input must be 16-byte for AES, and 8-byte for DES.
*** Sec Algorithm**	SecAlgorithm	Specifies the Security algorithm to use, AES, DES, Triple DES
*** String To Encrypt**	String	Specifiers the string to encrypt. The result is placed in the 'Encrypted Value' output
Salt	String	Specifies a random, 8-byte only string to use with the the password.

Output Values

Name	Type	Description
Encrypted Value	String	The encrypted value.

Servers in Environment Actions

Contents

Application Environment - Assign Servers

Assign servers to a server type in an Automation Studio application environment.

Input Values

Name	Type	Description
*** Application Name**	String	The application name.
*** Environment Name**	String	The environment name.
*** Server Identifiers**	String[]	An array of host names, IP addresses, or node IDs identifying the servers.
*** Server Type Name**	String	The server type name.
ASAP Password	Password	The Automation Studio user password.
ASAP User Name	String	The Automation Studio user requires permission to edit the environment. If no username is specified the executing user permissions is used.
Overwrite Existing Servers	Boolean	Indicates whether or not to overwrite the existing servers that are assigned to the server type. **Note:** Assigning servers that are already assigned when the flag is set clears any existing dependencies.

Output Values

Name	Type	Description
Unidentified Servers	String[]	A list of server identifiers that do not match any server in Automation Studio.

Application Environment - Get Assigned Servers

Retrieve assigned servers of a server type in an Automation Studio application environment.

Input Values

Name	Type	Description
*** Application Name**	String	The application name.
*** Environment Name**	String	The environment name.
*** Server Type Name**	String	The server type name.
ASAP Password	Password	The Automation Studio user password.
ASAP User Name	String	

Name	Type	Description
		The Automation Studio user requires permission to edit the environment. If no username is specified, the executing users permissions are used.

Output Values

Name	Type	Description
Server IPs	String[]	An array of IP addresses identifying the servers.
Server Names	String[]	An array of host names identifying the servers.
Server Node Ids	String[]	An array of node ids uniquely identifying the servers.

Application Environment - Unassign Servers

Unassign servers from a server type in an Automation Studio application environment.

Input Values

Name	Type	Description
*** Application Name**	String	The application name.
*** Environment Name**	String	The environment name.
*** Server Identifiers**	String[]	An array of host names, IP addresses, or node IDs identifying the servers.
*** Server Type Name**	String	The server type name.
ASAP Password	Password	The Automation Studio user password.
ASAP User Name	String	The Automation Studio user requires permission to edit the environment. If no username is specified, the the executing user permissions will be used.

SNMP Actions

Contents

- Send SNMPv3 Trap

Send SNMPv1 Trap

Sends an SNMPv1 Trap to a designated IP address.

Input Values

Name	Type	Description
*** Destination IP**	String	The IP address of the trap receiver.
Community Passphrase	Password	The community pass phrase.
Destination Port	Integer	The port of the trap receiver.
Error Index	Integer	The error index.
Error Status	Integer	The error status.
Request ID	Integer	The request ID.
Sender OID	String	The sender OID.
Timestamp	Long	The trap time stamp.
Transport Protocol	TransportProtocol	The transport protocol.
Trap Generic Type	Integer	The trap generic type.
Trap Specific Type	Integer	The trap specific type.
Variable Bindings	String	The variable bindings list. Each line defines a single variable binding and be in the form of: OID TYPE VALUE String values can contain spaces. **Example:** 1.3.6.1.4.54 INTEGER 17 or 1.3.6.1.4.1.2853 STRING a string. The available types are: STRING, COUNTER32, COUNTER64, GAUGE, INTEGER, ADDRESS, OID and TIMETICKS
Variable Bindings Path	String	A file path for the variable bindings that are used instead of Variable Bindings. **Note:** If variable bindings are set the value is ignored.

Send SNMPv2c Trap

Sends an SNMPv2c Trap to a designated IP address.

Input Values

Name	Type	Description
*** Destination IP**	String	The IP address of the trap receiver.
Community Passphrase	Password	The community passphrase.
Destination Port	Integer	The port of the trap receiver.
Error Index	Integer	The error index.
Error Status	Integer	The error status.
Request ID	Integer	The request ID.
Transport Protocol	TransportProtocol	The transport protocol.
Variable Bindings	String	The variable bindings list. Each line defines a single variable binding and be in the form of: OID TYPE VALUE String values can contain spaces. **Example:** 1.3.6.1.4.54 INTEGER 17 or 1.3.6.1.4.1.2853 STRING a string The available types are: STRING, COUNTER32, COUNTER64, GAUGE, INTEGER, ADDRESS, OID and TIMETICKS
Variable Bindings Path	String	A file path for the variable bindings that is used instead of Variable Bindings. **Note**: If Variable Bindings are set the value is ignored.

Send SNMPv3 Trap

Sends an SNMPv3 Trap to a designated IP address.

Input Values

Name	Type	Description
*** Destination IP**	String	The IP address of the trap receiver.

Name	Type	Description
*** User Name**	String	The user name.
Authentication Passphrase	Password	The authentication passphrase. Requires to be set if an authentication protocol has been set. Should be at least 8 characters long.
Authentication Protocol	AuthProtocol	The authentication protocol.
Destination Port	Integer	The port of the trap receiver.
Error Index	Integer	The error index.
Error Status	Integer	The error status.
Privacy Passphrase	Password	The privacy passphrase. requires to be set if an privacy protocol has been set. Should be at least 8 characters long.
Privacy Protocol	PrivacyProtocol	The privacy protocol.
Request ID	Integer	The request ID.
Transport Protocol	TransportProtocol	The transport protocol.
Variable Bindings	String	The variable bindings list. Each line defines a single variable binding and be in the form of: OID TYPE VALUE String values can contain spaces. **Example:** 1.3.6.1.4.54 INTEGER 17 or 1.3.6.1.4.1.2853 STRING a string The available types are: STRING, COUNTER32, COUNTER64, GAUGE, INTEGER, ADDRESS, OID and TIMETICKS.
Variable Bindings Path	String	A file path for the variable bindings that are used instead of Variable Bindings. **Note:** If Variable Bindings are set the value is ignored.

SSH Actions

Contents

- Get File (SSH)
- Put File (SSH)
- Run Command (SSH)

Check if File or Folder Exists (SSH)

Checks if a path exists on the target host using SSH protocol.

Input Values

Name	Type	Description
*** Full Path**	String	The path to the file or folder to check if exists. The input also accepts a File parameters. For Windows paths like c:\dir\file use pattern /C/dir/file.
*** Host Name**	String	The host name or IP address to connect to.
*** User Name**	String	The user name to use to authenticate with the remote host.
Keystore File	String	The private key file to use if the password is not used.
Keystore Password	Password	The password for the private key file if needed.
Password	Password	The password to use to authenticate with the remote host.
Should Exist	Boolean	Indicates whether to expect the file or folder to exist or not. True for exist or False for not exist.
Ssh Port	int	The port to use to connect to the remote host.

Get File (SSH)

Retrieve a file from a remote host using SSH/SFTP protocol.

Input Values

Name	Type	Description
*** Full Source File Path**	String	

Name	Type	Description
		The full path of the source file to retrieve from the target host. For Windows paths like c:\dir\file use pattern /C/dir/file.
*** Full Target File Path**	String	The full path of the destination on the local host.
*** Host Name**	String	The host name or IP address to connect to.
*** User Name**	String	The user name to use to authenticate with the remote host.
Keystore File	String	The private key file to use if the password is not used.
Keystore Password	Password	The password for the private key file if needed.
Password	Password	The password to use to authenticate with the remote host.
Preserve Permissions	Boolean	Determines whether or not to preserve file permissions and owner. **Note:** Applicable only to transfer files between two POSIX compatible (Unix/Linux) systems
Ssh Port	Int	The port to use to connect to the remote host.

Put File (SSH)

Copy a file to a remote host using SSH/SFTP protocol.

Input Values

Name	Type	Description
*** Full Source File Path**	String	The full path of the source file on the local host. For Windows paths like c:\dir\file use pattern /C/dir/file.
*** Full Target File Path**	String	The full path of the target file to put on the target host.
*** Host Name**	String	The host name or IP address to connect to.
*** User Name**	String	

Name	Type	Description
		The user name to use to authenticate with the remote host.
Keystore File	String	The private key file to use if the password is not used.
Keystore Password	Password	The password for the private key file if needed.
Password	Password	The password to use to authenticate with the remote host.
Ssh Port	int	The port to use to connect to the remote host.

Run Command (SSH)

Run a command on a remote host using SSH protocol.

Input Values

Name	Type	Description
*** Command**	String	The command to run on the target host.
*** Host Name**	String	The host name or IP address to connect to.
*** User Name**	String	The user name to use to authenticate with the remote host.
Error Output File Path	String	Full path of the file the error output is written to.
Expected Return Value	Integer	Expected return value of the command. If specified and the actual return value are not equal, the action fails.
Force Tty	Boolean	Force false TTY allocation for ssh
Keystore File	String	The private key file to use if the password is not used.
Keystore Password	Password	The password for the private key file if needed.
Output Max Size	int	The maximum size of the Standard Output and Error Output parameters.
Password	Password	The password to use to authenticate with the remote host.
Ssh Port	int	The port to use to connect to the remote host.

Standard Output File Path	String	Full path of the file the standard output is written to.
Timeout	int	Cancel the process if it passes the limit time in seconds. If 0 - no timeout

Output Values

Name	Type	Description
Command Error Output	String	The error output of the command.
Command Standard Output	String	The standard output of the command.
Return Value	int	The return value of the command.

System Actions

Contents

Check Free Disk Space

Checks the amount of free disk space on the specified drive or volume name.

Input Values

Name	Type	Description
*** Drive or Volume Name**	String	Name of partition or disk. **Example:** 'c:\', '/mnt/hda1', '/tmp'.
*** Expected Disk Space**	long	Number of free megabytes to compare with the actualDiskSpace free megabytes of disk space
Operator	ComparisonOp	Comparison operator to apply on the actual value vs. expected. Options can be selected from the list.

Output Values

Name	Type	Description
Actual Disk Space	long	The actual disk space as measured in the host.

Check number of CPUs

Checks the amount of CPUs on the host the step runs in.

Input Values

Name	Type	Description
*** Expected Number Of CP Us**	int	Number of CPUs to check for.
Operator	ComparisonOp	Comparison operator to apply on actual value vs. expected. Options can be selected from the list.

Output Values

Name	Type	Description
Actual Number Of CP Us	int	The actual number of CPUs measured in the host.

Check the Amount of Physical Memory

Checks the total physical memory amount.

Input Values

Name	Type	Description
*** Expected Memory Amount**	long	Amount of the total physical memory in megabytes to compare with the actual megabytes of RAM.

Operator	ComparisonOp	Comparison operator to apply on the actual value vs. expected. Options can be selected from the list.

Output Values

Name	Type	Description
Actual Memory Amount	long	Amount of the total physical memory in megabytes.

Get Agents for Execution Server

Retrieves the connected agents for an execution server.

Input Values

Name	Type	Description
*** Node Id**	String	Execution server Node ID

Output Values

Name	Type	Description
Agent IPs	String[]	An array of connected agents IPs.
Agent Names	String[]	An array of connected agents host names.
Agent Node IDs	String[]	An array of connected agents Node IDs.

Get Current Date

Used to set the current date to a parameter. The result date is the time when the step executes.

Input Values

Name	Type	Description
Amount To Add	long	Amount of time to add to the date. (To subtract use a negative number).
Date Format	String	String format to use. Default is yyyy/MM/dd HH:mm:ss .
Time Unit	TimeUnit	The time unit which the Amount To Add field refers to.

Output Values

Name	Type	Description
Date As Long	long	Date as number of milliseconds from 00:00:00 GMT, January 1, 1970.
Date As String	String	Date as string. String format is specified by the Date Format input.

Get Current Date and Time

Provides information on the current Data and Time of the computer it is executed on.

Input Values

Name	Type	Description
Date Format	String	The date format to use to set formattedDate.
Time Format	String	The time format to use to set formattedTime.

Output Values

Name	Type	Description
Current Date	Date	The current Date as a Date Object.
Formatted Date	String	The current date formatted as a String.
Formatted Time	String	The current time formatted as a String.
Time In Millis	Long	The number of milliseconds since January 1, 1970, 00:00:00 GMT represented by the date.

Get Unreachable Agents

Retrieves the unreachable agents.

Output Values

Name	Type	Description
Agent IPs	String[]	An array of unreachable agents IPs
Agent Names	String[]	An array of unreachable agents host names
Agent Node IDs	String[]	An array of unreachable agents Node IDs

Restart Agent

Restarts an agent process

Input Values

Name	Type	Description
Update Properties Path	String	The path of the update properties file. Use explicit values or file parameters.

Restart Host

Restarts the local computer that the step runs on. Supports Linux, Unix, and Windows systems.

Input Values

Name	Type	Description
Event Log Description	String	The description is entered in the Windows Event Log as the description of the restart operation.
Reboot Command	String	Comand line to execute for reboot in POSIX systems. Leave empty for autodetection.

Update Environment Variable

Updates the environment variables. In POSIX system updates only the process environment. In Windows it also updates the system environment.

Input Values

Name	Type	Description
*** Variable Name**	String	The name of the environment variable to update.
*** Variable Value**	String	The new value for the environment variable.
Create If Not Exists	Boolean	Whether to create the environment variable if it does not exist.

Text Actions

Contents

Add Text to a File

Inserts text into a file. Specify the text that is to be searched in the file and add the Text To Add input after it.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file where the text is added.
*** Text To Add**	String	The text to be added. The input also accepts one of the String Parameters except for the String Array Parameter.
Add After This Text	String	A text to search for in the file. The specified Text To Add input is added after.

Name	Type	Description
Add Text At The Beginning	Boolean	Whether to add the text at the beginning, True for beginning, False for end. (Relevant only if Add After This Text is empty).
Added new line character at the end of the text added	Boolean	Whether to add a new line character at the end of the text added.
Added text starts with new line.	Boolean	Whether to start the added text with a new line character.
Create File If Not Exist	Boolean	Create the file if it does not exist.
File Encoding	String	The encoding of the specified file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets
Is Regex	Boolean	Whether to treat Add After This Text as a regular expression.

Arrays - Read String Array from Text File

Reads a text file and puts every line into a new array element.

Input Values

Name	Type	Description
* **Filename**	String	The file name to read.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Output Values

Name	Type	Description
Result Array	String[]	The result array.

Arrays - Write Array to Text File

Writes an array to a file.

Input Values

Name	Type	Description
*** Array**	Object[]	The array to write.
*** Filename**	String	The file name to write to.
Append	Boolean	True to append the file or False to rewrite the file.
Create File If Not Exists	Boolean	Indicates whether to create the file if it does not exist.
File Encoding	String	The Encoding of the specified file. For example ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Extract Text from a File

Extracts text from a file by regular expression and stores the matched groups as defined by the result template input.

The extraction result is stored to a String Parameter.

Example: if a regular expression is "url=http://(server\d)/(path\d)", the result template is "$2 on $1" and file contains the line "url=http://server8/path5", the extraction result is "path5 on server8". Refer to http://www.regular-expressions.info/ for more details.

Input Values

Name	Type	Description
*** Filename**	String	The file from where the text is to be extracted.
*** Regular Expression**	String	The text to find in the file. Refer to http://www.regular-expressions.info/ for more details about regular expressions syntax.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets
Result Template	String	The template for the extraction result. Use $n for group n in regular expression input. Refer to step help for an example. To get character $ type \$.

Output Values

Name	Type	Description
Extraction Count	int	

Name	Type	Description
		The number of extraction results in the file.
Extraction Results Array	String[]	An array of the extraction results in the file.
First Extraction Result	String	The result of the extraction.

Find Text in File

Finds a word or a string in a text file.

Important! For regular expression searches, the supported files are limited to 10 MB.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file to search in.
*** Text To Find**	String	The text to find in the file. To search using Regular Expressions set the Is Regular Expression input to True. Refer to http://www.regular-expressions.info/ for more details.
Case Sensitive	Boolean	To consider the character case. True for case-sensitive or False for case-insensitive.
File Character Set	String	The file character set if it is known. **Example:** Unicode or Windows-1255. If the input is left blank, the step starts with the systems default character set and if the text is not found another try is done with Unicode.
Regular Expression	Boolean	Whether to search using Regular Expressions. If True, the Text To Find input is considered as a Regular Expression.
Text Should Exist	Boolean	Whether to expect the text to be found or not. True, to be found or False, not to be found.
Timeout	int	Timeout to wait before fail (action retries until success or timeout), the units are in seconds.

Find Text in HTML

Checks for the existence of a text in the HTML that is retrieved from the specified URL.

Input Values

Name	Type	Description
*** Target URL**	String	The URL to retrieve the HTML from.
*** Text To Match**	String	The text to search in the retrieved HTML.
Is Post	Boolean	Whether to use POST or GET method, default false for GET.
Is Unicode	Boolean	Whether the HTML that is retrieved from the specified URL is in Unicode, True for Unicode or False for not.
Maximum Time To Wait	int	Maximum time amount in seconds to wait. Is relevant if the Repeat Until Found input is set to True.
Parameters	String[]	URL parameters in case of POST HTTP method. Each line in array should be in form of key=value.
Password	Password	The user password to use.
Repeat Until Found	Boolean	Whether to repeatedly retrieve the HTML from the specified URL until the text is matched.
Time To Wait Between Retries	Long	The amount of time in seconds to wait between retries. Is relevant if the Repeat Until Found input is set to True.
Use Preemptive Authentication	Boolean	Whether to use Preemptive Authentication when using user name to log in to the web site.
User Name	String	The user name to use (leave empty to not use any user name).

Log to File on Center Machine

Allows the user to write a message to a CSV or a Text file on the center computer.

The CSV file format is: [time stamp],[agent],[message].

The Text file format is: [time stamp]tab[agent]tab[message].

Input Values

Name	Type	Description
*** File Path**	String	The path of a file. If the file does not exist it is created.

Name	Type	Description
File Type	FileType	Select the log file type. Selecting TXT the values delimiter is tab. Selecting CSV the values delimiter is "," and each of the values is wrapped with ".
Message	String	The message to be logged.
Messages array	String[]	Array of messages. The array values are separated by the file delimiter.

Read String from Text File

Reads a specified text file. Used to read the file contents in a String Parameter. The file size should not exceed 100 KB.

Input Values

Name	Type	Description
*** Filename**	String	The file name to read.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://www.iana.org/assignments/character-sets

Output Values

Name	Type	Description
Result String	String	The result string.

Replace Text in File

Replaces a string in a text file. The step fails if either the file does not exist or the text is not found. Files that are less then 10 MB are supported.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file to replace in.
*** Text To Find**	String	The text to find in the file. To search using Regular Expressions set the Is Regular Expression input to True. Refer to http://www.regular-expressions.info/ for more details.
*** Text To Replace**	String	

Name	Type	Description
		The text to replace the occurrences of Text To Find input in the file.
Escape Replacement	Boolean	Set to False if the replacement text is already escaped in case of regular expression.
Fail If Text Not Found	Boolean	Whether to fail the action if the text is not found.
File Encoding	String	The encoding of the file. **Example:** ISO-8859-1, US-ASCII, UTF-8, UTF-16. You can find more at http://docs.oracle.com/javase/6/docs/technotes/guides/intl/encoding.doc.html
Is Regular Expression	Boolean	Whether to search using Regular Expressions. if True, the Text To Find input is considered as a Regular Expression.

Replace Text in Multiple Files

Replaces a string in a many text files. The step fails if one of the files does not exist. Files that are less then 10 MB are supported.

Input Values

Name	Type	Description
*** Files Paths**	String[]	The paths to the files.
*** Text To Find**	String	The text to find in the every file. To search using Regular Expressions set the Is Regular Expression input to True. Refer to http://www.regular-expressions.info/ for more details.
*** Text To Replace**	String	The text to replace the occurrences of Text To Find input in the files.
Escape Replacement	Boolean	Set to False if the replacement text is already escaped in case of regular expression.
File Encoding	String	The encoding of the file.
Is Regular Expression	Boolean	Indicates whether to search using Regular Expressions, if true the Text To Find input will be considered as a Regular Expression.
May Not Contain The Text	Boolean	

Name	Type	Description
		Whether step fails if text is not found. Set to True if some of the files do not contain the text to replace.

Strings - Check For Substring In Source

Checks if a substring appears in the source.

Input Values

Name	Type	Description
*** Source**	String	Input string to the source.
*** Substring**	String	Input string to the substring.
Ignore case	Boolean	True the action only checks the characters sequence and ignores cases.
Search Pattern	SubstringSearchPattern	The search pattern for running the action. Starts with, contains, and ends with.

Output Values

Name	Type	Description
Starting Index	int	The index in the source of the first occurrence of the specified substring.

Strings - Extract Regular Expression Text From String

Extracts text from a String based on Regular Expression.
The extraction result can be modified using the Result Template input.
Example: If the regular expression is "url=http://(server\d)/(path\d)", the result template is "$2 on $1" and Sourece String is "url=http://server8/path5", the extraction result is "path5 on server8".

Refer to http://www.regular-expressions.info/ for more details about regular expressions.

Input Values

Name	Type	Description
*** Regular Expression**	String	The regular expression to use.
*** Source String**	String	The String to extract the text form.
Ignore If Not Found	Boolean	Whether to ignore if the text is not found.
Result Template	String	

Name	Type	Description
		The template for the extraction result. If no template is specified, the result is not modified. Use $n for group n in regular expression input. Refer to step help for an example. To get character $ type \$.

Output Values

Name	Type	Description
Num of Occurrences	int	The number of occurrences of the Regular Expression in the String
Split String	String[]	The split string according to the Regular Expression provided.
Target String	String	The text matched by the Regular Expression specified.
Target Strings	String[]	An array of the text occurrences matched by the regular expression.

Strings - Trim String

Removes the prefix and suffix whitespace characters, such as spaces, new lines, and tabs.

Input Values

Name	Type	Description
*** String**	String	Input string to trim

Output Values

Name	Type	Description
Trimmed String	String	The new string

Web Actions

Contents

- REST Operation
- SOAP Request

Access URL

Accesses the specified URL with an optional validation check of the HTTP response code.

Input Values

Name	Type	Description
*** Url**	String	Specifies the HTTP URL.
Authentication Method	AuthMethod	Specifies the authentication method of the HTTP servers or proxies. Specify one of the following methods: ▪ Basic ▪ Digest ▪ NTLM If no authentication is required, specify NONE.
Check Response Code	Boolean	Indicates whether the response code should be match an expected response. If the flag is set to true and the response code does not match the expected response, the action fails.
Connection Timeout (seconds)	Integer	Specifies the time to wait for a connection to establish. The default value is 60 seconds. Use 0 seconds to prevent a connection timeout.
Domain	String	Specifies the server domain. The Domain applies only if the Authentication Method is not set to NONE.
Expected Code	Integer	Specifies an expected HTTP response code. The action fails if the expected code is not returned. If you do not specify an Expected Code, any response code in the range 200-299 succeeds.
Is Post	Boolean	

Name	Type	Description
		Indicates whether to use the POST or GET method. The default value is false to specify GET. Use true to specify POST.
Parameters	String[]	Specifies URL parameters for the POST HTTP method. Specify each line in the array in key=value format.
Password	Password	Specifies the user password. The Password applies only if the Authentication Method is not set to NONE.
Proxy Server Name	String	Specifies the name of the proxy server.
Proxy Server Port	Integer	Specifies the port of the proxy server.
Read Timeout (seconds)	Integer	Specifies the time to wait for data to return. Use 0 seconds to prevent a read timeout.
Realm	String	Specifies the realm to use to authenticate the user. The Realm applies only if the Authentication Method is BASIC or DIGEST.
Response File Path	String	Specifies the file path to save the response to.
Use Preemptive Authentication	Boolean	Indicates whether to use Preemptive Authentication when the user logs in.
User Name	String	The user name for authentication. (only relevant if Authentication Method is NOT "NONE")

Output Values

Name	Type	Description
HTML Response	String	The response as a string.
HTTP Response Code	int	The HTTP response code. If no connection is made, the HTTP Response Code is -1.

Find Text in HTML

Checks for the existence of a text in the HTML that is retrieved from the specified URL.

Input Values

Name	Type	Description
*** Target URL**	String	The URL to retrieve the HTML from.
*** Text To Match**	String	The text to search in the retrieved HTML.
Is Post	Boolean	Whether to use POST or GET method, default false for GET.
Is Unicode	Boolean	Whether the HTML that is retrieved from the specified URL is in Unicode, True for Unicode or False for not.
Maximum Time To Wait	int	Maximum time amount in seconds to wait. Is relevant if the Repeat Until Found input is set to True.
Parameters	String[]	URL parameters in case of POST HTTP method. Each line in array should be in form of key=value.
Password	Password	The user password to use.
Repeat Until Found	Boolean	Whether to repeatedly retrieve the HTML from the specified URL until the text is matched.
Time To Wait Between Retries	Long	The amount of time in seconds to wait between retries. Is relevant if the Repeat Until Found input is set to True.
Use Preemptive Authentication	Boolean	Whether to use Preemptive Authentication when using user name to log in to the web site.
User Name	String	The user name to use (leave empty to not use any user name).

REST Operation

Performs REST operation for RESTful API calls.

Input Values

Name	Type	Description
*** URI**	String	REST URI to access. URI must contain Scheme (http/https/..)

Name	Type	Description
Authentication Method	AuthMethod	Can be set to each of the following methods: Basic, Digest or NTLM. Can be used to authenticate with http servers or proxies. NONE means no authentication is required.
Body	String	HTTP Body in case of POST/PUT.
Check Response Code	Boolean	The flag indicates whether actual response code should be matched against expected. If the flag is true and the actual response code is not equal to expected, the step fails.
Connection Timeout (seconds)	Integer	Specifies the time to wait for a connection to establish. The default value is 60 seconds. Use 0 seconds to prevent a connection timeout.
Domain	String	The server's domain. (Relevant if Authentication Method is not NONE)
Expected Code	Integer	Expected HTTP return code. Fails if expected code is not returned. If left empty will accept any code in the range 200-299
Http Headers	String[]	HTTP headers. Each line in array should be in form of key:value.
Include HTTP Headers In Response	Boolean	Set to true if you want HTTP headers to be included in the response.
Parameters	String[]	HTTP Parameters in case of POST. Each line in array should be in form of key=value.
Password	Password	The users password. (Relevant if Authentication Method is not NONE)
Proxy Server Name	String	The name of the proxy server
Proxy Server Port	Integer	The port of the proxy server
Read Timeout (seconds)	Integer	Specifies the time to wait for data to return. Use 0 seconds to prevent a read timeout.
Realm	String	The realm that authenticates the user. (Relevant if Authentication Method is "BASIC" or "DIGEST")

Name	Type	Description
Response File Path	String	File path to save the response to.
Rest Verb	HTTPMethodType	REST verb to perform.
Use Preemptive Authentication	Boolean	Whether to use Preemptive Authentication when using the username to log on o the web site.
User Name	String	The user name for authentication. (Relevant if Authentication Method is not "NONE")

Output Values

Name	Type	Description
HTTP Response Code	int	HTTP return code. -1 will be returned if no HTTP connection could be made.
Response As String	String	The response as string.
Response Headers As String	String	Response headers as string.

SOAP Request

Accesses WebService endpoint with the specified SOAP request and saves the response in a file.

Input Values

Name	Type	Description
*** End Point**	String	WebService endpoint URL to access. URL must contain Scheme (http/https/..)
*** Request Filename**	String	Filename of the XML file that holds the SOAP request.
*** Response Filename**	String	Filename to store the response.
Additional Headers	String[]	Additional HTTP headers. Each line in the array should be in the form of key:value.
Authentication Method	AuthMethod	Can be set to each of the following methods: Basic, Digest or NTLM. Can be used to authenticate with http servers or proxies. "NONE" means no authentication is required.
Connection Timeout (seconds)	Integer	Specifies the time to wait for a connection to establish.

Name	Type	Description
		Default value is 1 minute. A value of zero means the timeout is not used.
Domain	String	The server's domain. (only relevant if Authentication Method is not "NONE")
Expected Code	Integer	Expected HTTP return code. Fails if expected code is not returned. If left empty it will accept any code in the range 200-299
Password	Password	The users password. (only relevant if Authentication Method is not "NONE")
Proxy Server Name	String	The name of the proxy server
Proxy Server Port	Integer	The port of the proxy server
Read Timeout (seconds)	Integer	Specifies the time to wait for data to return.. Use 0 seconds to prevent a read timeout..
Realm	String	The realm that authenticates the user. (Relevant if Authentication Method is "BASIC" or "DIGEST")
SOAPAction header	String	The optional SOAPAction header element, required for some web services
Use Preemptive Authentication	Boolean	Whether to use Preemptive Authentication when using username to log onto the web site.
User Name	String	The user name for authentication. (Relevant if Authentication Method is not "NONE")

Output Values

Name	Type	Description
HTTP Response Code	int	HTTP return code. -1 is returned if no HTTP connection could be made.
Response	String	Output parameter that holds the response of the request.

XML Actions

Contents

Delete an XML Attribute Using XPath

Deletes an XML attribute selected by a XPath query.

Input Values

Name	Type	Description
*** X Path Query**	String	The XPath query to select the element that the specified attribute is deleted from.
*** XML Attribute Name**	String	The name of the XML attribute to delete.
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	Prefix the default namespace is mapped to.
Deletion Method	XpathDeleteAttribute$DeleteMode	Indicates which the action to take if the XPath Query returns several elements: First element found, Last element found, or all of the elements that are found.
Is Namespace Aware	Boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace please use the Default Namespace Prefix parameter.

Output Values

Name	Type	Description
XML Attribute Found	Boolean	Indicates whether the XML attribute was found and deleted.

Delete an XML Element Using XPath

Deletes an XML Element selected by a XPath query.

Input Values

Name	Type	Description
*** X Path Query**	String	The XPath query to select the element for deletion by.
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	Prefix the default namespace is mapped to.
Deletion Method	XpathDelete$DeleteMode	Indicates which action to take if the XPath Query returns several elements: First element found, Last element found, or all of the elements that were found.
Is Namespace Aware	Boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace please use the Default Namespace Prefix parameter.

Output Values

Name	Type	Description
XML Element Found	Boolean	Indicates whether the XML Element was found and deleted.

Execute an XPath Query

Executes the specified XPath query on the specified XML file and returns the XML Elements Values.

Input Values

Name	Type	Description
*** X Path Query**	String	The XPath query to execute.

Name	Type	Description
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	Prefix the default namespace is mapped to.
Is Namespace Aware	Boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace, please use the Default Namespace Prefix parameter.

Output Values

Name	Type	Description
First XML Element Value	String	The first value of the first XML Element returned by the exection of the specified XPath query
XML Elements Count	int	Number of XML Elements found.
XML Elements Values	String[]	An array of the values of the XML Elements returned by the execution of the specified XPath query.

Execute Multiple Queries Using XPath

Executes multiple XPath queries on the specified XML file, and returns the XML Elements Values (in multiple results).The main query is a mandatory input parameter to retrieve the main element. The other queries are intended for sub elements queries relative to the main element. For example, given the following XML: <item><name>example</name></item>

and Xpath Main Query of '//item' and XPath Query1 'name', then XPath Result1 would contain: <name>example</name>

Input Values

Name	Type	Description
*** XML Content/ File Path**	String	The path to the xml file or a string in xml format.
*** Xpath Main Query**	String	Xpath to the main target element. The entire can be executed relatively to the result of the query.
Default Empty Query Result	String	The default value result for any sub query that is returned with no result.
Fail If Element Is Not Full	Boolean	

Name	Type	Description
		Fail the action if one of the main or sub queries does not return a value.
Ignore Namespaces	Boolean	When set to True Xpath can be written without the namespace prefix. If set to False - Do not use the default namespace within the target XML.
Xpath Query1	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query10	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query2	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query3	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query will be irrelevant.
Xpath Query4	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query5	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query6	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query7	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query8	String	

Name	Type	Description
		Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.
Xpath Query9	String	Xpath Query which can be written relatively to the main query. If an absolute path is written (e.g. using '//') the main query is irrelevant.

Output Values

Name	Type	Description
Has Empty Result	Boolean	Whether one of the queries (Main or Sub) did not return a result.
Xpath Result1	String	Xpath Result of sub query. 1
Xpath Result10	String	Xpath Result of sub query. 10
Xpath Result2	String	Xpath Result of sub query. 2
Xpath Result3	String	Xpath Result of sub query. 3
Xpath Result4	String	Xpath Result of sub query. 4
Xpath Result5	String	Xpath Result of sub query. 5
Xpath Result6	String	Xpath Result of sub query. 6
Xpath Result7	String	Xpath Result of sub query. 7
Xpath Result8	String	Xpath Result of sub query. 8
Xpath Result9	String	Xpath Result of sub query. 9

Insert an XML Attribute Using XPath

Insert the specified attribute in the element selected by the specified XPath query.

Input Values

Name	Type	Description
*** X Path Query**	String	The XPath query to select the element that the specified attribute is inserted to.
*** XML Attribute Name**	String	The name of the XML attribute to insert.
*** XML Attribute Value**	String	The value to set to the inserted attribute.
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	

Name	Type	Description
		Prefix the default namespace is mapped to.
Insertion Method	XpathInsertAttribute$InsertMode	Indicates which action to take if the XPath Query returns several elements: First element found, Last element found, or all of the elements that were found.
Is Namespace Aware	boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace please use the Default Namespace Prefix parameter.

Output Values

Name	Type	Description
XML Attribute Found	Boolean	Indicates whether the XML attribute was inserted.

Insert an XML Element Using XPath

Inserts the specified XML Element as a child of XML Element selected by XPath query.

Input Values

Name	Type	Description
*** X Path Query**	String	The XPath query to select the element to insert the specified element in.
*** XML Element**	String	The XML Element to insert in string format, explicit values can be set or use String Value Parameter. If inserting a comment specify just the body of the comment without the &lt!-- and -->.
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	Prefix the default namespace is mapped to.
Inserted Element Location	XpathInsert$InsertPlace	Indicates where to insert the new XML Element relative to the selected XML Element childrent.
Insertion Method	XpathInsert$InsertMode	

Name	Type	Description
		Indicates which action to take if the XPath Query returns several elements: First element found, Last element found, or all of the elements that were found.
Is Comment	Boolean	Indicates whether it is a actually a comment that is being inserted.
Is Namespace Aware	boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace, please use the 'Default Namespace Prefix' parameter.

Output Values

Name	Type	Description
XML Element Was Found	boolean	Indicates whether the XML Element was found.

Update an XML Attribute Using XPath

Updates the specified attribute's value on the selected XML Element by XPath query.

Input Values

Name	Type	Description
*** New XML Attribute Value**	String	The new value to set to the specified attribute.
*** X Path Query**	String	The XPath query to select the element that the specified attribute will be inserted to.
*** XML Attribute Name**	String	The name of the XML attribute to update.
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	Prefix the default namespace is mapped to.
Is Namespace Aware	Boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace please use the Default Namespace Prefix parameter.
Update Method	XpathUpdateAttribute$UpdateMode	Indicates which action to take in case the XPath Query returns several elements: First element

Name	Type	Description
		found, Last element found, or all of the elements that were found.

Output Values

Name	Type	Description
XML Attribute Found	boolean	Indicates whether the XML attribute was updated.

Update an XML Element using XPath

Updates the value of an XML Element selected by XPath query.

Input Values

Name	Type	Description
*** New XML Element Value**	String	The new value to set to the XML Element.
*** X Path Query**	String	The XPath query to select the element to update.
*** XML File Path**	String	The path to the XML file to perform the operation on.
Default Namespace Prefix	String	Prefix the default namespace is mapped to.
Is Namespace Aware	Boolean	Determines whether the XPath query is namespace aware or not. When using a default namespace please use the Default Namespace Prefix parameter.
Update Method	XpathUpdate$UpdateMode	Indicates which action to take if the XPath Query returns several elements: First element found, Last element found, or all of the elements that were found.

Output Values

Name	Type	Description
XML Element Was Found	Boolean	Indicates whether the XML Element was found.

Deprecated Actions

Contents

Check if Nolio Agent has Root Permissions (deprecated)

Checks if the process that the step runs in has root permissions. The step supports Unix/Linux systems only.

Check Service Status (deprecated)

Checks whether a certain service exist and can also check whether it is in a certain status

Input Values

Name	Type	Description
*** Check Status**	Boolean	True if the service status is to be checked
*** Fail If In Status**	Boolean	

Name	Type	Description
		True if the action is to fail if it is in the expected status
*** Service Name**	String	The service name or display name of service to start. The name specified is case-insensitive and must be the exact service name or exact display name.
*** Service Should Exist**	Boolean	True if the service exists or False if not
Expected Status	CheckServiceStatus$ServiceStatus	The Expected status of the service

Output Values

Name	Type	Description
Actuall Status	String	The Actual Service Status

Copy Single File or Folder (deprecated)

Copies the file or folder from the source path to the target path.

Input Values

Name	Type	Description
*** Source Path**	String	The path of the source file or folder. Explicit values can be set or use file parameters.
*** Target Path**	String	The path of the target file or folder.

Delete File (deprecated)

Deletes a file in the file system.

Input Values

Name	Type	Description
*** File Path**	String	The path to the file to delete.
Fail If Not Exist	boolean	Indicates whether to fail if the path does not exist, True to fail or False to ignore.

Delete Folder (deprecated)

Deletes a folder in the file system.

Input Values

Name	Type	Description
*** Folder Path**	String	The path to the folder to delete.
Fail If Not Empty	Boolean	Indicates whether to fail if the folder is not empty or to delete its content, True to fail or False to delete its content.
Fail If Not Exist	Boolean	Indicates whether to fail if the path does not exist, True to fail or False to ignore.

Output Values

Name	Type	Description
Failed File Names	String[]	Array of file names that could not be deleted. (Might be a partial list only if many files could not be deleted)

Execute an XPath Query (deprecated)

Executes the specified XPath query on the specified XML file, and returns the XML Elements Values.

Input Values

Name	Type	Description
*** X Path Query**	String	The XPath query to execute.
*** XML File Path**	String	The path to the XML file to perform the operation on.

Output Values

Name	Type	Description
First XML Element Value	String	The first value of the first XML Element returned by the exection of the specified XPath query
XML Elements Values	String[]	An array of the values of the XML Elements returned by the execution of the specified XPath query.

Name	Type	Description
XML Elements Were Found	boolean	Indicates whether any XML Elements were found.

Find Files or Folders (deprecated)

Returns the paths of files or folders that match the wildcard.

Input Values

Name	Type	Description
*** Name Format**	String	Pattern of names to search for. The pattern can contain regular characters: '?' and '*'. '?' replaces any single character and '*' replaces any string.
*** Root Folder Path**	String	The path of the folder to start the search from.
After Date	String	Date in format yyyy/MM/dd HH:mm:ss. Files last modified before, won't appear in the result.
Before Date	String	Date in format yyyy/MM/dd HH:mm:ss. Files last modified after, won't appear in the result.
Files Or Folders	Boolean	Indicates whether to search for files or folders, True for files or False for folders.
Find Recursively	Boolean	Indicates whether to search the specified pattern recursively, True for recursive or False for root folder only.
Ignore Case	Boolean	Indicates whether to ignore case of file names or not., True to ignore case or False to perform case sensitive search.
Return Full Path	Boolean	Indicated whether to return the file name along with the full path. You may encounter duplication if there are files with the same name in different directories
Sort By Last Modified	Boolean	Indicates whether to sort the found entries by the last modified date, set true to sort, the most recent will be the first, the sort if set is done first.
Sort By Name	Boolean	

Name	Type	Description
		Indicates whether to sort the found entries by the name, set True to sort, the order is ascending, the sort if set is done after the last modified sort.
Sort By Size	Boolean	Indicates whether to sort the found entries by their size on disk, set True to sort, the biggest is the first, the sort if set is done after the type sort.
Sort By Type	Boolean	Indicates whether to sort the found entries by their file extension, set True to sort, the order is ascending, the sort if set is done after the name sort.

Output Values

Name	Type	Description
First Name Found	String	The first name found that matches the search pattern. Is for more convenient usage if only one file is expected.
List Of Found Entries	String[]	An array of files or folders that matched the search pattern
Number Of Entries	int	The amount of files or folders that matched the search pattern.
Was Found	Boolean	Flag will become True if at least one file or folder matched the search pattern.

Get Value from an INI File (deprecated)

Finds a key in a section of an INI file and returns its value. If the INI file, section, or key is missing then the actionl fails. If the needed key is not in any section the INI Section Name input can be left blank.

Input Values

Name	Type	Description
*** INI File Path**	String	The path to the INI file.
*** INI Key Name**	String	The name of the key to find.
Encoding	String	Character encoding of ini file.
INI Section Name	String	The name of section that contains the key.

Output Values

Name	Type	Description
INI Key Value	String	The value of the found key. (ASCII, UTF-8, UTF-16 etc.)

Kill Process (deprecated)

The action is deprecated. Please use "Kill process" action instead.

Kills a process identified by its name. The step fails if either the process name could not be found or the running user does not have permission to kill the process.

Input Values

Name	Type	Description
*** Process Name**	String	The name of the process to kill.
Fail If Not Exist	boolean	Indicates whether to fail if the process could not be found or not enough permissions, True to fail or False to ignore.
Timeout	int	The number of seconds to wait for the process to shutdown. Minimum timeout is 1 second.

Manipulate INI File (deprecated)

Adds or updates an INI file. It is possible to specify a section that the added or update keys belong to. Leave the INI Section Name input blank if no section is required and in addition set the Section Should Exist input to False.

Input Values

Name	Type	Description
*** INI File Path**	String	The path to the INI file.
*** Key Value Array**	String[]	An array of strings, each string is in the format of 'key=value'.
Encoding	String	Character encoding of ini file. (ASCII, UTF-8, UTF-16 etc.)
INI Section Name	String	The name of the INI section that contains the keys.
Key Value Separator	String	Indicates the character used as a separator in each string in the Key Value Array input.
Section Should Exist	Boolean	Indicates whether to create the specified section if it does not exist. True for ignore or False for create.

Rename File or Folder (deprecated)

Renames the file or folder to the specified name. Only one file name can be renamed.

Input Values

Name	Type	Description
*** New File Name**	String	New file name. Can not contain any path, and is created in the original file location
*** Source File**	String	Path to the file or folder that needs to be renamed

Rename File or Folder (deprecated)

Copies the file or folder from the source path to the target path.

Input Values

Name	Type	Description
*** Source Path**	String	The path of the source file or folder. Explicit values can be set or use file parameters.
*** Target Path**	String	The path of the target file or folder.

Run Command Line (deprecated)

Executes the command line as if it was executed from command line shell. It is possible to execute the command line in the background and not wait for it to finish its execution by setting the Wait For Process To Finish input to False

Input Values

Name	Type	Description
*** Command Line**	String	The command line to run.
*** Work Directory**	String	The working directory to run from.
Environment Variables Names	String[]	A list of environment variables names to be set as the environment variables for the created process.
Environment Variables Values	String[]	A list of environment variables values to be set as the environment variables for the created process, the values should match the names order in the Environment Variables Names input.

Expected Return Value	Long	Expected return value of the finished process. If specified, and actual return value is not equal to the expected value, the step fails. The field is used only if one return code is expected.
Expected Return Values	Long[]	Expected return values of the finished process. The field is used if more than one possible return codes are expected.
Std Err File Path	String	The file path to use to write the error stream of the process. If not set, the standard error is redirected to the standard output
Std Out File Path	String	The file path to use to write the output stream of the process.
Success Return List	Boolean	If set to True, the expected return value list is used to determine if the command executed successfully. If set to False, the expected return value list is used to determine if the command failed.
Time Out Duration	long	Time to wait for the process in seconds. Relevant only if Wait For Process To Finish is set to True.
Wait For Process To Finish	Boolean	Indicates whether to wait for the process to finish its execution or not. True for wait or False not to wait. On non-windows systems the step may fail if the input is set to False.

Output Values

Name	Type	Description
Return Value	Long	Return value of the finished process. Long Parameter can be used with the output.
Std Err Output	String	Standard error of the process. Valid only if Wait for process to finish is set. A maximum of 256 characters is returned..
Std Out Output	String	Standard output of the process. Valid only if "Wait for process to finish" is set. A maximum of 256 characters is returned.

Name	Type	Description
Success Return List	Boolean	If set to True, the expected return value list is used to determine if the command executed successfully. If set to False, the expected return value list is used to determine if the command failed.

Run Process as User (deprecated)

Executes an executable as a specified user.

The user the steps process (CA Deployer process) runs under requires the appropriate permissions to execute a process as a different user.

On Windows systems, the user needs to have the Replace a process level token permission.

The permission can be set as follows: Administrative Tools -> Local Security Policy -> Local Policies -> User Rights Assignment -> Replace a process level token. On non-windows systems, the user needs to have 'root' permissions.

It is possible to start the process in the background and not wait for it to finish execution, by setting the Wait For Process To Finish input to False. Supported on Windows systems only.

On non-windows systems the step fails if this input is set to False.

Input Values

Name	Type	Description
* **Executable Path**	String	The path to the executable to run.
* **User Name**	String	The user name to run the process as.
* **User Password**	Password	The user password to run the process as.
* **Work Directory**	String	The working directory for the executable.
Enviroment Variables Names	String[]	A list of environment variable names to be set as the environment variables for the created process.
Enviroment Variables Values	String[]	A list of environment variable values to be set as the environment variables for the created process. The values should match the names order in the Environment Variables Names input.
Executable Arguments	String[]	

Name	Type	Description
		The list of arguments to pass to the executable.
Expected Return Value	Long	If set, the action fails if the process return value is different than the expected return value.
Std Err File Path	String	The file path to use to write the error stream of the process.
Std Out File Path	String	The file path to use to write the output stream of the process.
Wait For Process To Finish	boolean	Indicates whether to wait for the process to finish execution or not. True for wait or False not to wait. On non-windows systems the step fails if this input is set to False.

Output Values

Name	Type	Description
Return Value	long	Return value of the finished process. Long Parameter can be used with this output.

Rapid Development Kit 1.0

This Rapid Development Kit (RDK) is a complementary tool to CA Release Automation which enables non-JAVA programmers to create Action Packs easily.

Features:

- Supports Command Line, Scripts, and RESTful APIs
- Generates ready-to-use Action Packs with support for Windows and Linux
- Generates Action Pack documentation templates

Use of this tool assumes that end-users:

- Have an intermediate knowledge of CA Release Automation
- Be an SME on the specific application or system that is orchestrated through RA
- Be skilled in using CLI/Scripting, such as for cmd, bash, and PowerShell, and parsing output data through RegEx
- Be skilled in using RESTful Web Services and parsing output data through JSONPath or XPath

RDK Workflow

The Rapid Development Kit (RDK) interface allows for simple navigation and input across the workflow in creating and maintaining action packs and their respective actions. A basic workflow is as follows:

- Install and Configure the RDK
- Create or Modify Action Packs in RDK 1
- Create or Modify Actions in RDK 1 and maintain their properties:
  - Define input parameters
  - Define the execution payload you want to run and map the input parameters as arguments to the execution payload.
  - (Optional) Define additional output.
  - (Optional) Define additional execution results.
- Publish Action Packs in RDK 1:
  - Publish by exporting as a jar file.
  - (Optional) Export a documentation .xml file.

Install and Configure the RDK

Contents

Requirements

Verify that you meet at least the minimum requirements for installing the CA Release Automation Rapid Development Kit.

System requirements:

- Windows 2008 R2 or later
- RHEL 6.4 or later
- CA Release Automation 4.7.1 or later
- Oracle JDK 7 64-bit
- JAVA_HOME variable set to JDK installation folder
- Apache Tomcat 7 64-bit. Do not use the Tomcat instance included with CA Release Automation.

Install the Rapid Development Kit

Before installing the CA Release Automation Rapid Development Kit (RDK):

- Verify the system requirements
- Ensure that you have the following files for the setup procedure:
  - ca_ra_rdk_windows-x64_1_0_0.exe
  - ca_ra_rdk_unix_1_0_0.sh
  - ca_ra_rdk_third-party_windows-x64_1_0_0.exe
  - ca_ra_rdk_third-party_unix_1_0_0.sh

Note: Do not install the RDK on any production system where Release Automation is installed. Only install the RDK on a development or test system.

Installation is a two-part installer that has the main installer launching a third-party component installer.

Follow these steps:

1. Run for Windows or Linux, respectively:
   - ca_ra_rdk_windows-x64_1_0_0.exe
   - ca_ra_rdk_unix_1_0_0.sh

   The installer for the third-party components runs automatically.
2. When the installer prompts you, specify the Tomcat 7 installation folder.
3. When the installation is complete, restart the Tomcat server.

Launch the Rapid Development Kit

To launch the RDK, perform one of the following steps:

- Access the hostname of the tool, such as http://<host name>:8080/rdk.

  Note: Do not access RDK using localhost or FQDN for hostname.
- Select the desktop shortcut. By default, a shortcut is created during installation.

Upon launching, if you are not logged in, you are prompted to do so. Upon installation, the default user name and password are superuser and suser, respectively. Optionally, change the user name and password by modifying the **tomcat-users.xml** file in the conf folder in the Tomcat installation folder.

Settings

The Settings function allows you to reference or modify certain directory settings for the RDK on your system.

To launch the Settings, click the gear icon in the upper right of the UI. The information for reference includes:

- The RDK version information.
- The named Base Package with applicable action packs.
- The JAVA Home location.

Other fields include:

- A Working Directory storing all your action packs and files, including distribution, libraries, and source.
- The Execution Directory from which the CLI commands are executed locally.

Create or Modify Action Packs in RDK

Through the Release Automation RDK interface, you have the options of adding a new action pack or modifying the properties of an existing action pack. Optionally, an existing action pack can be duplicated and used as a template for a new action pack or deleted.

- To create an action pack and assign its properties, click Add Action Pack.
- To open the properties of an existing action pack, click its action pack icon.
- To delete an action pack, select the action pack and click the red X above its icon.
- To copy an action pack, including its assigned actions, select the action pack and click the Duplicate option. Provide a name for the duplicate action pack and click Duplicate. This function lets you make subsequent versions of an action pack and actions while preserving the original version/template.
- (Optional) Update the cube icon with, for example, an applicable vendor icon. To select an alternative .jpeg file for display, click the Browse button.

Properties

The properties that are designated for action packs in the RDK correspond to the properties used in the Release Automation interface. These properties include its category, description, and version. Release Automation uses the Primary Category when browsing for actions.

If the Primary Category for the action pack is left blank, the Action Category that is assigned to an action is used. This way, one action pack can contain actions that are split across multiple categories. Optionally, use . (dot) annotation to create nested categories. If a Primary Category is specified at the action pack level, this category becomes the parent category for all actions within that action pack.

Create or Modify Actions in RDK

Actions are the building blocks that are grouped in a defined sequence within a flow. You add actions or modify existing actions within the action pack. The RDK lets you assign a name to the action, a description, and, optionally, an Action Category. The Action Category is a subcategory of any Primary Category that is defined at the Action Pack level. Typically, when you integrate with a vendor product, they support only one type, such as CLI. In these cases, all the actions that are defined in the action pack are CLI actions. Alternatively, different action types can exist within a single action pack.

Each action is assigned:

- One or more Input Parameters
- An execution payload that can be run on Windows or Linux
- Mapped Output Parameters and filtering

Once an action is created in the RDK, the options below each action let you delete the action, or move and copy the action between action packs.

To open the Action Details pane, click Add Action and select CLI, SCRIPT, or RESTful. Alternatively, choose existing action to modify.

- **Name**. This name is displayed in the RA Actions browser.
- **Description**. This description is displayed in the RA Actions browser.
- **Action Category**. This field lets you group actions in subcategories by type or class. By default, an action is associated to the **Primary Category**, as defined for the parent action pack.

Define Input Parameters in RDK

Each action is defined with one or more Input Parameters. These parameters are flexible. You can, for example, use an entire URL as an input parameter. The parameters include:

- Name
- Type (String, Password, Boolean, Integer, Float, or Selection)
- Required flag
- Array, add items to the array
- Description
- Default Value

To create an additional Input Parameter, click the + button.

Note: The order in which input parameters are added correspond to their appearance in the RA action properties. Drag and drop to change their order.

Define CLIs or Scripts to Run

After you define Input Parameters, you determine what CLI or script to run. You can create Windows and Linux CLI/Scripts and can also upload your own pre-made script. While the Input and Output parameters stay relatively static across the Action Types, the scripting, the content of the action, varies. Creating action packs through the RDK does not require knowledge of JAVA or programming. The tool does assume a basic knowledge of scripting by the user.

CLI

You have two options for scripting: manual input or script upload. The scripting pane elements are:

- **Execution Path**: This field defines the directory from which the script is executed.
- **Available Input parameters**: This listing includes the parameters that are defined through the Input Parameters section, such as appName, location, and password. Drag and drop a defined Input Parameter into the string in the Command field. Placed Input Parameters can be arranged in a specific order. If no value is entered, a default value is provided.

- **Command**: The script to run for the action pack.
  By default, RDK executes the commands or script for which an interpreter is present. For example, if running Python, then you use the CLI Action Type. Then, you call explicitly the Python interpreter and provide it a command or script file reference and arguments as input. For example, python -c "print ('hello')".

- **Upload script**: This function allows you to upload a script or some other scripting language. Once uploaded, the script is referenced as ${SCRIPT_NAME} in the Command option.

- **Execute**: This function allows you to run the script locally and validate its output for accuracy.

After you define Windows script commands, you can similarly define for Linux, as necessary.

Advanced options

- **Wait for process to finish**. By default, there are no time constraints on a running script process. If the system times out during the process, a duration time (in seconds) can be set here. Clear the check box and set the **Timeout duration**. If the time-out lasts longer than the set time, the script exits and is marked as a failure.

- **Overwrite output file**

- **Environment variables** (Name = Value)

SCRIPT

To clarify, there are fundamental differences between the CLI and SCRIPT action types.

- A CLI action type executes both commands and scripts, and assumes the script accepts command line arguments.

- A SCRIPT action type executes a script directly against the OS default interpreter (cmd.exe or bash) or PowerShell (Windows only) and enables inline variable substitution.

For SCRIPT action types, the script pane elements vary depending on your selection of Default OS or PowerShell. Other elements, such as Available Input Parameters, remain the same.

Default OS

- **Execution Path**: This field defines the directory from which the script is executed.

- **Available Input parameters**: Drag and drop a defined Input Parameter into the script field. Placed Input Parameters can be arranged in a specific order. If no value is entered, a default value is provided.

- **Script**: The script to run for the action pack.

- **Upload script**: This function allows you to upload a script or some other scripting language.

- **Execute**: This function allows you to run the script locally and validate its output for accuracy. By default, RDK executes the commands or script for which an interpreter is present.

After you define Windows script commands, you can similarly define for Linux, as necessary.

- **Advanced options**

- **Wait for process to finish**. By default, there are no time constraints on a running script process. If the system times out during the process, a duration time (in seconds) can be set here. Clear the check box and set the **Timeout duration**. If the time-out lasts longer than the set time, the script exits and is marked as a failure.
- **Overwrite output file**
- **Environment variables** (Name = Value)

PowerShell

- **Working Directory**: The directory from which the script is executed.
- **Advanced options**
  - **Execution Timeout** (in seconds)
  - **PowerShell Options** (Name = Value) For example, ExecutionPolicy=Bypass

RESTful

Elements in the RESTcall section, such as Available Input Parameters, are the same across the action types. Others are unique to the RESTful type.

- **Available Input parameters**: Drag and drop a defined Input Parameter into the Request URL field.
- **Execute RESTful Call**: This function allows you to execute locally and validate its output for accuracy. When executed, the RDK executes the given request method on the Request URI.

Target

- **Request URI:** The Uniform Resource Identifier with which to communicate.
- **Request Method**: GET, POST, PUT, or DELETE.
- **Request Timeout**: The time limit for the request, in seconds.
- **Request Headers**
- **Body Content** / Content Headers. These do not display with GET requests.
  Under Request Parameters, guided works with key value pairs, such as myKey=myValue.
  When sending structured data, such as JSON or XML, use RAW instead:

  1. Under Body Content, click the Switch to RAW button.
     The Advanced Rest Client screen displays.
  2. Under the RAW tab, enter complex structured data, such as JSON or XML.
     The JSON data displays in the RAW Body field.

Authorization

- An Authorization header must be chosen among the four: No Authentication, Basic Authentication, NTLM, or Digest Authentication. If no authentication is chosen and you attempt to save the action, then the system flags the action as incomplete. Basic Authentication adds automatically a user and password to the Input Parameters listing.

Define Output Parameters and Filtering in RDK

Each action is defined with one or more Output Parameters.

Note: If nothing is defined, then the default output parameter is Execution Results and the action produces all output.

Once defined, the subsequent action inputs use the output values. In defining the output, you assign value to the output. This output can be validated and displayed in the Script Execution Results. The parameters are:

- Output Parameter Name
- Type (for example, String)
- Array
- Description
- Filter Type (XPATH, JSONPath, Reg Exp) (Note: N/A if the source is Exit Code)
- Source (Standard Out, Standard Error, Exit Code)
- Find Using Regular Expression
- (Optional) Create input parameter for output file location

To create an additional Output Parameter, click the + button.

Define Execution Results and Error Messages in RDK

The Results and Error Conditions section gives you control over how you are to define and display results for evaluation. This function is useful when, for example, you run a successful API Call that is nevertheless tagged as failed based on the output.

You define an error condition that is based on the action results, such as determining under which conditions an action is marked as a failure. The defined output parameters are combined with an Operator and Value, such as:

Execution Output (parameter) -- **Contains** (operator) -- **Exception** (value)

After you execute a script and the script returns, it shows success or failure that is based on the error conditions. If none of the error conditions match, then it returns the success message. If the conditions are not defined, the system defaults to successful.

Publish Action Packs in RDK

You can export an action pack using the Export function on the action pack interface.

- **Export as Jar**. This function generates code and prompts you to download a Jar file locally. You do not need knowledge of JAVA code or source generation in compiling the Jar file. Exporting an action pack as a Jar file lets you import the action pack directly into the Release Automation environment. Through Action Management in RA, you can browse, open, and import the Jar file.

- **Export Documentation**. This function generates code and prompts you to download the documentation XML file. The file contains action pack properties, and details for each action. These details include name, description, input parameters, and description.

RDK - Acknowledgements

The Rapid Development Kit (RDK) uses the following third-party software under license:

Apache Commons Codec

This product includes Apache Commons Codec 1.3 and is distributed in accordance with the Apache Software License v.2.0.

Apache HttpComponents Core

Copyright 2005-2011 The Apache Software Foundation

Apache Commons httpmime

This product includes Apache Commons httpmime 4.1

Apache HttpComponents HttpMime

Copyright 1999-2011 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS

Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

Apache Mime4J

This product includes Apache Mime4J 0.6, which is distributed in accordance with the following license agreement:

Apache JAMES Mime4j

Copyright 2004-2009 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Software Foundation

Parts of this product include software developed by the Apache Software Foundation. The Apache software is distributed in accordance with the following license agreement:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices

normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Apache wss4j

This product includes Apache wss4j 1.5.8

Apache WebServices - WSS4J

Copyright 2004-2009 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

This product includes software Copyright University of Southampton IT Innovation Centre, 2006 (http://www.it-innovation.soton.ac.uk).

Apache XMLSchema

This product includes Apache XMLSchema 1.4.2

This product includes software developed by The Apache Software Foundation (http://www.apache.org/). Portions Copyright 2006 International Business Machines Corp.

Apache xmlsec

This product includes Apache xmlsec 1.4.4

This product contains software developed by The Apache Software Foundation (http://www.apache.org/). It was originally based on software copyright (c) 2001, Institute for Data Communications Systems, <http://www.nue.et-inf.uni-siegen.de/>.

The development of this software was partly funded by the European commission in the <WebSig> project in the ISIS Programme.

CA Inc

CA, Inc. ("CA")

End User License Agreement (the "Agreement") for the CA software product that is being installed as well as the associated documentation and any SDK, as defined below, included within the product ("the Product").

Carefully read the following terms and conditions regarding your use of the Product before installing and using the Product. Throughout this Agreement, you will be referred to as "You" or "Licensee."

You are:

(I) Representing that you are not a minor, and have full legal capacity and have the authority to bind yourself and your employer, as applicable, to the terms of this Agreement;

(II) Consenting on behalf of yourself and/or as an authorized representative of your employer, as applicable, to be bound by this Agreement.

Castor

This product includes Castor software distributed in accordance with the following license agreement:

Copyright 2004-2005 Werner Guttmann

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

google-gson

This product includes google-gson 1.7.1, which is distributed in accordance with the Apache Software License v.2.0.

httpclient

This product includes httpclient 4.1.2, which is distributed in accordance with the following license agreement:

Apache HttpComponents HttpClient

Copyright 1999-2011 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS

Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

HttpComponents HttpCore

This product includes Apache HttpComponents HttpCore 4.1, which is distributed in accordance with the following license agreement:

Apache HttpComponents HttpCore

Copyright 2005-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Install4j

Install4J runtime libraries under license from ej-technologies (http://www.ej-technologies.com).

Custom Actions SDK

The Custom Actions Software Development Kit (SDK) enable software engineers to develop actions that are used in designing Automation Studio applications.

The Custom Actions SDK consists of:

- Separate zip file containing a CA plug-in for Eclipse IDE.
- Javadoc for describing API NolioAction classes for use in developing actions.
- Action template examples available within the plug-ins.

Features:

- Each custom action template is contained in a Java Archive (JAR) file.
- Custom action templates behave just like CA Release Automation action templates.
- Automatic distribution of new or changed actions to all agents.
- Overrides user changes during upgrade.
- During process execution, agents have access to both CA Release Automation actions and custom actions and checks both for the action.
- Any suitable process can use custom actions.

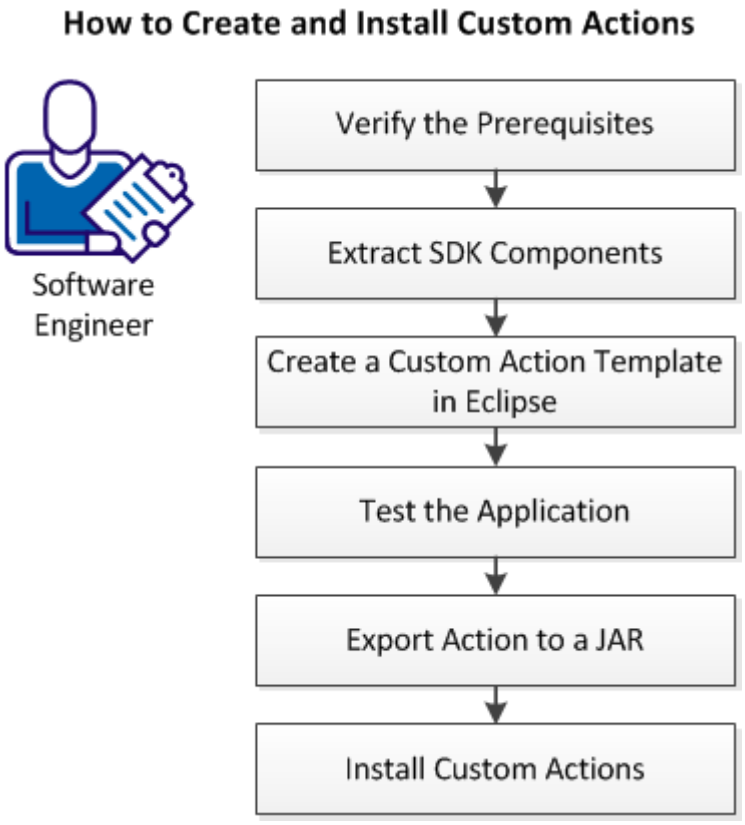
This section contains the following topics:

How to Create and Install Custom Actions

Contents

As a software engineer, you create custom actions by designing templates that are relevant to your installed base. Custom-built actions are installed and executed within Automation Studio. A custom action is a predefined operation that is customized for application components. CA Release Automation provides you with the tools to develop your own actions that are tailored to the workflow and tools of your organization. The creation of custom actions helps you achieve a deployment process that is integrated within the powerful CA Release Automation framework. (TK - lots of text, little value...we should rewrite this opening)

Use this scenario to guide you through the process:



how to create and install custom actions

1. Verify the Prerequisites.
2. Extract SDK Components.
3. Create a Custom Action Template in Eclipse.
4. Test the Application.
5. Export Action to a JAR.
6. Install Custom Actions.

Verify the Prerequisites

System processing and disk space must accommodate one of the development tools, Eclipse, the Custom Actions SDK and API Javadocs. Verify that you meet at least the minimum requirements for creating and installing custom actions.

Minimum version requirements are:

- Eclipse Indigo (for information about Eclipse IDE, see www.eclipse.org)
- Java 6 or 7
- CA Release Automation 4.7.0
- CA Release Automation Custom Actions SDK installed

Extract SDK Components

CA Release Automation provides development tool plug-ins for creating custom actions. Extract the following components to set up the custom actions development environment:

- Eclipse Plug-in
- NolioAction Classes

Follow these steps:

1. Extract Eclipse_Actions_SDK_4.7.zip to the Eclipse plugin folder.

   Note: Eclipse_Actions_SDK_4.7.zip is available under the CustomActionSDK folder of the CA Release Automation installation media.

2. Extract NolioAction Classes from the SDK_Java_Doc.zip (TK - Why is this needed?)

   Note: Available from the CA Web site.

Create a Custom Action in Eclipse

Creating custom action templates consists of creating a project in Eclipse then developing the actions. Custom templates function exactly like CA Release Automation action templates. All new or updated actions are automatically distributed to all agents. During a process execution, agents have access to both CA Release Automation actions and custom actions. Any suitable process can use custom actions. Actions are developed in the supported Java IDE using **NolioAction** classes. (TK - lots of text, little value...I would take this all out).

Follow these steps:

1. In Eclipse, Select File, New, Other, and select Java Project.
2. To create an action, right-click the project, select New, Other and select Nolio Action.
3. Enter Package and Name information.
4. Select Generate example to generate a complete example.
5. Click Finish.

Develop and Test the Custom Action

Once the action template is created, develop the code that the custom action is to run and test the action directly from Eclipse.

To test the action, follow these steps:

1. In Eclipse, click the Run button or hit Ctrl-F11.
2. In the console, at the prompt, supply all the inputs of the actions.
3. Verify that the action returns the expected result and parameter values.

Export Custom Action to a JAR

After the custom action is developed and tested, export it into a JAR file so that it can be imported into CA Release Automation.

Follow these steps:

1. In Eclipse, right-click the project.
2. Select Export and navigate to the location.
3. Enter the JAR Export criteria and click Finish.

Install Custom Actions

To add the custom actions CA Release Automation, import them from the Automation Studio.

Follow these steps:

1. Navigate to the Automation Studio installation directory and copy the custom action JAR file to the customerActions folder. (TK - I don't think this is required. Can you check with engineering?)

   Note: If a custom action JAR file depends on third-party JARs, the third-party JARs must also be imported via the Automation Studio.

2. Log in to Automation Studio.
3. Click the Administration tab and select Action Management.
4. Click the + icon.
   The Open window displays.
5. Select the action pack container files and click Open.
6. Click Import.

7. Click Close when the import completes.
   You can now use the actions included in the action pack to build processes and flows.

Best Practices for Creating Custom Actions

This section outlines best practices when developing custom actions:

Action Characteristics

The fundamental principle of the CA Release Automation framework is that actions are self-contained objects with an independent logic. Actions are state-less. Each connection and resource that are used during the execution of an action should be closed before the action ends. There is no need for other actions to proceed or follow any particular action. Actions should not be created with static variables that require other actions to read or modify them. There is no need to create critical segments that enforce synchronization. The framework distributes any custom action and executes it in different processes.

Proper action development enables:

- Utilization of many framework features
- Insertion of actions in various contexts
- Compatibility with prior versions of the product

The goal of creating custom actions is to share them with a community of CA Release Automation action developers.

Serialization

An action is a hierarchy object. If there is a parameter in the parent object, it is derived into the sub action. All actions implement serialization. Add transient keyword to class variables.

Naming Conventions

Take the following into consideration when naming custom actions:

- Custom actions cannot have the same action as one under CA Release Automation actions.
- Custom actions cannot have the same JAR file name as a CA Release Automation action.
- Use proper category for an action.

Parameters

Action parameters should be annotated with ParameterDescriptor. The parameter can be the following types:

- string
- int
- long
- double

- float
- char
- password

You can use an array for all of the types that are displayed in the list. For each primitive, you can use its wrapper. The user can specify an empty value in the interface, so the wrappers are suggested.

Note: A default value must be a concrete value, not a product parameter value. For example, Application Parameter\My Global Param.

- Enum

Inputs and Outputs

- Provide accurate descriptions with details and examples. When creating an action, define all possible inputs and outputs. Clearly explain what value you want to pass in the description. Output parameters will only need to be defined if you want to pass value returning from the action execution to the next action.
- HTML tags can be inserted in the action input name, description, and execution result message. Action properties that get displayed in Automation Studio are displayed as HTML. You can use
 to insert new line in your message to break it into multiple lines.
- Input and output parameters should be 10 based to allow more to be added later.
- Define the output. Consider how the user utilizes the data. For example, should the output be in XML, JSON, or HTML?
- Define input and output type, for example, string, or string array.
- Consider the usability of an action. If the action returns values that can be exposed to other actions as inputs, define the output parameters in the action.
- Use names of inputs and outputs associated by the target application.
- If an output repeats the same value, provide a default value.
- Use consistent input names.

Related Actions

Combine related actions and operations into a single action when practical.

Error Handling

Handle all important errors and output readable error messages. If the cause of the error is known, the output should provide suggestions.

Validation

- Validate all user inputs before running the action. If a host name input should not contain spaces, then add code to check for spaces in a string and call trim () to remove extra spaced after the host name. If the name cannot contain certain restricted characters, add to the code to check for it.
- Proper logging and execution logic of action for example, SOAP or CLI. Running logs are the input and output tracking mechanism used to determine how the inputs are behaving.
- Proper validation of a result - externalize all the messages. Users may not care about external messages.
- Test the actions.

Descriptors

There are two descriptors:

- ActionDescriptor
- ParameterDescriptor

For the implementation of an interface or customer action, the ActionDescriptor should annotate each action in the hierarchy that contains the parameters. Only annotated classes and parameters are treated by the action framework (UI and backend). Ensure there are no collisions between the ParameterDescriptor#order in the whole action hierarchy. You should only have UI implications.

For example:

```
@ParameterDescriptor(defaultValueAsString="Hi")
```

The parameter should be the initialization value of:

```
String myParam = "hi"
```