Universal Controller 7.0.x

Available Integrations

Built-In Universal Templates

Introduction

Specific Universal Templates are delivered with the Universal Controller installation package.

These Built-in Universal Templates are delivered READ ONLY; if you require changes to a Built-In Universal Template, you have to copy the Built-in Universal Template to create a new Universal Template, which you then can edit with the ops_universal_template_admin security role. Stonebranch does not provide support or updates for new Universal Templates created from the Built-in Universal Templates.

To use a Built-In Universal Template, you first must load it from the List/Load Built-In Universal Templates Server Operation.

Open source Universal Templates will continue to be made available via the Stonebranch Marketplace. Customers can access the the Stonebranch Marketplace from the Stonebranch Customer Portal.

Available Built-In Universal Templates

The following tabled identifies the Built-In Universal Templates that are available for Universal Controller release 6.9.0.0.

The Name of each Universal Template is a link to a separate page containing detailed information about that template.

(For additional information about Universal Templates, see Universal Templates.)

Name	Description	Available from Release
UAC - SSH Tasks	Execute Command or Script on Remote SSH Server with no UC Agent Installed, Remote Server Requires an SSH Server.	UC 6.8.0.0
UAC - UDM Gateway Tasks	UDM Gateway Integration Tasks.	UC 6.8.0.0
UAC - Docker Image	Universal Task to manage Docker Images, Build, Remove, Pull, Push, and Tag Functions.	UC 6.9.0.0
UAC - Docker Container	Universal Task to manage Docker Containers, Run, Create, Start, Stop and Remove Functions.	UC 6.9.0.0
UAC - Docker Compose	Universal Task for Docker Compose Functions, Build, Up, Down, Start, Stop Functions.	UC 6.9.0.0
UAC - Powershell	Universal Task to Run Powershell Scripts.	UC 6.9.0.0
UAC - Kubernetes	Kubernetes List (Get), Create, Delete and Replace Functions.	UC 6.9.0.0
UAC - UA Install	Install UA via SSH Server, Download UA install from SB Website.	UC 6.9.0.0
UAC - Remote Controller	Run Task or Workflow on a Remote Universal Controller.	UC 6.9.0.0
UAC - UC Report	Run a Universal Controller report and deliver output to a specified server and file location.	UC 6.9.0.0

UAC - SSH Tasks

Overview

Universal Task to run a command or Universal Controller "data" script on a remote ssh server.

Enables the execution of commands and scripts from the Universal Controller script library on a remote Unix. Linux, or Windows SSH server.

- The Universal Task will end with the exit code from the remote command / script.
- Both stdout and stderr from the remote task are returned to the Universal Task's stdout. Stderr is used for messages from the Universal Task itself.
- Requires Python 3.6 or higher with the ssh2-python module installed. Tested with the Universal Agent bundled Python distribution.
- You can set different log-levels for the Universal task, providing you more or less information to suit your needs.

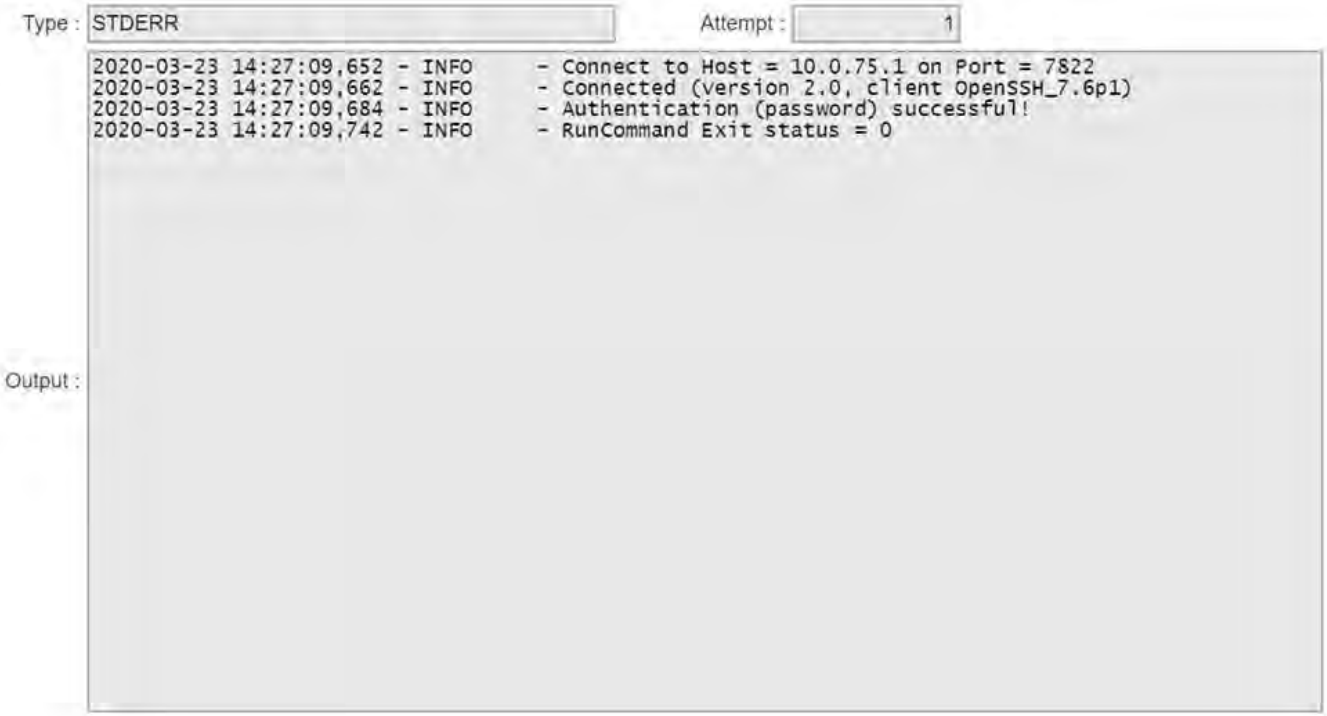
An SSH task is unable to provide the following functionality that is available for tasks that execute on a Universal Agent.

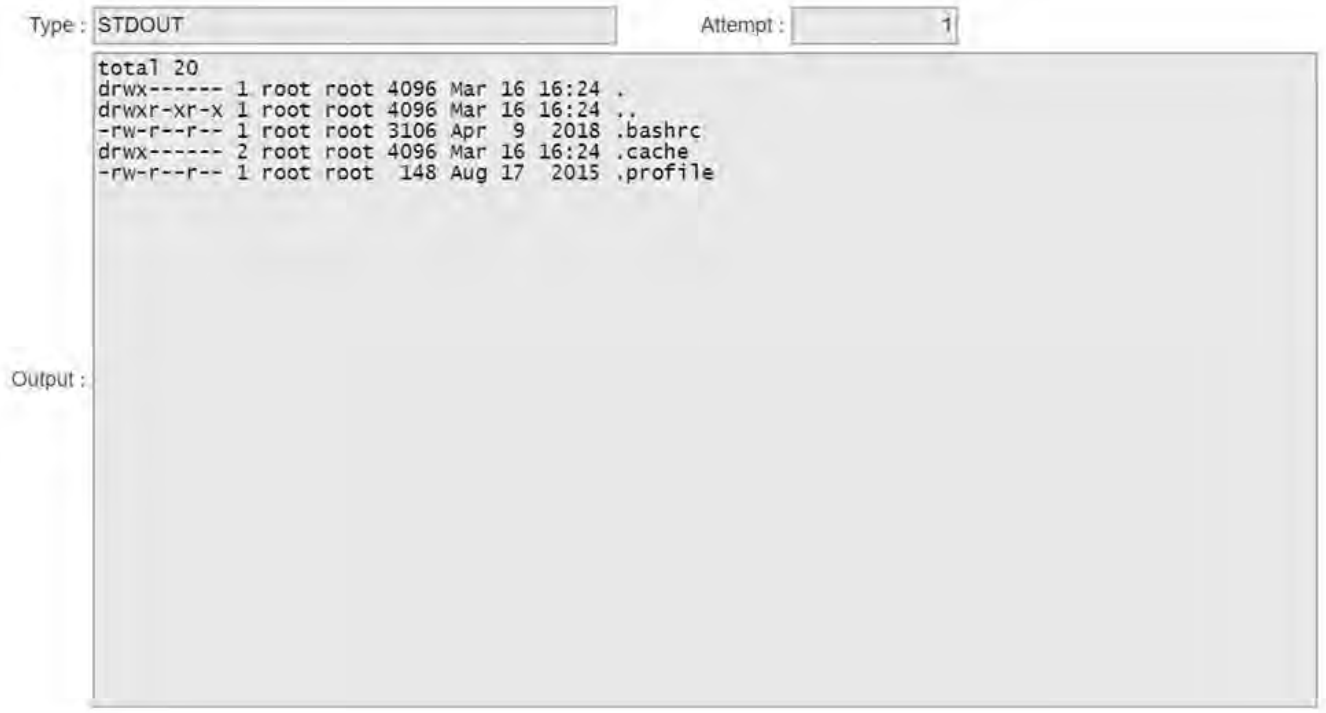
- No Agent visibility, status, or alerting is available
- The remote command or script cannot be cancelled from the controller UI, cancelling the SSH task will only cancel the Universal Task Python script that is initiating the SSH session
- No fault tolerant functionality
- No clustering, load balancing, or broadcast functionality

The Universal Task performs the following:

1. Connects to the Remote SSH Server
2. Authenticates to the Remote SSH Server via Userid and Password or SSH Public Key.
3. Execute Command on SSH Server or Transfer UC Data Script to Remote SSH Server and Execute.

Example Output


Type :
STDERR
Attempt :
1
Output :
2020-03-23 14:27:09,652 - INFO - Connect to Host = 10.0.75.1 on Port = 7822
2020-03-23 14:27:09,662 - INFO - Connected (version 2.0, client OpenSSH_7.6p1)
2020-03-23 14:27:09,684 - INFO - Authentication (password) successful!
2020-03-23 14:27:09,742 - INFO - RunCommand Exit status = 0



Universal Task Details Fields

Field Label	Description	Example	Required
Remote SSH Server	Specify the Hostname for the Remote SSH Server	mysshserver	Y
Remote SSH Port	Specify the SSH port for the Remote SSH Server	22	Y
Remote SSH Credential	Select the Credential Definition to access the Remote SSH Server. If the Credential specifies a Key Location, Public Key Authentication will be performed.		Y
Command or Script	Specify whether to execute a Command, or Transfer and Execute a UC Data Script.		Y

Command	Command To be Executed on the Remote SSH Server	ls -la	Y (If Command or Script = Command)
Script	UC Data Script to be transferred and Executed on the Remote SSH server		Y (If Command or Script = Script)

UAC - UDM Gateway Tasks

Overview

For integration of WLA / Job Scheduling with the UDM Gateway File Transfer solution. Universal Controller customers can implement the provided Universal Task.

- Universal Task (requires Universal Controller version 6.7.0.0 or higher).
- Uses the UDM Gateway Rest API (requires UDM Gateway Version 11 or higher).
- Requires Python 3.6 or higher with the requests module installed. Tested with the Universal Agent bundled Python distribution.
- Stdout returns information from the UDM Gateway. Stderr is used for messages from the Universal Task / Script itself.
- You can set different log-levels, providing you more or less information to suit your needs.

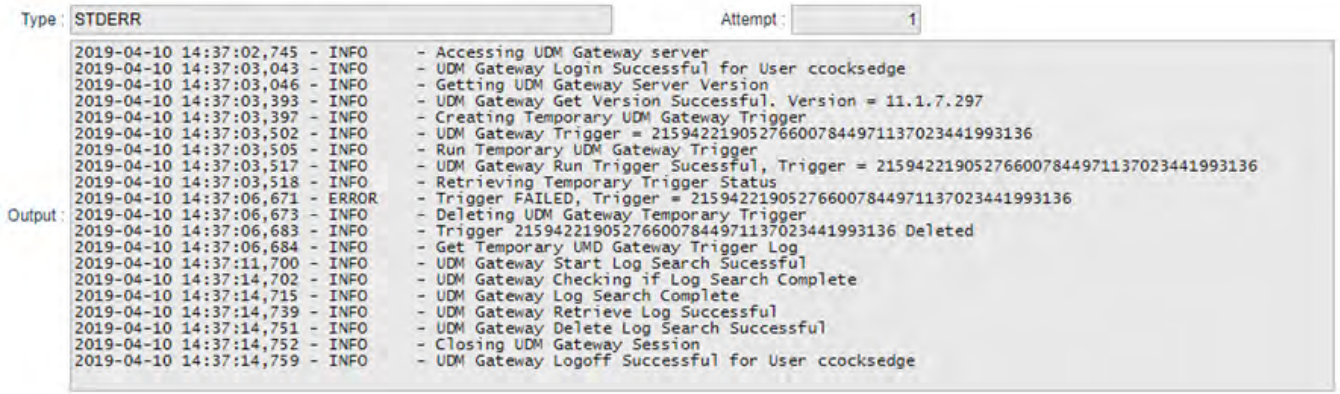
The Universal Task can automate the following UDM Gateway functions:

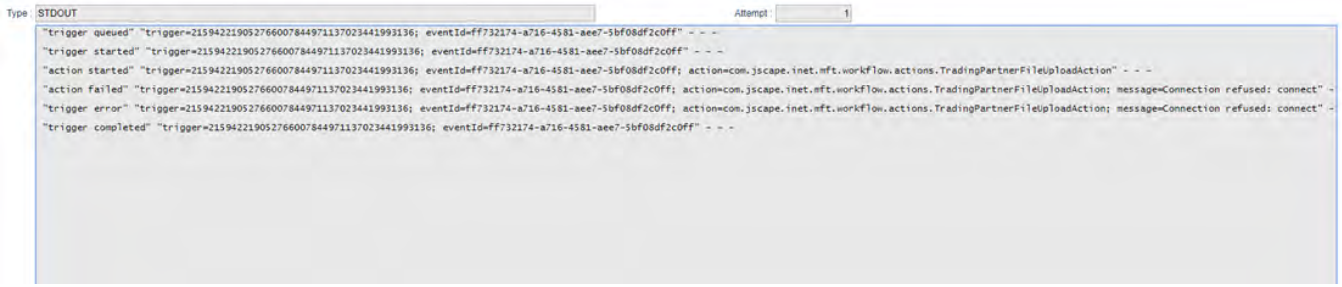
- PGP Encrypt
- PGP Decrypt
- Run a UDM Gateway Trigger
- SFTP File Upload
- SFTP File Download
- Trading Partner File Upload
- Trading Partner File Download
- Trading Partner File Upload using a Regex or Generic Filename Pattern
- Trading Partner File Download using a Regex or Generic Filename Pattern

The Universal Task performs the following:

1. Login to UDM Gateway Server
2. Check UDM Gateway Server Version
3. Create Temporary UDM Gateway Trigger with a unique name to perform the desired action. Except Run Trigger which copies the existing trigger to the temporary trigger.
4. Run Temporary UDM Gateway Trigger.
5. Get Temporary UDM Gateway Trigger Status
6. Delete Temp Trigger
7. Retrieve Trigger Log
8. Close UDM Gateway Session

Example Output



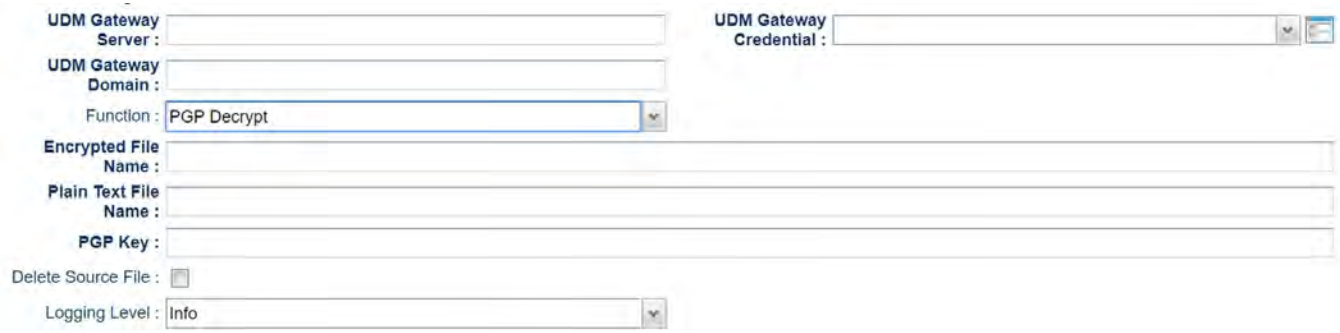


Universal Task Details Fields

Function = PGP Encrypt

Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the PGP Encrypt Function		Y
Encrypted File Name	The name of the target encrypted file	/demo/my-data.pgp	Y
Plain Text File Name	The name of the source plaintext file	/demo/my-data.txt	Y
PGP Key	PGP Key to use for encryption	secret: LocalDemo <stonebranch.demo@stonebranch.com>	Y
Delete Source File	Specify if the source plaintext file is deleted		Y
Compress	Specify if the target encrypted file is compressed		Y
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = PGP Decrypt



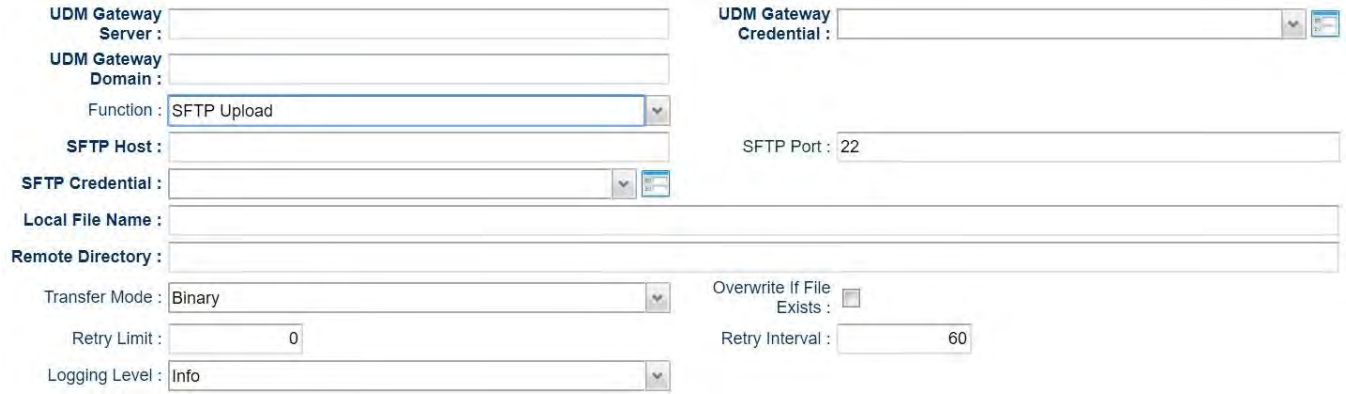
Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the PGP Decrypt Function		Y

Encrypted File Name	The name of the target encrypted file	/demo/my-data.pgp	Y
Plain Text File Name	The name of the source plaintext file	/demo/my-data.txt	Y
PGP Key	PGP Key to use for encryption	secret: LocalDemo <stonebranch.demo@stonebranch.com>	Y
Delete Source File	Specify if the source plaintext file is deleted		Y
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = Run Trigger

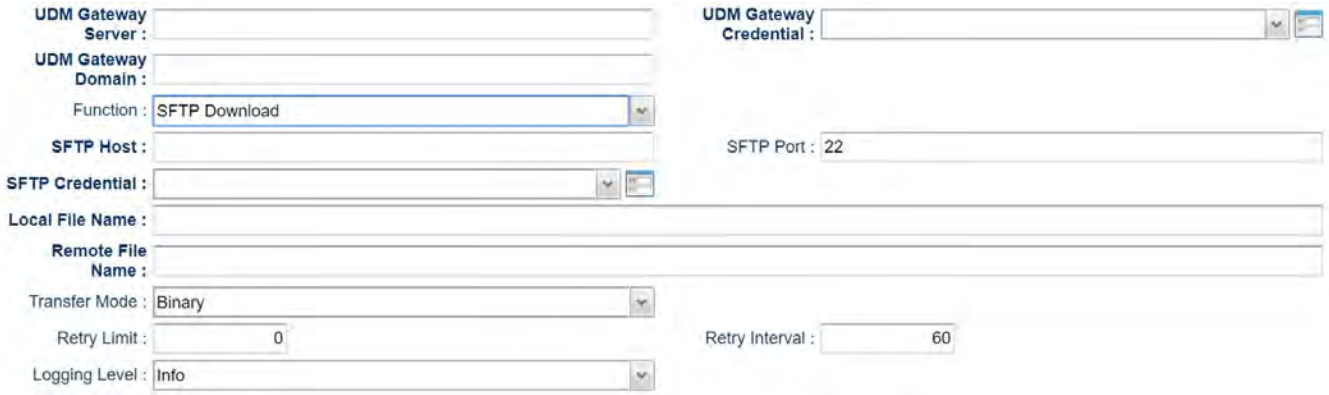
Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the Run Trigger Function		Y
Trigger Name	Provide the name of an existing UDMG Trigger	Test Trigger	Y
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = SFTP Upload



Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the SFTP Upload Function		Y
SFTP Host	Remote SFTP server Host Name or IP	127.0.0.1	Y
SFTP Port	Remote SFTP server port	22	N
SFTP Credential	Select the Credential Definition to access the remote SFTP server		Y
Local File Name	Location of the Local File	/demo/my-data.pgp	Y
Remote Directory	Remote SFTP Server Directory	/	Y
Transfer Mode	Transfer Mode (Ascii, Binary, Auto)	Ascii	N
Overwrite if File Exists	Overwrite destination file(s) if it exists		N
Retry Limit	Maximum Retry Attempts	0	N
Retry Interval	Interval in Seconds Between Retries	60	N
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = SFTP Download



Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the SFTP Download Function		Y
SFTP Host	Remote SFTP server Host Name or IP	127.0.0.1	Y
SFTP Port	Remote SFTP server port	22	N
SFTP Credential	Select the Credential Definition to access the remote SFTP server		Y
Local File Name	Location of the Local File	/demo/my-data.pgp	Y
Remote File Name	Remote SFTP File	/my-data.pgp	Y
Transfer Mode	Transfer Mode (Ascii, Binary, Auto)	Ascii	N
Retry Limit	Maximum Retry Attempts	0	N
Retry Interval	Interval in Seconds Between Retries	60	N
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

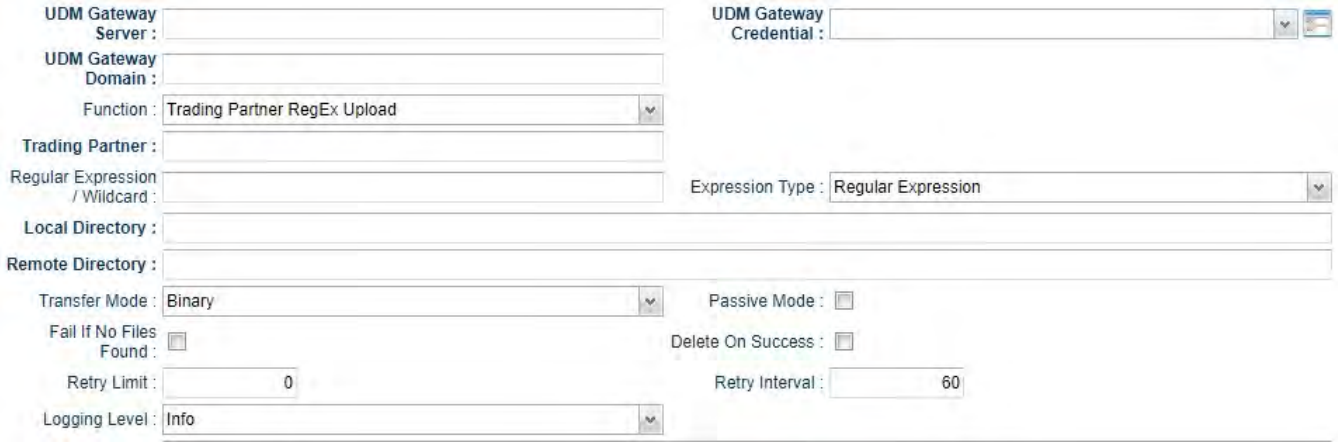
Function = Trading Partner Upload

Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the Trading Partner Upload Function		Y
Trading Partner	UDM Gateway Trading Partner definition	Customer A	Y
Local File Name	Location of the Local File	/demo/my-data.pgp	Y
Remote Directory	Remote Trading Partner Directory	/	Y
Transfer Mode	Transfer Mode (Ascii, Binary, Auto)	Ascii	N
Passive Mode	Passive Mode (true, false)		N
Retry Limit	Maximum Retry Attempts	0	N
Retry Interval	Interval in Seconds Between Retries	60	N
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = Trading Partner Download

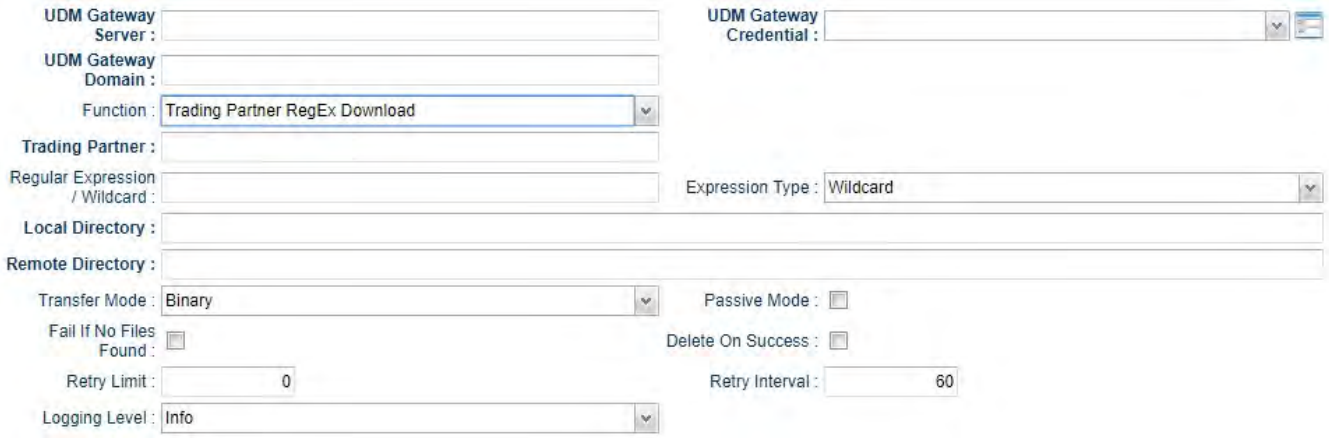
Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the Trading Partner Download Function		Y
Trading Partner	UDM Gateway Trading Partner definition	Customer A	Y
Local File Name	Location of the Local File	/demo/my-data.pgp	Y
Remote File Name	Remote Trading Partner File	/my-data.pgp	Y
Transfer Mode	Transfer Mode (Ascii, Binary, Auto)	Ascii	N
Passive Mode	Passive Mode (true, false)		N
Retry Limit	Maximum Retry Attempts	0	N
Retry Interval	Interval in Seconds Between Retries	60	N
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = Trading Partner RegEx Upload



Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the Trading Partner RegEx Upload Function		Y
Trading Partner	UDM Gateway Trading Partner definition	Customer A	Y
Regular Expression / Wildcard	Specify the Regular Expression or Wildcard	*.txt	N
Expression Type	Select either "Regular Expression" or "Widlcard"		Y
Local Directory	Location of the Local File	/demo	Y
Remote Directory	Remote Trading Partner Directory	/	Y
Transfer Mode	Transfer Mode (Ascii, Binary, Auto)	Ascii	N
Passive Mode	Passive Mode (true, false)		N
Fail if No Files Found	Specify whether the task should fail if no files match the specified expression (Regular Expression or Wildcard)		N
Delete on Success	Specify whether files should be deleted after successful transfer		N
Retry Limit	Maximum Retry Attempts	0	N
Retry Interval	Interval in Seconds Between Retries	60	N
Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

Function = Trading Partner RegEx Download



Field Label	Description	Example	Required
UDM Gateway Server	Specify the URL for the UDM Gateway Server	https://localhost:11880	Y
UDM Gateway Credential	Select the Credential Definition to access the UDM Gateway Server		Y
UDM Gateway Domain	UDM Gateway Server Domain	Local	Y
Function	Select the Trading Partner RegEx Download Function		Y
Trading Partner	UDM Gateway Trading Partner definition	Customer A	Y
Regular Expression / Wildcard	Specify the Regular Expression or Wildcard	*.txt	Y
Expression Type	Select either "Regular Expression" or "Widlcard"		Y
Local Directory	Location of the Local File	/demo	Y
Remote Directory	Remote Trading Partner Directory	/	Y
Transfer Mode	Transfer Mode (Ascii, Binary, Auto)	Ascii	N
Passive Mode	Passive Mode (true, false)		N
Fail if No Files Found	Specify whether the task should fail if no files match the specified expression (Regular Expression or Wildcard)		N
Delete on Success	Specify whether files should be deleted after successful transfer		N
Retry Limit	Maximum Retry Attempts	0	N
Retry Interval	Interval in Seconds Between Retries	60	N

Logging Level	Controls messages issues from the Universal Task Script • None • Info • Debug • Warning • Critical	None	Y

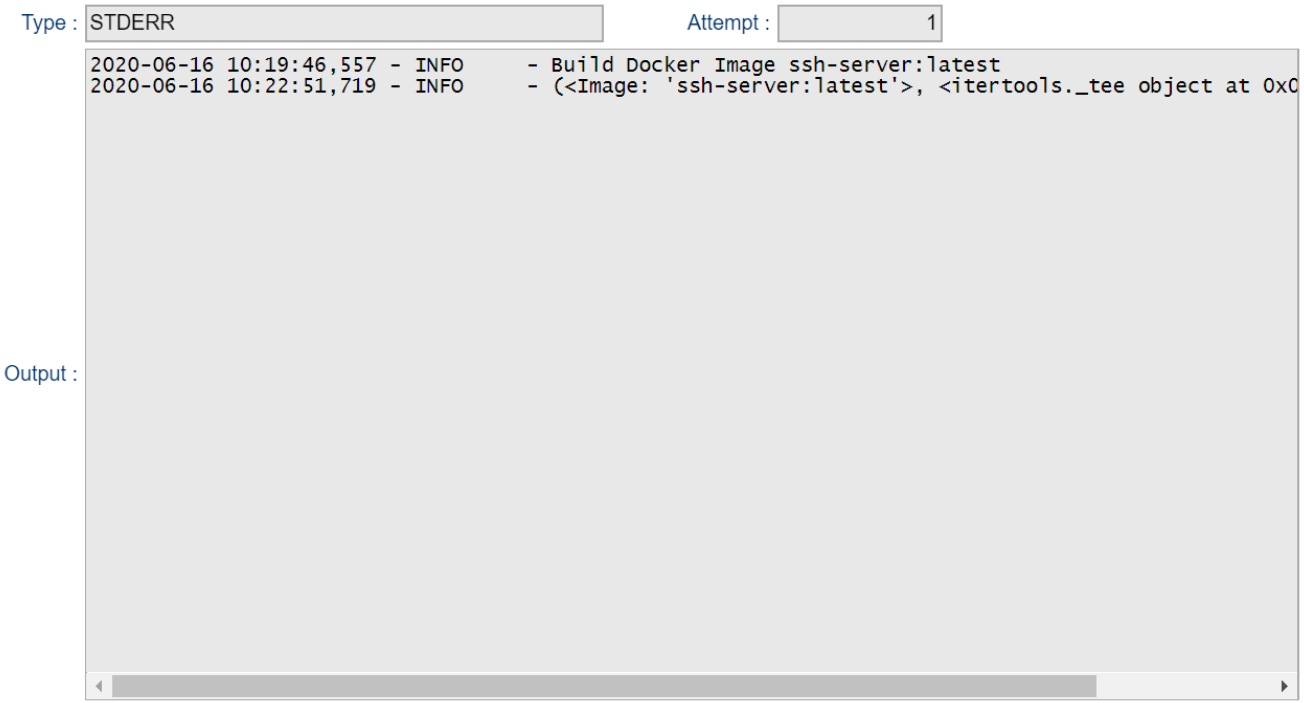
UAC - Docker Image

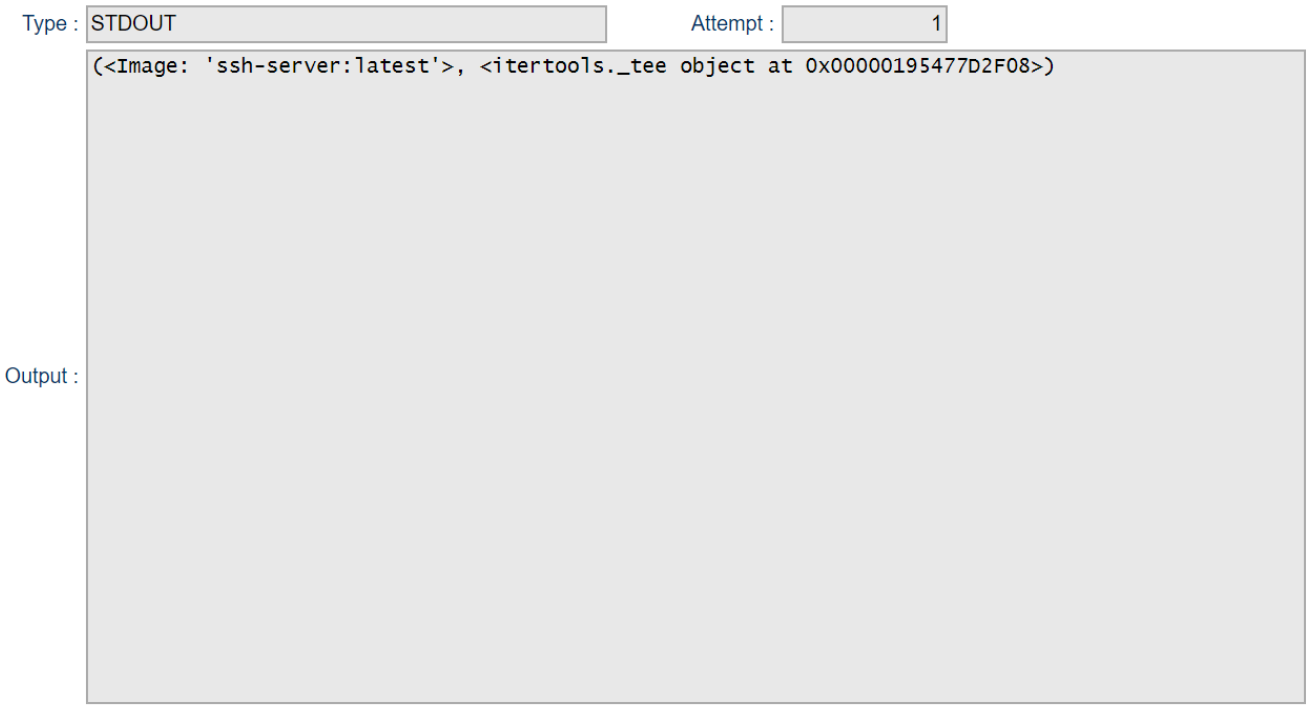
Universal Task to Manage Docker Images; Run, Create, Start, Stop, and Remove Functions

This Universal Task can automate the following Docker Image functions:

- Build
- Remove
- Pull
- Push
- Tag

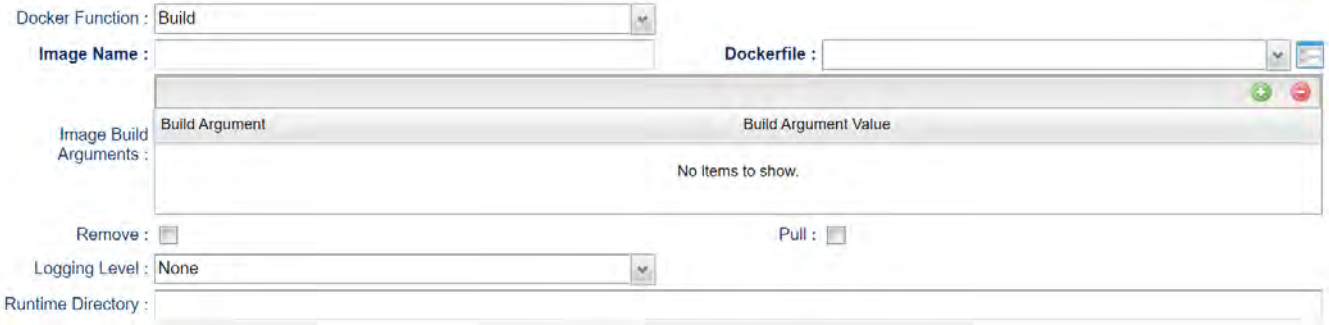
Example Output


Type :
STDERR
Attempt :
1
Output :
2020-06-16 10:19:46,557 - INFO - Build Docker Image ssh-server:latest
2020-06-16 10:22:51,719 - INFO - (<Image: 'ssh-server:latest'>, <itertools._tee object at 0x0



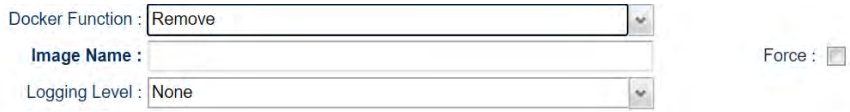
Universal Task Details Fields

Docker Function = Build



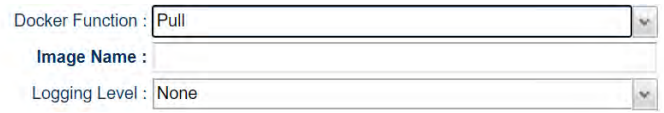
Field Label	Description	Example
Image Name	Docker Image Name (including tag)	myimage:latest
Dockerfile	Select the required dockerfile from the UC script library	
Image Build Arguments	Specify options to pass to the docker image build	
Remove	Always remove intermediate containers, even after unsuccessful builds	
Pull	Always attempt to pull a newer version of the base image	
Logging Level	Controls messages issued from the Universal Task: • None • Info • Debug • Warning • Critical	None
Runtime Directory	Will be used as the build context	/docker/builds/myimage

Docker Function = Remove



Field Label	Description	Example
Image Name	Docker Image Name (including tag)	myimage:latest
Force	Force the removal of a running container or image	
Logging Level	Controls messages issued from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Pull

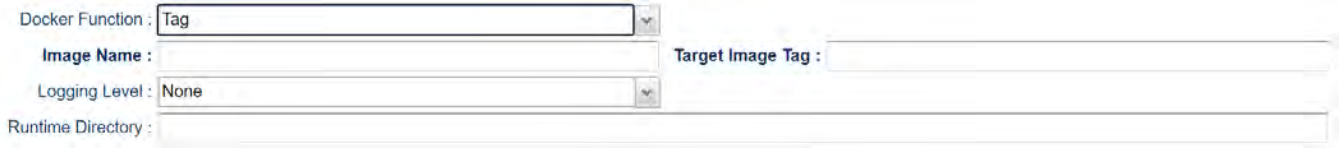


Field Label	Description	Example
Image Name	Docker Image Name (including tag)	stonebranch/universal-agent:latest
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Push

Field Label	Description	Example
Image Name	Docker Image Name (including tag)	stonebranch/universal-agent:latest
Repository Credential	Docker Repository Credential	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Tag



Field Label	Description	Example
Image Name	Docker Image Name (including tag)	universal-agent:6.8.0.0
Target Image Tag	Target Image that refers to source image	universal-agent:latest
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

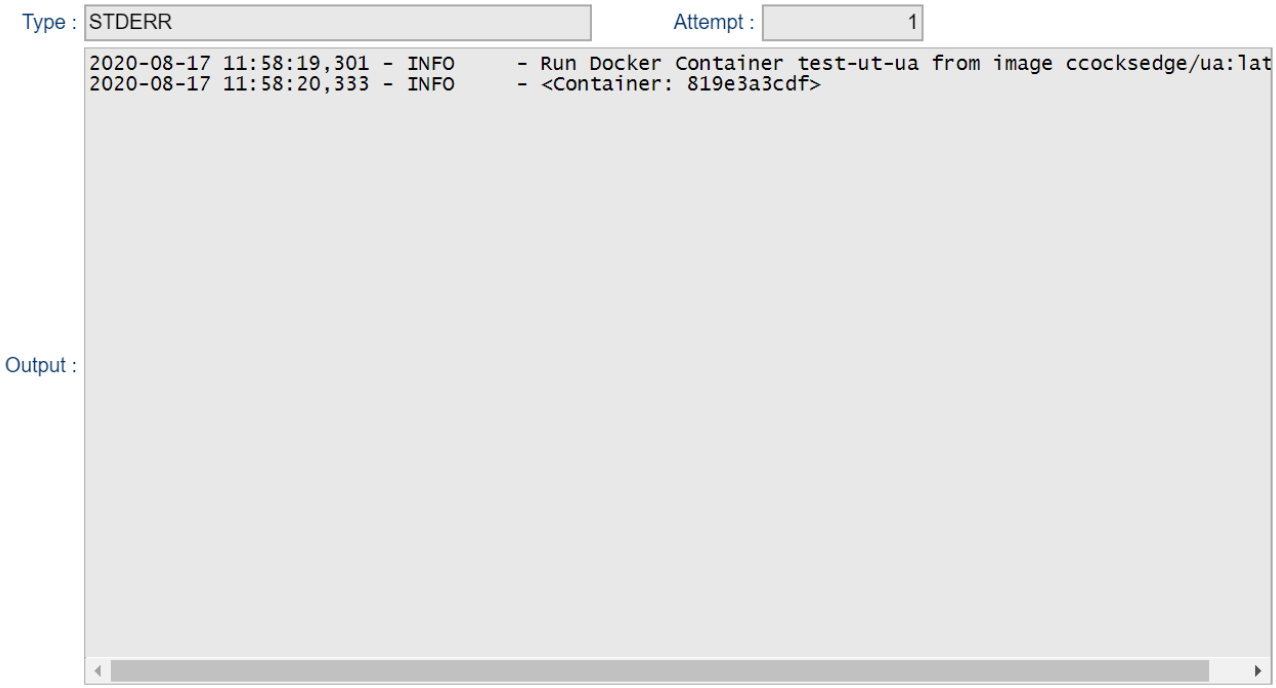
UAC - Docker Container

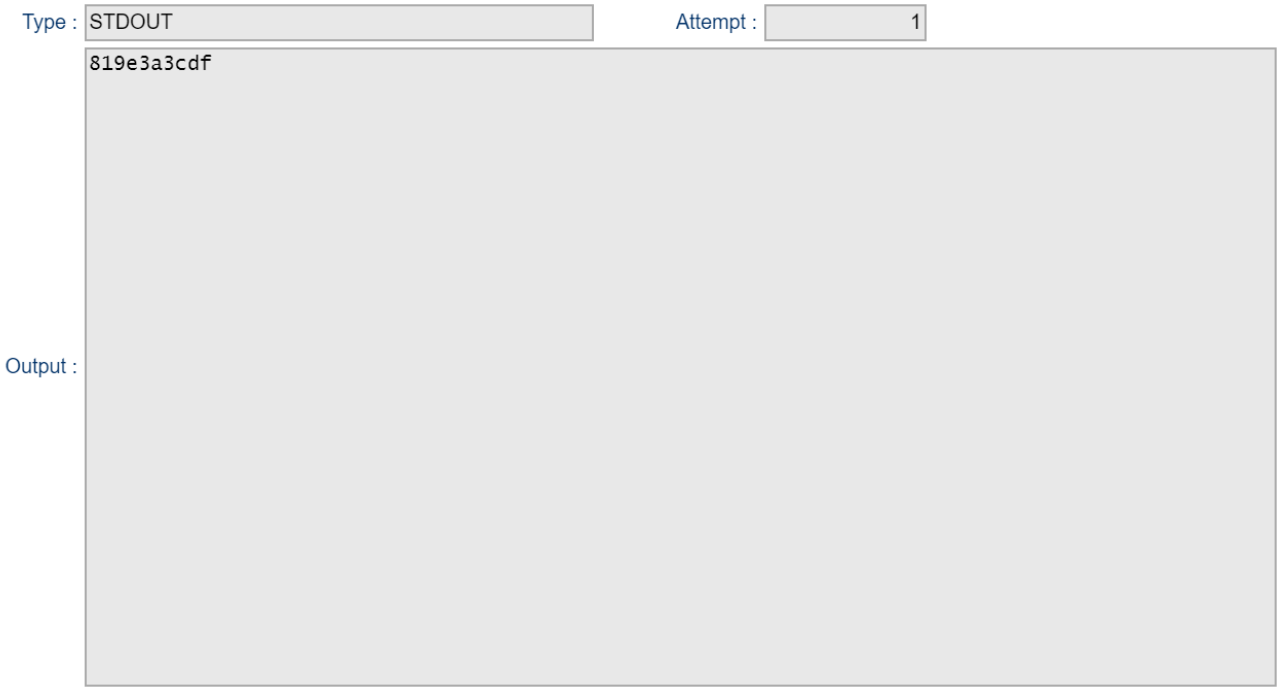
Universal Task to Manage Docker Container; Run, Create, Start, Stop and Remove Functions

The Universal Task can automate the following Docker Container functions:

- Run
- Create
- Start
- Stop
- Remove

Example Output


Type :
STDERR
Attempt :
1
Output :
2020-08-17 11:58:19,301 - INFO - Run Docker Container test-ut-ua from image ccocksedge/ua:lat
2020-08-17 11:58:20,333 - INFO - <Container: 819e3a3cdf>



Universal Task Details Fields

Docker Function = Run

Field Label	Description	Example
Image Name	Docker Image Name (including tag)	myimage:latest
Container Name	Docker Container Name	mycontainer
Hostname	Docker Container Hostname	myhost
Detach	Run Container in background and print Container ID	
Remove	Automatically remove the container	
Command	Command to Run in the Container	echo "Hello World"
Container Environment Variables	Set Container Environment Variable Values	
Container Port Mapping	Specify specific Container ports to map to specific Host ports	

Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Create

Field Label	Description	Example
Image Name	Docker Image Name (including tag)	myimage:latest
Container Name	Docker Container Name	mycontainer
Hostname	Docker Container Hostname	myhost
Command	Command to Run in the Container	echo "Hello World"
Container Environment Variables	Set Container Environment Variable Values	
Container Port Mapping	Specify specific Container ports to map to specific Host ports	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Start

Field Label	Description	Example
Container Name	Docker Container Name	mycontainer
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Stop

Field Label	Description	Example
Container Name	Docker Container Name	mycontainer
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Remove

Field Label	Description	Example
Container Name	Docker Container Name	mycontainer
Force	Force the removal of a running container (uses SIGKILL)	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

UAC - Docker Compose

Universal Task for Docker Compose Functions; Build, Up, Down, Start, Stop Functions

The Universal Task can automate the following Docker Compose functions:

- Build
- Up
- Down
- Start
- Stop

Example Output

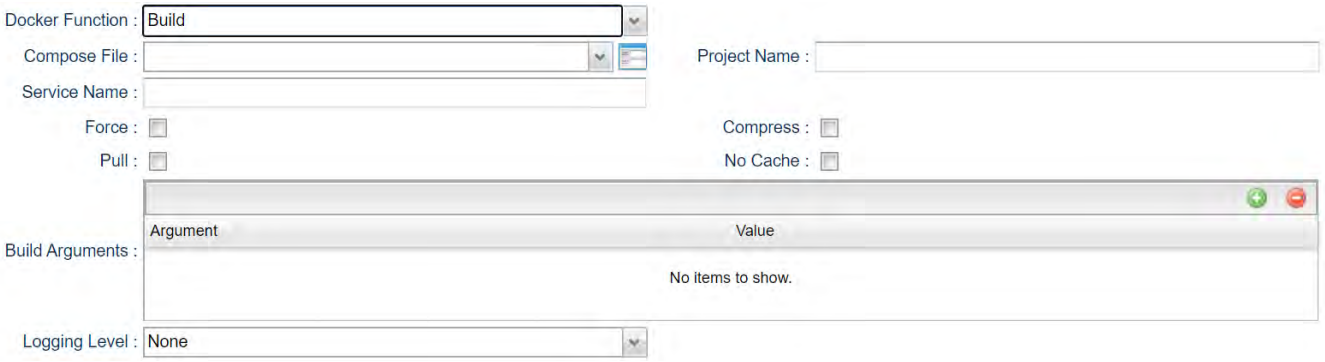
Type : STDERR Attempt : 1

Output :

```
2020-08-17 12:44:36,270 - INFO     - Docker Project uacdev Action Stop
2020-08-17 12:44:47,688 - INFO     - Docker Project uacdev Action stop
```

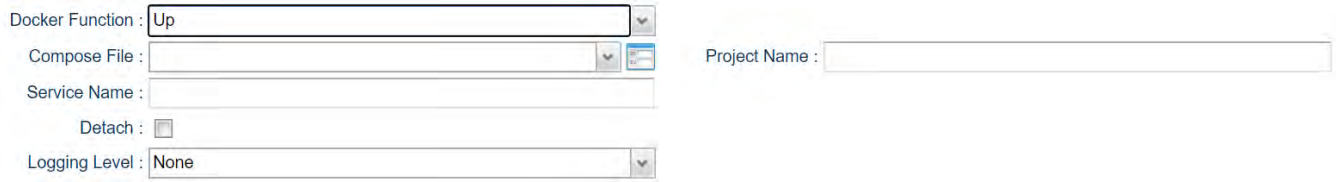
Universal Task Details Fields

Docker Function = Build



Field Label	Description	Example
Compose File	Universal Controller Script (Type = Data) that contains the required compose file YAML statements	
Project Name	Docker Compose Project Name	myproject
Service Name	Optional Service Name as defined in the Compose File	myservice
Force	Always remove intermediate containers, even after unsuccessful builds	
Compress	Compress the build context using gzip	
Pull	Always attempt to pull a newer version of the base image	
No Cache	Do not use Cache when building the image	
Build Arguments	Specify any build arguments	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None
Runtime Directory	Will be used as the build context	/docker/builds/myimage

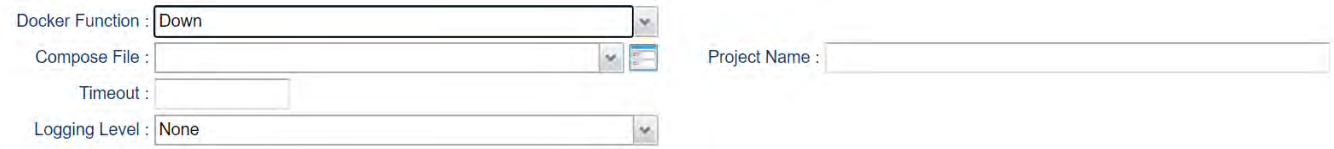
Docker Function = Up



Field Label	Description	Example
Compose File	Universal Controller Script (Type = Data) that contains the required compose file YAML statements	
Project Name	Docker Compose Project Name	myproject
Service Name	Optional Service Name as defined in the Compose File	myservice
Detach	Run Project / Service in background	

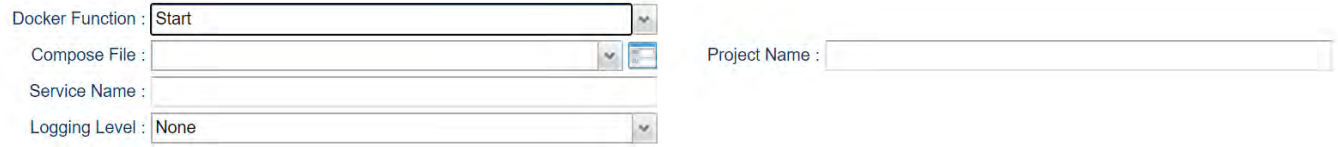
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Down



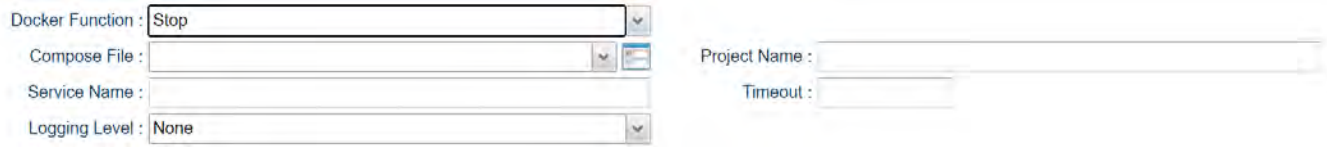
Field Label	Description	Example
Compose File	Universal Controller Script (Type = Data) that contains the required compose file YAML statements	
Project Name	Docker Compose Project Name	myproject
Timeout	Specify a shutdown timeout in seconds (default: 10)	25
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Start



Field Label	Description	Example
Compose File	Universal Controller Script (Type = Data) that contains the required compose file YAML statements	
Project Name	Docker Compose Project Name	myproject
Service Name	Optional Service Name as defined in the Compose File	myservice
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Docker Function = Stop



Field Label	Description	Example
Compose File	Universal Controller Script (Type = Data) that contains the required compose file YAML statements	
Project Name	Docker Compose Project Name	myproject
Service Name	Optional Service Name as defined in the Compose File	myservice
Timeout	Specify a shutdown timeout in seconds (default: 10)	25
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

UAC - Powershell

Universal Task to Run Powershell Scripts

Universal Task Details Fields

Powershell Script Location = UC Script Library

This option executes a powershell script from the Universal Controller Script Library. Scripts to be executed from this task must be Script Type = Data and the Script Name must end in .ps1.

Field Label	Description	Example
Powershell Script	Select the required Powershell Script fro the Universal Controller Script Library.	
Powershell Script Options	Specify any required options and their values to be passed to the Powershell Script. Note: • Powershell Script options should be prefixed with a "-". • It may be required to quote any values that contain spaces or special charaters.	

Powershell Script Location = Server Disk

This option executes a Powershell Script from the Agent server's files system.

Field Label	Description	Example
Powershell Script	Specify the script on the Agent server's files system. Note: You can optionally use the Run Time Directory field to specify the script's location or use a fully qualified path/name in this field.	C:\scripts\myscript.ps1

Powershell Script Options	Specify any required options and their values to be passed to the Powershell Script. Note: • Powershell Script options should be prefixed with a "-". • It may be required to quote any values that contain spaces or special charaters.	

UAC - Kubernetes

Universal Task to Run Kubernetes; List (Get), Create, Delete, and Replace Functions.

The Universal Task can automate the following Kubernetes functions:

- List
- Create
- Delete
- Replace

Example Output

Type : STDOUT Attempt : 1

Output :

```
{'api_version': 'v1',
 'kind': 'Namespace',
 'metadata': {'annotations': None,
              'cluster_name': None,
              'creation_timestamp': datetime.datetime(2020, 6, 2, 14, 0, 34, tzinfo=tzutc()),
              'deletion_grace_period_seconds': None,
              'deletion_timestamp': None,
              'finalizers': None,
              'generate_name': None,
              'generation': None,
              'initializers': None,
              'labels': {'name': 'stonebranch'},
              'managed_fields': None,
              'name': 'stonebranch',
              'namespace': None,
              'owner_references': None,
              'resource_version': '856519',
              'self_link': '/api/v1/namespaces/stonebranch',
              'uid': 'af6c9ad9-cbad-4938-8dda-2ad23088db88'},
 'spec': {'finalizers': ['kubernetes']},
 'status': {'phase': 'Active'}}
```

Universal Task Details Fields

Function = List (Get)

Function : List (Get)

Kubernetes Object : Pod

Logging Level : None

Field Label	Description	Example

Kubernetes Object	Type of Kubernetes Object to perform the selected function against: • Pod • Deployment • Namespace	Pod
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

Function = Create

Field Label	Description	Example
Kubernetes Object	Type of Kubernetes Object to perform the selected function against: • Pod • Deployment • Namespace	Pod
Object Name	The Name of the Kubernetes Object (Pod, Deployment, Namespace) to be created	
Namespace	The Namespace of the Kubernetes Object to be created. NOT used if Kubernetes Object = Namespace in this case the Namespace name is specified as the Object Name	default
YAML Definition	UC Data Script containing the Kubernetes Object definition in yaml format	
Logging Level	Controls messages issues from the Universal Task Script: • None • Info • Debug • Warning • Critical	None

Function = Delete

Field Label	Description	Example
Kubernetes Object	Type of Kubernetes Object to perform the selected function against • Pod • Deployment • Namespace	Pod
Object Name	The Name of the Kubernetes Object (Pod, Deployment, Namespace) to be deleted	
Namespace	The Namespace of the Kubernetes Object to be deleted. NOT used if Kubernetes Object = Namespace in this case the Namespace name is specified as the Object Name	default
Logging Level	Controls messages issues from the Universal Task • None • Info • Debug • Warning • Critical	None

Function = Replace

Field Label	Description	Example

Kubernetes Object	Type of Kubernetes Object to perform the selected function against: • Pod • Deployment • Namespace	Pod
Object Name	The Name of the Kubernetes Object (Pod, Deployment, Namespace) to be Replaced	
Namespace	The Namespace of the Kubernetes Object to be replaced. NOT used if Kubernetes Object = Namespace in this case the Namespace name is specified as the Object Name	default
YAML Definition	UC Data Script containing the Kubernetes Object definition in yaml format	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

UAC - UA Install

Universal Task to Install Universal Agent via SSH Server; Download Universal Agent Install from Stonebranch Website

Note

Requires an SSH Server running on the target Server where the Universal Agent is to be installed.

The Universal Task automates the following functions:

- Download

This function downloads the Universal Agent installation package for the selected OS platform and version from the downloads.stonebranch.com site to the Runtime directory specified.

- Install

This function performs the following:

- Uploads a previously downloaded Universal Agent installation package for the selected OS platform and version from the specified Runtime directory to the target installation server.
- Installs the Universal Agent.

Example Output

Type : STDERR Attempt : 1

Output :

```
2020-06-08 12:27:49,556 - INFO     - Connect to Host = sbus08 on Port = 7822
2020-06-08 12:27:49,634 - INFO     - Connected (version 2.0, client OpenSSH_7.6p1)
2020-06-08 12:27:49,681 - INFO     - Authentication (password) successful!
2020-06-08 12:27:49,684 - INFO     - Transfer File = sb-6.8.0.0-linux-3-x86_64-deb.tar.Z
2020-06-08 12:27:49,760 - INFO     - [chan 0] Opened sftp connection (server version 3)
2020-06-08 12:28:00,444 - INFO     - [chan 0] sftp session closed.
2020-06-08 12:28:00,517 - INFO     - Installing from sb-6.8.0.0-linux-3-x86_64-deb.tar.Z with opt
2020-06-08 12:28:00,517 - INFO     - Executing Command zcat sb-6.8.0.0-linux-3-x86_64-deb.tar.Z |
2020-06-08 12:28:04,202 - INFO     - RunCommand Exit status = 0
2020-06-08 12:28:04,202 - INFO     - Executing Command ./unvinst --oms_servers 7878@10.0.75.1 --a
2020-06-08 12:28:06,774 - INFO     - RunCommand Exit status = 0
```

Type : STDOUT Attempt : 1

Output :

```
-- Output from Command : zcat sb-6.8.0.0-linux-3-x86_64-deb.tar.Z | tar xvf -
unv-6.8.0-0-linux-3-x86_64.deb
Readme.unv
unvinst
upimerge.sh
usrmode.inc
unv-python3.7.tar
unv-opscli-6.8.0-0-linux-3-x86_64.deb
unvfiles.tar


-- Output from Command : ./unvinst --oms_servers 7878@10.0.75.1 --ac_netname TestUAInstall
Group name 'ubroker' does not exist and will be created.
User name 'ubroker' does not exist and will be created.
Selecting previously unselected package unv.
(Reading database ... 9948 files and directories currently installed.)
Preparing to unpack unv-6.8.0-0-linux-3-x86_64.deb ...
Unpacking unv (6.8.0.0) ...
Setting up unv (6.8.0.0) ...
Running unv postinst ...
Package installed correctly.
Setting netname value to  TestUAInstall
Setting oms_servers value to  7878@10.0.75.1
Starting ubrokerd daemon.
```

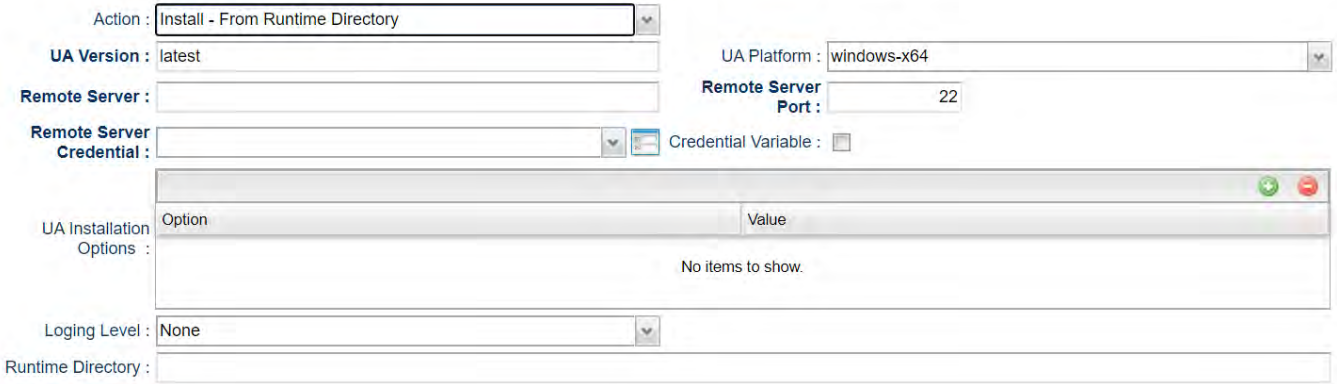
Universal Task Details Fields

Function = Download - To Runtime Directory

Field Label	Description	Example
UA Version	The Universal Agent version number to be downloaded. In the format v.r.m.m or latest.	6.8.0.0

UA Platform	Select the required Universal Agent OS platform from the list.	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None
Runtime Directory	Specify where the Universal Agent installation package will be downloaded to.	C:\UA\InstallFiles

Function = Install - From Runtime Directory



Field Label	Description	Example
UA Version	The Universal Agent version number to be uploaded and installed. In the format v.r.m.m or latest.	6.8.0.0
UA Platform	Select the required Universal Agent OS platform from the list. Note: This version of the UA Install task only supports the following Linux and Windows operating systems. • windows-x64 • windows-x64-um • linux-3.10-x86_64 • linux-3-x86_64-deb • linux-3.10-ppc64le	
Remote SSH Server	Specify the Hostname for the Remote SSH Server.	mysshserver
Remote SSH Port	Specify the SSH port for the Remote SSH Server.	22

Remote SSH Credential	Select the Credential Definition to access the Remote SSH Server. If the Credential specifies a Key Location, Public Key Authentication will be performed.	
Credential Variable	Check to specify a variable for the Remote SSH Credential .	
UA Installation Options	Specify any desired Installation Options to be passed to the install command. Note: • Options for Linux / Unix are prefixed with -- • Options for Windows are not prefixed • Options for Windows User mode are prefixed with - Refer to the documentation for the relevant installation package for more information:	
Logging Level	Controls messages issues from the Universal Task: • None • Info • Debug • Warning • Critical	None

UAC - Remote Controller

Run Task or Workflow on a Remote Universal Controller

Universal Task to launch, monitor, and return results from a task defined in a remote Universal Controller.

For customers who would like to manage tasks in one Universal Controller from another Universal Controller. Simply supply the target Universal Controller URL, valid credentials, and the name of a task defined to the target Universal Controller. Using the web service API's, the Universal Task launches the requested task by name, captures the new instance sysid, and uses this to track the status of the task. After completion, the output is retrieved if the task type supports this, the Universal Task complete with the same exit code as the launched task.

- Requires Universal Controller version 6.5.0.0 (or higher) on the target Universal Controller, and UC 6.7.0.0 or Higher on the Source Universal Controller.
- Running remote Universal Controller workflows is partially supported with the following limitations:
    - Tasks within the workflow are not individually tracked.
    - No output is returned.
    - Workflow status is returned, however workflows do not set a return code so you will need to define the Universal Task's exit code processing to handle the status returned in the stderr appropriately.
- Both stdout and stderr from the remote task are returned to the Universal Task's stdout. Stderr is used for messages from the Universal Task itself.
- Requires Python 3.6 or higher with the requests module installed. Tested with the Universal Agent bundled Python distribution.
- You can set different log-levels for the Universal task, providing you more or less information to suit your needs.
- We are also delivering a Python script for customers who would like to run Universal Controller tasks externally from a command line or other Job Scheduling tool.
    - Example command line:

python run_universal_controller_task.py --controllerurl https://localhost:8080/opswise --username controlleruser --password controlleruserpassword --task "sleep 0" --loglevel info

    - Example stderr messages:

2018-08-07 09:33:08,976 - INFO - Launching Task : Sleep 0

2018-08-07 09:33:09,049 - INFO - Task Instance Sysid : 1533578492639002557VEO7SQNCIHTCF

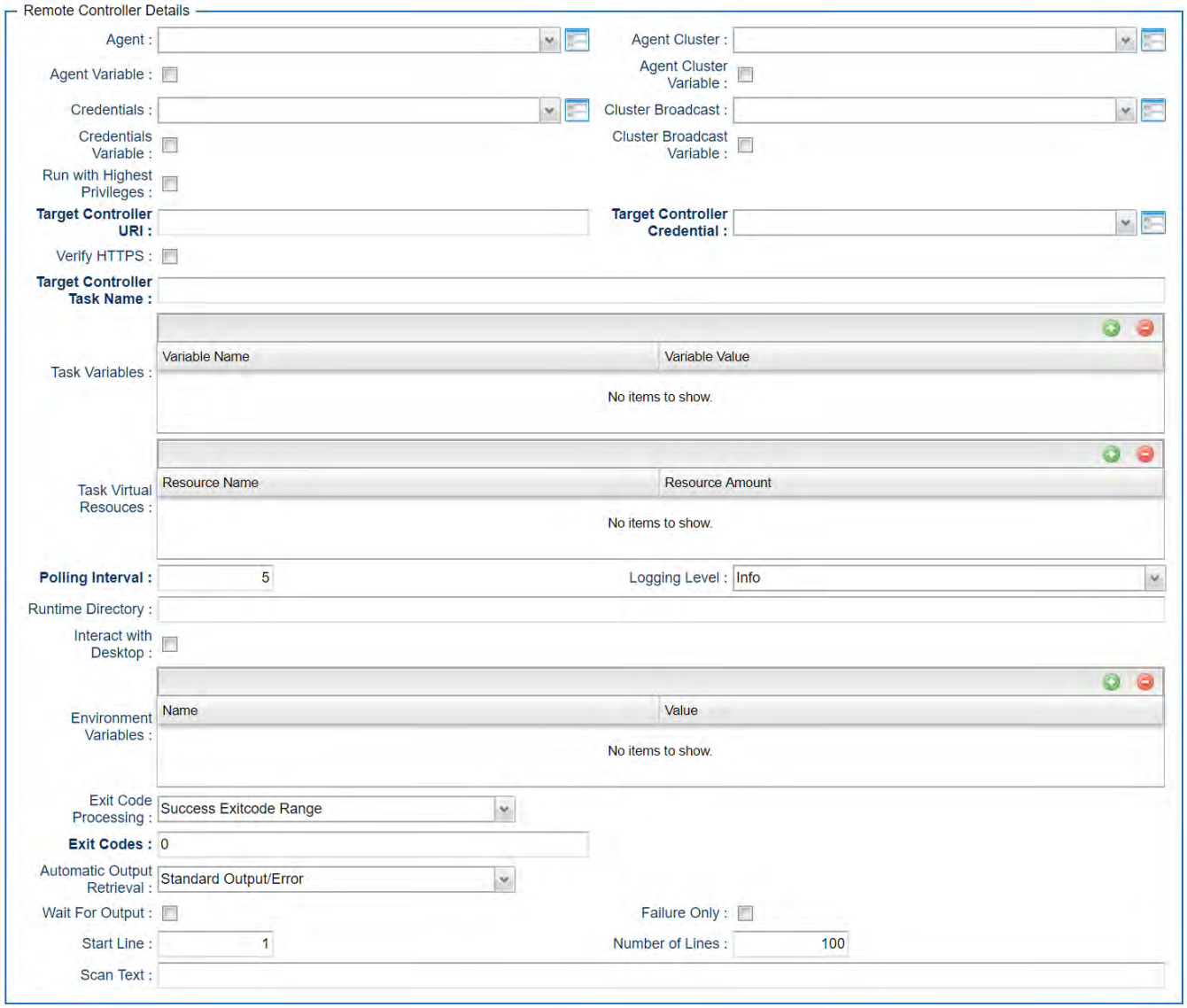
2018-08-07 09:33:09,049 - INFO - Monitoring Task : Sleep 0

2018-08-07 09:33:15,087 - INFO - Task Complete : Status = SUCCESS, Exit Code = 0

2018-08-07 09:33:40,530 - INFO - Retrieving Available Output : Sleep 0

2018-08-07 09:33:40,539 - INFO - Output Retrieval Not Valid for Task Type : Timer

Universal Task Field Descriptions



Field Label	Description	Example	Required
Target Controller URL	Specify the URL for the Target Universal Controller. This specifies where the Task will be Launched.	https://localhost:8080/opswise	Y

Target Controller Credential	Select the Credential Definition to access the Target Universal Controller		Y
Verify HTTPS	For https connections, the Universal Task supports host validation with a certificate or CA certificate, check this option to provide the location of the certificate or CA certificate. If unchecked no host certificate validation will be performed.		N
Certificate File Location	Select on of the following: • Utility Agent Path : If the certificate or CA certificate is located on the Utility Agent File System. • Script File : if the certificate or CA certificate is stored as a Script Data object on the Source Universal Controller		Y (if Verify HTTPS is True)
Certificate File	Specify the location of the certificate or CA certificate on the Utility Agent file system.	/etc/certs/ca-cert.cer C:\security\certs\ca-cert.cer	Y (if Certificate File Location is Utility Agent Path)
Certificate File	Select the Script Data object that contains the certificate or CA certificate.		Y (if Certificate File Location is Script Data)
Target Controller Task Name	Specify the Task to Launch on the Target Universal Controller. The Task name specified must exist of the Target Universal Controller.	Sleep 0	Y
Task Variables	Specify any Variables to pass to the task.		N
Task Virtual Resources	Specify any Virtual Resources and the Amount that the Task will consume.		N
Polling Interval	Provide the polling interval in seconds for checking the Task's status. Default is 5 seconds.		Y
Logging Level	Select the level of messages returned from the script. Valid options are: • None • Info • Debug • Warning • Error • Critical		N

Command Line Script Parameters

Option	Description	Example	Required
--controllerurl	Specify the URL for the Target Universal Controller. This specifies where the Task will be Launched.	--controllerurl https://localhost:8080/opswise	Y
--username	Specify the user to access the Remote Universal Controller	--username ops.admin	Y
--password	Specify the password for the user specified via --username	--password secret	Y

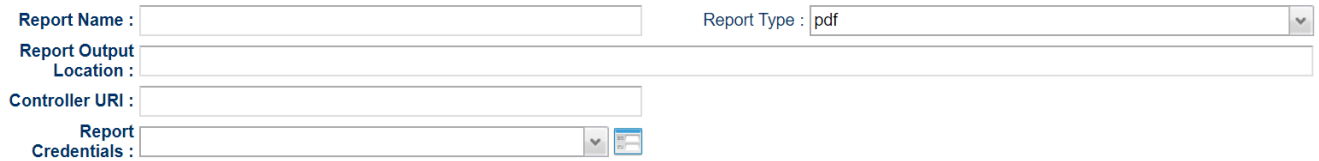
--verfiy	For https connections, the Universal Task supports host validation with a certificate or CA certificate. To Bypass host validation specify False. If not specified host validation will be performed.	--verify False	N
--certfile	Specify the location of the certificate or CA certificate	--certfile /etc/certs/uc-ca.cer	N
--taskname	Specify the Task to Launch on the Target Universal Controller. The Task name specified must exist of the Target Universal Controller.	--taskname "Sleep 0"	Y
--vararray		--vararray '{"var1":"value1","var2":" value2"}'	N
--resarray		--resarray '{"Resource1":"1"," Resource2":"6"}'	N
--interval	Provide the polling interval in seconds for checking the Task's status.	--interval 30	Y
--loglevel	Specify the level of messages returned from the script. Valid options are: • None • Info • Debug • Warning • Error • Critical	--loglevel info	N

UAC - UC Report

Run Universal Controller Report and Deliver Output to a Specified Server and File Location

Universal Task to execute a Universal Controller report and write the output file to the specified location on Agent Server

Universal Task Details Fields



Field Label	Description	Example
Report Name	The Name of the Universal Controller Report to Run	UAC - Audit Failed Logins
Report Type	Type of the Report File that will be created: • pdf • png • csv • tab • xml	
Report Output Location	Fully qualified File Name where the report will be written	c:\temp\myreport.pdf
Controller URI	URL of the Universal Controller	https://localhost:8443/uc
Report Credentials	Credential that will run the report	

Downloadable Universal Templates

Available Downloadable Universal Templates

The following tabled identifies the Downloadable Universal Templates that are available for Universal Controller release 7.0.0.0.

The Name of each Universal Template is a link to a separate page containing detailed information about that template.

(For additional information about Universal Templates, see Universal Templates.)

Name	Description	Available from Release
Ansible Integration	Manage Ansible task execution through the Universal Controller user interface.	UC 6.8.0.0 and later
AWS Create EC2 Instance (with Universal Agent)	Create an EC2 instance with parameters, either in task form, or by simply creating an EC2 instance from the existing AWS launch template.	UC 6.8.0.0 and later
AWS EC2 - Start, Stop, Terminate, and Manage Universal Task Instances	Start, stop, terminate, and manage AWS EC2 instances on demand, simply by providing one or more instance IDs as input.	UC 6.8.0.0 and later
AWS S3	Securely transfers files from, to and between AWS S3 cloud storage buckets and folders.	UC 6.8.0.0 and later
AWS SQS	Create, send and monitor AWS SQS messages and automatically trigger a task in Universal Controller once a message has been received.	UC 6.8.0.0 and later
Azure Blob Storage	Securely transfers files from, to and between Azure Blob Storage container and folders.	UC 7.0.0.0 and later
Azure Data Factory Integration	Schedule, trigger, and monitor the Azure Data Factory pipeline process directly from Universal Controller.	UC 6.8.0.0 and later
Azure Logic Apps	Trigger and monitor the execution of Azure Logic workflows and retrieve the execution of Azure Logic workflow output.	UC 6.8.0.0 and later
Azure Virtual Machines Start-Stop-Terminate-Instance	Utilize Azure Virtual Machine (VM) name, resource group, subscription ID, and access token as inputs for the start, stop, terminate, list, and check status of Azure VMs.	UC 6.8.0.0 and later
Container File Monitoring	Dynamic File Monitoring and File Transfer solution for containerized applications running in any container management solution.	UC 7.0.0.0 and later
Databricks	Perform end-to-end Orchestration and Automation of Jobs & Clusters in Databricks environment, either in AWS or Azure.	UC 6.8.0.0 and later
GitHub	Perform server operations, such as importing/exporting Universal Automation Center objects and integrating with GitHub, as well as allowing the import/export of Universal Automation Center objects using the Universal Controller script library.	UC 6.8.0.0 and later
Google BigQuery	Schedule, trigger, monitor, and orchestrate the Google BigQuery process directly from Universal Controller.	UC 6.8.0.0 and later

Informatica Cloud	Schedule any Data Integration Task or Linear Taskflow in the Informatica Cloud.	UC 6.8.0.0 and later
Informatica PowerCenter	Schedule Informatica PowerCenter Workflows and Tasks, including retrieval of the workflow and session log.	UC 6.8.0.0 and later
Inter-Cloud Data Transfer	Transfer data to, from, and between any of the major private and public cloud providers like AWS, Google Cloud, and Microsoft Azure.	UC 7.0.0.0 and later
Jenkins Integration	Improves the functionality of Jenkins when orchestrated from Universal Controller.	UC 6.8.0.0 and later
JSCAPE MFT	Manage and integrate JSCAPE Managed File Transfer Server processes within UAC automation processes and workflows.	UC 7.0.0.0 and later
Microsoft Teams Integration	Send messages to an existing channel of Microsoft Teams, allowing you to integrate this solution in UAC to notify users for UAC result or send approval notifications on Microsoft teams.	UC 6.8.0.0 and later
PagerDuty	Notify PagerDuty (Incident management platform) in the event of job Failure or long run of a job or early finish of a job or any other event in Universal Controller.	UC 6.8.0.0 and later
Power BI	Perform end-to-end Orchestration and Automation of Jobs & Clusters in Databricks environment, either in AWS or Azure.	UC 7.0.0.0 and later
Salesforce	Create contact and lead objects in Salesforce, as well as execute Salesforce Object Query Language (SOQL) queries.	UC 6.8.0.0 and later
SAP Batch Input	Schedule and execute batch input sessions in SAP.	UC 6.0.0.0 and later
SAP Calendar Import	Import the SAP Factory Calendar and the related Holiday Calendar into the Universal Controller.	UC 6.8.0.0 and later
SAP Data Services	Execute an SAP Data Services “ETL” Job using the “AL_RWJobLauncher.exe”.	UC 6.9.0.0 and later
SAP Event History Monitor	Queries the SAP Event history table for a selected SAP Event & Parameter. If the Event is found, it gets confirmed, so that it is not triggered again. Optionally, a task can be launched based on the occurrence of an Event & Parameter.	UC 7.0.0.0 and later
SAP Extract Job Definitions	Export SAP Job definitions from SAP into one flat file for each Job selected for extraction.	UC 6.8.0.0 and later
ServiceNow Integration	Create incident tickets, problem tickets, and change requests in ServiceNow straight from the Universal Controller.	UC 6.8.0.0 and later
Slack Notifications	Sends job status notifications to a Slack channel and enables users to send interactive messages in Slack for Universal Controller manual task approvals.	UC 6.8.0.0 and later
Snowflake	Orchestrate, schedule, trigger, and monitor the Snowflake load and unload process from different data sources (cloud storage or local VM’s) directly from Universal Controller.	UC 6.8.0.0 and later
SQL	Execute SQL scripts and functions against a MySQL, PostgreSQL, Microsoft SQL Server, Oracle, and SAP HANA database.	UC 6.8.0.0 and later
UiPath for UAC Integration	Schedule, trigger, and monitor the UiPath (RPA) process directly from the Universal Controller.	UC 6.8.0.0 and later

Ansible Integration

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

Ansible is an opensource tool that is used primarily for:

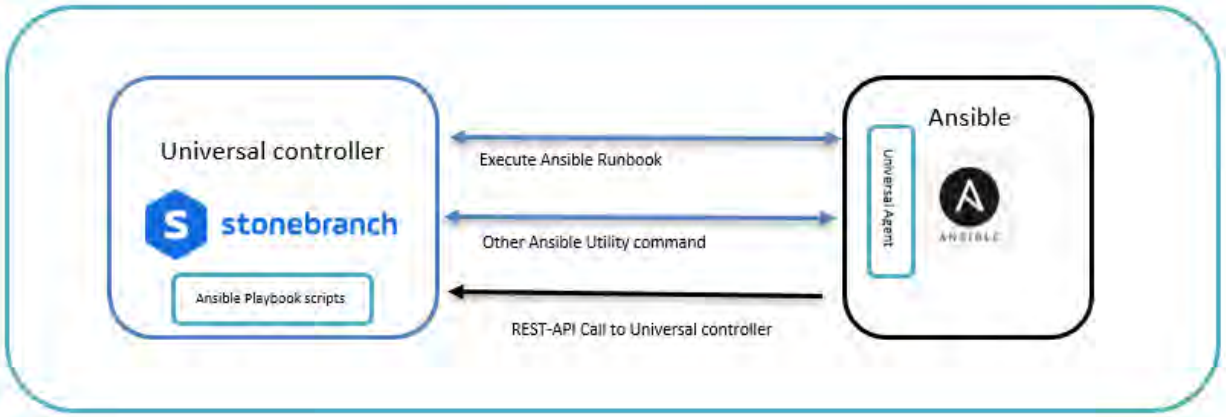
- Application deployment
- Updates on workstations and servers
- Cloud provisioning
- Configuration management
- Intra-service orchestration.

Ansible does not depend on agent software and has no additional security infrastructure.

Universal Controller can be integrated easily with Ansible through a Linux/Unix Universal Agent, as Ansible software can only be installed in Unix/Linux machines. So Universal Controller can manage all Ansible task execution through an intuitive user interface controlling Ansible playbook execution/host inventory details / other Ansible utility.

Overview

- Manage Ansible task execution through the intuitive Universal Controller user interface.
- Ansible playbooks can either be centrally stored and maintained in the Universal Controller script library, or Universal Controller can call the relevant playbook that is residing in Ansible host.
- This Universal Task also enables the execution of other Ansible commands.



Software Requirements

Linux Universal Agent installed in an Ansible host.

Software Requirements for Universal Template and Universal Task

This integration requires an Universal Agent installed in an Ansible server where there is access to use Ansible CLI.

Software Requirements for Universal Agent

- Universal Agent for Linux Version 6.5.0.0 and later.

Software Requirements for Universal Controller

- Universal Controller Version 6.5.0.0 and later.

Software Requirements for the Application to be Scheduled

This Universal Task can work with any of the Ansible Version (tested with 2.8.0).

Technical Considerations

- Accepts input parameters like Ansible utility, Ansible playbook path, Ansible host inventory, Script library (Yaml)
- Universal agent should be installed in the Ansible hosted machine and register to a Universal Controller.
- The Universal Task supports only Universal Agent for Linux.

Ansible Integration Key Features

Feature	Description
Ansible Playbook	This feature help to execute a Ansible playbook that resides in the remote ansible host.
Other Ansible Utility commands	This may be used other Ansible CLI commands.

Import Ansible Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Ansible Integration Universal Task

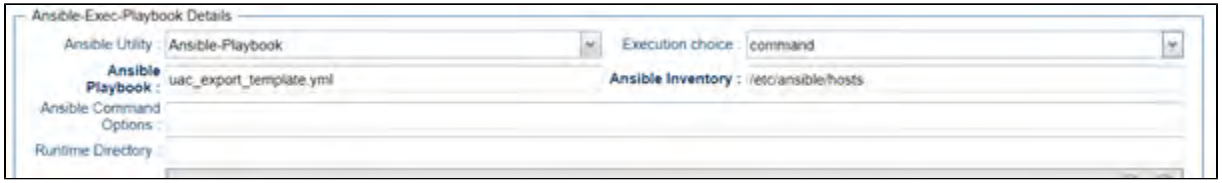
For this Universal Task type, create a new task and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Ansible Integration Universal Task

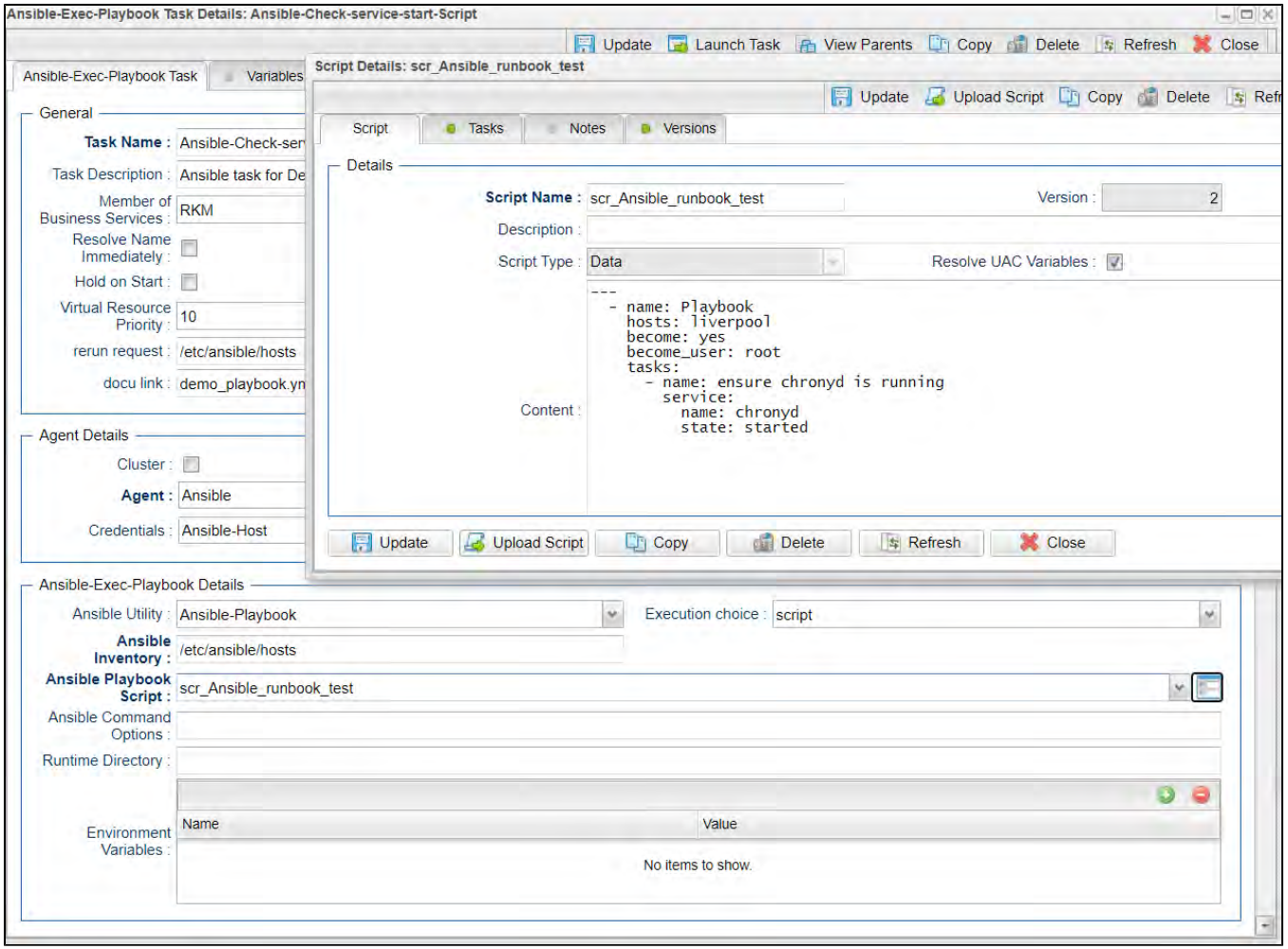
Field	Description
Ansible utility	1. Ansible-Playbook (select to execute Ansible playbook) 2. Other-Ansible-Utility (Other Ansible commands)
Execution Choice	• Command: provide Ansible command that will be used to execute in Ansible machine where playbook and inventory is stored in Ansible host. • Script: Select this option if you want to store the Ansible playbook in controller script library.
Ansible Inventory	Provide the Ansible Inventory source from the Ansible host.
Ansible Playbook Script	Provide a Uiversal Controller script name where you have the Ansible playbook (YAML).
Ansible command options	Any additional commands if needed in playbook execution.

Examples for Ansible Integration Universal Tasks

Invoke an Ansible Playbook that Resides in a Remote Ansible Server



Creating and Executing an Ansible Playbook from Controller



Document References

This document references the following documents:

Name	Location	Description

Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

AWS Create EC2 Instance (with Universal Agent)

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

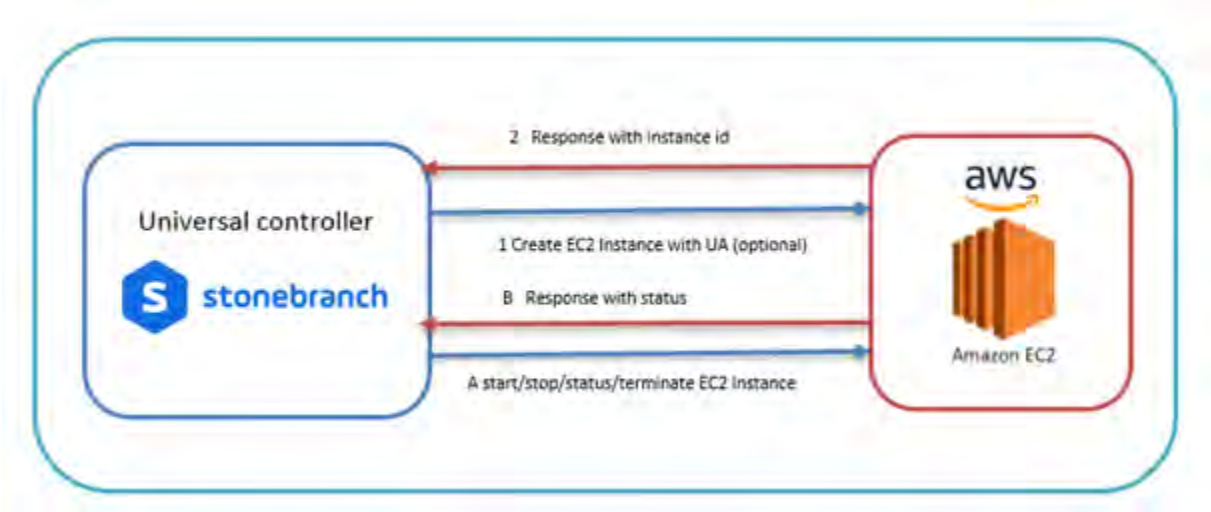
Introduction

This Universal Task allows customers to create an EC2 instance with parameters, either in task form, or by simply creating an EC2 instance from the existing AWS launch template. This task also offers the option to additionally install a Linux/UNIX Universal Agent in the newly provisioned EC2 Instance.

Overview

- The task interacts with the AWS platform via a Python boto3 module.
- All AWS credentials remain encrypted.
- Customers can also install/configure a Linux Universal Agent for each EC2 instance, enabling the Universal Controller to instantly communicate with the newly created instance. (NOTE: only Linux Universal Agent is supported at the moment.)
- This task also lets customers create multiple EC2 instances with the same configuration. New instances can also be tagged.
- It allows customers to create a new keypair or use an existing one for the new EC2 instance.
- This task also enables options for additional EBS volume and encryption, as well as detailed monitoring.

AWS EC2 Task High-Level Overview



Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against AWS EC2 Instance.

Software Requirements for Universal Template and Universal Task

- Requires Python 2.7 or Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests
    - Boto3

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.6.0.0 and later with python options installed
- Universal Agent for Linux Version 6.6.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.7.0.0 and later

Software Requirements for the Application to be Scheduled

- The Server Running the Universal Agent needs to have Python 2.7.x or 3.6.x installed
- AWS IAM Credentials -Access Key, Secret Access key and Region with EC2 set of permissions

- This universal task for the AWS EC2-start-stop-terminate has been tested with the agent bundled with python 3.6 and boto3 module

Technical Considerations

- Consider using this universal task either with universal agent bundled with python(uapy) and also having boto3 module within this environment or a python environment (py) in a host where universal agent is installed with boto3 module in it.
- AWS IAM credentials (Access Key, Secret Access key and Region) should be with the Appropriate access for handling AWS EC2 instances.
- With the current version of this Universal Task, Universal Agent can be installed only in Linux EC2 Instance.

AWS Create EC2 Instance (with Universal Agent) Key Features

Feature	Description
Create New EC2 Instance	Creates a EC2 Instance based on the parameters that are provided in the form
Launch EC2 from template	Create a EC2 Instance based on a template in AWS

Import AWS Create EC2 Instance (with Universal Agent) Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure AWS Create EC2 Instance (with Universal Agent) Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

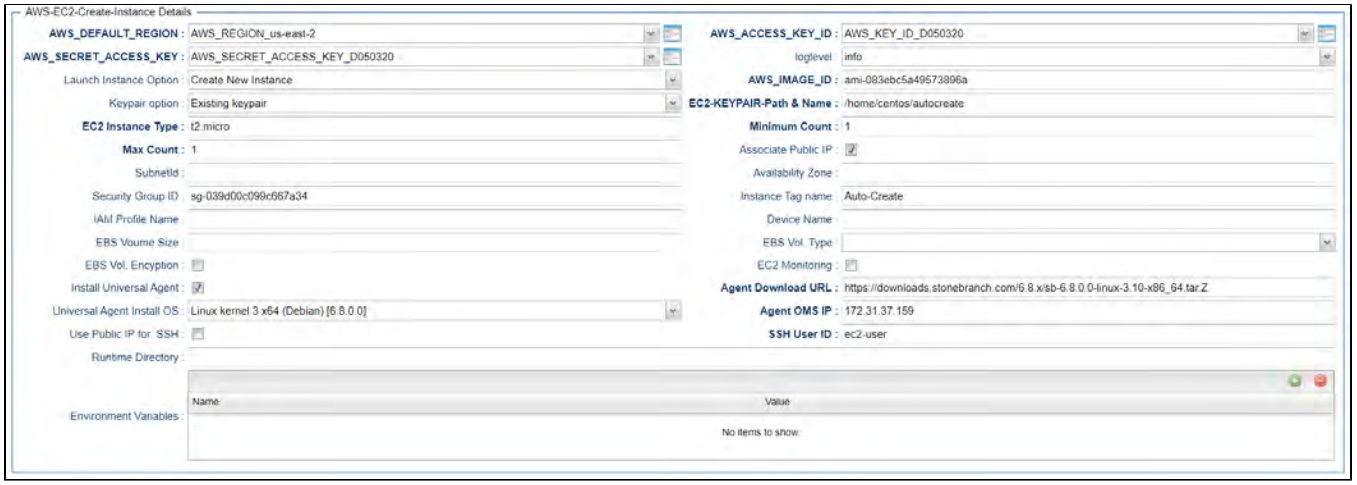
Field Descriptions for AWS Create EC2 Instance (with Universal Agent) Universal Task

Field	Description
AWS-DEFAULT-REGION	AWS Region kept as credential
AWS-SECRET-ACCESS-KEY	AWS Secret Key
AWS-ACCESS-KEY-ID	AWS Access Key
Launch Instance Option	Select either launch from template or create a brand new ec2 instance with the parameter supplied in the form
LaunchTemplateName	Mandatory if launch_instance_option=” Launch from template”
AWS_IMAGE_ID	Provide the AWS machine ID ,Mandatory if launch_instance_option=” new_instance”

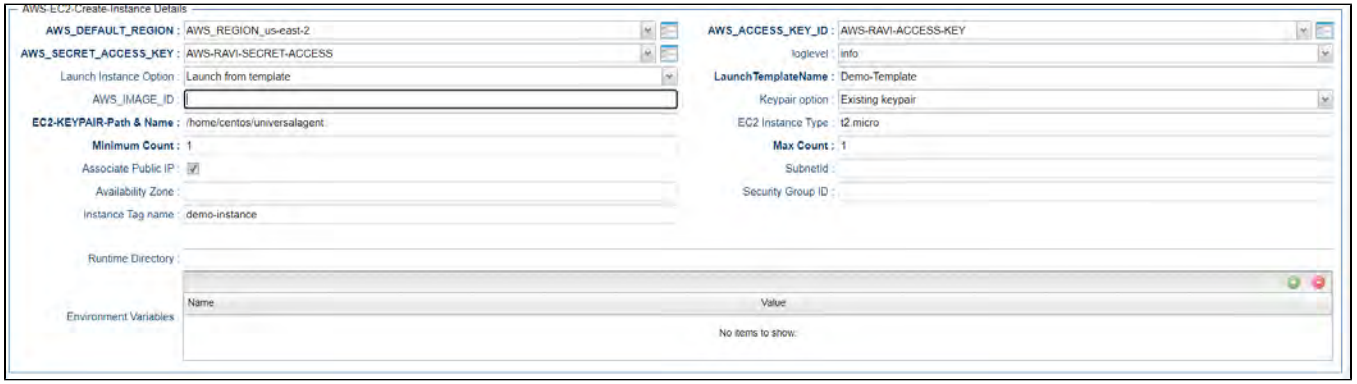
Keypair option	PEM file creation choice , Select either existing Key pair or New Key pair
EC2-KEYPAIR-Path & Name	Provide Keypair file name and the path (Do not give the extension) for new and for existing keypair just the name
EC2 Instance Type	provide ec2-instance type like t2. Micro , if Launch from template = “Create New Instance”
Minimum Count	Minimum Count of instance that need to be created, if Launch from template = “Create New Instance”
Max Count	Max count of instance that needs to be created, if Launch from template = “Create New Instance”
associate_public_ip	If a public IP needs to be created when a instance is created
SubnetId	Provide subnetID where the instance to be associated within AWS
Availability Zone	Provide Availability Zone where the instance to be associated within AWS
Security Group ID	provide security group ids, if multiple ID's then separate by comma
Instance Tag name	EC2 Instance Tag Name
iam_instance_profile_name	If applicable provide the IAM Instance Profile Name
device_name	Provide the device name; for example, /dev/sda1
ebs_volume_size	Provide EBS Volume size
EBS Vol. Type	Select either standard or io1 or gp2 or sc1 or st1
EBS Vol. Encyption	Check if encryption needs needed
EC2 Monitoring	Check this box if detailed monitoring required
Install Universal Agent	Check this box if you would need to install universal agent with this new EC2 instance created
Agent Download URL	Provide the path to download the agent URL, if install universal agent option is selected
Universal Agent Install OS	select the OS where universal agent needs to be installed
Agent OMS IP	Provide the OMS server IP for the universal agent to be connected after installation , if install universal agent option is selected
Use Public IP for SSH	Select if you would need to use the public or provide IP for SSH
os_user_id	Provide the OS user ID that will be used to make SSH connection

Examples for AWS Create EC2 Instance (with Universal Agent) Universal Tasks

New EC2 Instance Creation



Launch Instance with Launch Template



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

AWS EC2 - Start, Stop, Terminate, and Manage Universal Task Instances

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

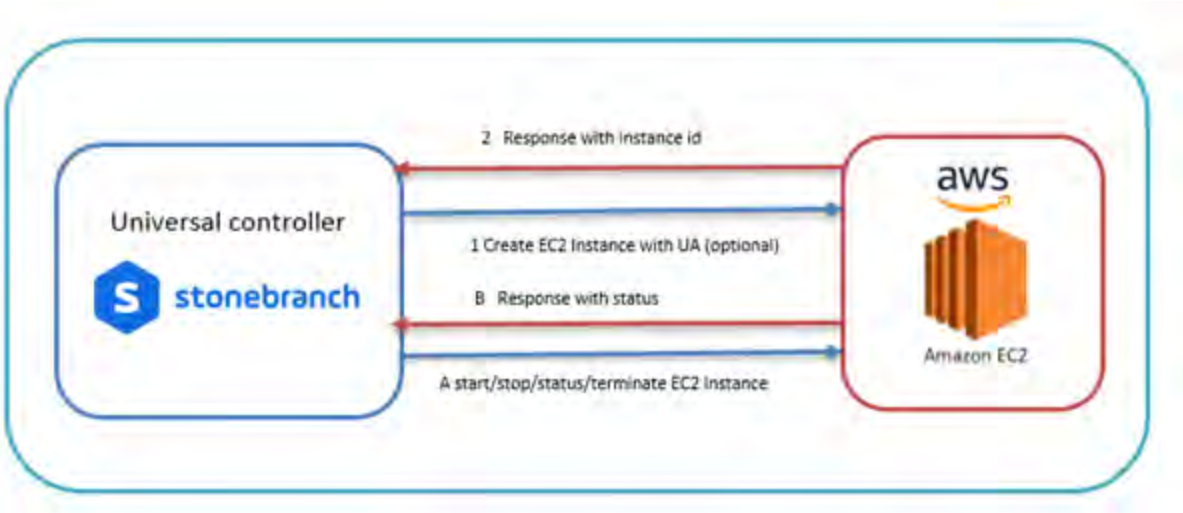
Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows users to start, stop, terminate, and manage AWS EC2 instances on demand, simply by providing one or more instance IDs as input.

Overview

- This task uses python boto3 to interact with the AWS platform using the credentials supplied within the task.
- It supports multiple EC2 instances at once.
- In Universal Controller this task goes to the success state until the EC2 instance is completely spun up or terminated.
- Scheduling this task using Universal Controller workflow spins up and tears down EC2 instances based on the business needs, complete with correct set up and dependencies.
- It dynamically manages EC2 operations, offering the potential to reduce EC2 operations costs in the cloud.



Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against AWS EC2 Instance.

Software Requirements for Universal Template and Universal Task

- Requires Python 2.7 or Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests
    - Boto3

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.6.0.0 and later with python options installed
- Universal Agent for Linux Version 6.6.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.7.0.0 and later

Software Requirements for the Application to be Scheduled

- The Server Running the Universal Agent needs to have Python 2.7.x or 3.6.x installed
- AWS programmatic Credentials -Access Key, Secret Access key and Region
- This Universal Task for the AWS EC2-start-stop-terminate has been tested with the agent bundled with python 3.6 and boto3 module

Technical Considerations

- Consider using this Universal Task either with universal agent bundled with python(uapy) and also having boto3 module within this environment or a python environment (py) in a host where Universal Agent is installed with boto3 module in it.
- AWS IAM credentials (Access Key, Secret Access key and Region) should be with the appropriate access for handling AWS EC2 instances

Start, Stop, Terminate, and Manage AWS EC2 Instance Key Features

Feature	Description
Start EC2 Instance	Start one or Multiple EC2 instance
Stop EC2 Instance	Stops one or Multiple EC2 instance
Terminate EC2 Instance	Terminate one or Multiple EC2 instance
EC2 status	Provides one or Multiple EC2 instance status

Import Start, Stop, Terminate, and Manage AWS EC2 Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Start, Stop, Terminate, and Manage AWS EC2 Instance Universal Task

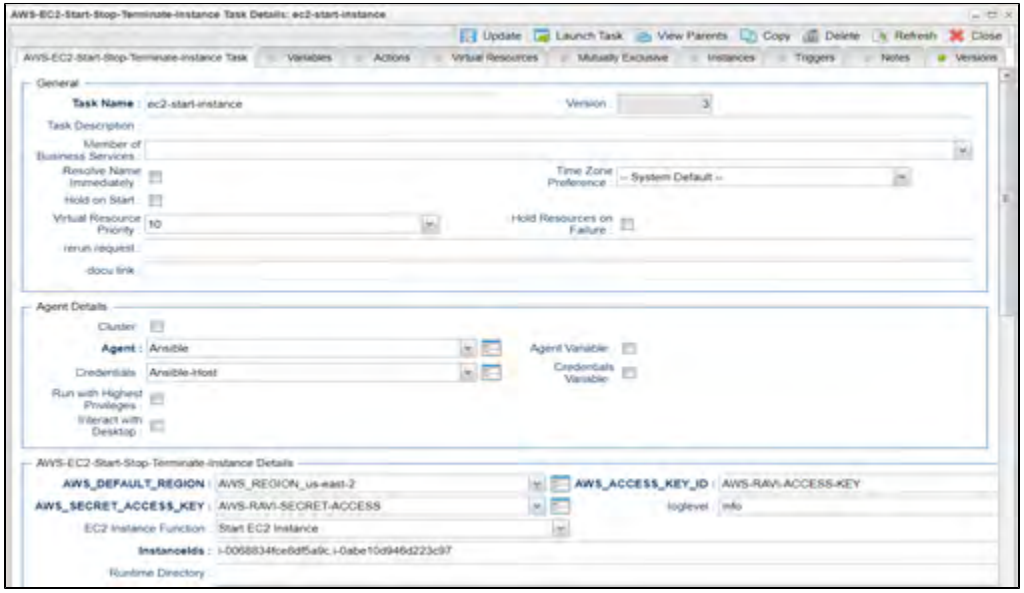
For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Start, Stop, Terminate, and Manage AWS EC2 Instance Universal Task

Field	Description
AWS-DEFAULT-REGION	AWS Region kept as credential
AWS-SECRET-ACCESS-KEY	AWS Secret Key
AWS-ACCESS-KEY-ID	AWS Access Key
EC2 Instance Function	Select either start or stop or terminate or check ec2 instance
Instance ID’s	Provide the Instance ID which you would need to start or stop or term

Examples for Start, Stop, Terminate, and Manage AWS EC2 Instance Universal Tasks

Start EC2 Instances



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

AWS S3

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

Storing data in the cloud becomes an integral part of most modern IT landscapes. With Universal Automation Center you can securely automate your AWS, Azure, Google and MinIO File Transfers and integrate them into your existing scheduling flows.

As security is one of the key concerns when moving to the cloud, the provided solution supports multi-level of security:

- Credentials for AWS S3 (Access Key, Secret Access key and Region) are stored in an encrypted form in the database
- IAM Role-Based Access Control (RBAC) is supported
- Communication to AWS is done via the HTTPS protocol
- A Proxy Server connection to AWS with basic authentication is supported
- Secure access to AWS S3 buckets using AWS bucket policies can be configured in the AWS console
- Restrict sending files only to specific buckets using AWS End Points can be configured in the AWS console

This Universal Task focuses on the AMAZON AWS S3 file transfer, including support for MinIO. MinIO is an Open Source object storage server for private cloud environments based on Amazon's S3 API. All file transfer scenarios supported for AMAZON AWS S3 are also support for MinIO. The scenarios described in this documentation are also valid for MinIO.

A similar solution as for AWS S3 is also available for Microsoft Azure Blob Storage and Google Cloud Storage.

Overview

The Universal Task for AWS S3 allows to securely transfers files from, to and between AWS S3 cloud storage buckets and folders.

The Universal Task for AWS S3 support the following main features:

- The following file transfer commands are supported:
    - Upload a file(s) to an S3 bucket
    - Download of file(s) from an S3 bucket
    - Transfer files between S3 buckets
    - List objects in an S3 bucket
    - Delete object(s) in an S3 bucket
    - List S3 bucket names
    - Create an S3 bucket
- Monitor for a key(s) in a bucket.
- File Transfer can be trigger by a third-party application using the Universal Automation Center RESTfull Webservice API: REST API.
- Universal Task for AWS S3 can be integrated into any existing scheduling workflow in the same way as any standard Linux or Windows Task type.
- Security is ensured by using the HTTPS protocol with support for an optional Proxy Server.
- AWS IAM Role Based Access (RBCA) is supported.
- No Universal Agent needs to be installed on the AWS Cloud – the communication goes via HTTPS.

Software Requirements

Software Requirements Universal Agents and Universal Controller

- Universal Agent for Linux or Windows Version 6.9.0.0 or later are required
- The Universal Agent needs to be installed with python option (--python yes)

Software Requirements Universal Controller

- Universal Controller 6.9.0.0. or later is required

Software Requirements for the Application to be scheduled

The Universal Task has been tested for the AWS S3 SDK for python (boto3) V1.15.6

Universal Task for AWS S3 Key Features

The solution supports the following file transfer scenarios:

Name	Description

Upload a file(s) to a bucket (Copy or Move)	• A single or multiple files can be uploaded to a bucket.
	• Move or copy are supported for the upload
	• Unix filename pattern matching support e.g. wild card support “*” to upload multiple files
	• A single or multiple files can be Uploaded to a folder using a prefix
	• It can be decided via an the *Upload Write Options*:
	  • to overwrite an existing object (*Replace existing Object*)
	  • to cancel the operations in case an object with a similar name exists (*Do not overwrite existing Object*)
	  • to add a timestamp to the uploaded Object (*Timestamp*)

Download of file(s) from a bucket (Copy or Move)	• A single or multiple files should be downloaded from a bucket.
	• Move or copy must be supported for the download
	• Unix filename pattern matching support e.g. wild card support “*” to download multiple files
	• Download to a specific folder is supported
	• It can be decided via an the *Download Write Options*:
	  • to overwrite an existing file (*Replace existing File*)
	  • to cancel the operations in case a File with a similar name exists (*Do not overwrite existing File*)
	  • to add a timestamp to the uploaded file (*Timestamp*)
	  • to perform the default Windows behaviour for copying files (*Default Windows behaviour*)
	  • If a file with a similar name exists, the file names that are similar will be edited so that the files you copied have a number appended at the end of them. For example, if you’re copying a file named image.png to a folder that already has a file named image.png in it, the copied file will be named image (1).png.

Copy object to a bucket	• An object can be copied from one AWS S3 bucket to another. • Folders are support
List objects in a bucket	• Show all Objects in a bucket • Show all Objects in a folder • Unix filename pattern matching support e.g. wild card support “*” to narrow down the objects to display
Delete object(s) in a bucket	• Delete one or multiple objects in a bucket • Delete one or multiple objects in folder in a bucket • Unix filename pattern matching support e.g. wild card support “*” to narrow down the objects to be deleted
List buckets	List all bucket in an AWS S3 account

Create an S3 bucket	Create a new S3 bucket
Delete an S3 bucket	Delete an S3 bucket
Monitor Key	Monitor if a certain key(s) exists in an S3 bucket. Unix filename pattern matching support; for example, wild card support “*” to narrow down the keys to be monitored.
Proxy Server Connection	A proxy server connection with or without basic authentication can be configured
Integration into 3rd Party Applications	An AWS S3 file transfer can be triggered via the Universal Automation Center RESTfull Webservice API within an application.
Self-Service through Web-Client	The S3 Task can be fully configured, monitored and updated via the Universal Controller Web-GUI

Import AWS S3 Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure AWS S3 Universal Tasks

For the new Universal Task type, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for AWS S3 Universal Task - Action

The AWS S3 Task provides multiple different file transfer actions. For each action the specific fields are described.

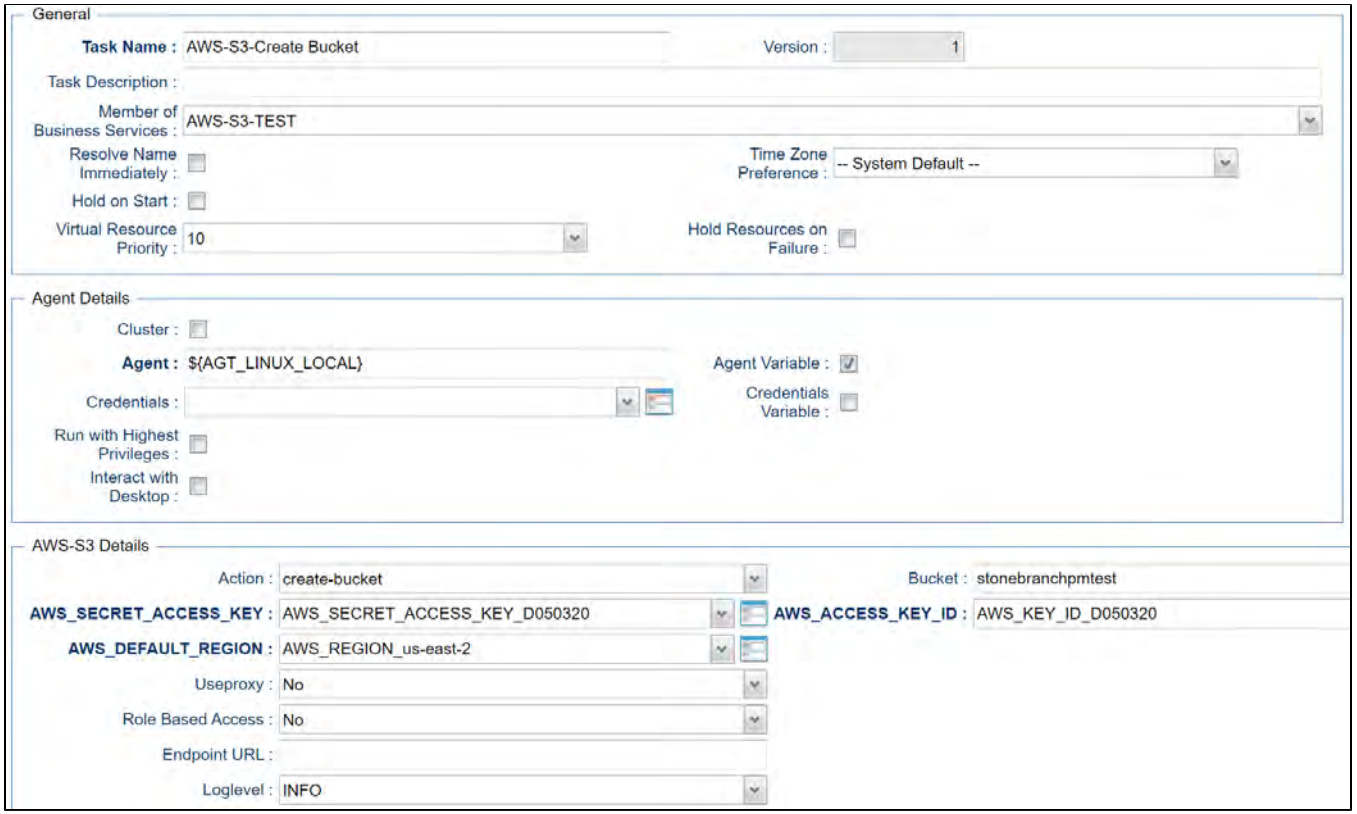
Create Bucket - Action

The Action Creates a new S3 bucket.

Field	Description
Action	create-bucket action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES]

	If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection; for example, rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Bucket	Name of the AWS Bucket to create

Example for AWS S3 Universal Tasks - Create Bucket



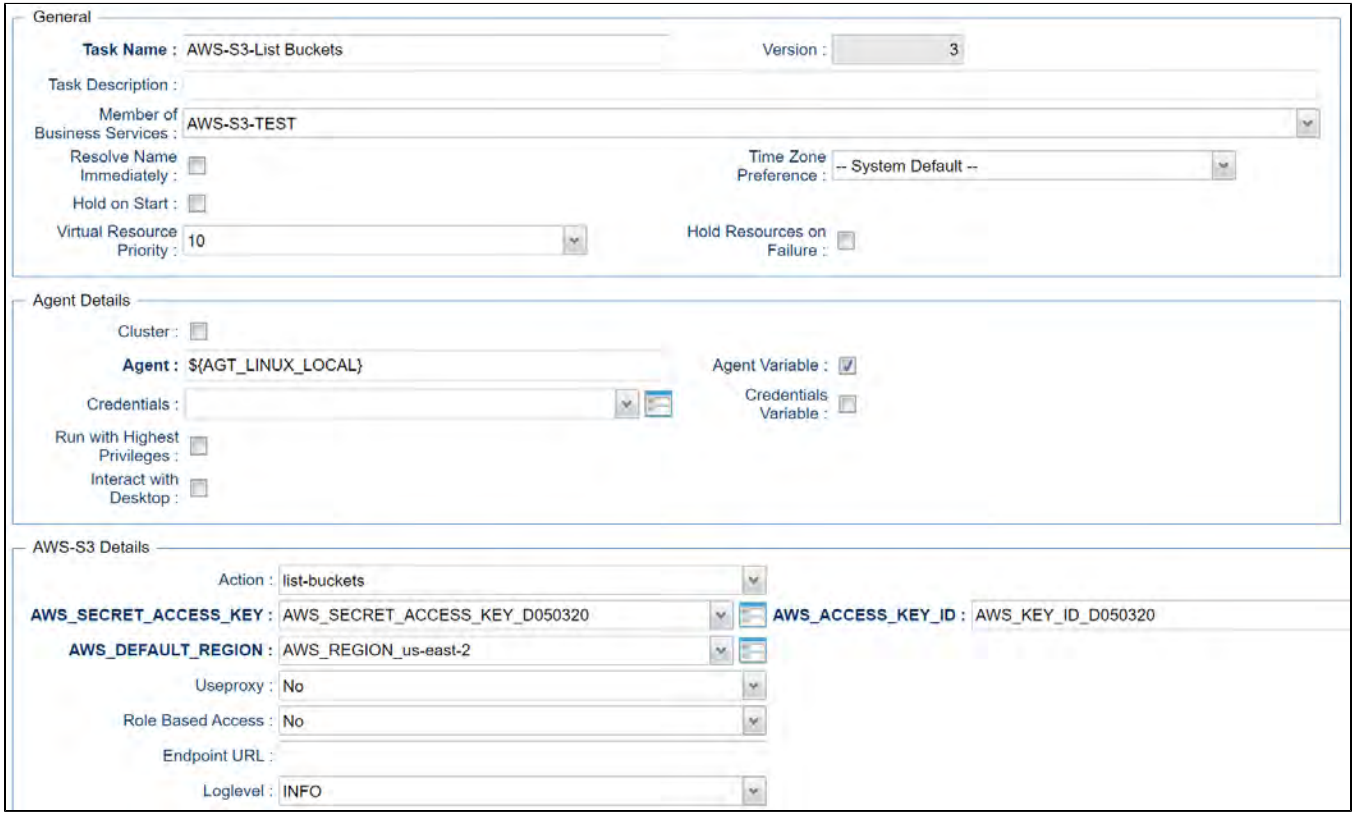
List Buckets - Action

The Action list all buckets of an AWS Account.

Field	Description
Action	list-buckets action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)

Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system

Example for AWS S3 Universal Tasks - List Buckets

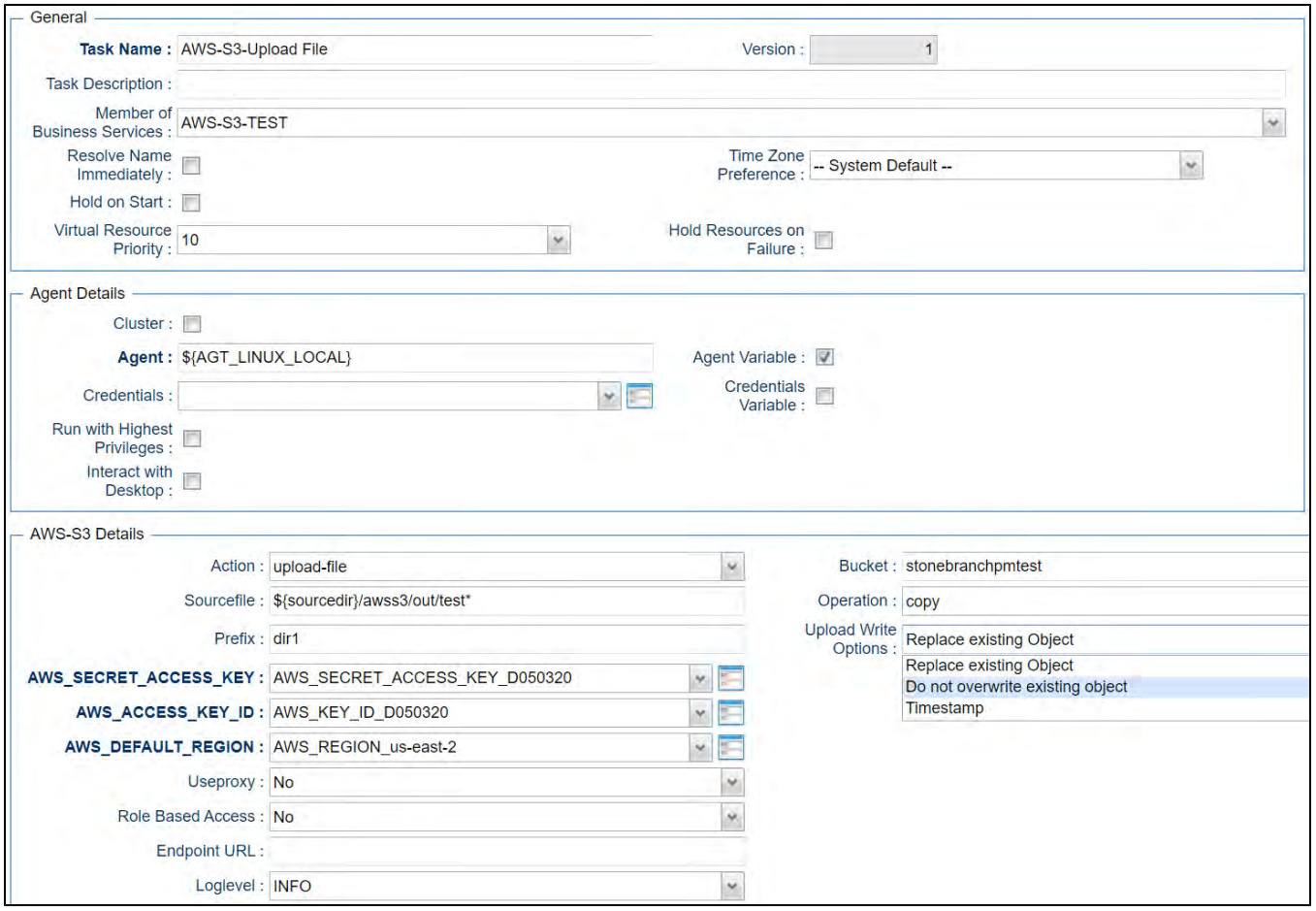


Upload File - Action

The Action is used to upload a single or multiple files from a Windows or Linux server to an AWS bucket or a folder in an AWS bucket.

Field	Description
Action	Upload-file action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage
Sourcefile	Source file(s) to be uploaded to a bucket or specific folder in a bucket. Unix filename pattern are supported to upload a selection of files. Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • "*" matches everything
Operation	[copy | move] In case of a "move" the source file(s) is/are deleted after the upload to the bucket.
Prefix	A folder in a bucket is called prefix in AWS. In the field Prefix only the name of the folder needs to be provided no "/" after the folder name.
Upload Write Options	Upload Write Options [Replace existing Object | Do not overwrite existing Object | Timestamp] • Replace existing Object: Overwrite an existing object • Do not overwrite existing Object: Cancel the operations in case an object with a similar name exists • Timestamp: Add a timestamp to the uploaded Object

Example for AWS S3 Universal Tasks - Upload File



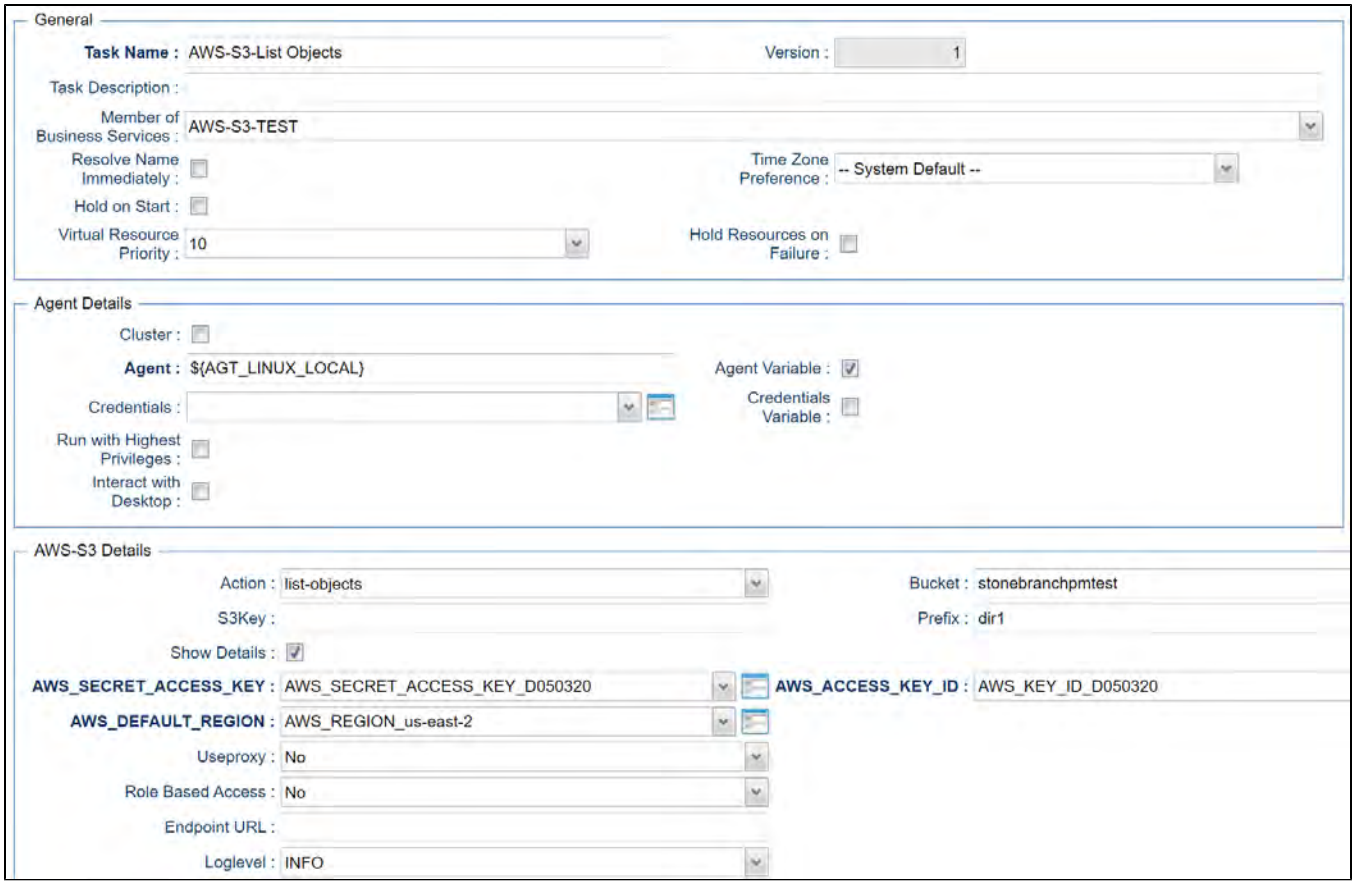
List Objects - Action

The Action is used to display objects in a bucket or a specific bucket folder (prefix).

Field	Description
Action	List-objects action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key

Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Bucket	Bucket name in AWS
Prefix	A folder in a bucket is called prefix in AWS. In the field Prefix only the name of the folder needs to be provided no “/” after the folder name. If a prefix is provided only objects in the folder with the prefix name are listed in the output.
S3key	Objects matching the given S3key are listed. Unix filename pattern are supported to list only a selection of files: Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything Example: S3key = test* : matches everything starting with test S3key = test[1-2].txt : matches test1.txt, test2.txt S3key = test[!1].txt: does not match test1.txt S3key = test?.txt: matches test1.txt, test2.txt etc.
Show Details	Show details like creation timestamp in the output

Example for AWS S3 Universal Tasks - List Objects



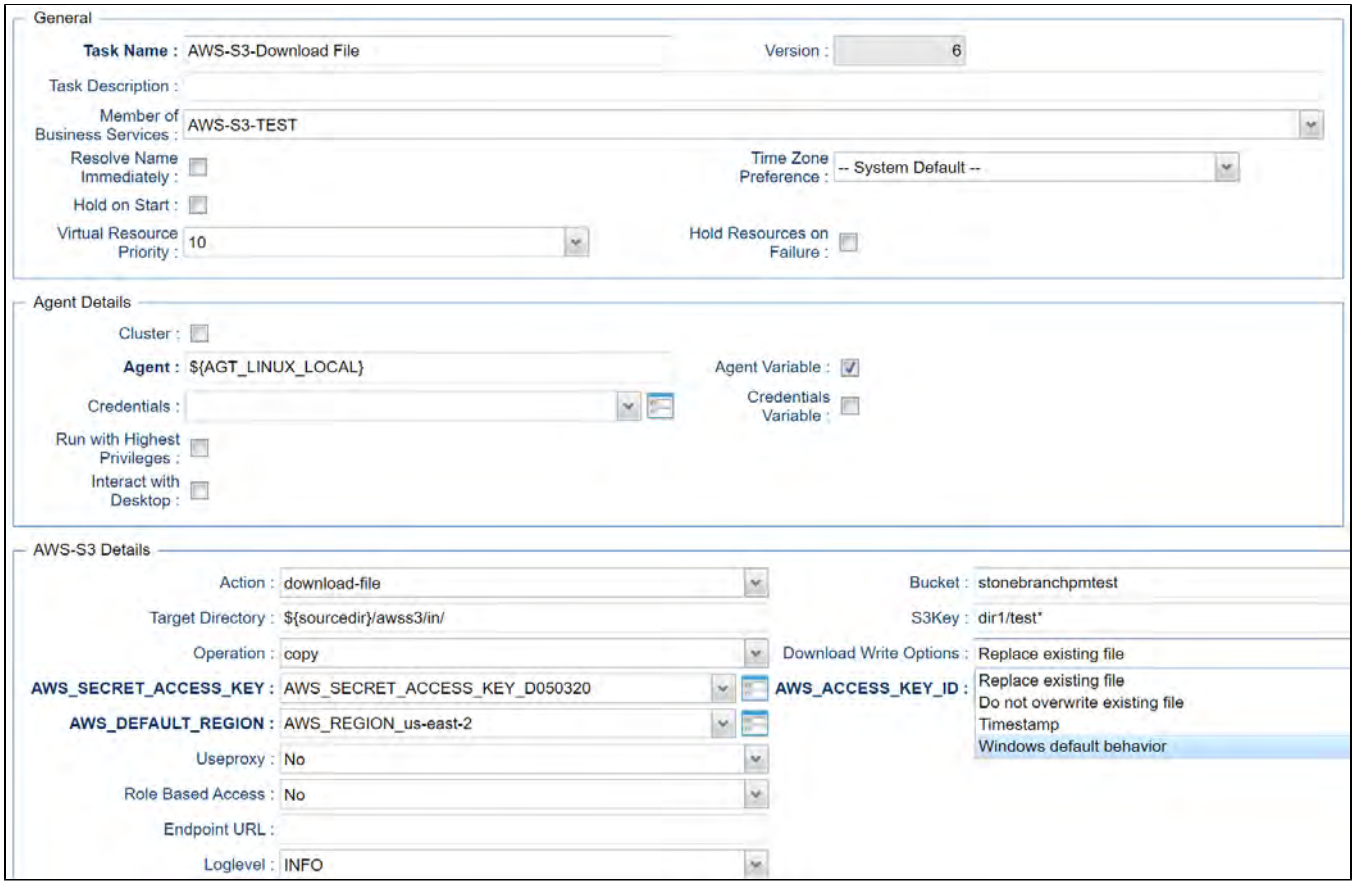
Download File - Action

This Action downloads one or multiple files from a bucket to a Linux or Windows folder

Field	Description
Action	Download-file action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key

Useproxy (default is NO)	[*NO* | *YES*] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Target Directory	Linux or Windows Target Directory For example, C:\tmp\ or /home/ubuntu/download Unix filename pattern are supported to download a selection of files. Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything
Operation	[copy | move] In case of a “move” the objects are deleted after they have been download from the bucket.
Download Write Options	Download Write Options: [Replace existing File | Do not overwrite existing File | Timestamp | Default Windows behaviour] • Replace existing File : overwrite an existing file • Do not overwrite existing File: cancel the operations in case a File with a similar name exists • Timestamp: add a timestamp to the uploaded file • Default Windows behaviour: perform the default Windows behaviour for copying files. If a file with a similar name exists, the file names that are similar will be edited so that the files you copied have a number appended at the end of them. For example, if you’re copying a file named image.png to a folder that already has a file named image.png in it, the copied file will be named image (1).png.

Example for AWS S3 Universal Tasks - Download File



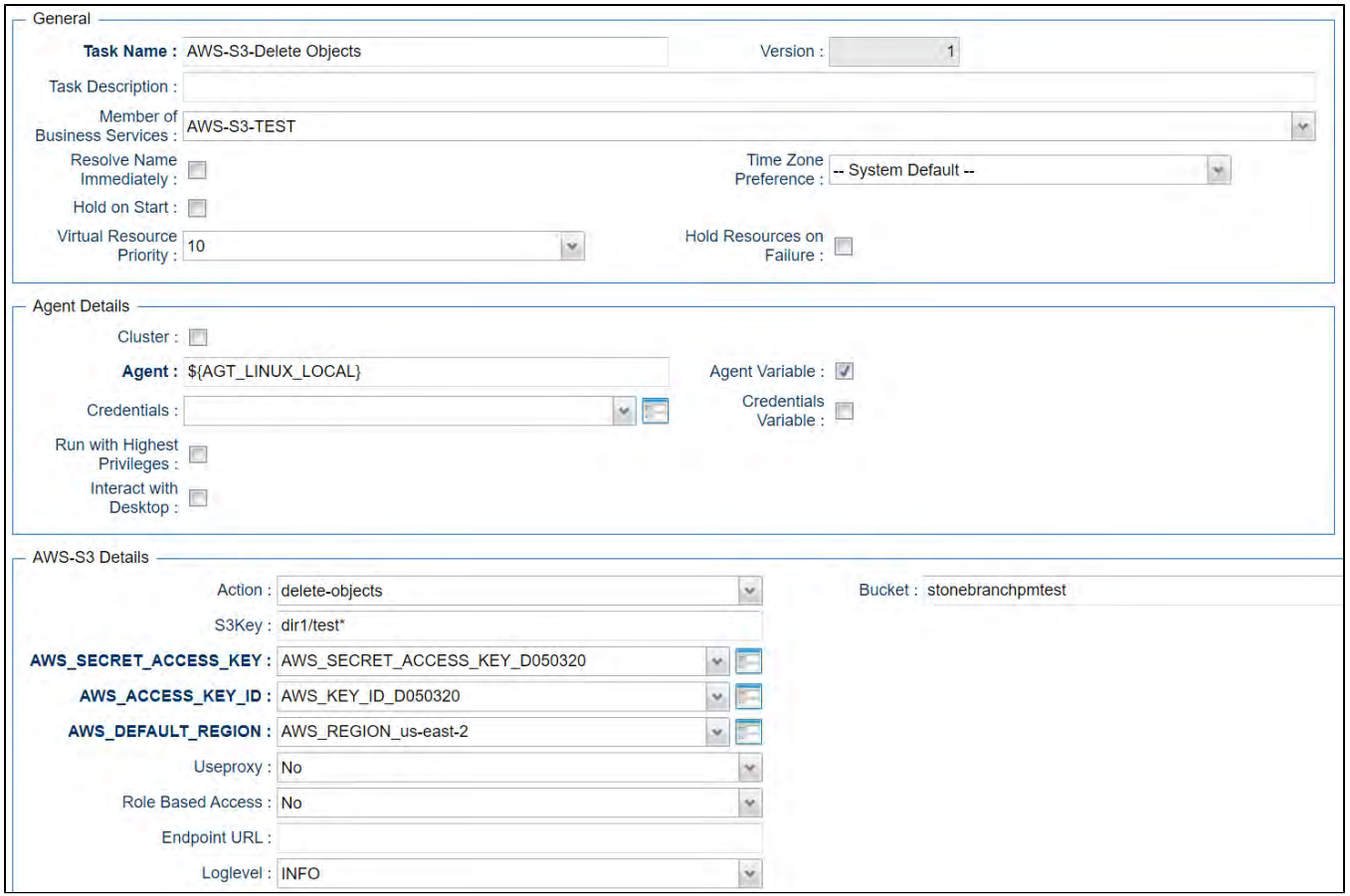
Delete Objects - Action

The Actions is used to delete an object in a bucket or folder.

Field	Description
Action	Delete-objects action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname

	• Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Bucket	AWS bucket name
S3key	Key to be deleted in AWS Note: Due to security reasons wild card is only support, if at least one character is provided e.g. t* would delete all files starting with at “t”.

Example for AWS S3 Universal Tasks - Delete Objects



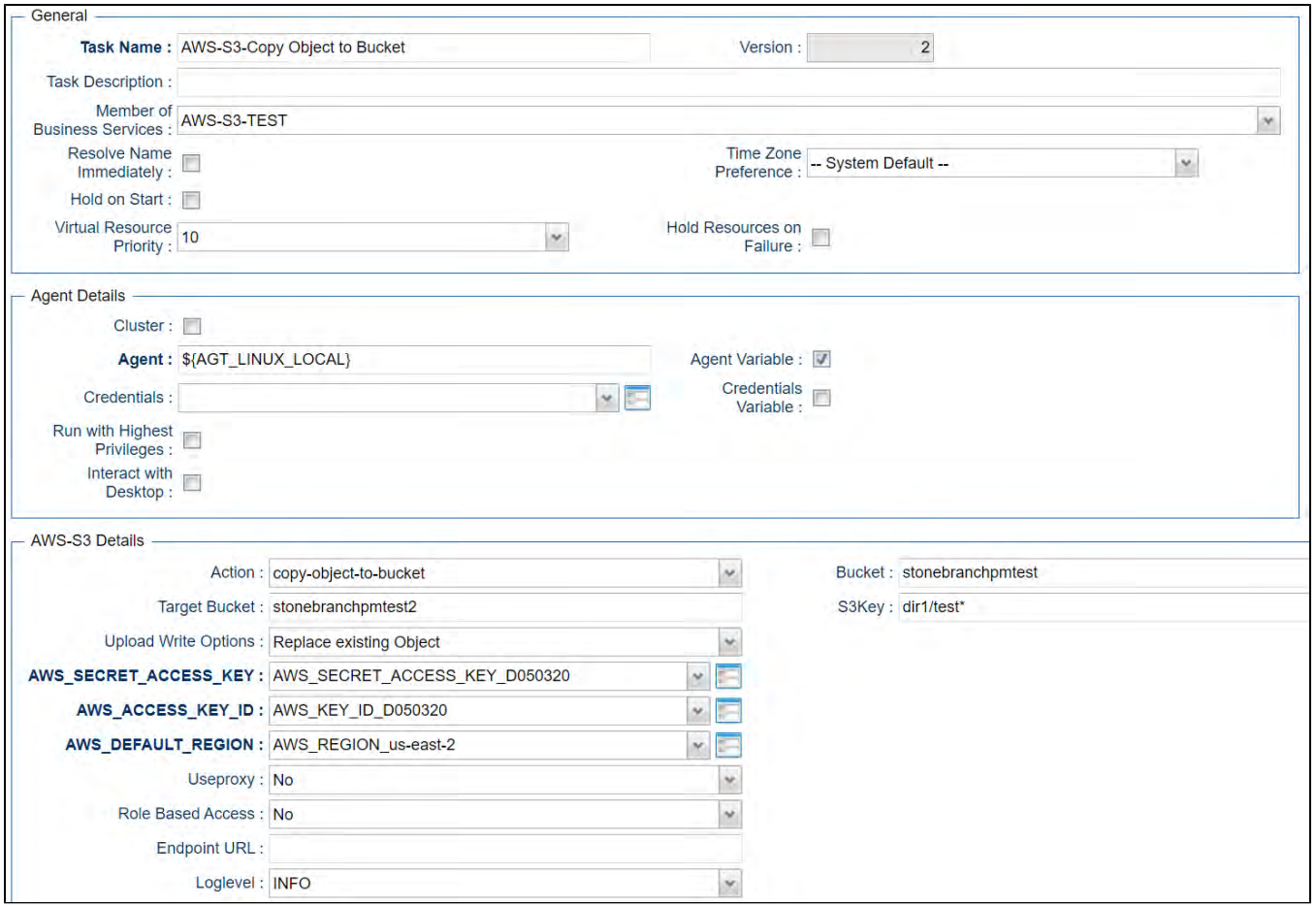
Copy Object to Bucket - Action

This Action is used copy a single or multiple objects from one bucket to another bucket in AWS

Field	Description
Action	Copy-object-to-bucket action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed:

	• Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Bucket	AWS bucket name
Target Bucket	target Bucket, where the object(s) will be copied to
S3key	File to be copied from one bucket to another
Upload Write Options	Upload Write Options [Replace existing Object | Do not overwrite existing Object | Timestamp] • Replace existing Object: Overwrite an existing object • Do not overwrite existing Object: Cancel the operations in case an object with a similar name exists • Timestamp: Add a timestamp to the uploaded Object

Example for AWS S3 Universal Tasks - Copy Object to Bucket



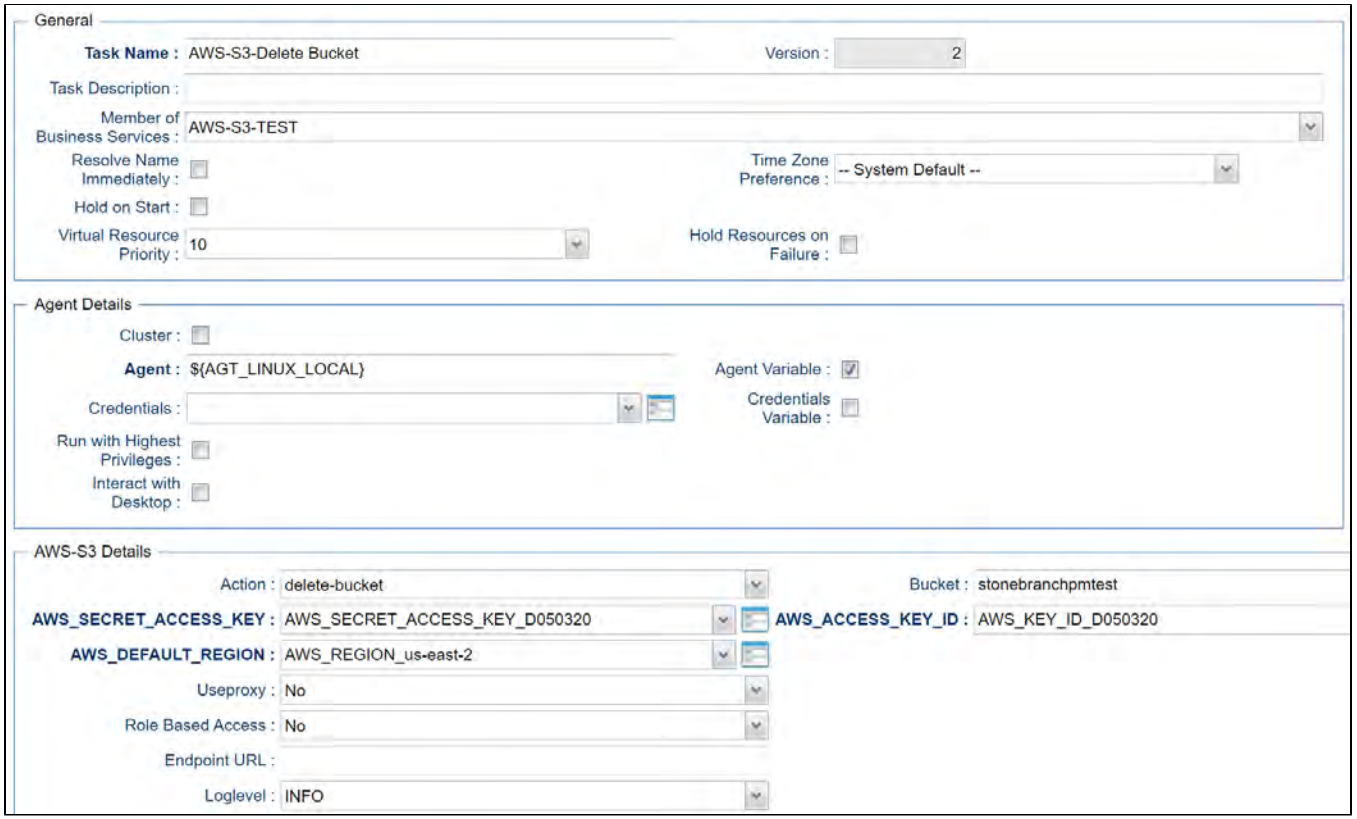
Delete Bucket - Action

This action is used to delete a bucket

Field	Description
Action	Delete-bucket action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES]

	If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Bucket	Name of the bucket to be deleted

Example for AWS S3 Universal Tasks - Delete Bucket



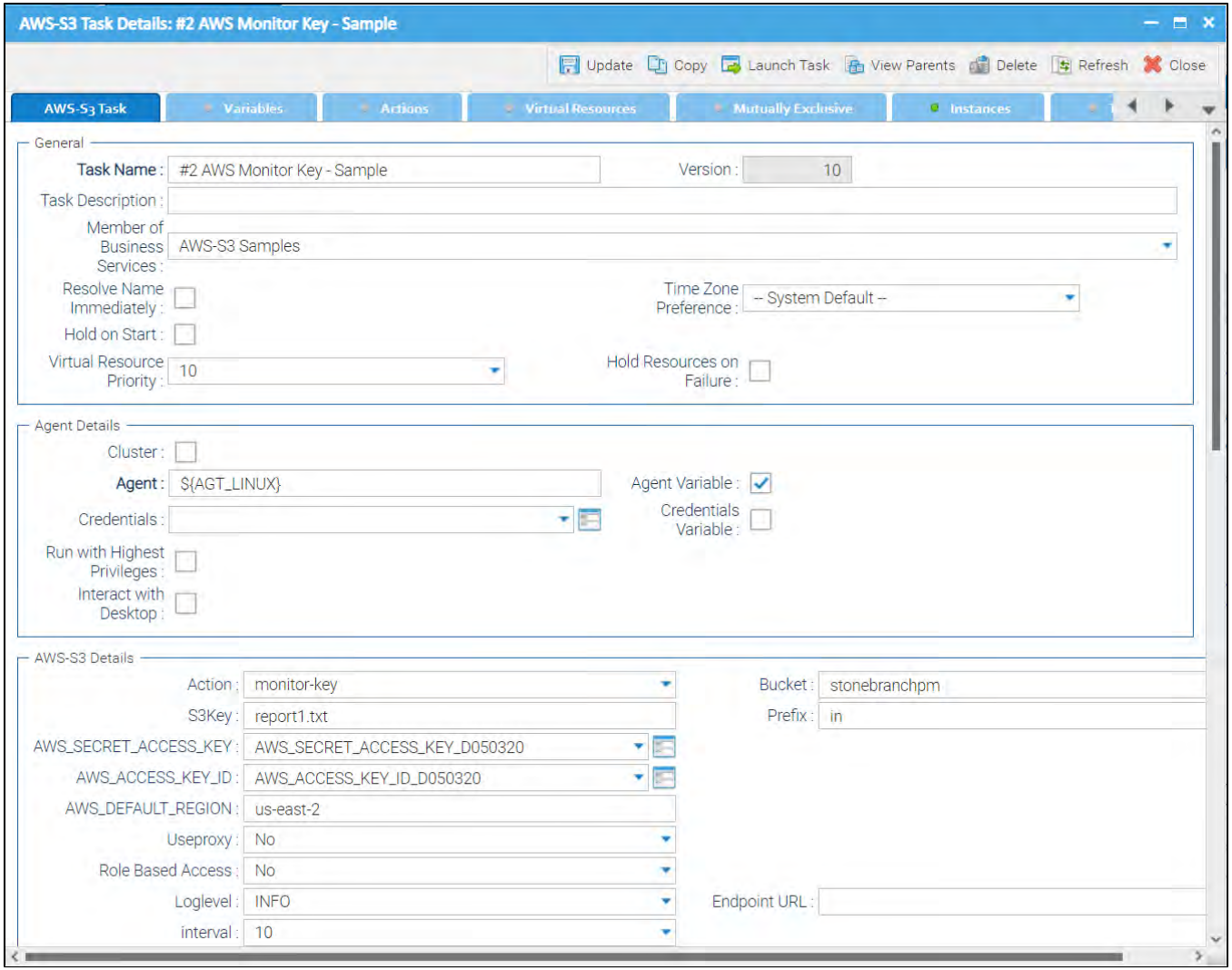
Monitor Key - Action

This Action is used to monitor for a key(s) in a bucket

Field	Description
Action	monitor-key action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account credentials Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname

	• Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection e.g. rn:aws:iam::111222333444:role/SB-AWSS3 • Service Name [STS | S3] , default is STS STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system
Bucket	Bucket name in AWS
Prefix	A folder in a bucket is called prefix in AWS. In the field Prefix only the name of the folder needs to be provided no “/” after the folder name. If a prefix is provided only objects in the folder with the prefix name are listed in the output.
S3key	Objects matching the given S3key are listed. Unix filename pattern are supported to list only a selection of files: Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything Example: S3key = test* : matches everything starting with test S3key = test[1-2].txt : matches test1.txt, test2.txt S3key = test[!1].txt: does not match test1.txt S3key = test?.txt: matches test1.txt, test2.txt etc.
Interval	[10 | 60 | 180] Monitoring Interval in seconds Example: An Interval of 60s means every 60s the bucket is scanned for the key(s) to monitor.

Example for AWS S3 Universal Tasks - Monitor Key


AWS-S3 Task Details: #2 AWS Monitor Key - Sample
Update
Copy
Launch Task
View Parents
Delete
Refresh
Close
AWS-S3 Task
Variables
Actions
Virtual Resources
Mutually Exclusive
Instances
General
Task Name :
#2 AWS Monitor Key - Sample
Version :
10
Task Description :
Member of Business Services :
AWS-S3 Samples
Resolve Name Immediately :
Time Zone Preference :
-- System Default --
Hold on Start :
Virtual Resource Priority :
10
Hold Resources on Failure :
Agent Details
Cluster :
Agent :
${AGT_LINUX}
Agent Variable :
Credentials :
Credentials Variable :
Run with Highest Privileges :
Interact with Desktop :
AWS-S3 Details
Action :
monitor-key
Bucket :
stonebranchpm
S3Key :
report1.txt
Prefix :
in
AWS_SECRET_ACCESS_KEY :
AWS_SECRET_ACCESS_KEY_D050320
AWS_ACCESS_KEY_ID :
AWS_ACCESS_KEY_ID_D050320
AWS_DEFAULT_REGION :
us-east-2
Useproxy :
No
Role Based Access :
No
Loglevel :
INFO
Endpoint URL :
interval :
10

AWS SQS

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

Overview

Using SQS, you can send, store, and receive messages between software components.

The Universal Task for SQS allows you to create, send and monitor AWS SQS messages and automatically trigger a Task in Universal Controller once a message has been received.

Software Requirements

Software Requirements Universal Agents and Controller

- Universal Agent for Linux or Windows Version 6.9.0.0 or later are required

Software Requirements Universal Controller

- Universal Controller 6.9.0.0. or later is required
- A Universal Controller license key with support for SAP connector is required

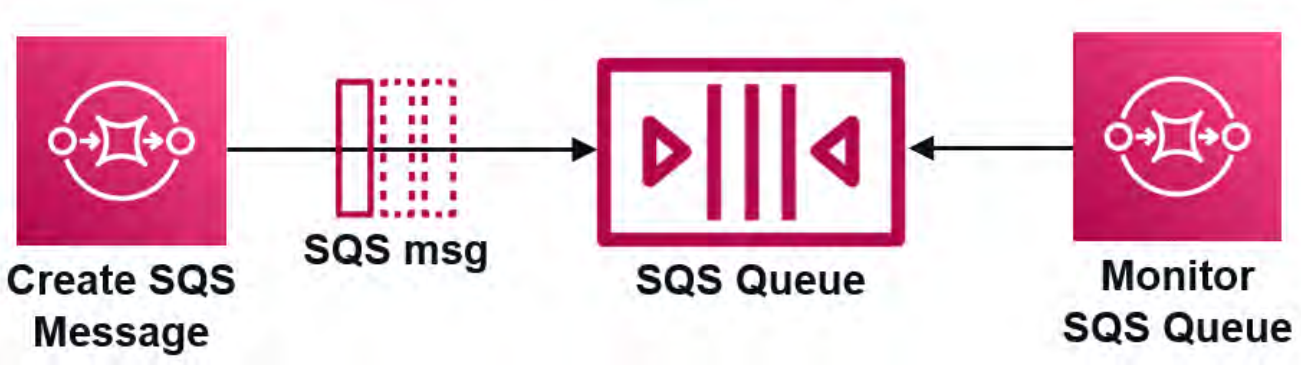
Software Requirements for the Application to be Scheduled

In order to connect to the SAP System the SAP NetWeaver RFC SDK 7.50 libraries are required from SAP.

Those can be downloaded from the SAP Software Download: SAP NetWeaver RFC SDK 7.50

Universal Task for AWS SQS Key Features

Some details about the Universal Tasks for AWS SQS:



- The Universal Tasks for SQS allows you to create, send and monitor for new AWS SQS messages.
- The Universal Task for SQS can trigger a Task in Universal Controller upon each arrival of a new message
- Credentials for AWS S3 are stored in an encrypted format in the database
- IAM Role-Based Access Control (RBAC) is supported
- Communication to AWS is done via the HTTPS protocol
- A Proxy Server connection to AWS with basic authentication is supported
- You can create and send a SQS message out of any modern application by calling the Universal Controller Remote Webservice API
- The new SQS task can be integrated into any existing or new automation workflow like any other task.
- The Universal template script is using the Python Boto3 Module. This allows to quickly introduce new AWS Service and to update the current SQS Task when new requirements occur.
- You can set different log-levels for the Universal task, providing you more information in case of issues

Import AWS SQS Virtual Machine Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure AWS SQS Universal Tasks

For the new Universal Task type AWS SQS, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for AWS SQS Universal Task - Actions

The AWS SQS Task provides three different Actions.

- receive-message
- list-queues
- send-message

For each action, the specific fields are described and an example is provided.

list-queues - Action

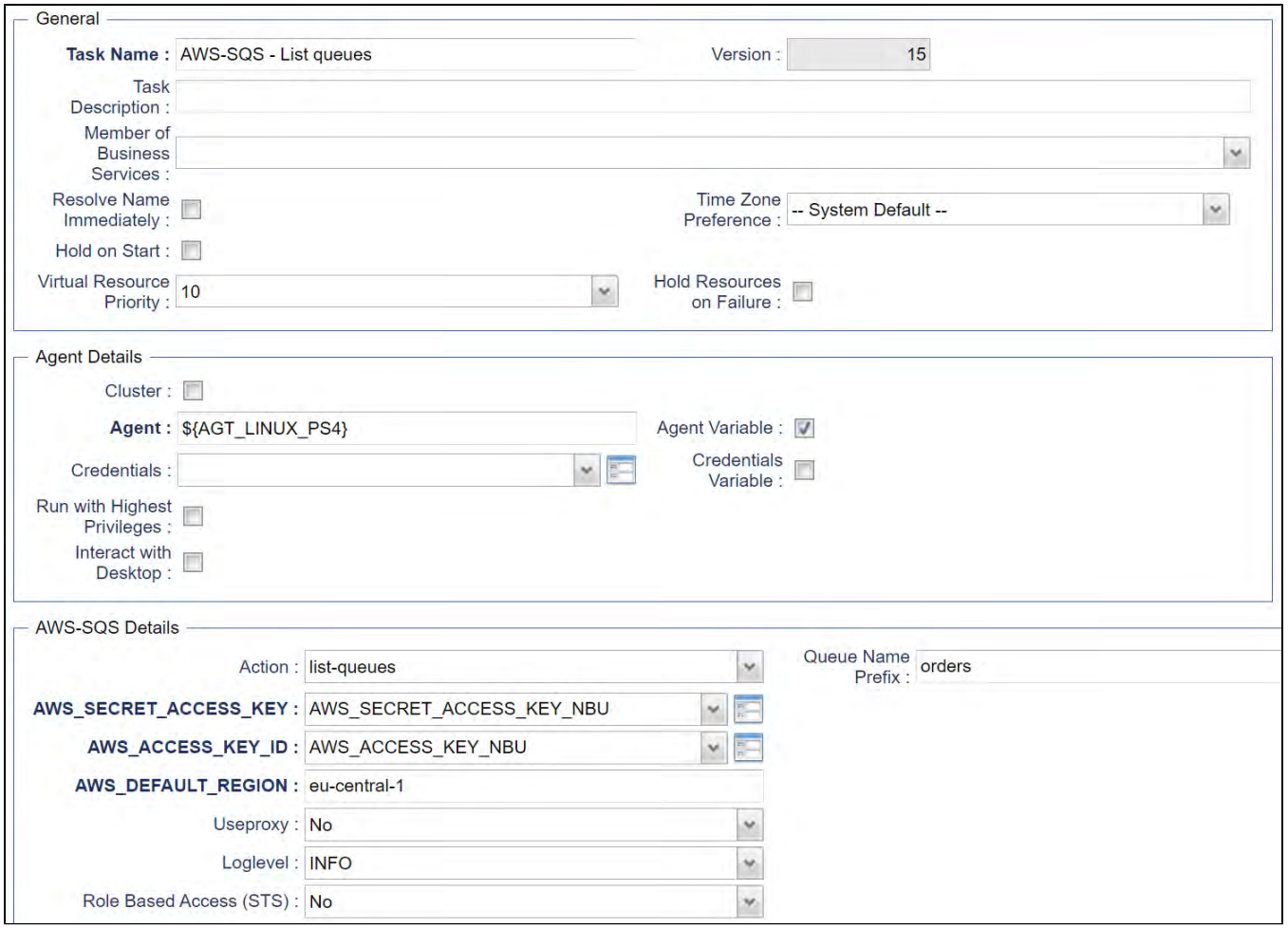
The Action list the available SQS queues for the given AWS Account and Role ARN (optional).

Field	Description
Action	list-queues action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Queue Name Prefix	A string to use for filtering the list results. Only those queues whose name begins with the specified string are returned. Queue URLs and names are case-sensitive.

	e.g. Queue Name Prefix = orders , will list all Queues starting with the name orders
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access (STS)	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection Example RoleArn: *arn:aws:iam::119322085622:role/SB-SQS-ReadOnly* STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources like SQS Queues.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system

Examples for AWS SQS Universal Tasks - Action: list-queues

List-Queues - Action



Send-message - Action

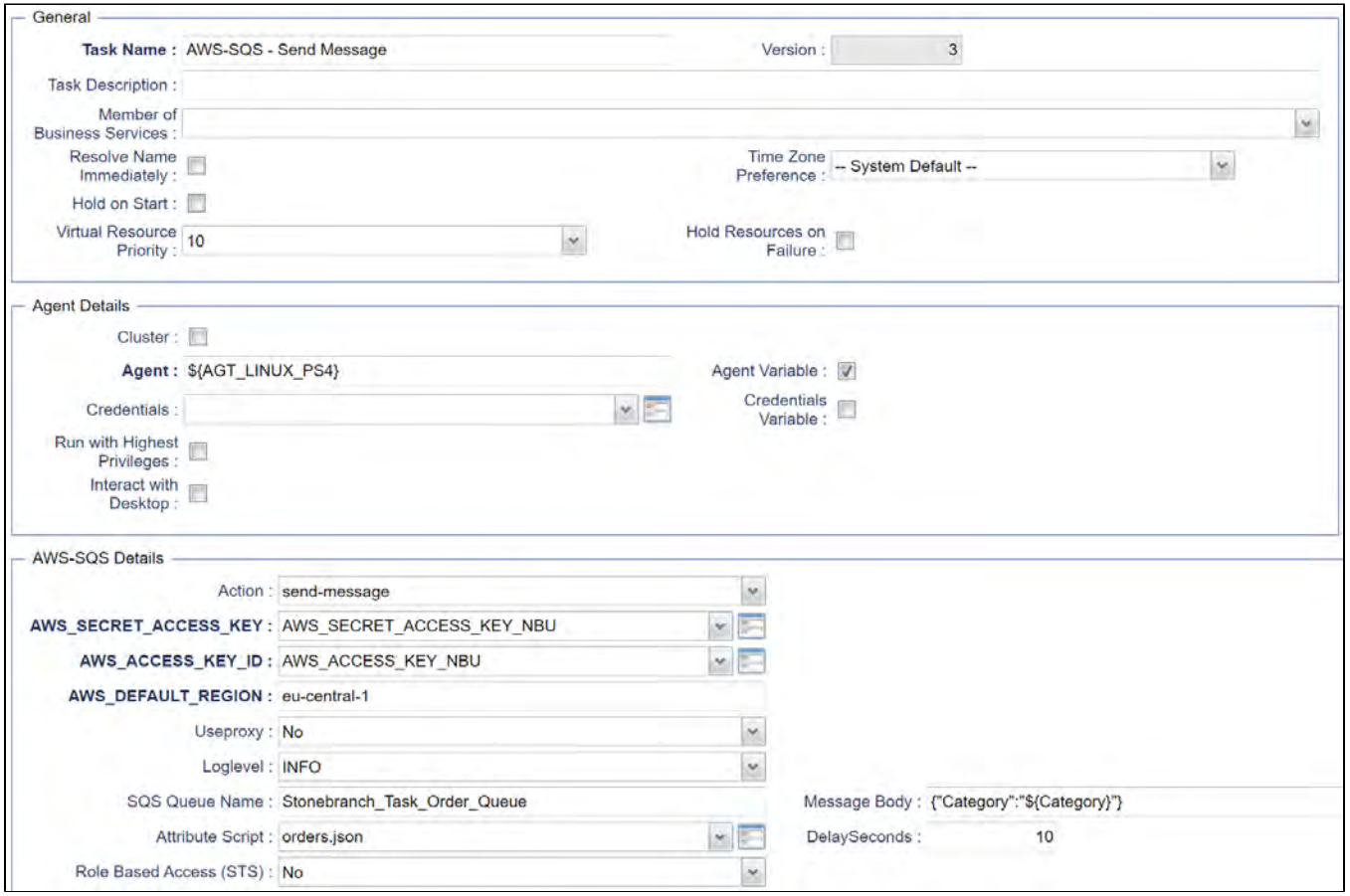
This Action inserts a message into the given AWS SQS queue. The message Body and Attributes are configurable.

Field	Description
Action	send-message action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account Region

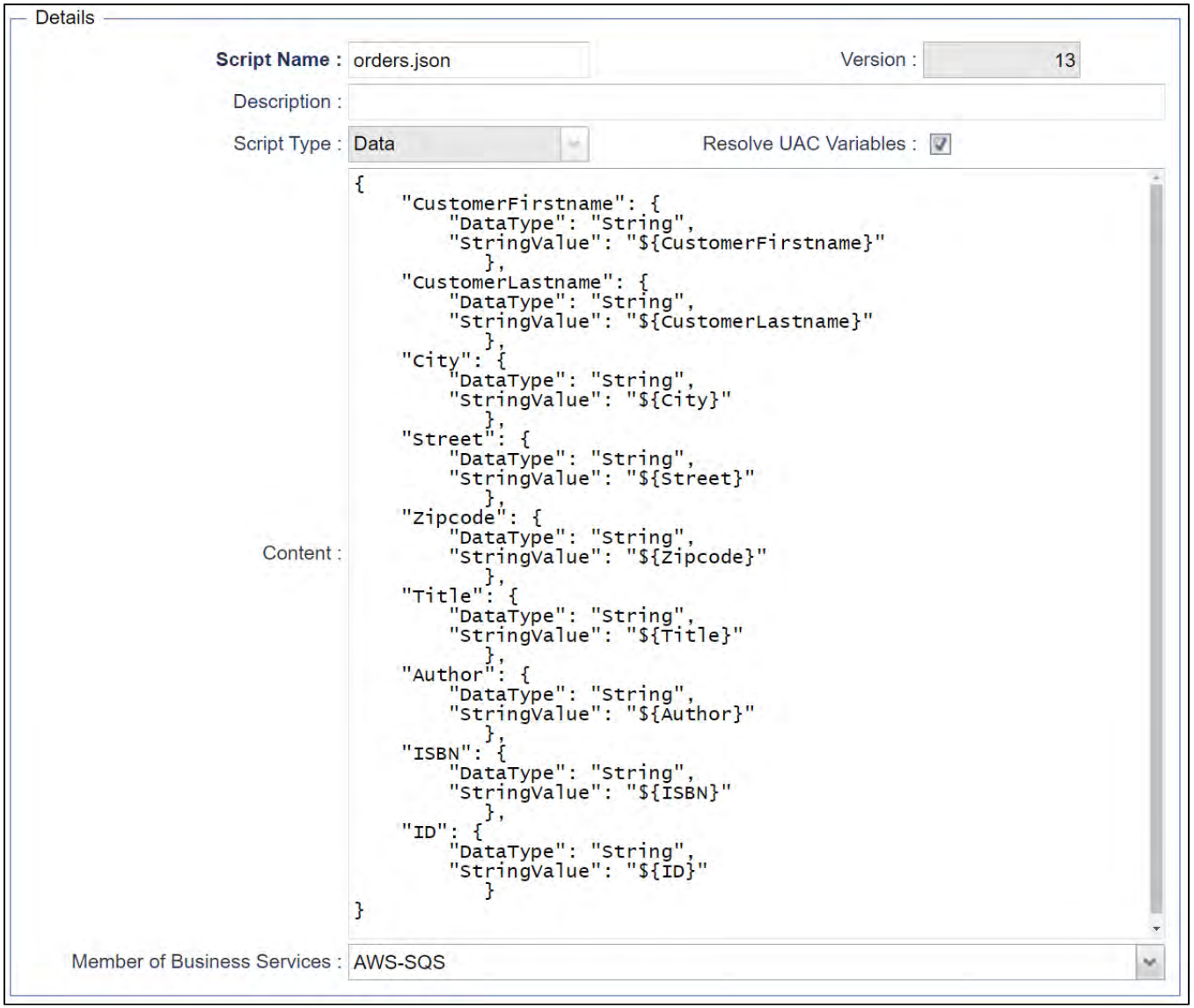
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
SQS Queue Name	Name of the SQS Message queue Queue names are case-sensitive.
Message Body	SQS Message Body The message must contain the parameter Message Body. Parameter is of Type string. Example: {"Category":"Books"}
Attribute Script	The Attribute Script is a script in json format, which is saved in the Controller script library. Each message attribute consists of a `Name` , `Type` , and `Value` . The message attributes can be provided using JSON format. Example of Attribute Script with 2 Attributes: `{ "CustomerFirstname": { "DataType": "String", "StringValue": "Nils" }, "CustomerLastname": { "DataType": "String", "StringValue": "Buer" } }` Amazon SQS lets you include structured metadata (such as timestamps, geospatial data, signatures, and identifiers) with messages using *message attributes*. Each message can have up to 10 attributes. Message attributes are optional and separate from the message body (however, they are sent alongside it).
Delay Seconds	The length of time, in seconds, for which to delay a specific message. Valid values: 0 to 900. Maximum: 15 minutes. Messages with a positive DelaySeconds value become available for processing after the delay period is finished. If you don't specify a value, the default value for the queue applies.
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access (STS)	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection Example RoleArn: *arn:aws:iam::119322085622:role/SB-SQS-ReadOnly* STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources like SQS Queues.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system

Examples for AWS SQS Universal Tasks - Action: send-message

Send-message - Action



Attribute Script Example



Script Example: orders.json

```
{
    "CustomerFirstname": {
        "DataType": "String",
        "StringValue": "${CustomerFirstname}"
```

```
            },
        "CustomerLastname": {
            "DataType": "String",
            "StringValue": "${CustomerLastname}"
            },
        "City": {
            "DataType": "String",
            "StringValue": "${City}"
            },
        "Street": {
            "DataType": "String",
            "StringValue": "${Street}"
            },
        "Zipcode": {
            "DataType": "String",
            "StringValue": "${Zipcode}"
            },
        "Title": {
            "DataType": "String",
            "StringValue": "${Title}"
            },
        "Author": {
            "DataType": "String",
            "StringValue": "${Author}"
            },
        "ISBN": {
            "DataType": "String",
            "StringValue": "${ISBN}"
            },
        "ID": {
            "DataType": "String",
            "StringValue": "${ID}"
            }
}
```

Receive-message - Action

The Action polls in configurable interval the provided SQS queue. If a message is found a Task can be launched (optional). Optionally it be configured after a message is received the task goes to success and does not continue to poll for new messages.

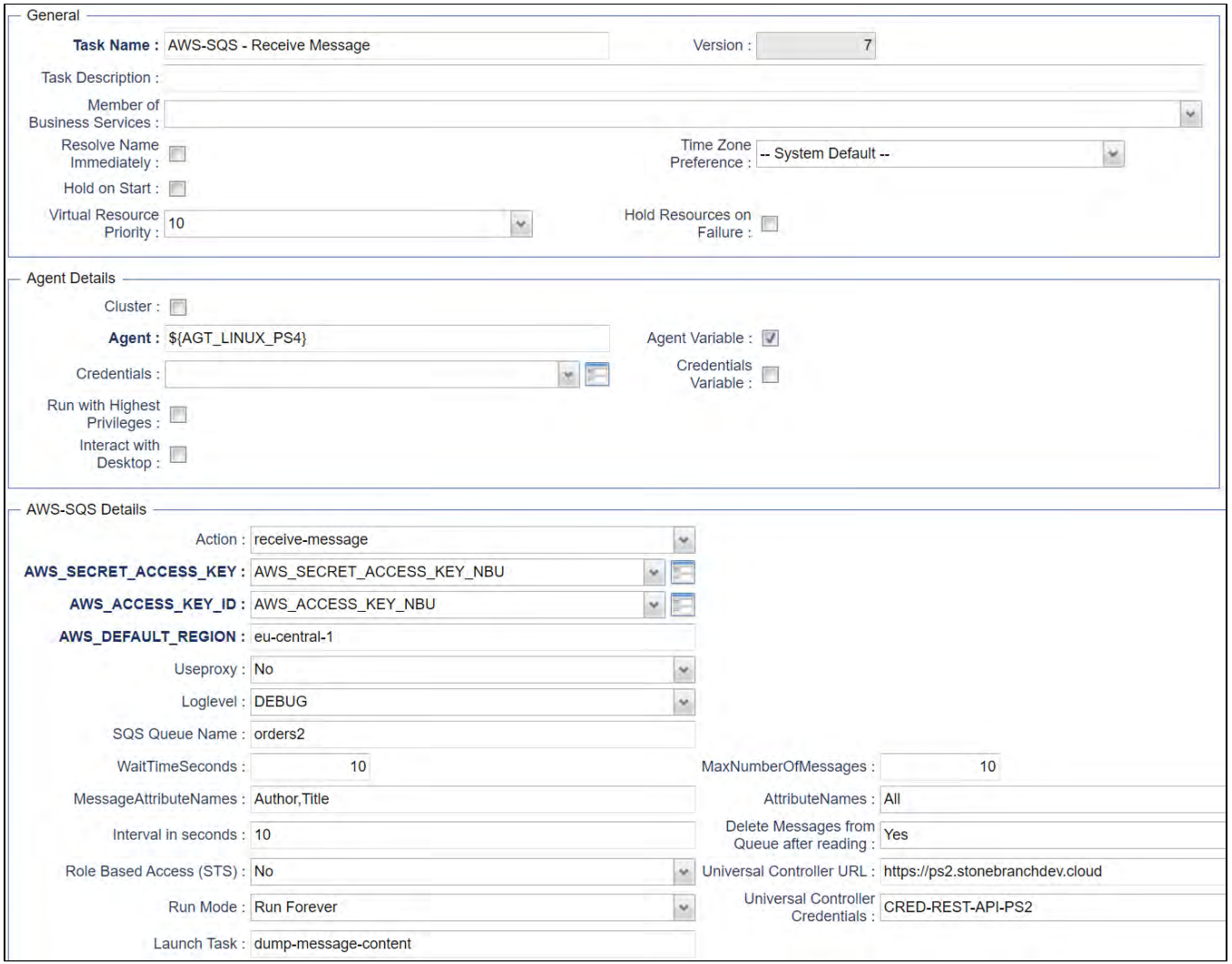
Field	Description
Action	receive-message action
AWS_ACCESS_KEY_ID	AWS Account credentials Access Key
AWS_DEFAULT_REGION	AWS Account Region
AWS_SECRET_ACCESS_KEY	AWS Account credentials Secret Access Key
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port

	• Proxy Server Credentials (optional)
Universal Controller URL	Universal Controller URL URL has no backslash “/” at the end. Example: http://192.168.88.10:8080/uc
Universal Controller Credentials	Universal Controller Credentials The Credentials need to have “Web Service Access” Permissions
Attribute Names	[All|Policy|VisibilityTimeout|MaximumMessageSize|MessageRetentionPeriod|ApproximateNumberOfMessages|ApproximateNumberOfMessagesNotVisible|CreatedTimestamp|LastModifiedTimestamp|QueueArn|ApproximateNumberOfMessagesDelayed|DelaySeconds|ReceiveMessageWaitTimeSeconds|RedrivePolicy|FifoQueue|ContentBasedDeduplication|KmsMasterKeyId|KmsDataKeyReusePeriodSeconds] For details on the attributes refer to: AWS-SQS-CLI-AttributeNames
MessageAttribute Names	[ALL | list of attributes in CSV format] Message Attributes to be returned ‘ALL’ returns all attributes of the message Example: MessageAttributeNames = Author, Title will return only the attribute for Author and Title.
WaitTimeSeconds	The duration (in seconds) for which the call waits for a message to arrive in the queue before returning. If a message is available, the call returns sooner than WaitTimeSeconds . If no messages are available and the wait time expires, the call returns successfully with an empty list of messages.
Launch Task	Taskname of the Task to launch in Universal Controller. The task will be always launched with two Task Variables: • {Body}: contains the message body in json format • {Attributes}: contains the message attributes in json format. The Variable content can be used in further processing. e.g. If a Linux task is launched by the AWS SQS Task than an *echo {Attributes}* command executed by the Linux task will print out all message attributes in json format to STDOUT.
Delete Messages from queue after reading	[Yes | No] If “Yes” the received Messages is deleted from the queue after reading
Interval in seconds	Message Polling Interval in seconds
Run Mode	[Run Once | Run Forever] • Run once : if a message is found the task goes to success • Run Forever : each time a new message is found the configured task is launched Example: • Run once can be used in Workflows • Run Forever can be used as Standalone Task, where each message triggers a task launch.
	The maximum number of messages to return. Amazon SQS never returns more messages than this value (however, fewer messages might be returned). Valid values: 1 to 10. Default: 1.

MaxNumberOfMessages	
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Role Based Access (STS)	[NO | YES] If set to YES, the fields to set-up the IAM Role Based Access Connection are displayed: • Role Arn: Amazon Role, which is applied for the connection Example RoleArn: *arn:aws:iam::119322085622:role/SB-SQS-ReadOnly* STS: AWS Security Token Service is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources like SQS Queues.
Endpoint URL	Only used in case of a MinIO server; connection Endpoint URL for the MinIO storage system

Examples for AWS SQS Universal Tasks - Action: receive-message

Receive-message - Action


General
Task Name : AWS-SQS - Receive Message
Version : 7
Task Description :
Member of Business Services :
Resolve Name Immediately :
Time Zone Preference : -- System Default --
Hold on Start :
Virtual Resource Priority : 10
Hold Resources on Failure :
Agent Details
Cluster :
Agent : ${AGT_LINUX_PS4}
Agent Variable :
Credentials :
Credentials Variable :
Run with Highest Privileges :
Interact with Desktop :
AWS-SQS Details
Action : receive-message
AWS_SECRET_ACCESS_KEY : AWS_SECRET_ACCESS_KEY_NBU
AWS_ACCESS_KEY_ID : AWS_ACCESS_KEY_NBU
AWS_DEFAULT_REGION : eu-central-1
Useproxy : No
Loglevel : DEBUG
SQS Queue Name : orders2
WaitTimeSeconds : 10
MaxNumberOfMessages : 10
MessageAttributeNames : Author,Title
AttributeNames : All
Interval in seconds : 10
Delete Messages from Queue after reading : Yes
Role Based Access (STS) : No
Universal Controller URL : https://ps2.stonebranchdev.cloud
Run Mode : Run Forever
Universal Controller Credentials : CRED-REST-API-PS2
Launch Task : dump-message-content

Azure Blob Storage

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

Storing data in the cloud becomes an integral part of most modern IT landscapes. With Universal Automation Center you can securely automate your AWS, Azure, Google and MinIO File Transfers and integrate them into your existing scheduling flows.

As security is one of the key concerns, when moving to the cloud, the provided solution supports multi-level of security:

- Credentials for Azure Keys are stored in an encrypted form in the database
- Support for Azure Token based Shared Access Signatures (SAS)
- Communication to Azure is done via the HTTPS protocol
- A Proxy Server connection to Azure with basic authentication is supported

This Universal Task focuses on file transfer to, from and between an Azure Blob Storage.

A similar solution as for Azure Blob Storage is also available for AWS S3 and Google Cloud Storage.

Overview

The Universal Task for Azure Blob Storage allows to securely transfers files from, to and between Azure Blob Storage container and folders.

The Universal Task for Azure Blob Storage supports the following main features:

- The following file transfer commands are supported:
    - Upload a file(s) to an Azure Blob Storage container
    - Download of file(s) from an Azure Blob Storage container
    - Transfer files between Azure Blob Storage containers
    - List objects in an Azure Blob Storage container
    - Delete object(s) in an Azure Blob Storage container
    - List Azure Blob Storage container names
    - Create an Azure Blob Storage container
    - Monitor a Blob in a Storage container
- File Transfer can be triggered by a third-party application using the Universal Automation Center RESTfull Webservice API: REST API
- Universal Task for Azure Blob Storage can be integrated into any existing scheduling workflow in the same way as any standard Linux or Windows Task type.
- Security is ensured by using the HTTPS protocol with support for an optional Proxy Server.
- Support for Azure Token based Shared Access Signatures (SAS)
- No Universal Agent needs to be installed on the Azure Cloud – the communication goes via HTTPS

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 7.0.0.0 or later are required
- The Universal Agent needs to be installed with python option (--python yes)

Software Requirements for Universal Controller

- Universal Controller 7.0.0.0. or later is required

Software Requirements for the Application to be Scheduled

The Universal Task has been tested for the python Azure Storage SDK: *azure-storage-blob version: 12.7.1*

Key Features

The solution supports the following file transfer scenarios:

Feature	Description

Upload a file(s) to a container (Copy or Move)	• One or more files can be uploaded to a container.
	• Move or copy are supported for the upload
	• Unix filename pattern matching support; for example, wild card support “*” to upload multiple files
	• One or more files can be Uploaded to a folder using a prefix
	• It can be decided via an the *Upload Write Options* to:
	  • Overwrite an existing object (*Replace existing Object*)
	  • Cancel the operations in case an object with a similar name exists (*Do not overwrite existing Object*)
	  • Add a timestamp to the uploaded Object (*Timestamp*)

Download of file(s) from a container (Copy or Move)	• One or more files should be downloaded from a bucket.
	• Move or copy must be supported for the download
	• Unix filename pattern matching support; for example, wild card support “*” to download multiple files
	• Download to a specific folder is supported
	• It can be decided via an the *Download Write Options* to:
	  • Overwrite an existing file (*Replace existing File*)
	  • Cancel the operations in case a File with a similar name exists (*Do not overwrite existing File*)
	  • Add a timestamp to the uploaded file (*Timestamp*)
	  • Perform the default Windows behaviour for copying files (*Default Windows behaviour*)
	  • If a file with a similar name exists, the file names that are similar will be edited so that the files you copied have a number appended at the end of them. For example, if you’re copying a file named image.png to a folder that already has a file named image.png in it, the copied file will be named image (1).png.

Copy object to a container	• An object can be copied from one container to another. • Folders are support Note: wild cards are not supported for this method.
List objects in a container	• Show all Objects in a container • Show all Objects in a folder • Unix filename pattern matching support e.g. wild card support “*” to narrow down the objects to display
Delete object(s) in a container	• Delete one or multiple objects in a container • Delete one or multiple objects in folder in a container • Unix filename pattern matching support e.g. wild card support “*” to narrow down the objects to be deleted
List buckets	List all container in an Azure account
Create a container	Create a new container
Monitor Blob	Monitor a Blob in a Storage container

Proxy Server Connection	A proxy server connection with or without basic authentication can be configured
Integration into 3rd Party Applications	An Azure Blob Storage file transfer can be triggered via the Universal Automation Center RESTfull Webservice API within an application.
Self-Service through Web-Client	The Azure Blob Storage Task can be fully configured, monitored and updated via the Universal Controller Web-GUI

Import Azure Blob Storage Universal Template

The Universal Template is provided as a Zip-file. This Zip-file can be imported from any local folder using the Universal Controller UI.

To load the Universal Template Zip-file perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy the Universal Template Zip-file to a local directory directory.
3. In the Universal Controller UI, select Administration > Universal Templates > Import Template.
4. Select the Universal Template Zip-file to import.

When the Universal Template Zip-file has been imported successfully, the Universal Template will appear on the list.

Configure Azure Blob Storage Universal Tasks

For the new Universal Task type, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for Azure Blob Storage Universal Task - Action

The Azure Blob Storage Task provides multiple different file transfer actions. For each action the specific fields are described.

Azure Blob Storage Container is abbreviated to **Container** in the following.

Create Container - Action

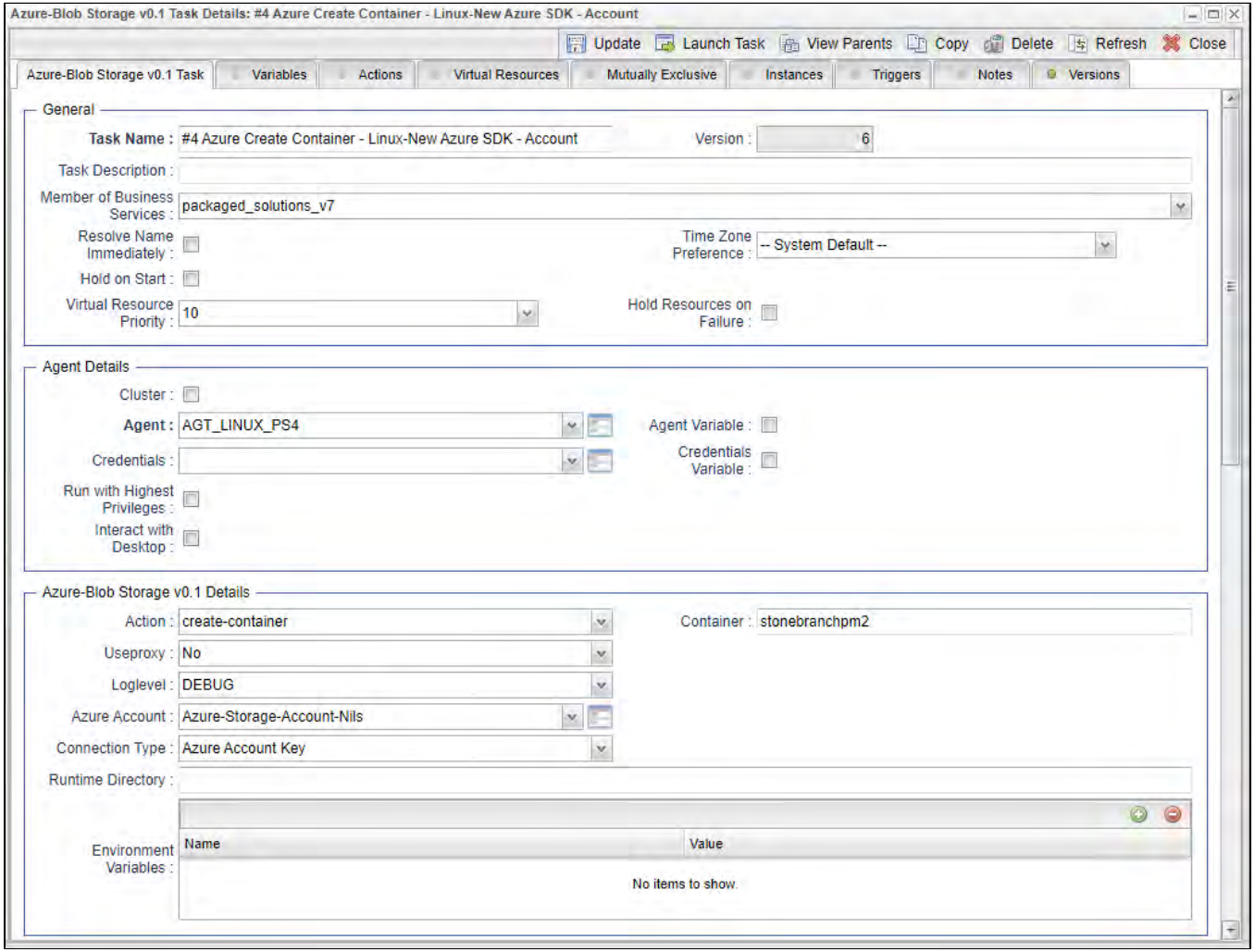
The Action Creates a new Container.

Field	Description
Action	create-container action
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Connection Type	[Azure Account Key | SAS Token]

	If set to 'Azure Account Key', the credential field 'Azure Account' will appear for configuration. If set to 'SAS Token', the credential field 'SAS Token' will appear for configuration.
Container	Name of the Container to create

Example for Azure Blob Storage Universal Tasks - Create Container

The following Task creates the Container: stonebranchpm2



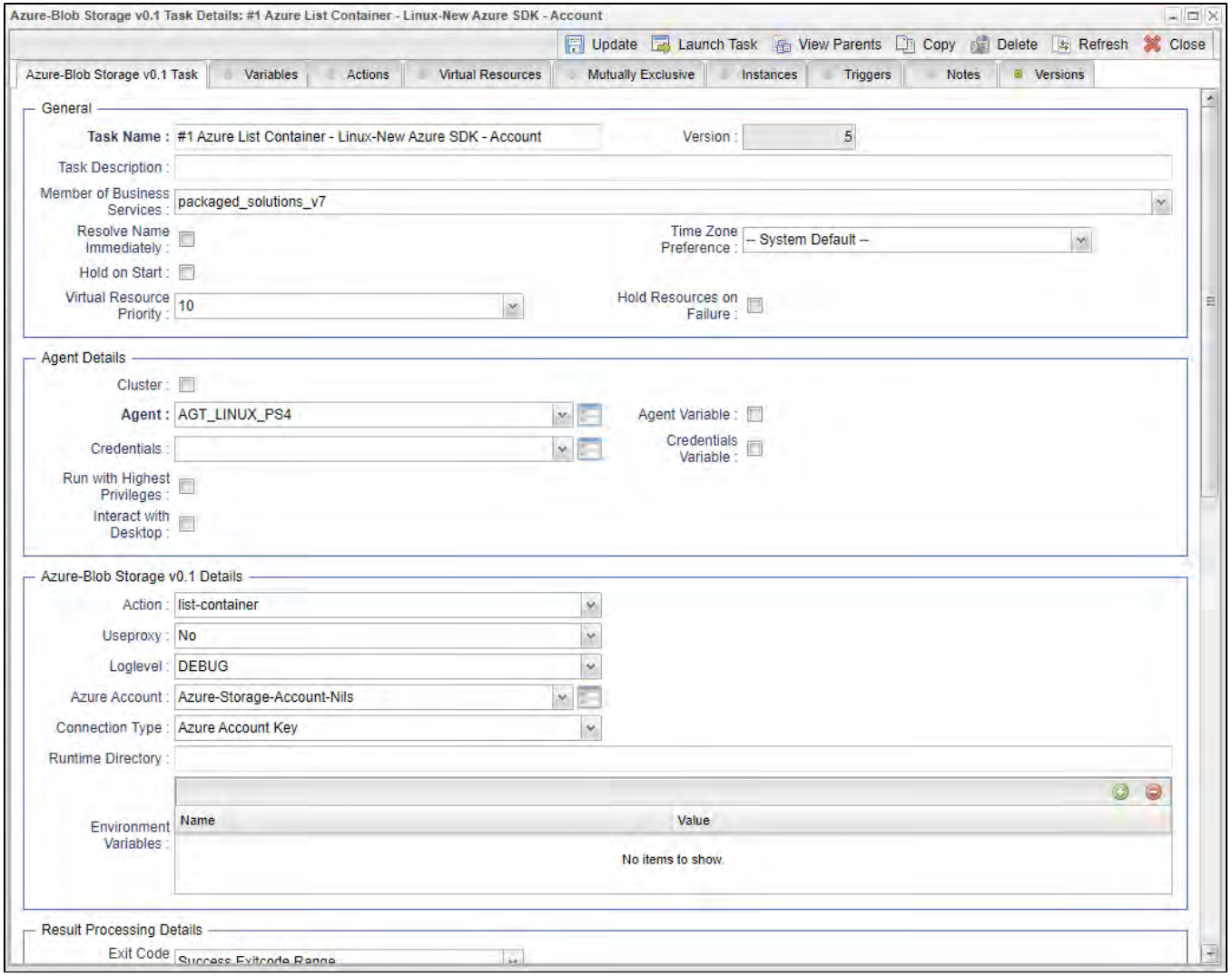
List Container - Action

The Action list all container of an Azure Account.

Field	Description
Action	list-container action
Connection Type	[Azure Account Key | SAS Token] If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration. If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example for Azure Blob Storage Universal Tasks - List Containers

The following Task list all container of the selected Azure Account.



Upload File - Action

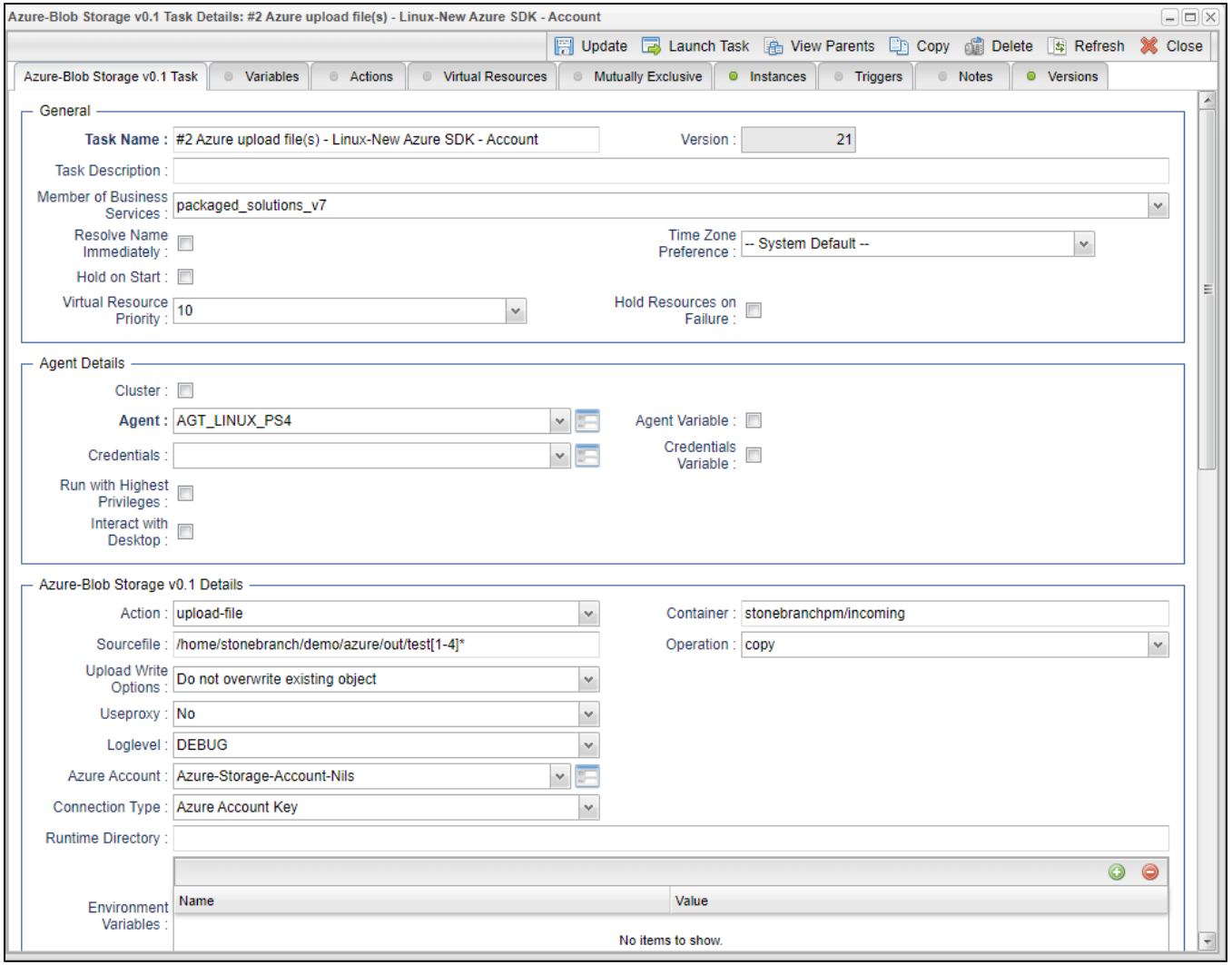
The Action is used to upload a single or multiple files from a Windows or Linux server to an container or a folder in a container.

Field	Description
Action	Upload-file action
Connection Type	[Azure Account Key | SAS Token]

	If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration. If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Sourcefile	Source file(s) to be uploaded to a bucket or specific folder in a container. Unix filename pattern are supported to upload a selection of files. Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything
Operation	[copy | move] In case of a “move” the source file(s) is/are deleted after the upload to the container.
Prefix	A folder in a container is called prefix. In the field Prefix only the name of the folder needs to be provided no “/” after the folder name.
Upload Write Options	Upload Write Options [Do not overwrite existing Object | Timestamp] • Do not overwrite existing Object: Cancel the operations in case an object with a similar name exists • Timestamp: Add a timestamp to the uploaded Object

Example for Azure Blob Storage Universal Tasks - Upload File

The following Task uploads from the Linux directory the files: */home/stonebranch/demo/azure/out/test[1-4]** to the the container *stonebranchpm*, folder *incoming*.



List Objects - Action

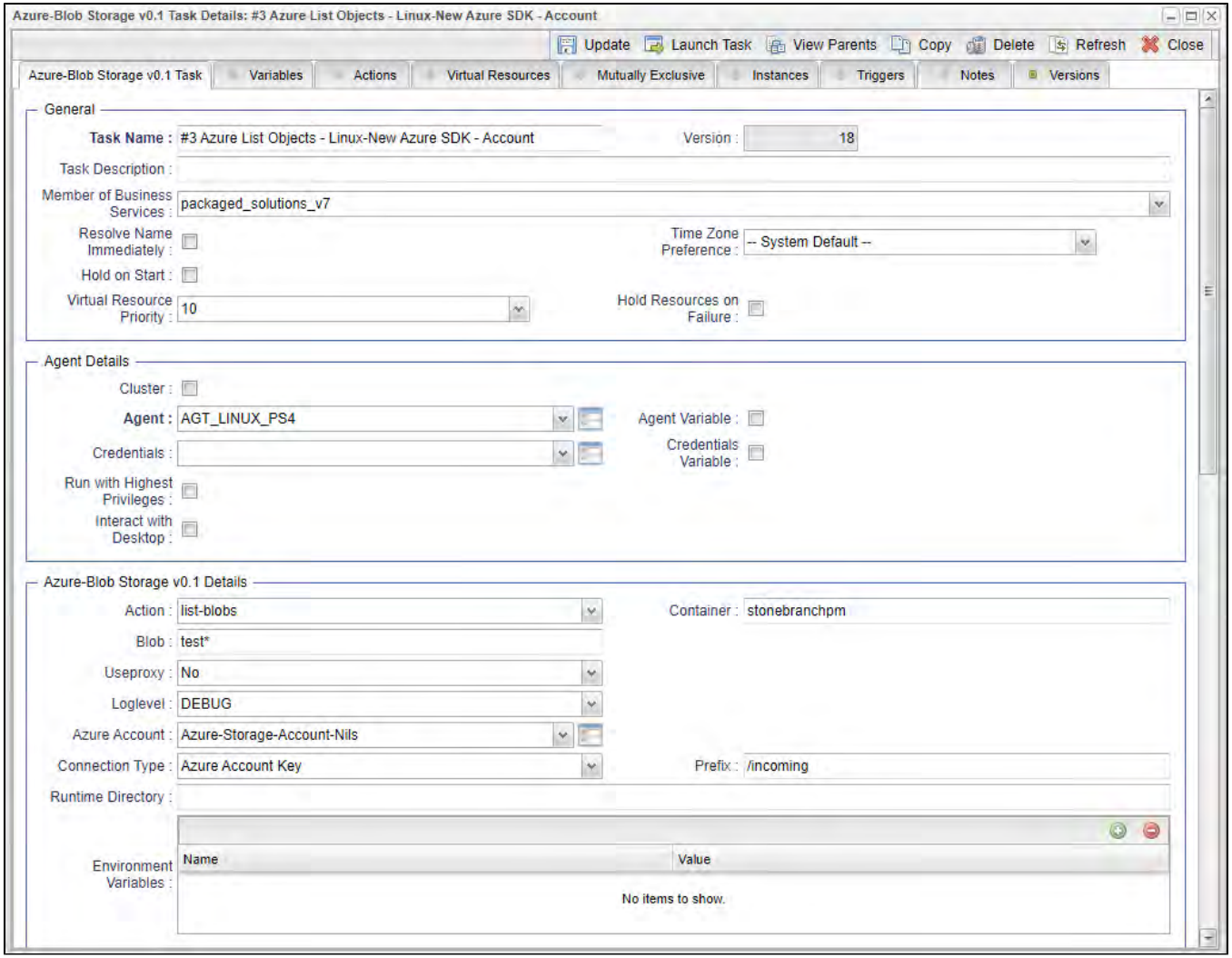
The Action is used to display objects in a Azure Container or a specific folder in an Azure Container (prefix).

Field	Description
Action	List-objects action
Connection Type	[Azure Account Key | SAS Token] If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration.

	If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Container	Container in which the objects should be listed
Prefix	A folder in a container is called prefix. In the field Prefix only the name of the folder needs to be provided no “/” after the folder name. If a prefix is provided only objects in the folder with the prefix name are listed in the output.
Blob	Objects matching the given Blob are listed. Unix filename pattern are supported to list only a selection of files: Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything Example: Blob = test* : matches everything starting with test Blob = test[1-2].txt : matches test1.txt, test2.txt Blob = test[!1].txt: does not match test1.txt Blob = test?.txt: matches test1.txt, test2.txt etc.
Show Details	Show details like creation timestamp in the output

Example for Azure Blob Storage Universal Tasks - List Objects

The following Task list all objects matching the criteria *test** in the container *stonebranchpm* in the folder *incoming*.



Download File - Action

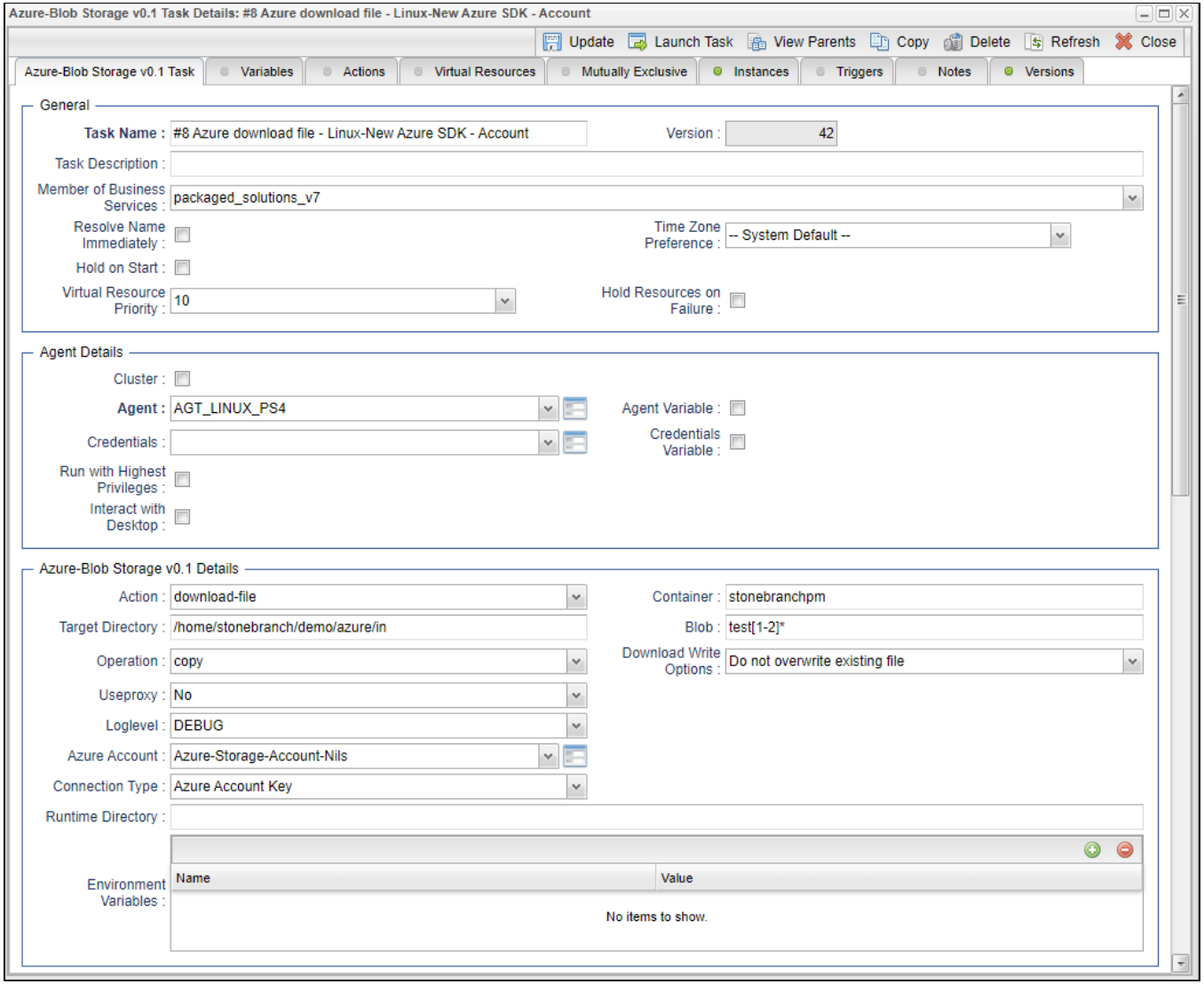
This Action downloads one or multiple files from an Azure Container to a Linux or Windows folder.

Field	Description
Action	Download-file action
Connection Type	[Azure Account Key | SAS Token] If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration.

	If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[*NO* | *YES*] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Target Directory	Linux or Windows Target Directory For example, C:\tmp\ or /home/ubuntu/download Unix filename pattern are supported to download a selection of files. Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything
Operation	[copy | move] In case of a “move” the objects are deleted after they have been download from the container.
Download Write Options	Download Write Options: [Replace existing File | Do not overwrite existing File | Timestamp | Default Windows behaviour] • Replace existing File : overwrite an existing file • Do not overwrite existing File: cancel the operations in case a File with a similar name exists • Timestamp: add a timestamp to the uploaded file • Default Windows behaviour: perform the default Windows behaviour for copying files. If a file with a similar name exists, the file names that are similar will be edited so that the files you copied have a number appended at the end of them. For example, if you are copying a file named image.png to a folder that already has a file named image.png in it, the copied file will be named image (1).png.

Example for Azure Blob Storage Universal Tasks - Download File

The following Task downloads from the container stonebranchpm all blobs matching the criteria *test[1-2]** to the Linux directory: */home/stonebranch/demo/azure/in*



Delete Objects - Action

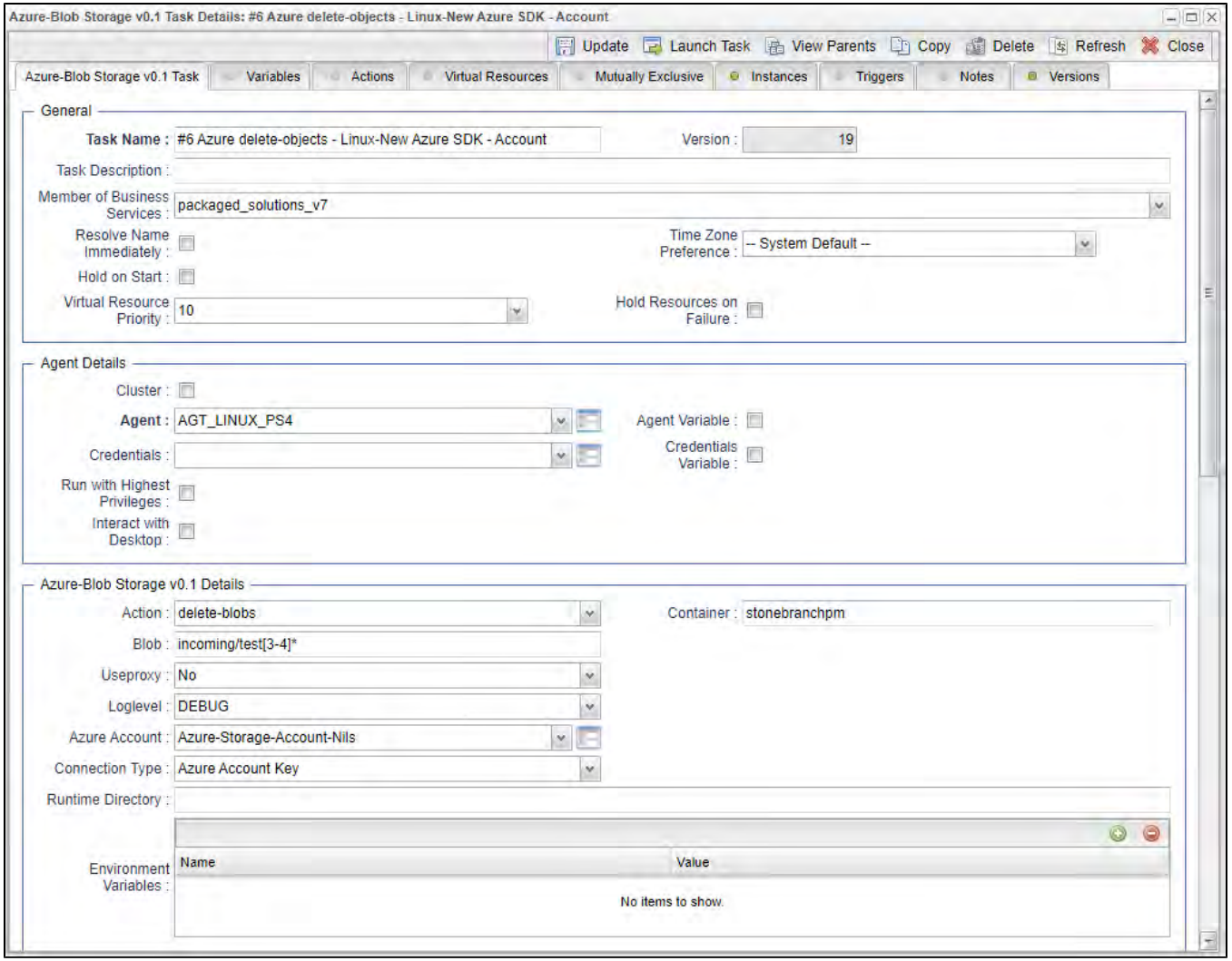
The Actions is used to delete an object in an Azure Container or folder

Field	Description
Action	Delete-objects action
Connection Type	[Azure Account Key | SAS Token]

	If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration. If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Container	Container in which the Blobs should be deleted.
Blob	Blobs to be deleted in the given container Note: Due to security reasons wild card is only support, if at least one character is provided e.g. t* would delete all files starting with at “t”.

Example for Azure Blob Storage Universal Task - Delete Objects

The following Task deletes in the container *stonbranchpm*, folder *incoming* all Blobs matching the criteria test[3-4]*.



Copy Object to Container - Action

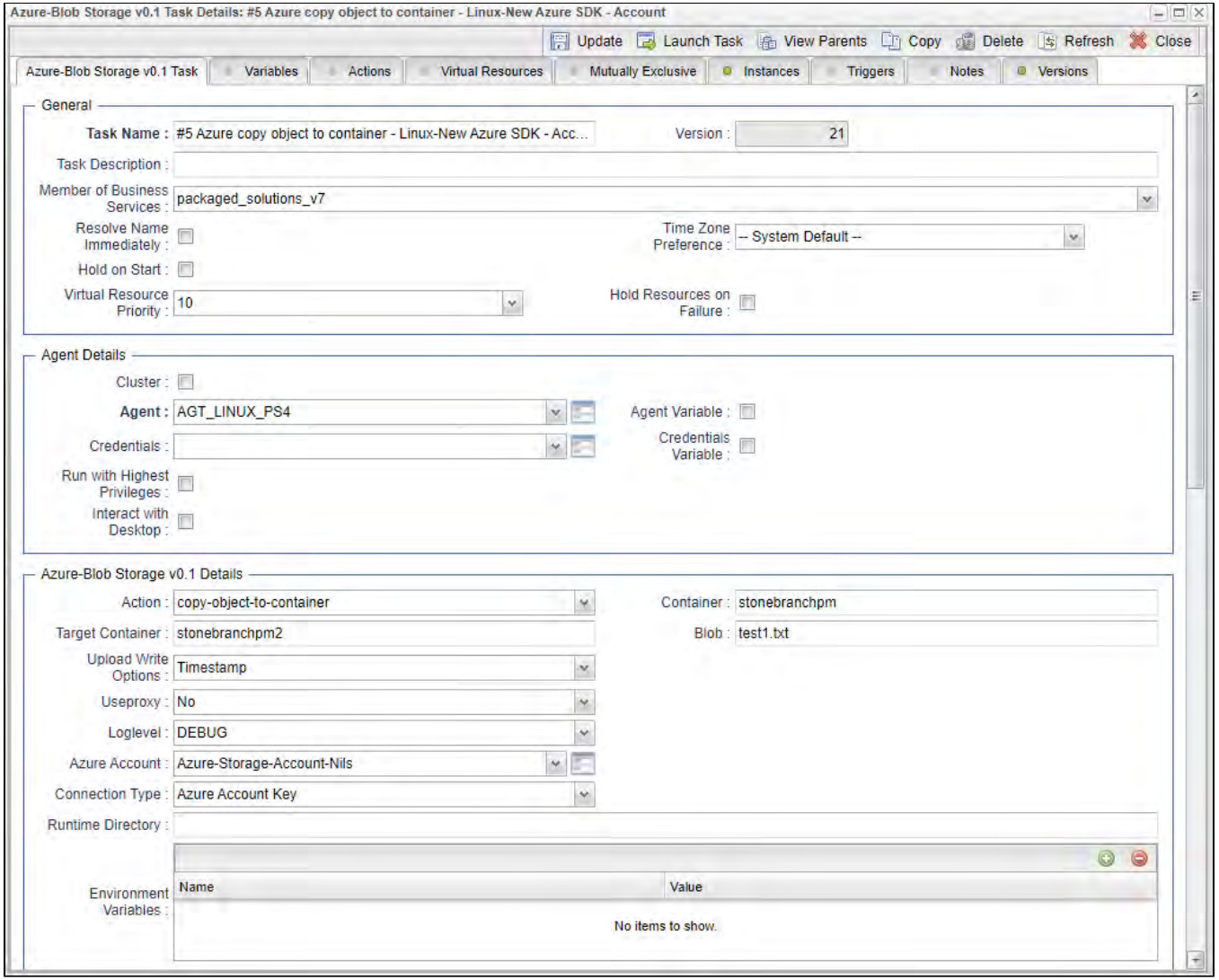
This Action is used copy an objects from one Azure Container to another Azure Container.

Field	Description
Action	Copy-object-to-bucket action
Connection Type	[Azure Account Key | SAS Token] If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration.

	If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Container	Source container
Target Container	Target container, where the object(s) will be copied to
Blob	File (Blob) to be copied from the source container to target container.
Upload Write Options	Upload Write Options [Do not overwrite existing Object | Timestamp] • Do not overwrite existing Object: Cancel the operations in case an object with a similar name exists • Timestamp: Add a timestamp to the uploaded Object

Example for Azure Blob Storage Universal Tasks - Copy Object to Container

The following Task copies the Blob: test1.txt from the source container: *stonebranchpm* to the target container: *stonebranchpm2*



Delete Container - Action

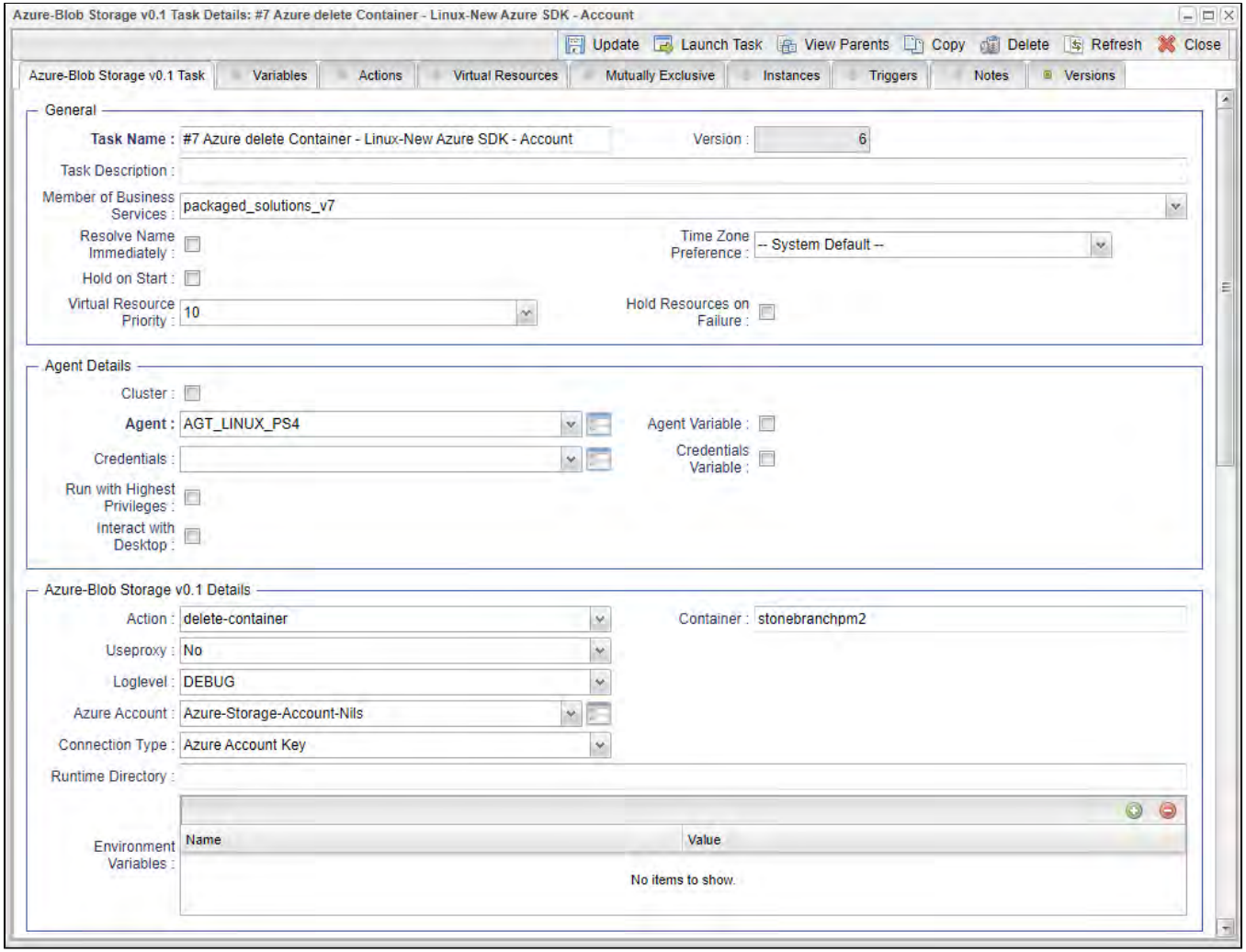
This action is used to delete an Azure Container

Field	Description
Action	Delete-container action
Connection Type	[Azure Account Key | SAS Token]

	If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration. If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Container	Name of the Container to be deleted Note: The container is delete including all objects.

Example for Azure Blob Storage Universal Tasks - Delete Container

The following Task delete the container stonebranchpm2



Upload File - Action

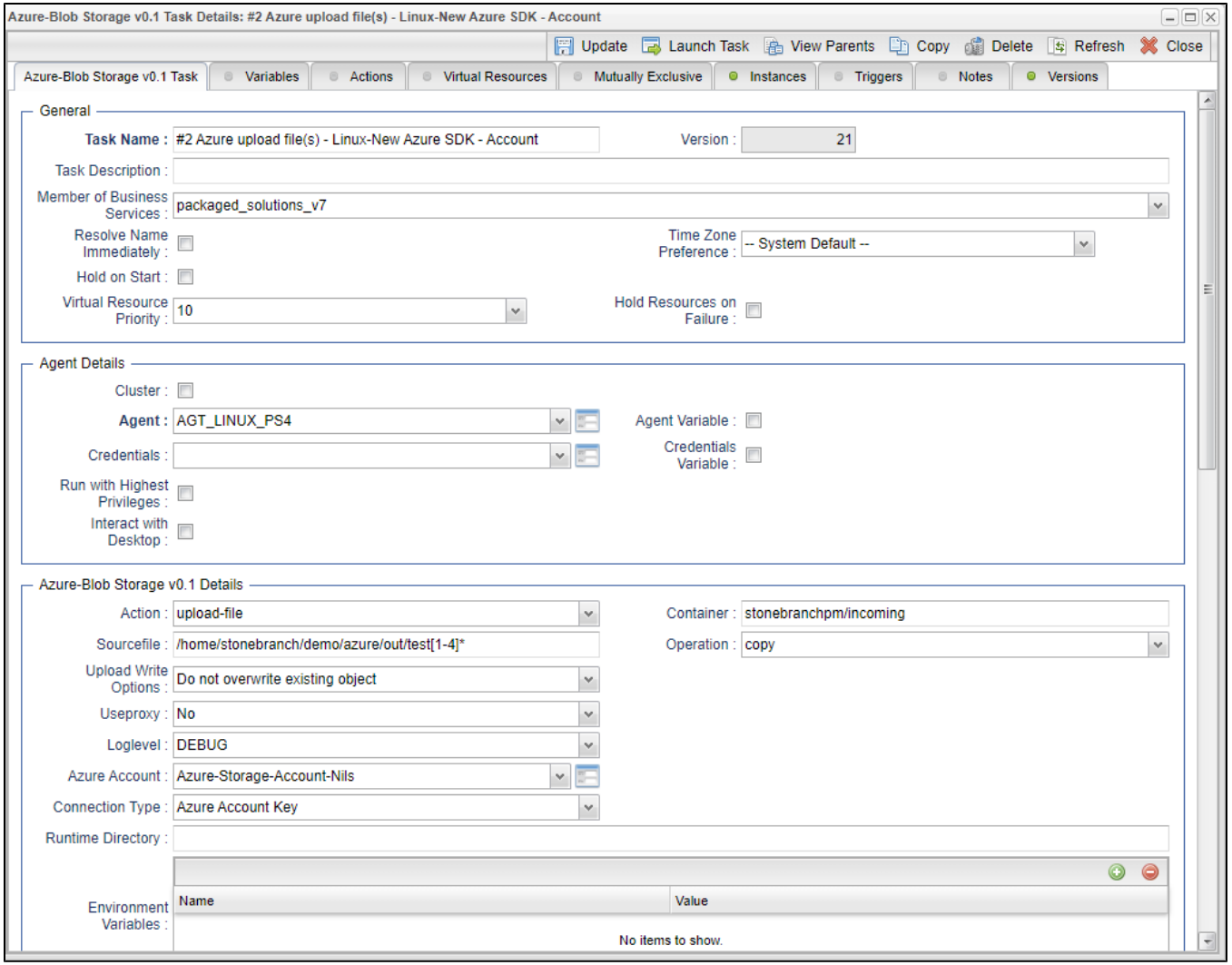
The Action is used to upload a single or multiple files from a Windows or Linux server to an container or a folder in a container.

Field	Description
Action	Upload-file action
Connection Type	[Azure Account Key | SAS Token] If set to ‘Azure Account Key’, the credential field ‘Azure Account’ will appear for configuration.

	If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Sourcefile	Source file(s) to be uploaded to a bucket or specific folder in a container. Unix filename pattern are supported to upload a selection of files. Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything
Operation	[copy | move] In case of a “move” the source file(s) is/are deleted after the upload to the container.
Prefix	A folder in a container is called prefix. In the field Prefix only the name of the folder needs to be provided no “/” after the folder name.
Upload Write Options	Upload Write Options [Do not overwrite existing Object | Timestamp] • Do not overwrite existing Object: Cancel the operations in case an object with a similar name exists • Timestamp: Add a timestamp to the uploaded Object

Example for Azure Blob Storage Universal Tasks - Upload File

The following Task uploads from the Linux directory the files: */home/stonebranch/demo/azure/out/test[1-4]** to the the container *stonebranchpm*, folder *incoming*.



Monitor Blob - Action

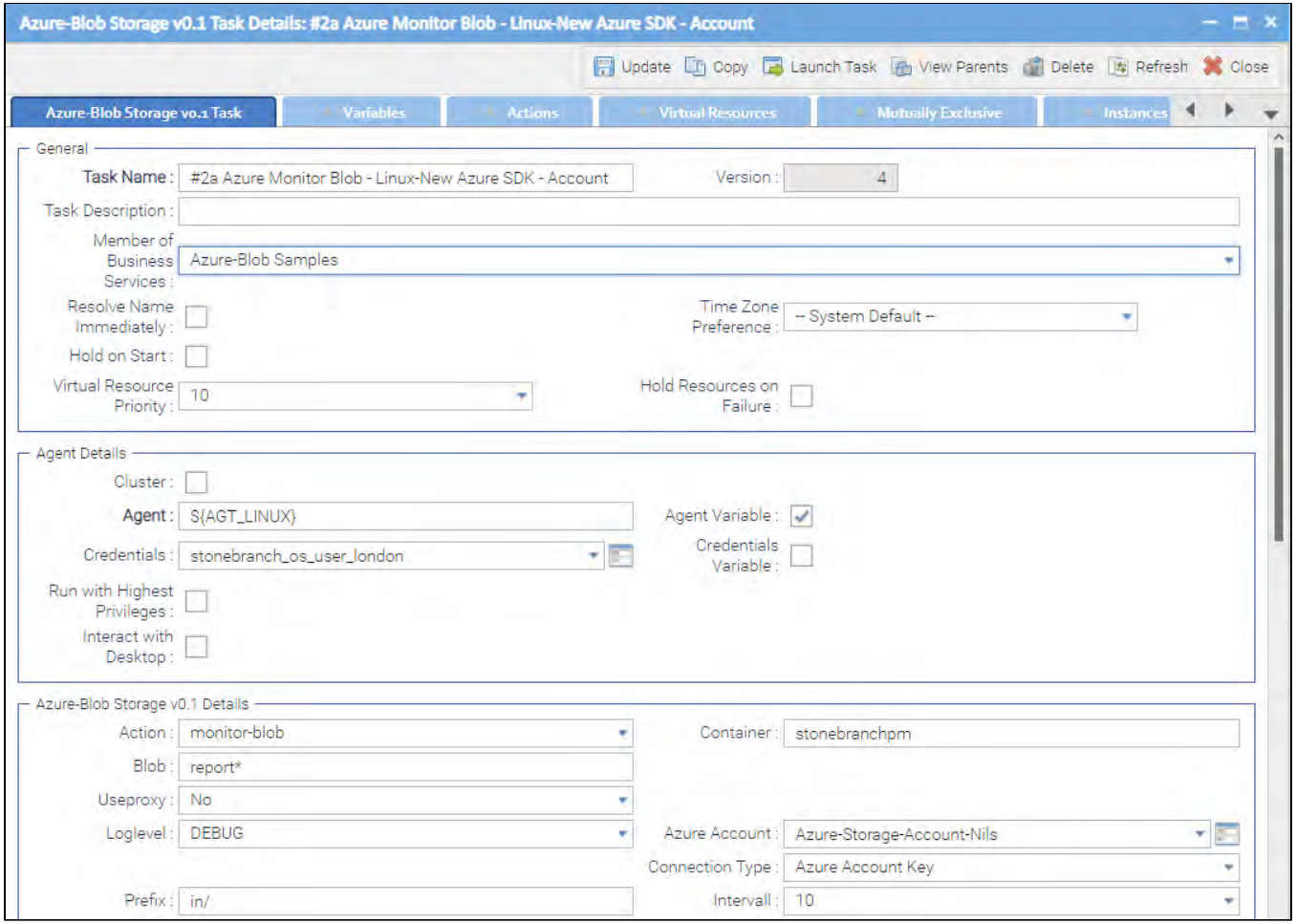
The Action is used to monitor a Blob in an Azure Container.

Field	Description
Action	Monitor-Blob action
Connection Type	[Azure Account Key | SAS Token] If set to 'Azure Account Key', the credential field 'Azure Account' will appear for configuration.

	If set to ‘SAS Token’, the credential field ‘SAS Token’ will appear for configuration.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
Container	Container in which the objects should be monitored
Prefix	A folder in a container is called prefix. In the field Prefix only the name of the folder needs to be provided no “/” after the folder name. If a prefix is provided only objects in the folder with the prefix name are listed in the output.
Blob	Blob to monitor. The task goes to status success, if the Blob is found. Unix filename pattern are supported to list only a selection of files: Supported wildcards are: • ? matches any single character • [seq] matches any character in seq • [!seq] matches any character not in seq • ”*” matches everything Example: Blob = test* : matches everything starting with test Blob = test[1-2].txt : matches test1.txt, test2.txt Blob = test[!1].txt: does not match test1.txt Blob = test?.txt: matches test1.txt, test2.txt etc.

Example for Azure Blob Storage Universal Tasks - Monitor Blob

The following Task monitors in the Container: *stonebranchpm* all Blob’s starting with the name *report*. If a Blob starting with the name *report* is found the task goes to task status *success*.


Azure-Blob Storage v0.1 Task Details: #2a Azure Monitor Blob - Linux-New Azure SDK - Account
Update
Copy
Launch Task
View Parents
Delete
Refresh
Close
Azure-Blob Storage v0.1 Task
Variables
Actions
Virtual Resources
Mutually Exclusive
Instances
General
Task Name :
#2a Azure Monitor Blob - Linux-New Azure SDK - Account
Version :
4
Task Description :
Member of Business Services :
Azure-Blob Samples
Resolve Name Immediately :
Time Zone Preference :
-- System Default --
Hold on Start :
Virtual Resource Priority :
10
Hold Resources on Failure :
Agent Details
Cluster :
Agent :
${AGT_LINUX}
Agent Variable :
Credentials :
stonebranch_os_user_london
Credentials Variable :
Run with Highest Privileges :
Interact with Desktop :
Azure-Blob Storage v0.1 Details
Action :
monitor-blob
Container :
stonebranchpm
Blob :
report*
Useproxy :
No
Loglevel :
DEBUG
Azure Account :
Azure-Storage-Account-Nils
Connection Type :
Azure Account Key
Prefix :
in/
Intervall :
10

Azure Data Factory Integration

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Integration allows Stonebranch users to schedule, trigger, and monitor the Azure Data Factory pipeline process directly from Universal Controller.

Overview

- This integration uses Python modules azure-mgmt-resource and azure-mgmt-datafactory to make REST-API calls to Azure Data Factory.
- This integration will use the Azure Tenant id , Subscription id , client id ,client secret, Resource group, and location for authenticating the REST-API calls to Azure Data Factory.
- User can perform the following Azure Data Factory operations:
    - Run a Pipeline.
    - Get information on a Pipeline.

    - List all Pipelines.
    - Cancel Pipeline run.
    - List factory by resource group.
- Also, with respect to Azure Data Factory triggers, users can perform the following operations from UAC:
    - Start Trigger.
    - Stop Trigger.
    - List Trigger by Factory.
- UAC also can restart a failed pipeline either from the failed step or from any activity name in the failed pipeline.

Software Requirements

This integration requires a Universal Agent and a Python runtime to execute the Universal Task against an Azure Data Factory.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.4 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - azure-mgmt-resource
    - azure-mgmt-datafactory

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.6 and later with Python options installed.
- Universal Agent for Linux Version 6.6 and later with Python options installed.

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later.

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with the Azure Data Factory Version 2.

Technical Considerations

- This integration uses the Python modules Azure resource Management Module and Azure Data Factory management to make REST-API calls with Azure Data Factory.
- Use Azure App Registration services to create an app, client ID, and client and associate it with Data Factory to provide appropriate roles (for example, Contributor).

Key Features

Feature	Description
Run a Pipeline	Execute a pipeline defined in Azure Data Factory. Schedule, trigger, and monitor the execution of a pipeline.

Get Information on a Pipeline	Get a information about the pipeline name .
List all Pipelines	List all pipelines that belongs to a Data Factory.
Cancel a Pipeline Run	Abort a Pipeline execution from UAC by providing the run ID for the Pipeline execution.
List Factory by resource group	List the available factory name that belongs to a resource group.
Start a Trigger	Start a trigger that will manage pipeline execution schedule in the Azure Data Factory.
Stop a Trigger	Stop a trigger that is in the Azure Data Factory.
List Trigger by Factory	List all the triggers that belongs to the Data Factory.

Import Azure Data Factory Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Azure Data Factory Integration Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

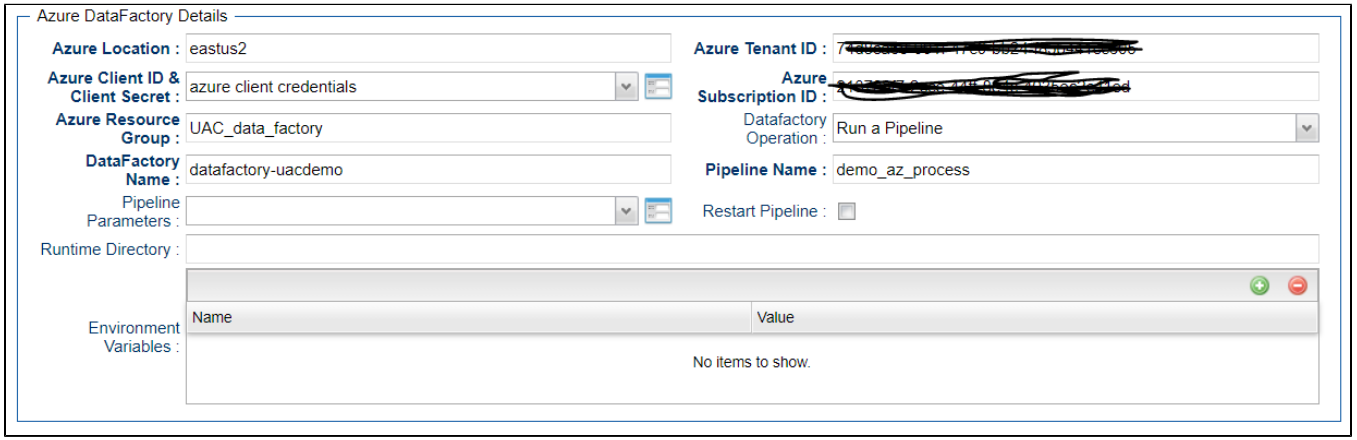
Field Descriptions for Azure Data Factory Integration Universal Task

Field	Description
Azure Location	Provide the Azure location.
Tenant ID	Provide the Azure Tenant ID or Directory ID.
Azure Client ID and Client Secret	Provide the Azure client ID and client secret (Client ID as user name and client secret as the password) - This comes from your Azure App registration services.
Subscription	Azure Subscription ID.
Azure Resource Group	Azure Resource group responsible for the Data Factory.
Data factory Operation	Select the required Azure Data Factory function for the task.
Data Factory Name	Provide the Azure Data Factory Name.
Pipeline Name	Pipeline name that needs to be executed in Azure Data factory.
Pipeline Parameters	Provide the pipeline run time parameters that needs to be passed for the pipeline execution.
Run ID	Run id for the Azure Data Factory Pipeline execution - Required in case of a restart.

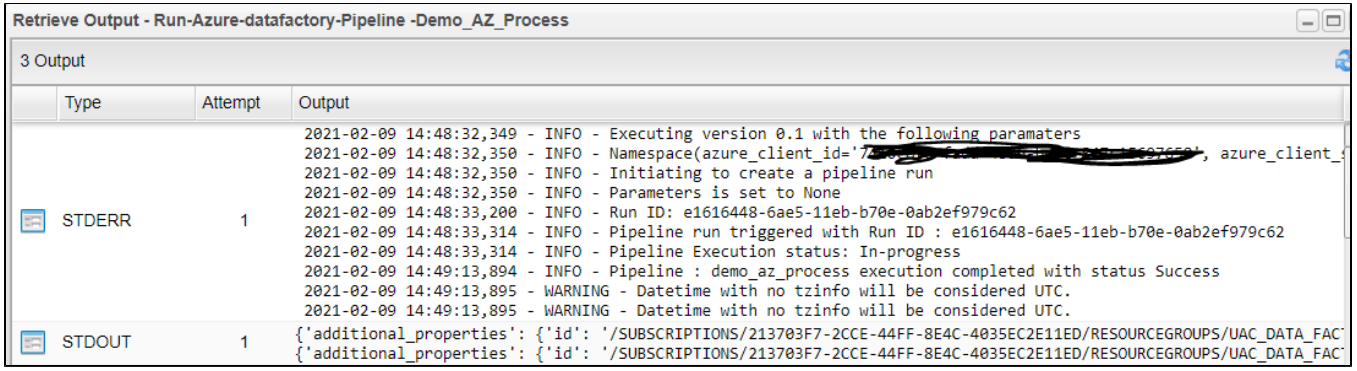
Restart Pipeline	Check if the pipeline execution needs to be restarted for a failure.
Start From Failure	Pipeline needs to be restarted from a Failure step.
Start Activity Name	If Start from Failure is not checked, specify if a failed pipeline needs to be restarted from a specific activity.
Run ID for Restart	Provide the Pipeline run ID for the restart.
Trigger Name	Trigger Name in the Azure Data Factory.
Polling	True for ARMPolling, False for no polling, or a polling object for personal polling strategy.

Examples for Azure Data Factory Integration Universal Tasks

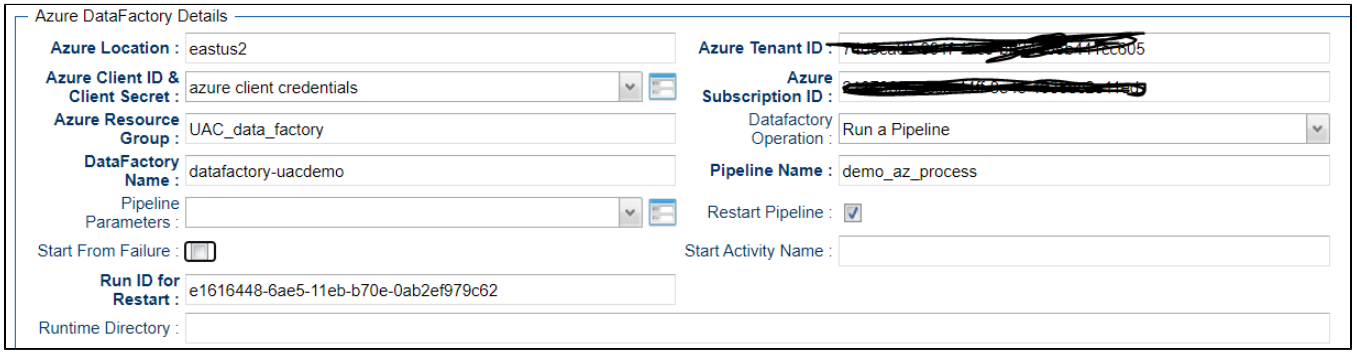
Run a Pipeline



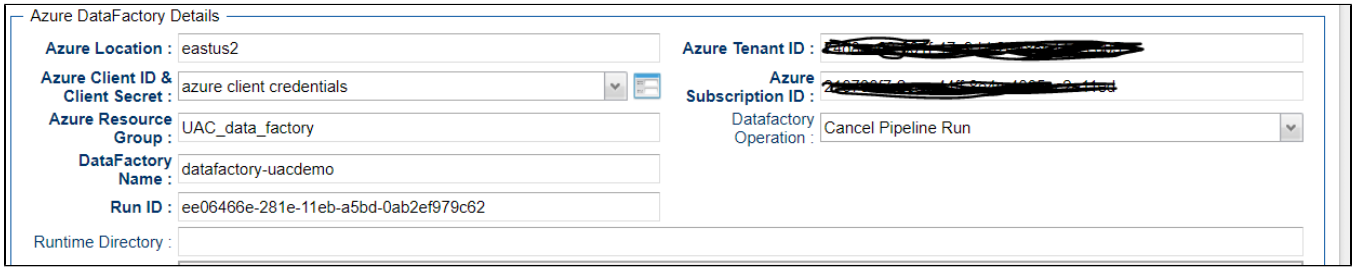
Pipeline Execution Logs



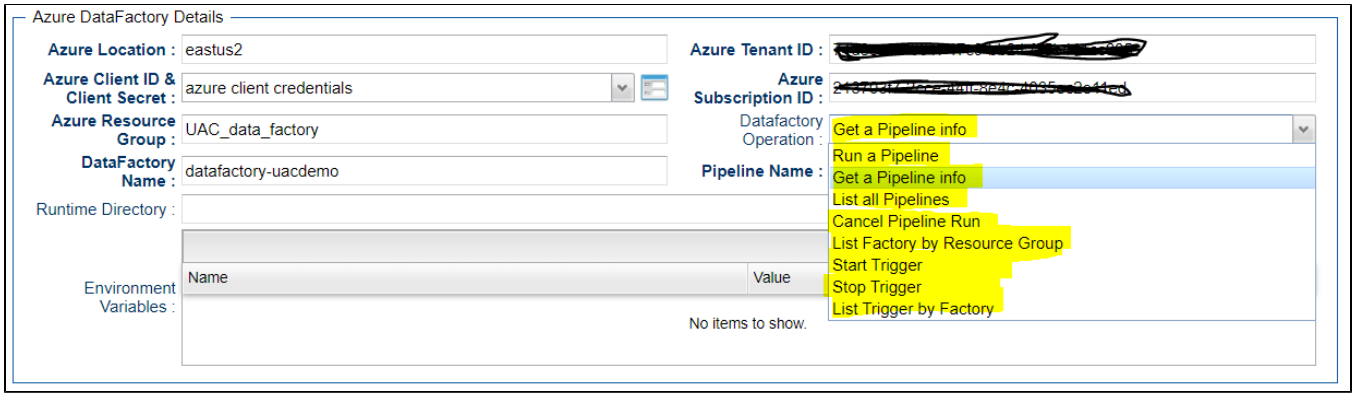
Restart a Failed Pipeline Execution



Cancel a Pipeline Run



Other Azure Data Factory Operation from UAC



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC69/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Azure Logic Apps

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task can trigger and monitor the execution of Azure Logic workflows and retrieve the execution of Azure Logic workflow output. The Universal Controller integrates with Logic Apps through Rest-APIs securely through the Azure Oauth2.0 authentication mechanism.

Overview

- This task passes dynamic input parameters (JSON format) to each Azure Logic app workflow.
- The task triggers a workflow, monitors it until the process is completed, and then delivers the results to Universal Controller.
- Customers can manage and control Logic App workflow execution from Universal Controller, with the capability to employ other dependencies like time triggers or event-based jobs/workflows.
- This task offers ITSM integration capability, enabling the auto-creation of incidents in case of Logic Apps workflow execution failure.

Software Requirements

Software Requirements for Universal Template and Universal Task

Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.

- Python modules required
    - requests
- UAC Global Variables should be set in order to run the UT properly.
    - Global variable: Azure_access_token to be set with Azure access token and this will be used to make the REST-API calls authenticated from universal controller to Azure logic apps
    - Azure_Access_token can be populated with new access token by setting up a web services task that can run periodically; for example, every hour.

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.5 and later with python options installed
- Universal Agent for Linux Version 6.5 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.4.7.0 and later

Software Requirements for the Application to be Scheduled

This Universal Task can schedule and execute Azure Logic Apps workflows with API Version 2016-06-01.

Azure Logic apps

This Universal Task can trigger and monitor the execution of Azure Logic workflows and retrieve the execution of Azure Logic workflow output. The Universal Controller integrates with Logic Apps through Rest-APIs securely through the Azure Oauth2.0 authentication mechanism.

Key Features

Feature	Description
Workflow Trigger Run	Trigger a logic apps workflow in Azure, monitor the execution, and pull the output results to Universal Controller.

Import Azure Logic apps Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.

4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Azure Logic apps Universal Task

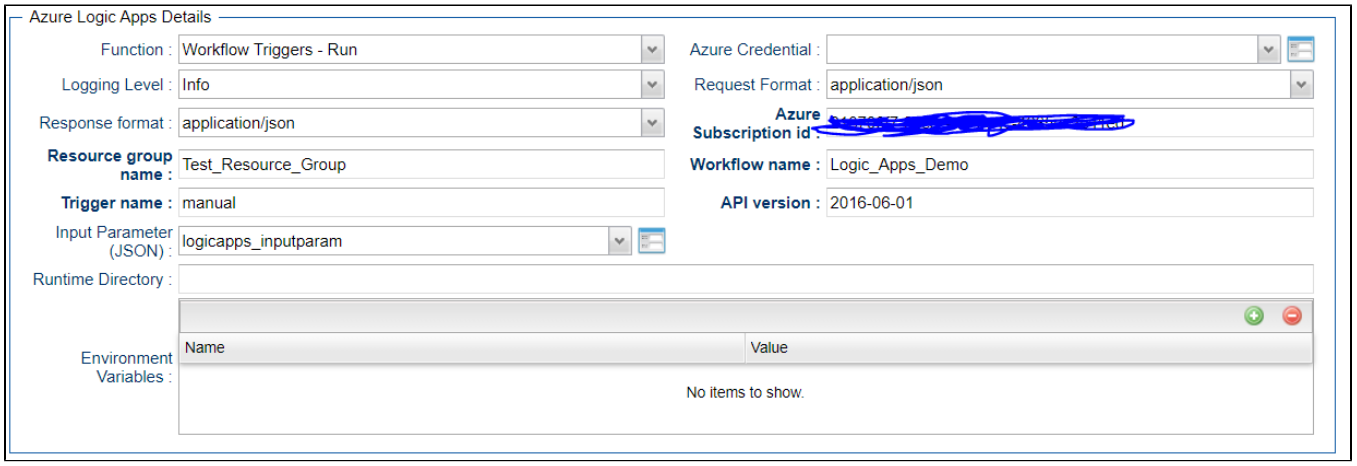
For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Azure Logic apps Universal Task

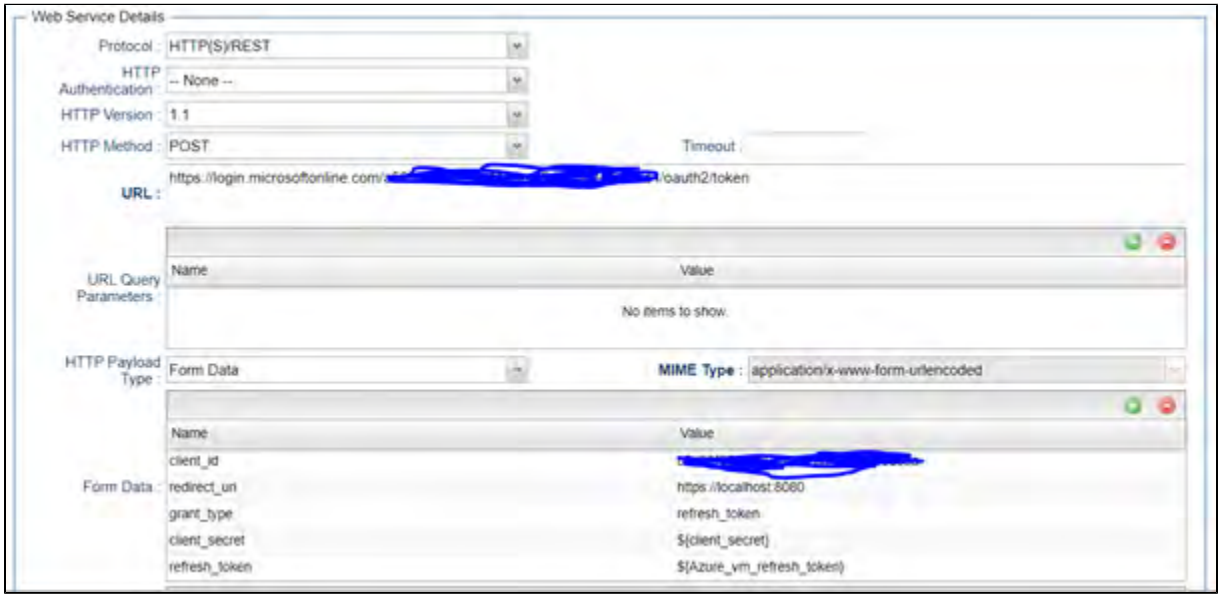
Field	Description
Function	Select the function for logic apps.
subscription Id	Azure Subscription ID.
Api-version	Provide the api version for Azure API.
Trigger Name	Should be set to “Manual“.
Workflow Name	Name of the Azure Logic Apps workflow.
Input Parameter	Run time input parameters(JSON) for the logic apps workflow.

Examples for Azure Logic apps Universal Tasks

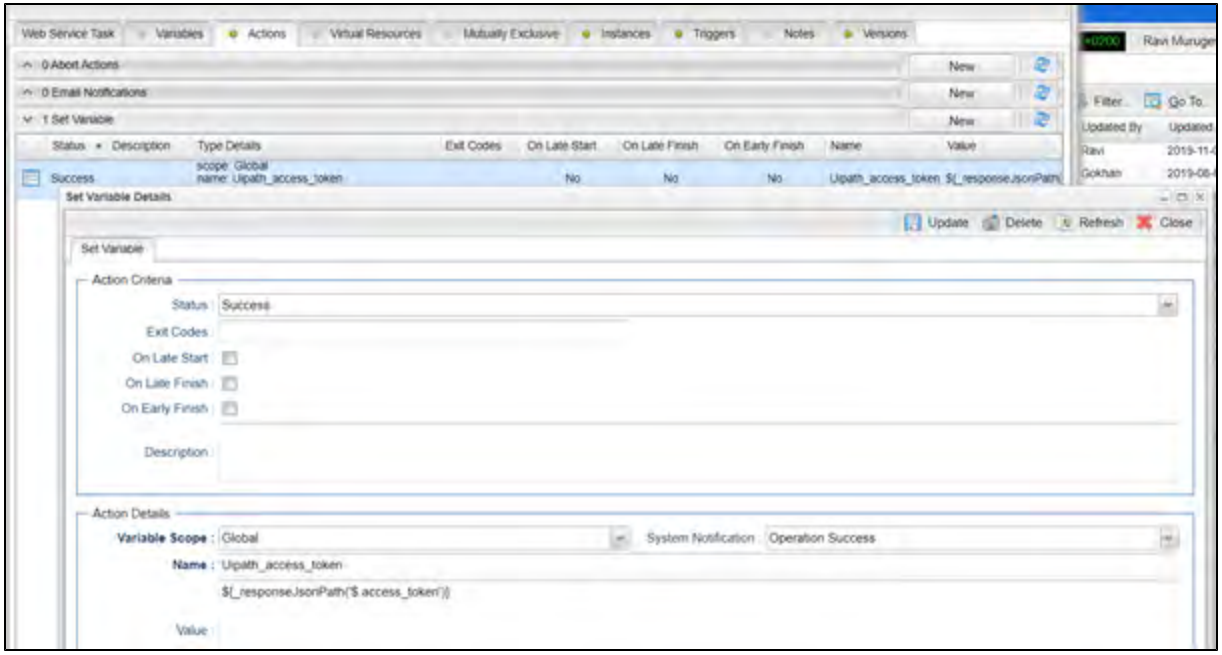
Trigger a Azure Logic apps Workflow



Oauth2.0 Webservices Task - Refresh Access Token



Webservices Task Actions to Assign New Access Token to Global Variable



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Azure Virtual Machines Start-Stop-Terminate-Instance

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

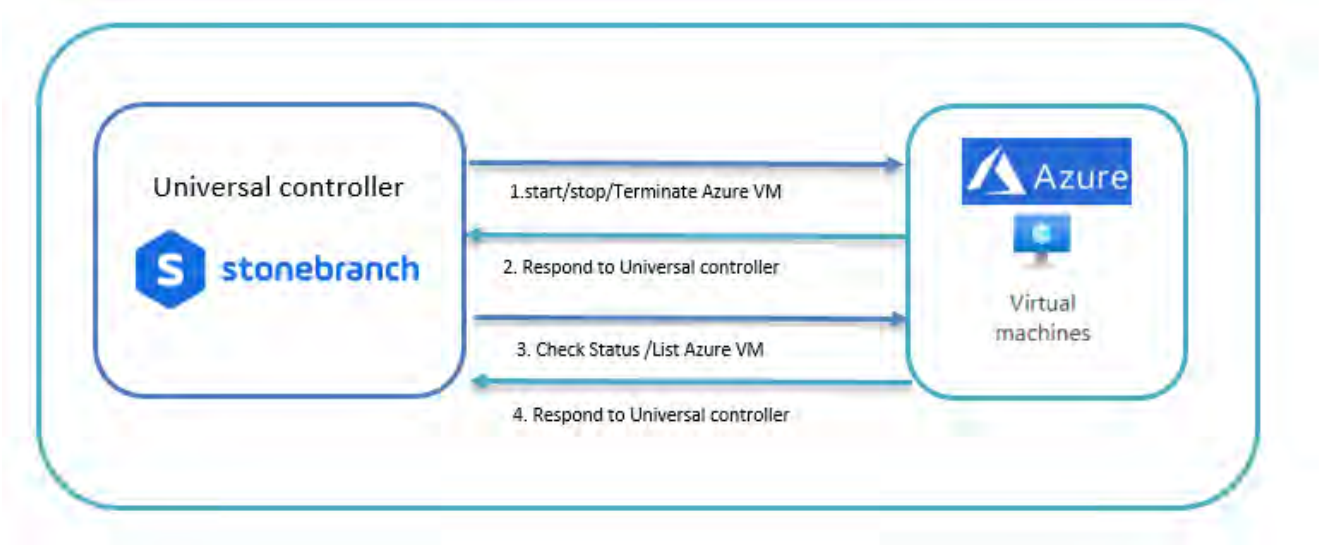
Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task enables users to utilize Azure Virtual Machine (VM) name, resource group, subscription ID, and access token as inputs for the start, stop, terminate, list, and check status of Azure VMs.

Overview

- This task uses python requests module to interact with the Azure cloud platform.
- It expands user ability to start/stop/terminate/check/list Azure VMs that belong to a subscription and resource group.
- In Universal Controller (UC), this task reaches and stays in the success state until the Azure instance is completely started, stopped, or terminated.
- Scheduling this task in Universal Controller with the right dependencies set up would start and stop EC2 instances based on business needs using a UC workflow.
- This task helps to dynamically manage VM operations. It could potentially reduce the Azure VM running cost in the cloud.



Software Requirements

This integration requires a Universal Agent and a Python runtime to execute the Universal Task against a Azure Virtual Machine.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher . Tested with the Universal Agent bundled Python distribution.
- Python modules required
    - requests
- UAC Global Variables with azure oauth2.0 access should be set in order to run the UT properly.
    - Create a global variable and set the value of that with the Azure access token and this will be used in job definition to make the REST-API calls authenticated from universal controller to Azure logic apps
    - Access token can be populated with a new access token by setting up a web services task that can run periodically; for example, every hour.

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.5 and later with python options installed
- Universal Agent for Linux Version 6.5 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.4.7.0 and later

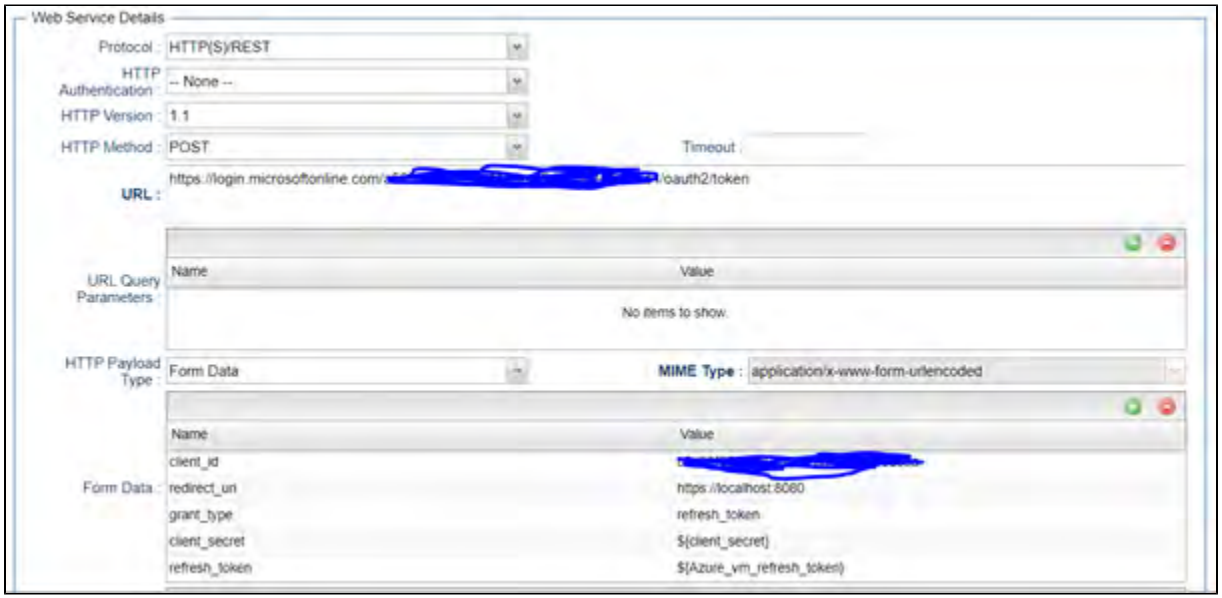
Software Requirements for the Application to be Scheduled

This Universal Task can schedule and execute Azure VM Operations with API Version 2019-12-01.

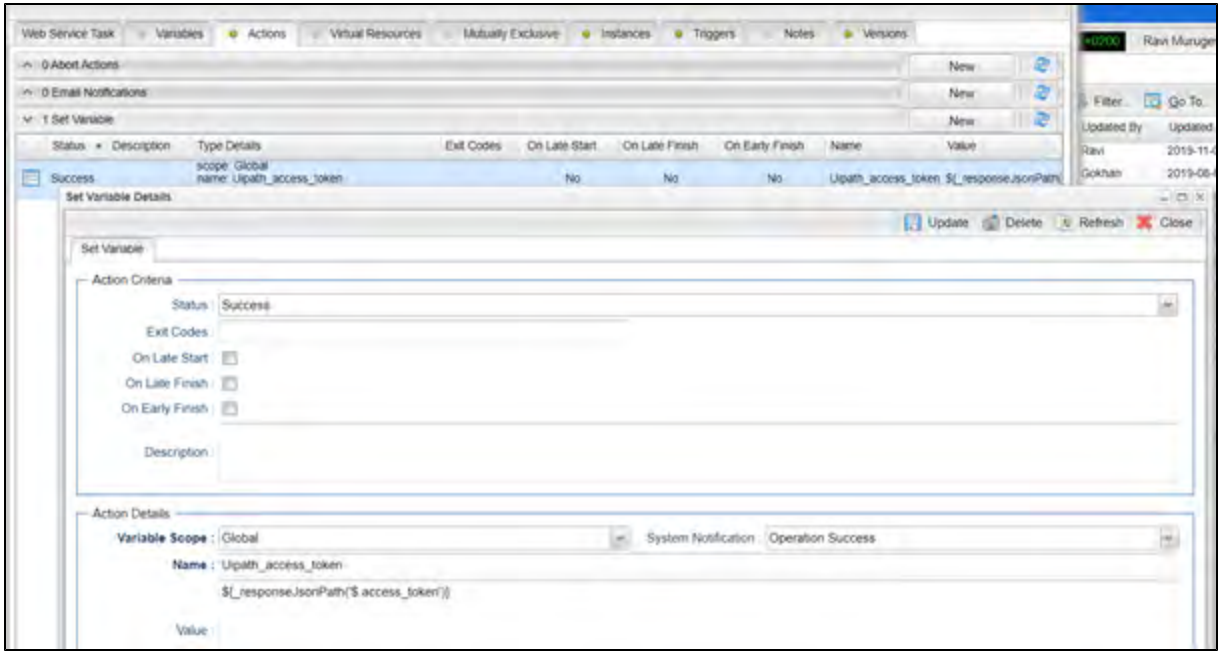
Technical Considerations

This task uses Azure Oauth2.0 access token for Azure API authentication. Users may need to use Universal Controller web services task to refresh the access token periodically.

Generate Access Token Using Sample webservice Task



Generated Access Token can be Stored in a Global Variable by Using the UAC Function for the Above webservices Task in Actions Set Variable



Azure Virtual Machines Key Features

Feature	Description
Start VM	Starts an Azure Virtual Machine
Stop VM	Stop an Azure Virtual Machine
Terminate VM	Terminate an Azure Virtual Machine
VM status	Check the status of an Azure Virtual Machine
List All VM	List the Azure VM's belongs to a subscription & Resource group

Import AWS Virtual Machine Start-Stop-Terminate Instances Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Azure Virtual Machine Start-Stop-Terminate Instances Universal Task

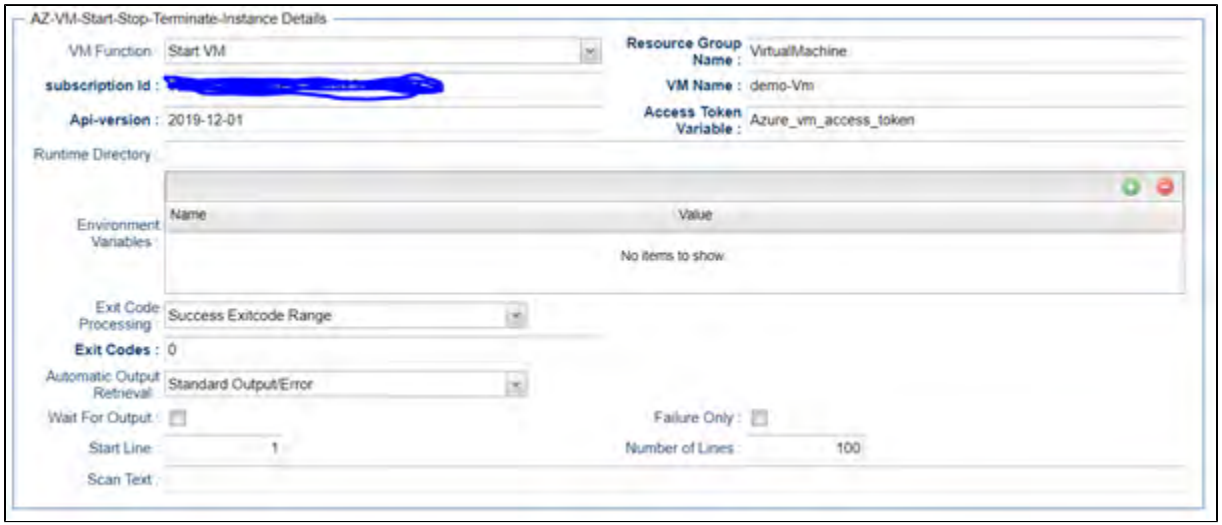
For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Azure Virtual Machine Start-Stop-Terminate Instances Universal Task

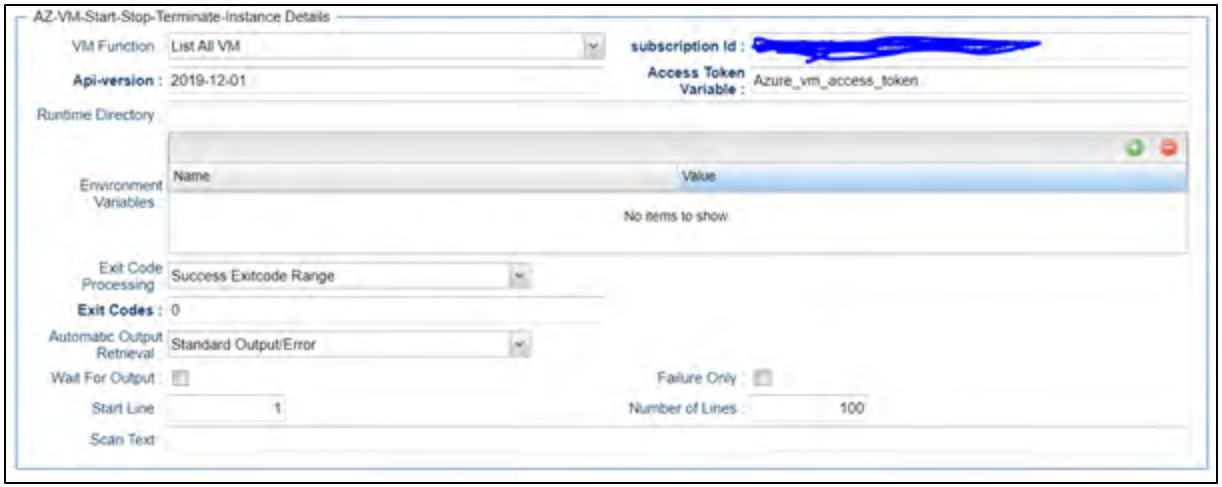
Field	Description
VM Function	Select the function to perform with the Azure VM , Either start or stop or terminate or check status or List VM's
Resource Group Name	Resource group name from the Azure Subscription
subscription Id	Azure Subscription ID
VM Name	Name of the Virtual Machine in Azure
Api-version	Api version in Azure
Access Token Variable	Bearer access that is store Universal controller global Variaböe

Examples for Azure Virtual Machine Start-Stop-Terminate Instances Universal Tasks

Starting an Azure Virtual Machine from Universal Controller



List All Virtual Machines for an Azure Subscription



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Container File Monitoring

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task and associated Task Templates provides a dynamic File Monitoring and File Transfer solution for containerized applications running in any container management solution (for example: OpenShift, Kubernetes).

Overview

For containers running a Universal Agent, or for application pods with a sidecar container running a Universal Agent, the container file system can be dynamically monitored and files automatically transferred from the container file system.

- Dynamically creates and enables an Agent File Monitor Trigger each time specific containers are started.
- Transfer files from the containers.
- Cleanup Agent File Monitor Triggers each time specific containers are stopped.

Software Requirements

This solution-pack requires a Universal Agent and a Python runtime to execute the Universal Task.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests version 2.22.0

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 7.0.0 and later with python options installed
- Universal Agent for Linux Version 7.0.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 7.0.0.0 and later

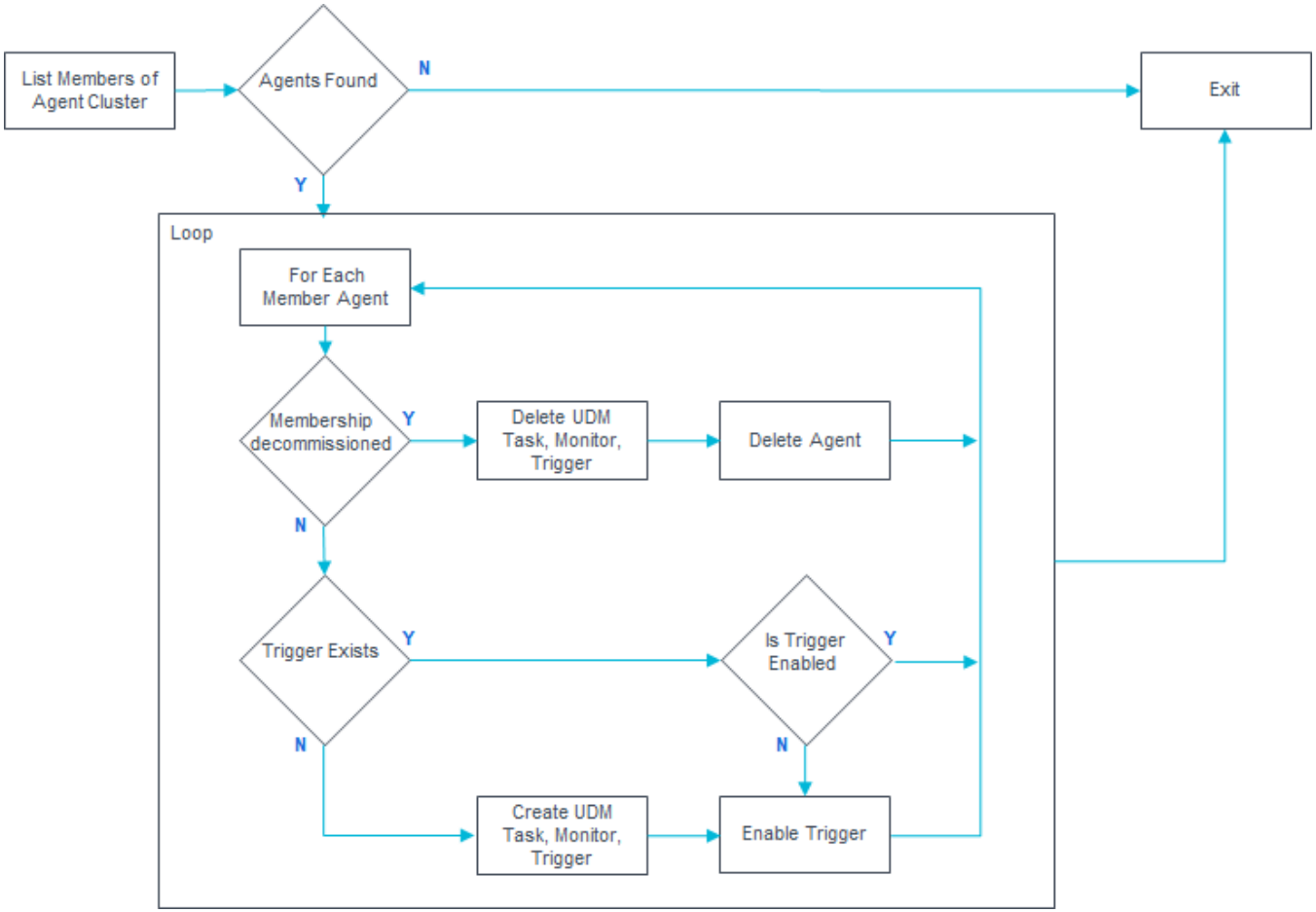
Software Requirements for the Application to be Scheduled

The requests Python module is used to make API call to a Universal Controller instance.

Solution-pack Description

Dynamic container File Monitoring and File Transfers can be achieved with this Universal Task and its associated Task Templates by simply configuring the containerized Universal Agents as Transient and ensuring that they register with a specific Agent Cluster.

Each time the Container File Monitor Universal Task runs, it lists the members of the specified Agent Cluster, and for each member Agent performs the functions detailed in the following flowchart.



Customers must make sure that the Universal Task is executed on a schedule that suits their requirements for how often they need to check whether new containerized Agents have been started or stopped.

This can be done by either:

- Simple Time Trigger based on the desired interval.
- Using a Recurring Task (requires Universal Controller 6.9.x or higher) to provide a loop function.

Import Universal Template

To use the Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.

2. Download the provided ZIP file.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Click Import Template.
5. Select the template ZIP file and Import.

When the template has been imported successfully, the Universal Template will appear on the list. Refresh your Navigation Tree to see these tasks in the Automation Center Menu.

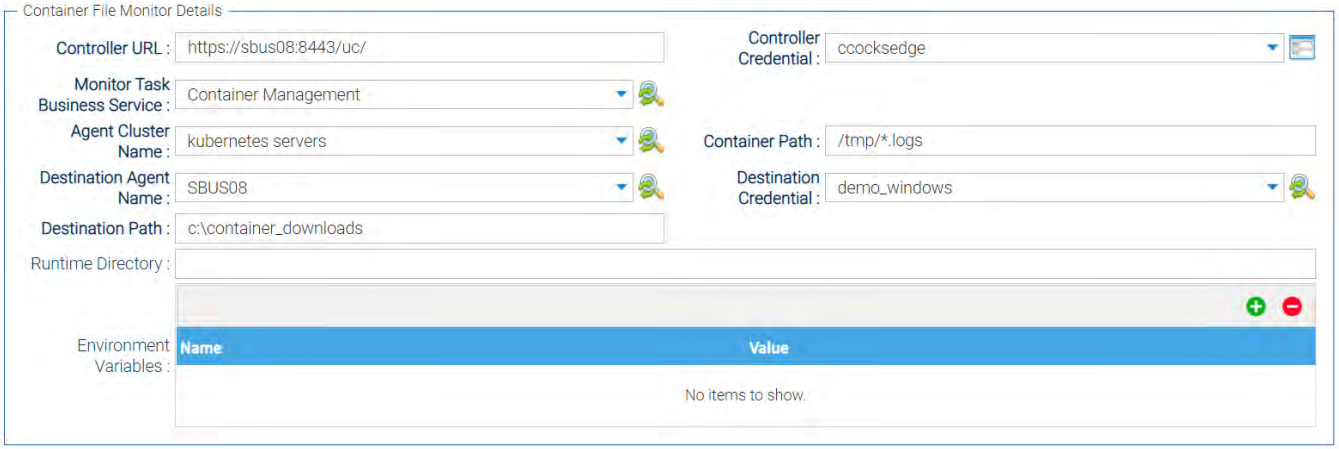
Configure Universal Task

Create a Container File Monitor Universal Task for each Agent Cluster.

Field Descriptions for Container File Monitoring Universal Task

Field	Description
Controller URL	Universal Controller URL; that is: https://localhost:8443/uc
Controller Credential	Universal Controller Credential, specified user will need the following access rights: • Read Access to the specified Agent Cluster • Ability the delete the container Agents • Ability to Create, Enable, Disable and Delete Agent File Monitors, File Transfer Tasks, and Agent File Monitor Triggers that are members of the Business Service specified in the Monitor Task Business Service field.
Monitor Task Business Service	Business Service required for the generated UDM File Transfer Task, Agent File Monitor Task, and Agent File Monitor Trigger definitions.
Agent Cluster Name	Agent Cluster to Monitor.
Container Path	Path on the Container to Monitor; that is `/tmp/*.txt` or `/tmp`. This value is passed to the template Agent File Monitor Task
Destination Agent Name	Destination Agent for the UD File Transfer. This value is used in the generated UDM File Transfer Task.
Destination Credential	Destination Credential for File Transfer. This value is used in the generated UDM File Transfer Task.
Destination Path	Destination Path for File Transfer. This value is used in the generated UDM File Transfer Task.

Example for Container File Monitoring Universal Tasks



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Databricks

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows Stonebranch users to perform end-to-end Orchestration and Automation of Jobs & Clusters in Databricks environment, either in AWS or Azure.

Overview

- This task will use the Databricks URL and the user bearer token to connect with the Databricks environment.
- Users can perform the following with respect to the Databricks jobs.
  - Create and list jobs
  - Get job details
  - Run now jobs
  - Run submit jobs

    - Cancel run jobs
- Also with respect to Databricks clusters, this Universal Task can perform the following operations:
    - Create, start and restart a cluster
    - Terminate a cluster
    - Get a cluster info
    - List clusters
- With respect to Databricks DBFS , this Universal Task also provides a feature to upload larger files.

Software Requirements

This integration requires a Universal Agent and a Python runtime to execute the Universal Task against a Databricks environment.

Software Requirements for Universal Template and Universal Task

Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.

- Python modules required
    - requests

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.6 and later with Python options installed
- Universal Agent for Linux Version 6.6 and later with Python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with the Azure Databricks environment -API version 2.0.

Technical Considerations

- This task uses Python modules requests to make REST-API calls to the Databricks environment.
- Databricks URL and user bearer token would be required as basic input for this Universal Task.
- Please refer to the Databricks API related to jobs and clusters in URL: https://docs.Databricks.com/dev-tools/api/latest/

Key Features

Feature	Description
Create Job	Create a job in a Databricks environment from Universal Controller. Here, a JSON input for job creation in Databricks environment will be used.

List jobs	List the jobs available within the Databricks environment.
Get Job details	Provides an existing job definition in Databricks by providing the job ID as input.
Run now Jobs	This feature helps to run an existing job in Databricks environment using the run time input parameters supplied in JSON from the universal task and the Universal Controller will be monitoring the execution of the job until it gets completed.
Run Submit jobs	This feature helps to run a job in Databricks environment that can be dynamically defined in JSON as an input parameter in the Universal Task and the Universal Controller will be monitoring the execution of the job until it gets completed.
Cancel Run job	Cancel a execution of job that is in running state within the Databricks environment.
Create Cluster	Create a cluster in Databricks environment. Input to be provided in the JSON in a script in this Universal Task.
List clusters	List the clusters available in the Databricks environment.
Start cluster	Start a cluster that is in stopped state in Databricks.
Restart cluster	Restart a cluster in the Databricks environment.
Terminate cluster	Terminate cluster in Databricks environment by providing cluster ID as input.
Get a Cluster info	Provides the definition of an existing cluster in Databricks environment in JSON.
Upload file to DBFS	Upload a file from local server to a Databricks file system DBFS.

Import Databricks Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Databricks Integration Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

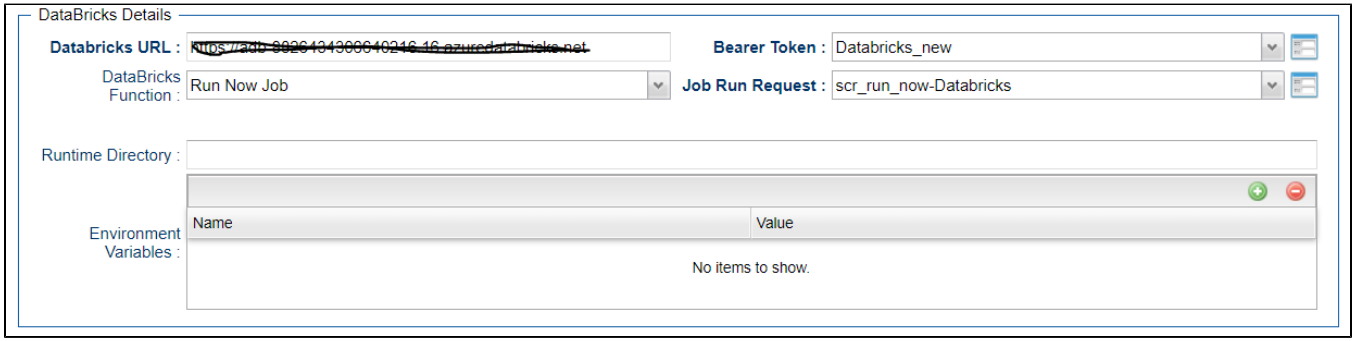
Field Descriptions for Databricks Universal Task

Field	Description
Databricks URL	Specify the Databricks URL.
Bearer Token	Provide the Databricks Personal token or the Azure AD token.

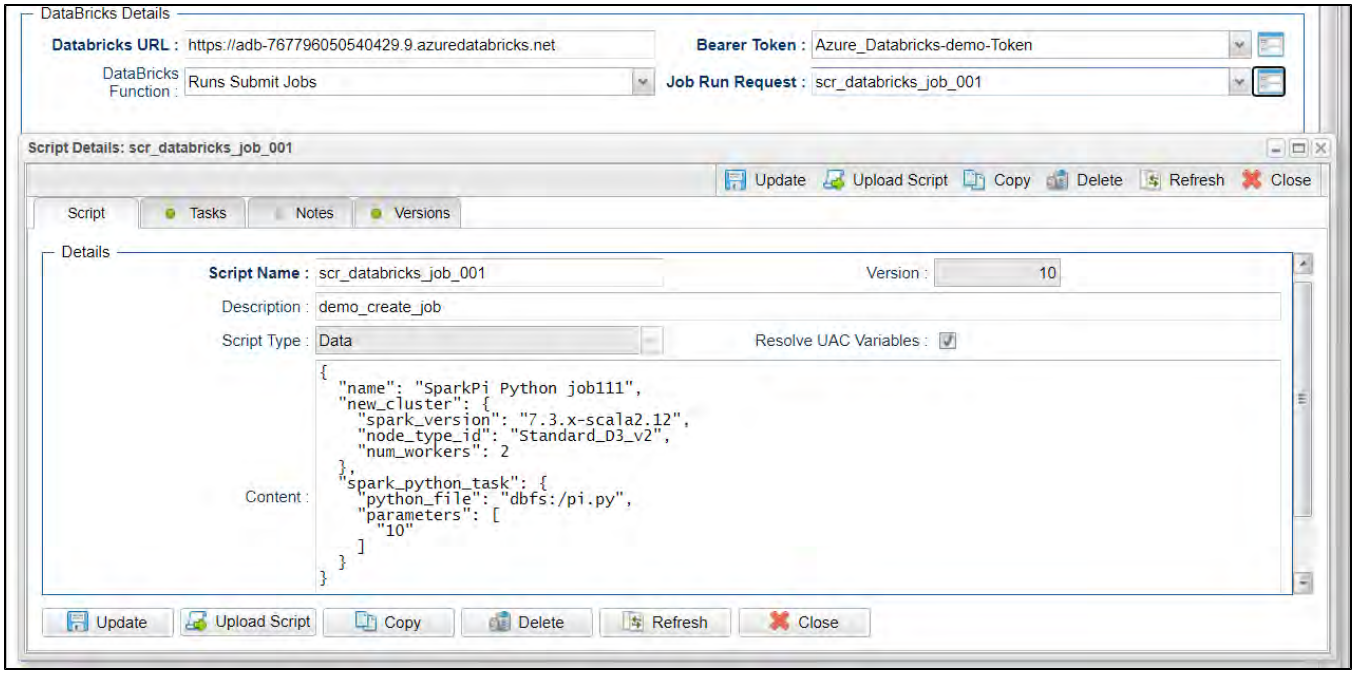
Databricks Function	Select a Function that would like to perform with Databricks.
Create Request Script	Feed the script for the new job creation or cluster in Databricks.
Job ID	Provide the Databricks Job ID.
Job Run Request	Specify the parameters for Jar or notebook or python or spark-submit or the Job submit run request.
Run ID	Specify the Databricks Run ID.
Cluster ID	Provide the cluster ID.
Local file name	Local file name with path.
DBFS file name	Provide the Databricks file path and name.
overwrite	Specify if the uploaded files need to overwritten in DBFS.

Examples for Databricks Integration Universal Tasks

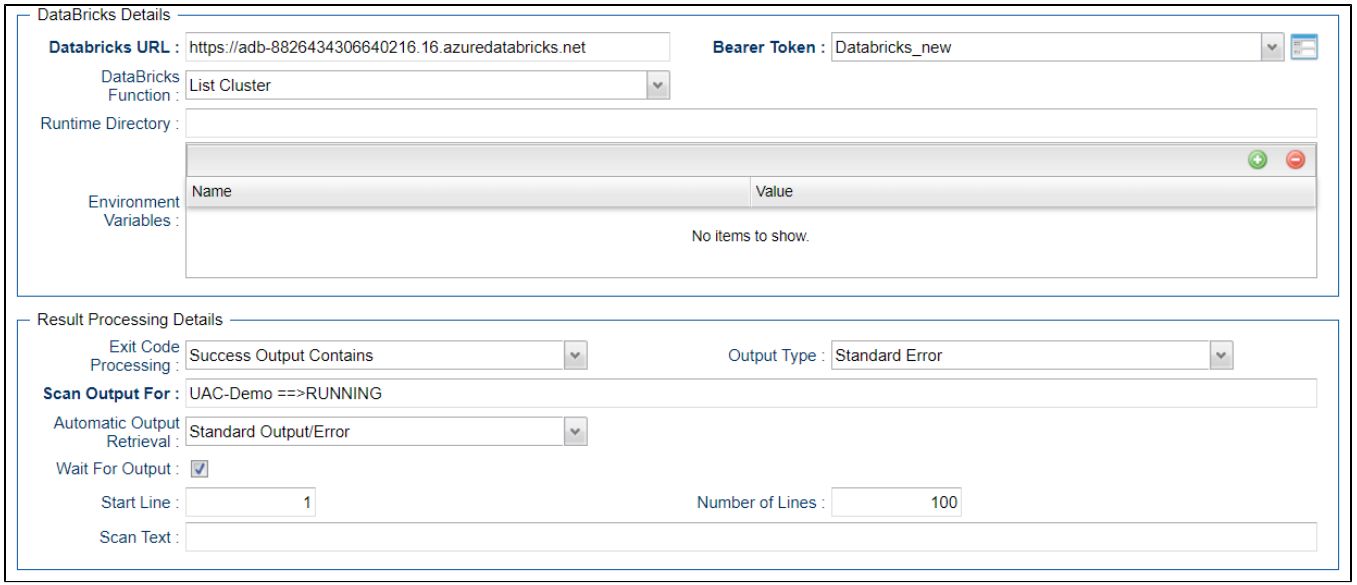
Run now Job



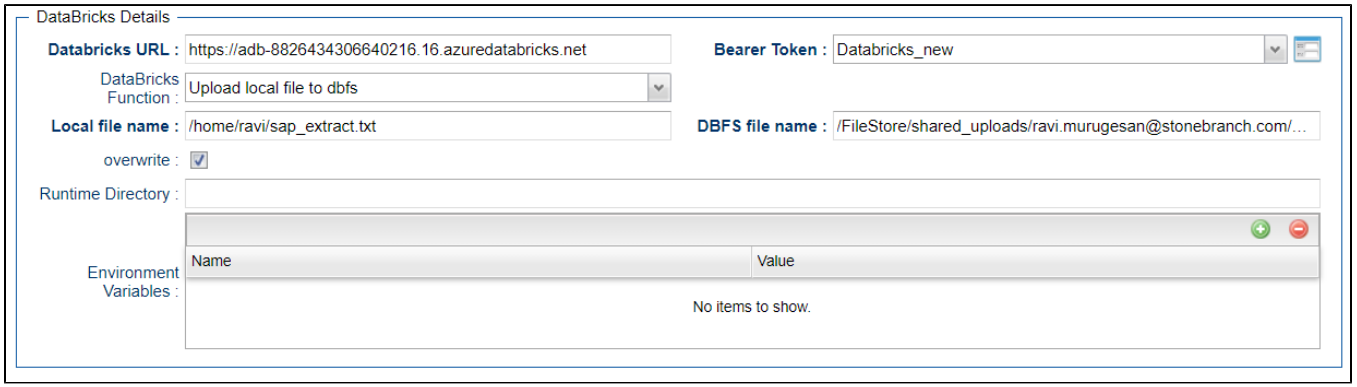
Run Submit Job



List Cluster



Upload Local File to DBFS



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

GitHub

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows you to perform server operations, such as importing/exporting Universal Automation Center objects and integrating with GitHub; for example, you could import a new Universal Template from GitHub into your Universal Controller.

It also allows you to import/export Universal Automation Center objects using the Universal Controller script library.

Software Requirements

Software Requirements for Universal Agent and Universal Controller

- Universal Agent for Linux or Windows Version 6.9.0.0 or later are required.
- For Universal Agent 6.9.0.0, the PyGitHub python module must be installed.

Software Requirements for Universal Controller

- Universal Controller 6.9.0.0 or later.

Software Requirements for the Application to be Scheduled

- A GitHub Account with a GitHub token is required,

A GitHub token can be created under GitHub / Settings / Developer settings / Personal access tokens

Universal Task for GitHub Details

Import	Import any Universal Controller objects such as tasks, calendar, scripts, and trigger from GitHub into Universal Controller; for example, import a new Universal Template in the Marketplace from GitHub to Universal Controller.
Import	Import any Universal Controller object such as Universal Template, tasks, calendar, scripts, and triggers from a script file into Universal Controller; for example, no Internet connection from Universal Controller to GitHub is supported.
Export	Export any Universal Controller object such as tasks, calendar, scripts, and triggers to GitHub from Universal Controller; for example, export a Universal Template to a GitHub repository.
Export	Export any Universal Controller object such as tasks, calendar, scripts, and triggers to a script object, so that the content of the script later can be used to import it on a Controller without needing the UAC import functionality.
Support	Support Stonebranch SaaS Universal Controller and on-premise Universal Controller customers.

Import GitHub Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure GitHub Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for the GitHub Integration Universal Task

Field	Description
Universal Controller URL	Universal Controller URL; for example, Local Universal Controller: http://192.168.88.10:8080/uc/ Stonebranch SaaS Cloud Universal Controller: https://superstore.stonebranchdev.cloud/
Universal Controller Credentials	Credentials of the Universal Controller Webservice API
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
UAC Base URL	The REST API URL for UAC. for example, http://192.168.88.40:8080/uc
UAC REST Credentials	The REST API credentials for UAC.
Operation Type	Operation Type can be one of the following: • **From UAC to GitHub** This operation gets the universal object information from UAC and sends it into GitHub repository. See Export Universal Objects from UAC to GitHub. • **From GitHub to UAC** This operation gets the universal object information from GitHub repository and sends it into UAC. It either creates these or updates the already existing ones. See Import Universal Objects from GitHub to UAC. • **UAC_to_Script** This operation gets the universal object information from UAC and exports it into a UAC script object, so that later the content of the script can be used to import it on a Controller without needing to the UAC import functionality. See Export Universal Objects from UAC to Script. • **Script_to_UAC** This operation gets the universal object information from UAC script and sends it into UAC. It either creates these or updates the already existing ones. See Import Universal Objects from Script to UAC.
Object Type	Object Types in UAC: [agent | agentcluster | businessservice | calendar | customday | credential | databaseconnection | emailconnection | peoplesoftconnection | sapconnection | task | trigger | script | virtual | variable | universaltemplate]
Object Sub Type	The type of the selected object. Refer to Task Types in the Product Documentation; for example, For Object Type tasks: Workflow, Timer, Windows, Linux/Unix, z/OS, …
Business Services	Comma Separated List of Business Services
GitHub Base URL	Should be filled in order to use local/own/private GitHub server. For GitHub please leave empty.
GitHub Token	Connection token for GitHub. Can be generated under GitHub / Settings / Developer settings / Personal access tokens
Repository Path	GitHub repository ; for example, stonebranch-marketplace/ut-sql-tasks used to: • Export Universal Package to GitHub • Import Universal Package from GitHub

Branch	Branch name ; for example, main, development
Task Folder Name	Folder name of the task to be put under the Branch; for example, export
Change Message	Commit definition. ${ops_execution_user}_${ops_agent_name}_app.version_%s_${ops_launch_time}_change_ is added in front of the message; for example, if Change Message field left empty: nbuer_AGT_LINUX_PS4_app.version_1.3_2021-01-20 10:25:48 +0000_change_ if Change Message field = “dev” nbuer_AGT_LINUX_PS4_app.version_1.3_2021-01-20 10:25:48 +0000_change_dev

Naming Conventions

The naming of the objects that are exported to a GitHub are as follows:

<Object Name**>.<**Object Type**>.<**Object Sub Type**>.json**

;for example, AWS Task.task.SAP.json

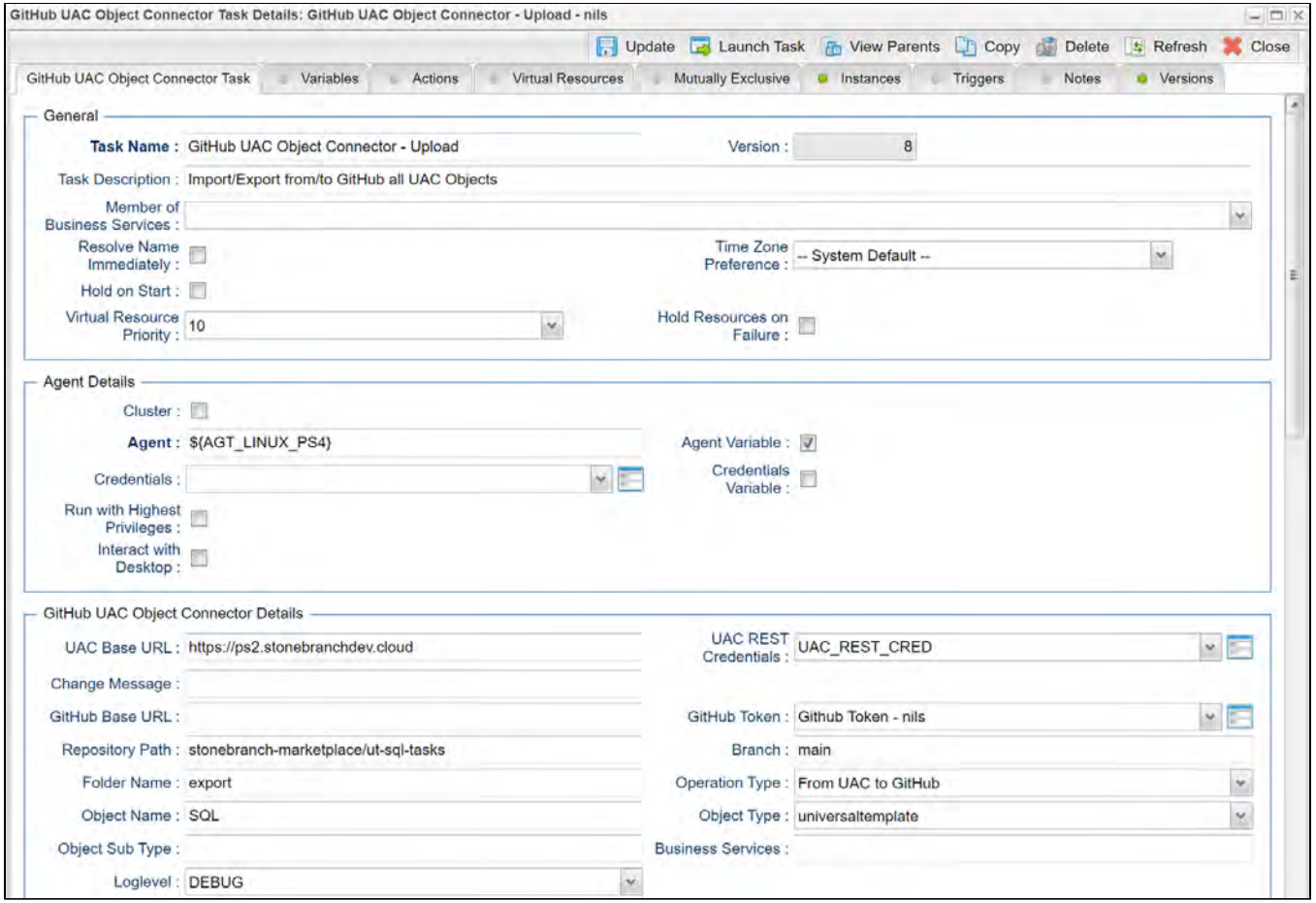
If there is no Object Sub Type, then:

;for example, AWS Task.universaltemplate.json

Examples for GitHub Universal Tasks

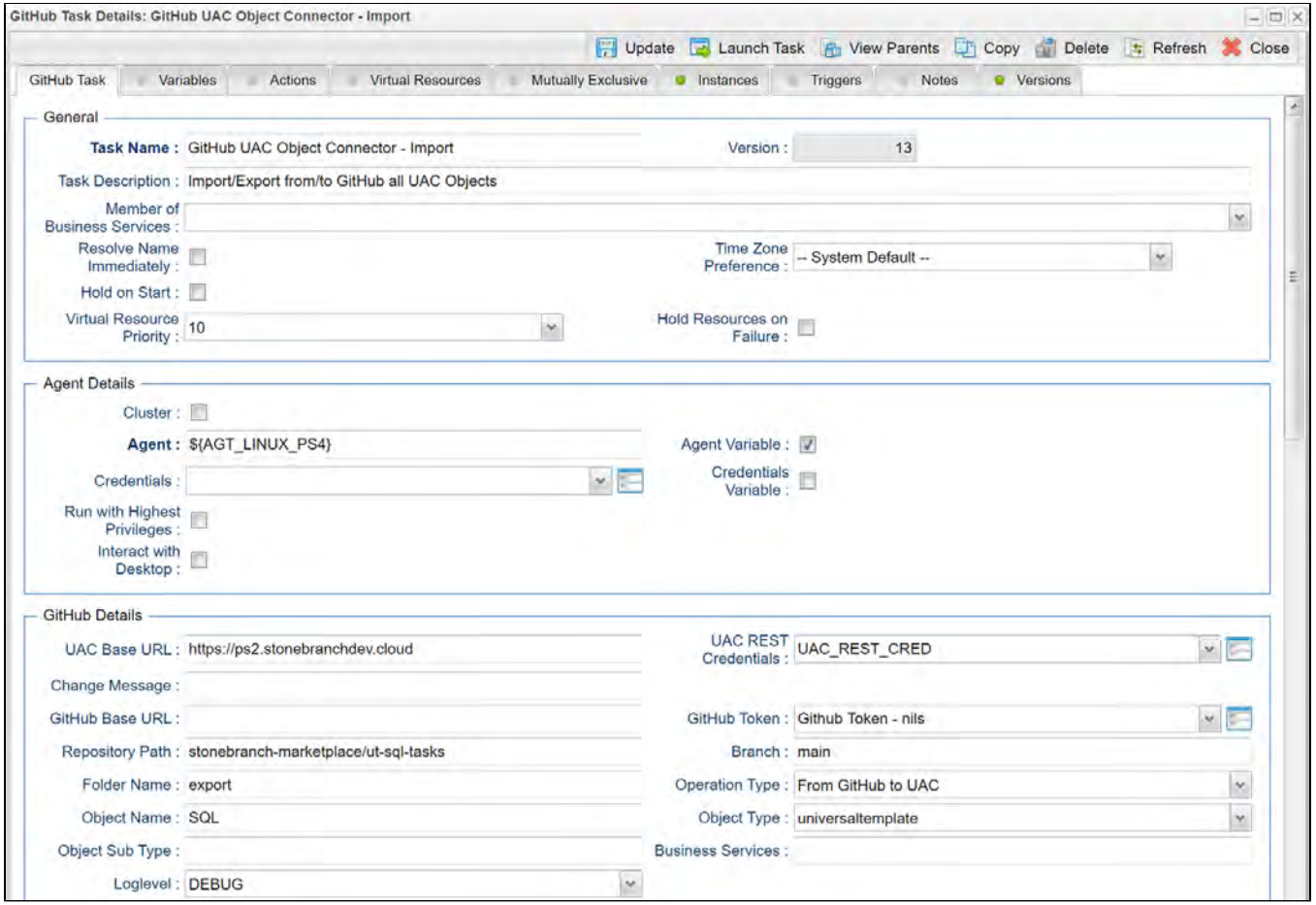
Export Universal Objects from UAC to GitHub

The following task exports the Universal Template **SQL** from Universal Controller to the main branch of the GitHub repository: **stonebranch-marketplace/ut-sql-tasks; folder: export**



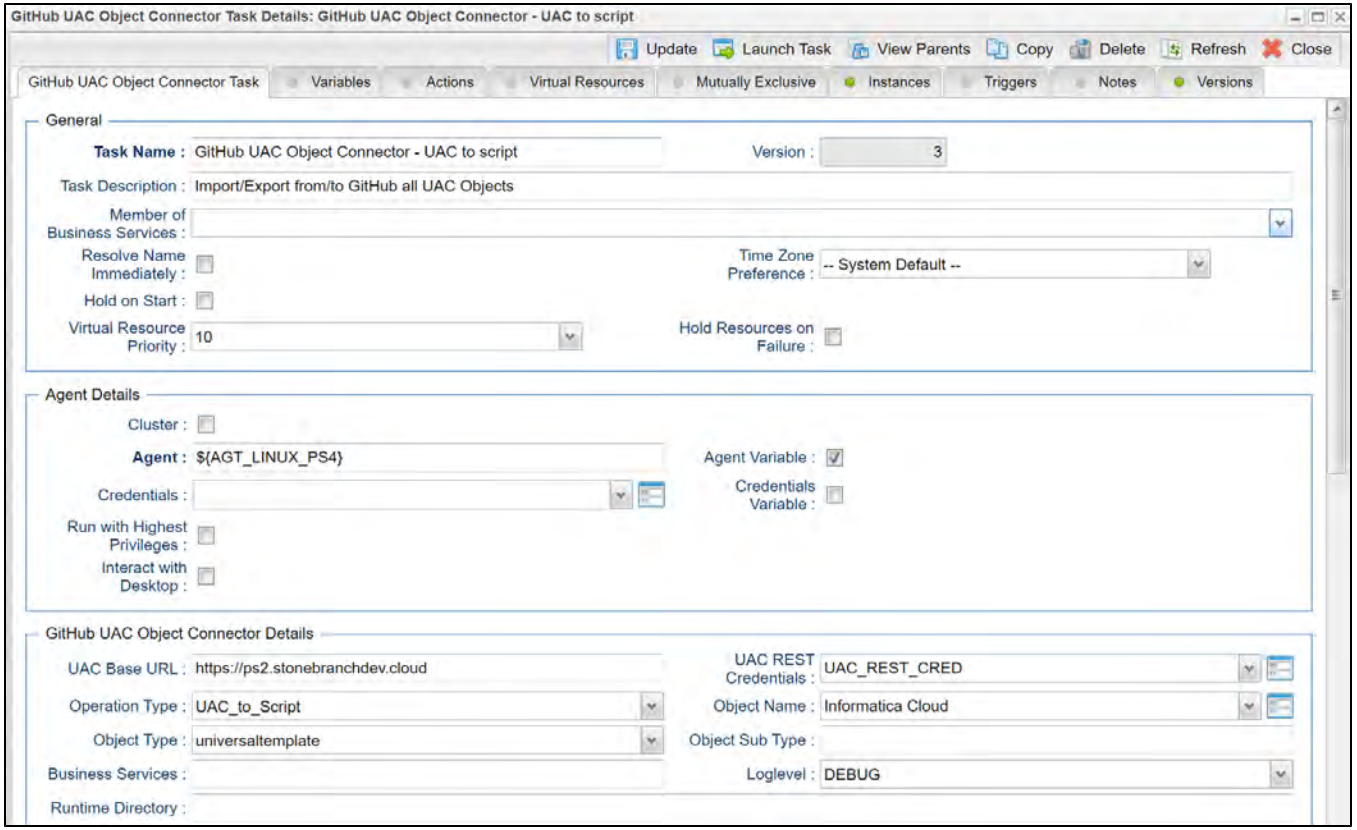
Import Universal Objects from GitHub to UAC

The following Task imports the universal template **SQL** from the main branch of the GitHub repository: **stonebranch-marketplace/ut-sql-tasks; folder export**



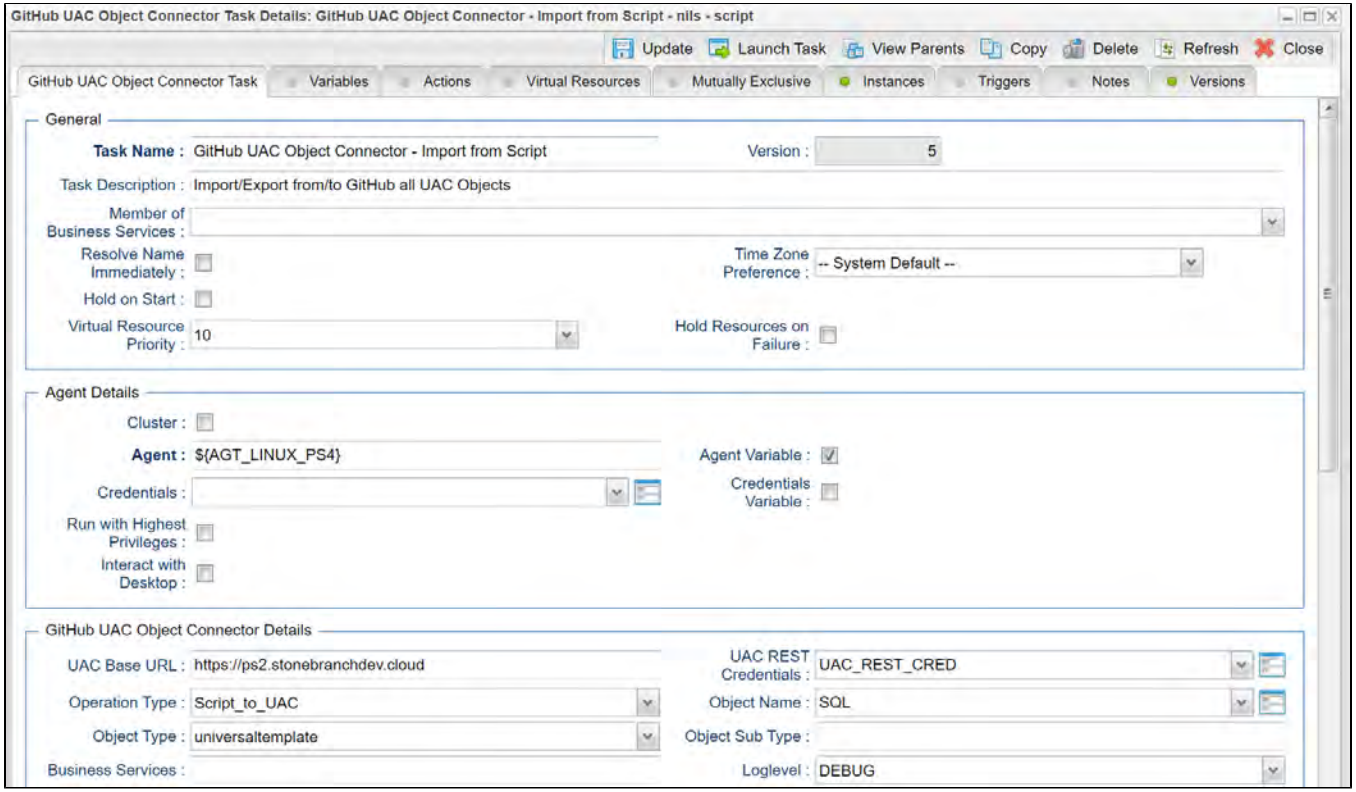
Export Universal Objects from UAC to Script

This task gets the object for the Universal Template “Informatica Cloud” from UAC and exports it into a UAC script object with the same name, so that later the content of the script can be used to import it on a Controller without needing the UAC import functionality.

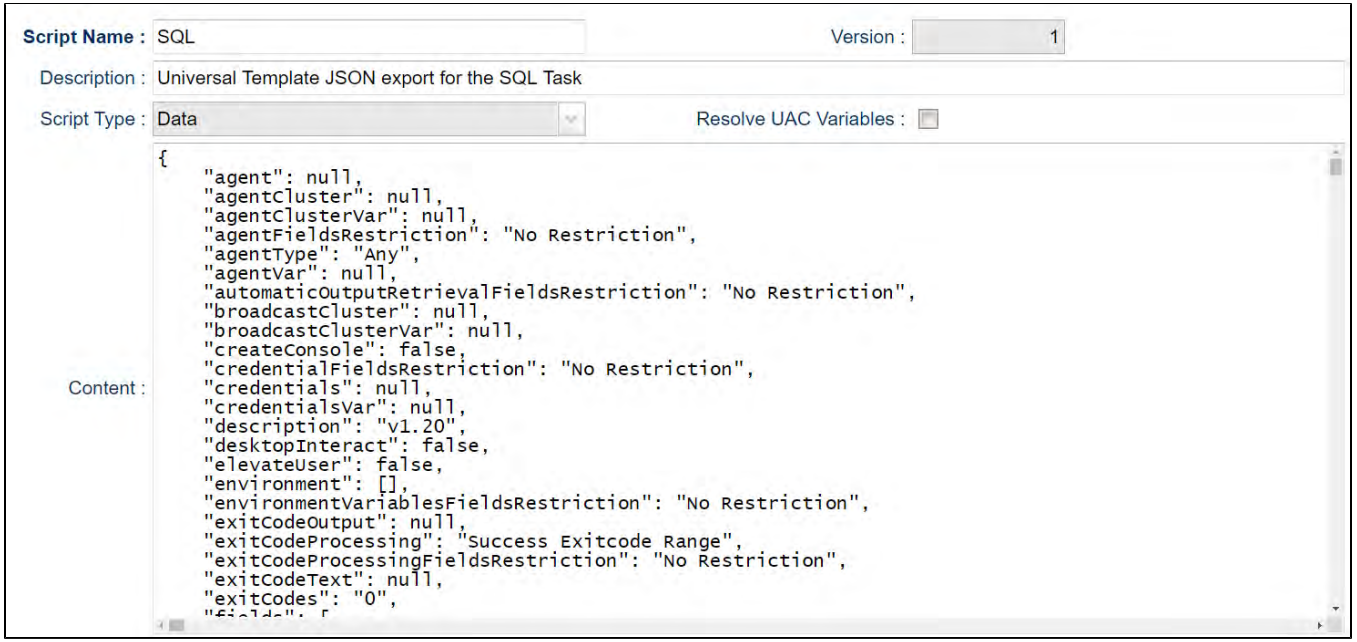


Import Universal Objects from Script to UAC

The following task imports the Universal Template as defined in the JSON script named **SQL.**



JSON Script that Defines Universal Template SQL


Script Name : SQL
Version : 1
Description : Universal Template JSON export for the SQL Task
Script Type : Data
Resolve UAC Variables :
Content :
{
"agent": null,
"agentCluster": null,
"agentClusterVar": null,
"agentFieldsRestriction": "No Restriction",
"agentType": "Any",
"agentVar": null,
"automaticOutputRetrievalFieldsRestriction": "No Restriction",
"broadcastCluster": null,
"broadcastClusterVar": null,
"createConsole": false,
"credentialFieldsRestriction": "No Restriction",
"credentials": null,
"credentialsVar": null,
"description": "V1.20",
"desktopInteract": false,
"elevateUser": false,
"environment": [],
"environmentVariablesFieldsRestriction": "No Restriction",
"exitCodeOutput": null,
"exitCodeProcessing": "Success Exitcode Range",
"exitCodeProcessingFieldsRestriction": "No Restriction",
"exitCodeText": null,
"exitCodes": "0",

Google BigQuery

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows Stonebranch users to schedule, trigger, monitor, and orchestrate the Google BigQuery process directly from Universal Controller.

Overview

- This task uses Python modules google-cloud-bigquery and google-auth to make REST-API calls to Google BigQuery
- This task will use the GCP Project ID, BigQuery SQL or Schema, Dataset ID, Job ID, Location, Table ID, Cloud Storage URI, and Source File Format as parameters of BigQuery function, and GCP KeyFile (API KEY) of Service account for authenticating the REST-API calls to Google BigQuery.

- User can perform the following Google BigQuery operations:
    - BigQuery SQL
    - List dataset
    - List tables in dataset
    - View job information
    - Create a dataset
    - Load local file to a table
    - Load cloud storage data to a table
    - Export table data

Software Requirements

This Universal Task requires a Universal Agent and a Python runtime to execute the Universal Task against a Google BigQuery data warehouse.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.4 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - google-cloud-bigquery
    - google-auth

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.6 and later with Python options installed.
- Universal Agent for Linux Version 6.6 and later with Python options installed.

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later.

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with the Google BigQuery data warehouse.

Technical Considerations

- This Universal Task uses the Python modules Google auth and Google BigQuery management to make REST-API calls with Google BigQuery
- Create a service account in your GCP identity management

Key Features

Feature	Description

BigQuery SQL	Runs a BigQuery SQL query and returns query results.
List dataset	Lists all existing datasets in a particular project.
List tables in dataset	List tables in a particular dataset.
View job information	Retrieve the information of a job for a given job ID.
Create a dataset	Create a dataset within a project.
Load local file to a table	Load a local file to a BigQuery table.
Load cloud storage data to a table	Load a file form cloud storage to BigQuery table.
Export table data	Export table to cloud storage bucket.

Import Google BigQuery Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Google BigQuery Integration Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

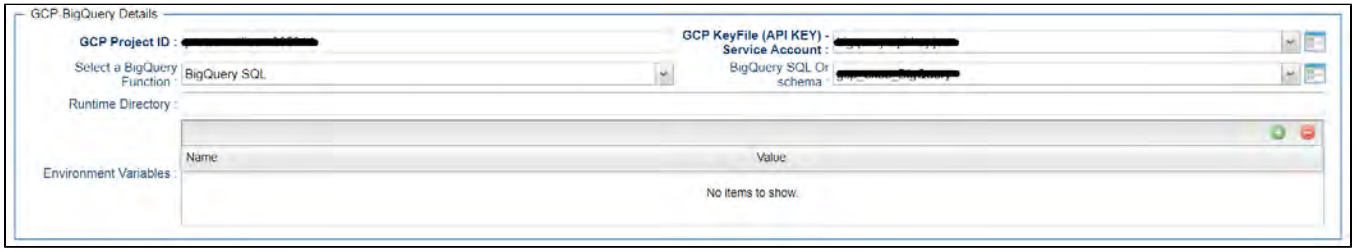
Field Descriptions for Google BigQuery Integration Universal Task

Field	Description
GCP Project ID	Provide Google Cloud Platform (GCP) project ID.
GCP KeyFile (API KEY) Service account	Provide Google Cloud Platform Service account authentication key in JSON format.
Select a BigQuery Function	Select a function for execution in GCP.
BigQuery SQL or Schema	Provide BigQuery SQL or schema as applicable for function to load data either from cloud or local storage.
Dataset ID	Provide GCP Dataset ID (name of the dataset) - Must be alphanumeric.
Job ID	Provide BigQuery Job ID.
Location	Provide GCP BigQuery location.
Table ID	Provide table ID(Name of the table where the file needs to be loaded).

Cloud Storage URI	Provide URI for the cloud storage.
Source File Format	Provide source file format for the data load in to table.
Local File Path	Provide local file path for the data load in to a table.
Header Row to Skip	Provide an integer indicating the number of header rows in the source data.
Schema Auto Detect	Option to enable schema auto detection when loading data applicable only for JSON and CSV format.

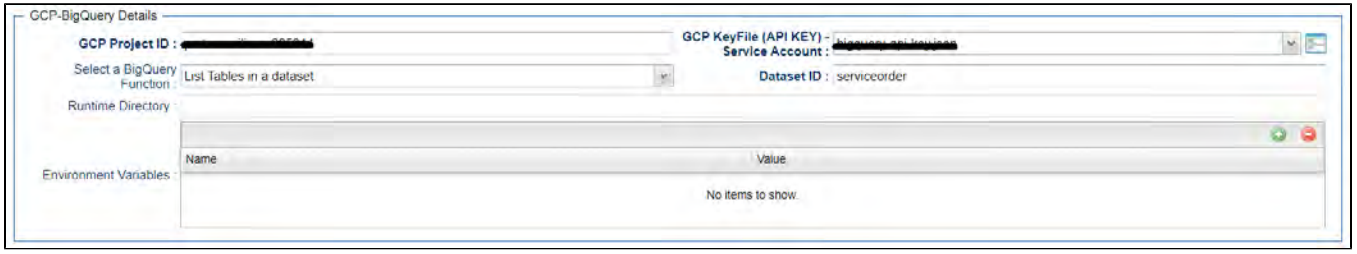
Examples for Google BigQuery Integration Universal Tasks

BigQuery SQL

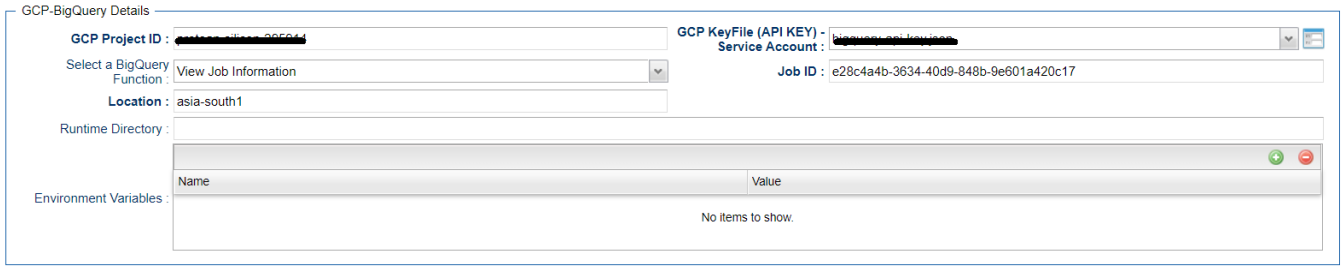


List dataset

List Tables in dataset

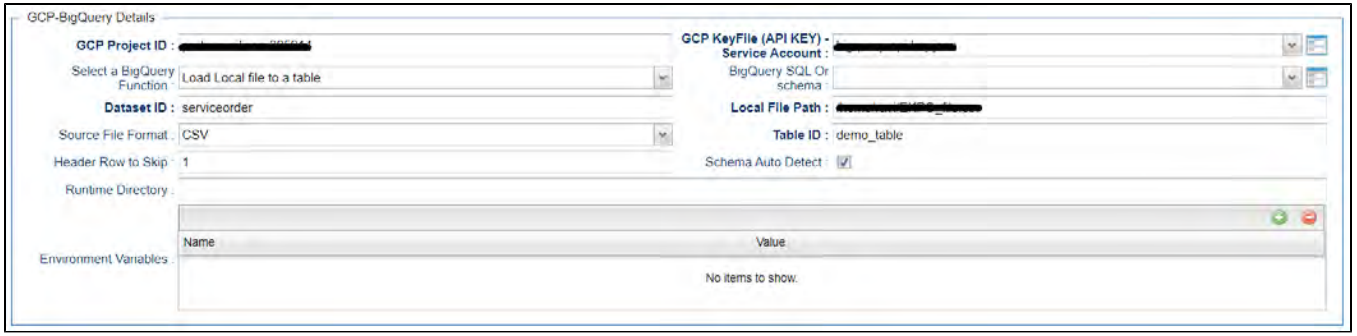


View Job Information

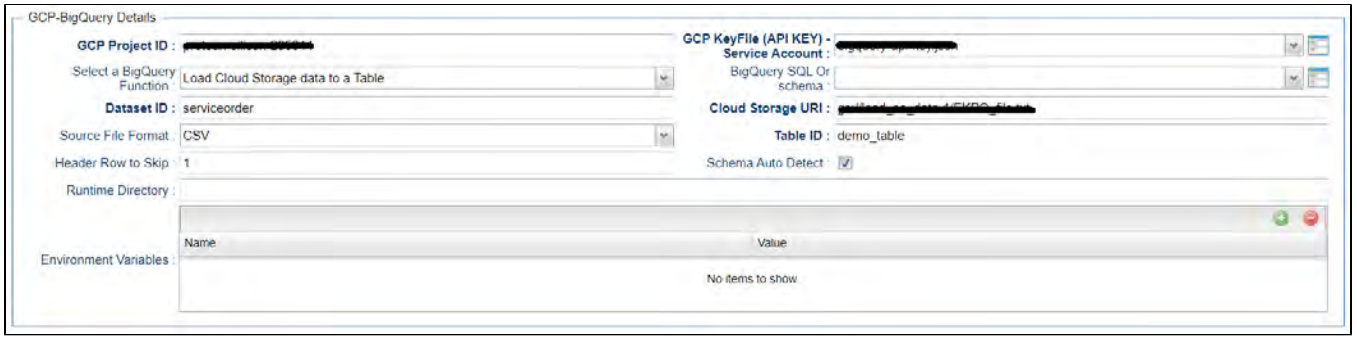


Create a dataset

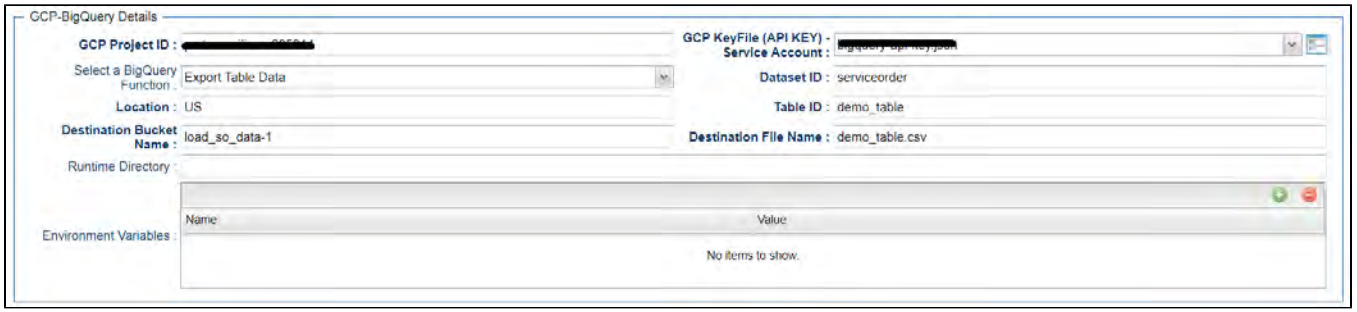
Load Local File to a Table



Load Cloud Storage Data to a Table



Export Table Data



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Informatica Cloud

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Overview

This Universal Task allows users to schedule any Data Integration Task or Linear Taskflow in the Informatica Cloud.

All communication is Web-Service based using the latest Informatica REST API version 2 & 3 with support for folders.

Log-files including activity-, session- and error-log are available from the Universal Controller Web UI in the same way as from the Informatica Monitoring Console.

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 6.9.0.0 or later is required.

Software Requirements for Universal Controller

- Universal Controller 6.9.0.0 or later.
- Universal Controller license key with support for SAP Connector is required.

Software Requirements for the Application to be Scheduled

In order to connect to the SAP System, the SAP NetWeaver RFC SDK 7.50 libraries are required from SAP.

Those can be downloaded from the SAP Software Download: SAP NetWeaver RFC SDK 7.50.

Technical Considerations

- Start a data integration task in Informatica Cloud.
- Support for folder by using latest REST API version 3.
- Automatic Retrieval of Activity, Session and Error-log *****.
- Supported task types for Data Integration:
    - Masking Task
    - Replication Task
    - Synchronization Task
    - Mapping Task
    - PowerCenter Task
    - Linear Taskflow
- Integrate the Informatica Task in any scheduling workflow.

*

The Error-log is retrieved only for Tasks. For workflows, the error message is provided.

Import Informatica Cloud Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Informatica Cloud Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Informatica Cloud Universal Task

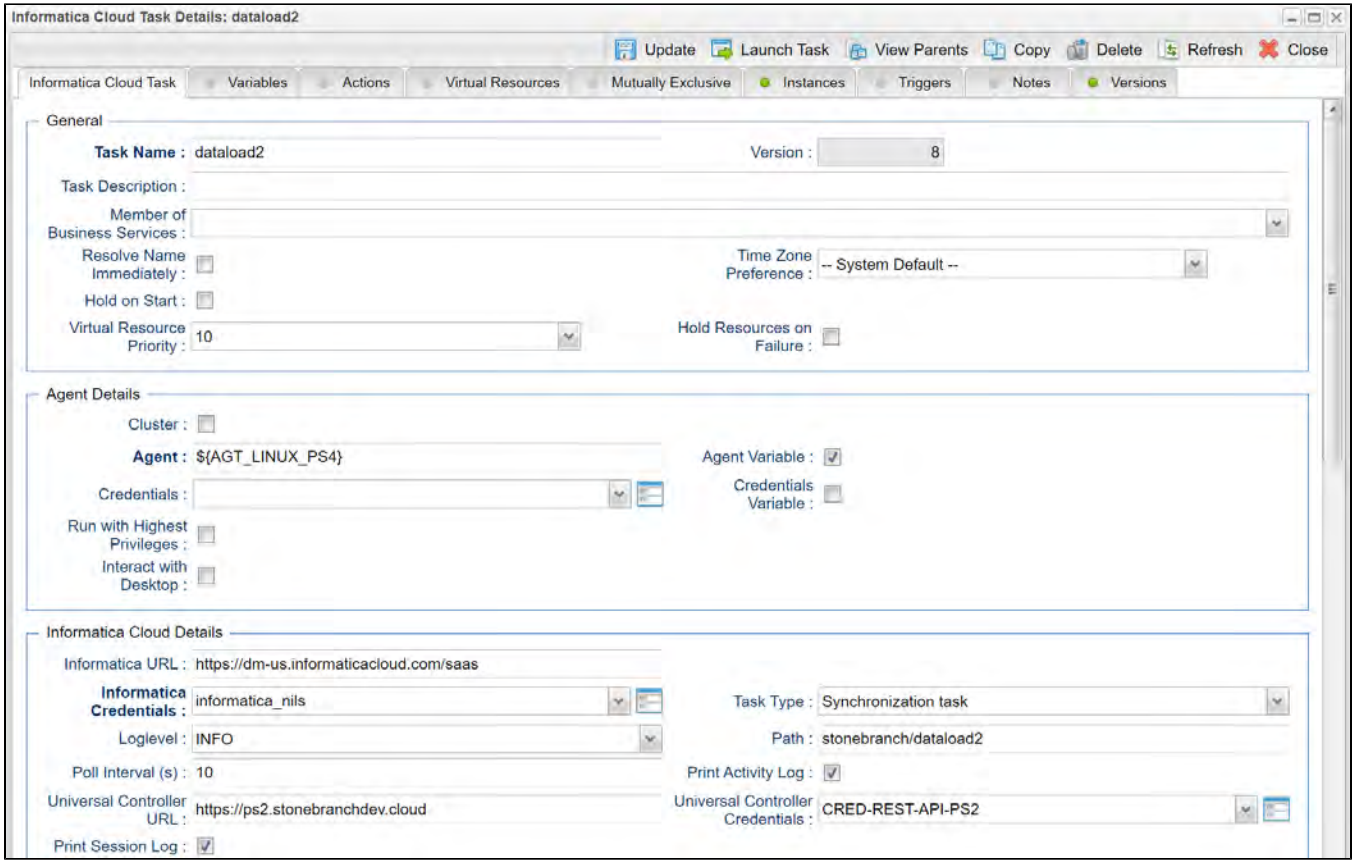
Field	Description
Informatica URL	Endpoint URL of the Informatica Cloud Rest API; for example, endpoint URL of the Informatica SaaS API: https://dm-us.informaticacloud.com/saas
Informatica Credentials	Informatica credentials for basic authentication with username and password.
Task Type	The following Task Types are available for selection: [Masking Task | Replication Task | Synchronization Task | Mapping Task | PowerCenter Task | Linear Taskflow]
Path	<path>/<taskname> (path is optional) Path and name of the task to start in Informatica Cloud; for example, *stonebranch/dataload2* starts the task *dataload2* in the folder *stonebranch*
Print Activity Log	Prints the Activity Log to the Output of the Task Instance. For an example refer to Activity Log screenshot. **Note**: The Error Log always will be printed to the Output of the Task Instance.
Print Session Log	Prints the Session Log to the Output of the Task Instance. **Note**: The Error Log always will be printed to the Output of the Task Instance.
Useproxy (default is NO)	[NO | YES] If set to YES, the fields to set-up the proxy server connections are displayed: • Proxy Server IP or hostname • Proxy Server Port • Proxy Server Credentials (optional)
Universal Controller URL	Universal Controller URL. URL has no backslash "/" at the end. Example: http://192.168.88.10:8080/uc
Universal Controller Credentials	Universal Controller Credential.s The Credentials must have "Web Service Access" Permissions.
Poll Interval (s)	Task Polling Interval in seconds; for example, a Poll Interval of 60 means that every 60 seconds, the Informatica Cloud will we queried if the started task has been finished.
Loglevel (default is INFO)	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following screen shows an example of an Informatica Cloud Universal Task, which will start the Synchronization task dataload2 in the folder stonebranch.

The Task will print the Session Log and Activity Log into the Output of the task instance.

Informatica Cloud Universal Task

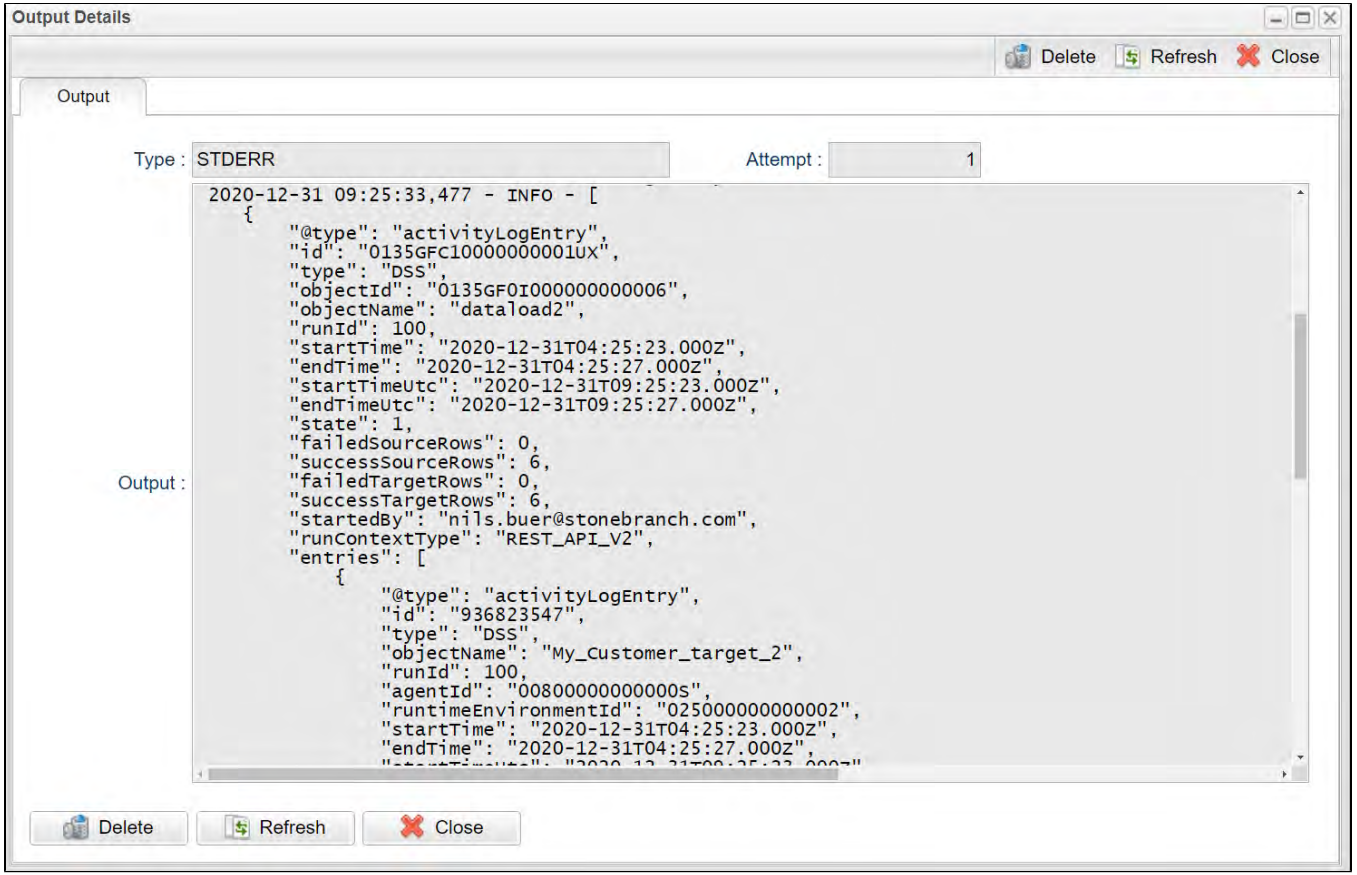


Log Files

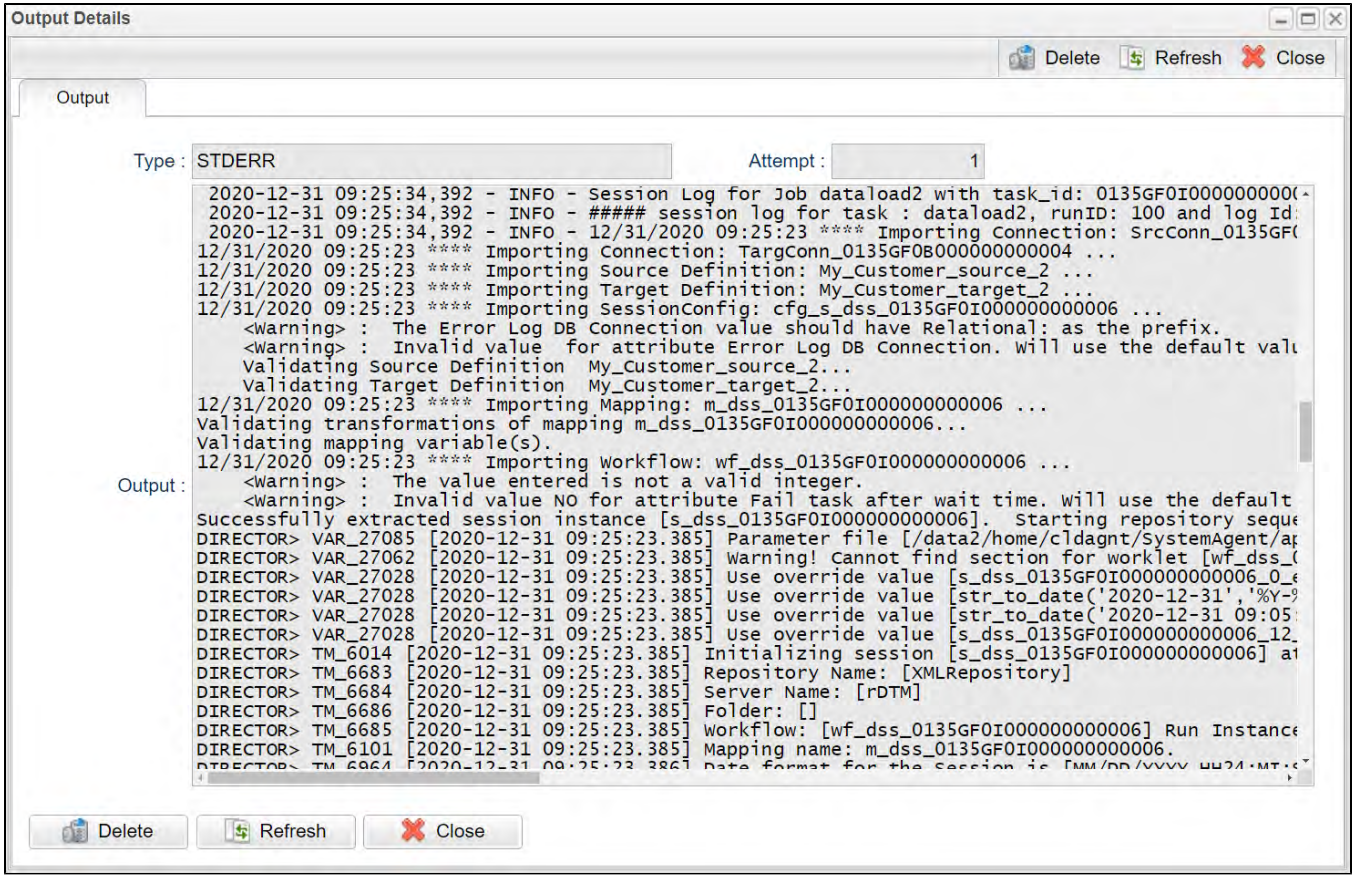
The Activity Log and Session Log will be provided if the corresponding Flags (Print Session Log, Print Activity Log) are set in the Universal Task.

The Error Log always will be printed to the Output of the Task Instance.

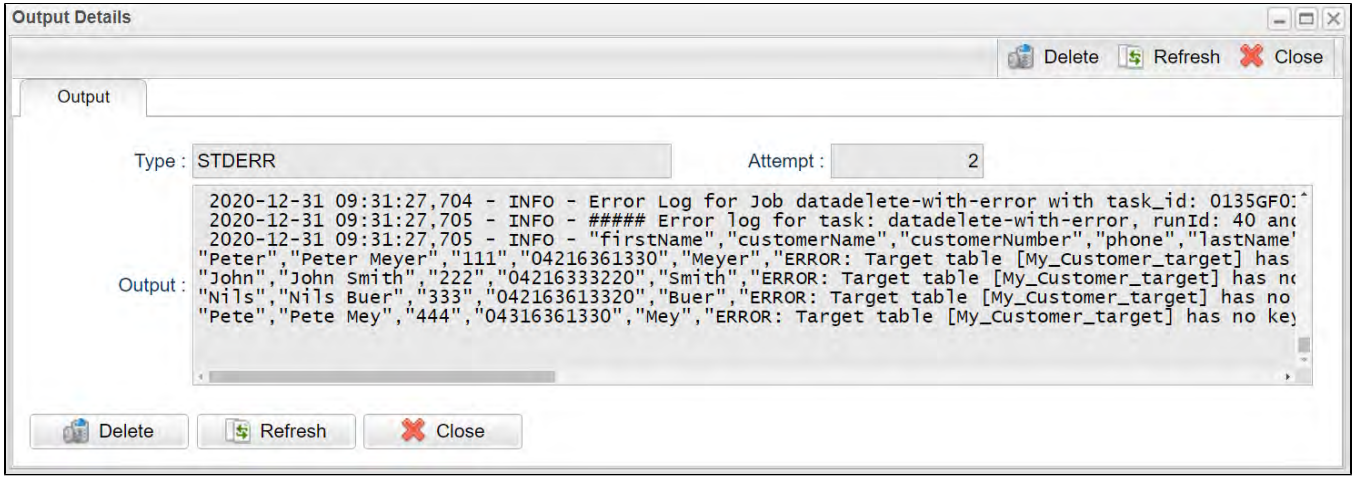
Activity Log


Output Details
Delete
Refresh
Close
Output
Type :
STDERR
Attempt :
1
Output :
2020-12-31 09:25:33,477 - INFO - [
{
"@type": "activityLogEntry",
"id": "0135GFC10000000001UX",
"type": "DSS",
"objectId": "0135GF0I000000000006",
"objectName": "dataload2",
"runId": 100,
"startTime": "2020-12-31T04:25:23.000Z",
"endTime": "2020-12-31T04:25:27.000Z",
"startTimeUtc": "2020-12-31T09:25:23.000Z",
"endTimeUtc": "2020-12-31T09:25:27.000Z",
"state": 1,
"failedSourceRows": 0,
"successSourceRows": 6,
"failedTargetRows": 0,
"successTargetRows": 6,
"startedBy": "nils.buer@stonebranch.com",
"runContextType": "REST_API_V2",
"entries": [
{
"@type": "activityLogEntry",
"id": "936823547",
"type": "DSS",
"objectName": "My_Customer_target_2",
"runId": 100,
"agentId": "00800000000000S",
"runtimeEnvironmentId": "025000000000002",
"startTime": "2020-12-31T04:25:23.000Z",
"endTime": "2020-12-31T04:25:27.000Z",
Delete
Refresh
Close

Session Log


Output Details
Delete
Refresh
Close
Output
Type : STDERR
Attempt : 1
Output :
2020-12-31 09:25:34,392 - INFO - Session Log for Job dataload2 with task_id: 0135GF0I000000000(
2020-12-31 09:25:34,392 - INFO - ##### session log for task : dataload2, runID: 100 and log Id:
2020-12-31 09:25:34,392 - INFO - 12/31/2020 09:25:23 **** Importing Connection: SrcConn_0135GF(
12/31/2020 09:25:23 **** Importing Connection: TargConn_0135GF0B000000000004 ...
12/31/2020 09:25:23 **** Importing Source Definition: My_Customer_source_2 ...
12/31/2020 09:25:23 **** Importing Target Definition: My_Customer_target_2 ...
12/31/2020 09:25:23 **** Importing SessionConfig: cfg_s_dss_0135GF0I000000000006 ...
<Warning> : The Error Log DB Connection value should have Relational: as the prefix.
<Warning> : Invalid value for attribute Error Log DB Connection. Will use the default valu
Validating Source Definition My_Customer_source_2...
Validating Target Definition My_Customer_target_2...
12/31/2020 09:25:23 **** Importing Mapping: m_dss_0135GF0I000000000006 ...
Validating transformations of mapping m_dss_0135GF0I000000000006...
Validating mapping variable(s).
12/31/2020 09:25:23 **** Importing Workflow: wf_dss_0135GF0I000000000006 ...
<Warning> : The value entered is not a valid integer.
<Warning> : Invalid value NO for attribute Fail task after wait time. Will use the default
Successfully extracted session instance [s_dss_0135GF0I000000000006]. Starting repository seque
DIRECTOR> VAR_27085 [2020-12-31 09:25:23.385] Parameter file [/data2/home/cldagnt/SystemAgent/ap
DIRECTOR> VAR_27062 [2020-12-31 09:25:23.385] Warning! Cannot find section for worklet [wf_dss_(
DIRECTOR> VAR_27028 [2020-12-31 09:25:23.385] Use override value [s_dss_0135GF0I000000000006_0_e
DIRECTOR> VAR_27028 [2020-12-31 09:25:23.385] Use override value [str_to_date('2020-12-31','%Y-%
DIRECTOR> VAR_27028 [2020-12-31 09:25:23.385] Use override value [str_to_date('2020-12-31 09:05:
DIRECTOR> VAR_27028 [2020-12-31 09:25:23.385] Use override value [s_dss_0135GF0I000000000006_12_
DIRECTOR> TM_6014 [2020-12-31 09:25:23.385] Initializing session [s_dss_0135GF0I000000000006] at
DIRECTOR> TM_6683 [2020-12-31 09:25:23.385] Repository Name: [XMLRepository]
DIRECTOR> TM_6684 [2020-12-31 09:25:23.385] Server Name: [rDTM]
DIRECTOR> TM_6686 [2020-12-31 09:25:23.385] Folder: []
DIRECTOR> TM_6685 [2020-12-31 09:25:23.385] Workflow: [wf_dss_0135GF0I000000000006] Run Instance
DIRECTOR> TM_6101 [2020-12-31 09:25:23.385] Mapping name: m_dss_0135GF0I000000000006.
Delete
Refresh
Close

Error Log


Output Details
Delete
Refresh
Close
Output
Type :
STDERR
Attempt :
2
Output :
2020-12-31 09:31:27,704 - INFO - Error Log for Job datadelete-with-error with task_id: 0135GF0:
2020-12-31 09:31:27,705 - INFO - ##### Error log for task: datadelete-with-error, runId: 40 and
2020-12-31 09:31:27,705 - INFO - "firstName","customerName","customerNumber","phone","lastName"
"Peter","Peter Meyer","111","04216361330","Meyer","ERROR: Target table [My_Customer_target] has
"John","John Smith","222","04216333220","Smith","ERROR: Target table [My_Customer_target] has no
"Nils","Nils Buer","333","042163613320","Buer","ERROR: Target table [My_Customer_target] has no
"Pete","Pete Mey","444","04316361330","Mey","ERROR: Target table [My_Customer_target] has no key
Delete
Refresh
Close

Informatica PowerCenter

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Overview

This Universal Task allows users to schedule Informatica PowerCenter Workflows and Tasks, including retrieval of the workflow and session log.

It also is possible to start a Workflow from a certain task onwards.

The Universal Task schedules Informatica PowerCenter via the Informatica PowerCenter Web Services Hub; therefore, no installation on any Informatica Informatica system is required.

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 6.9.0.0 or later is required.

Software Requirements for Universal Controller

- Universal Controller 6.9.0.0 or later.

Software Requirements for the Application to be Scheduled

- The Universal Task has been tested for PowerCenter 10.1.0.
- The Informatica PowerCenter Web Services Hub needs to be started in Informatica PowerCenter Administrator.

Technical Considerations

- This task It is based on the standard Informatica PowerCenter Web Services Hub using SOAP protocol.
- The Informatica PowerCenter Web Services Hub Interface is called from a Universal Agent running on a Linux Server or Windows Server.
- Start a Task in an Informatica PowerCenter Workflow.
- Start an Informatica PowerCenter Workflow.
- Start an Informatica PowerCenter Workflow from a given Task onwards.
- Different log-levels can be selected for example, Info and debug.
- The Workflow log is always provided.
- Http and Https connections are support (Note: the host certificate is not verified).

Import Informatica PowerCenter Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Informatica PowerCenter Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

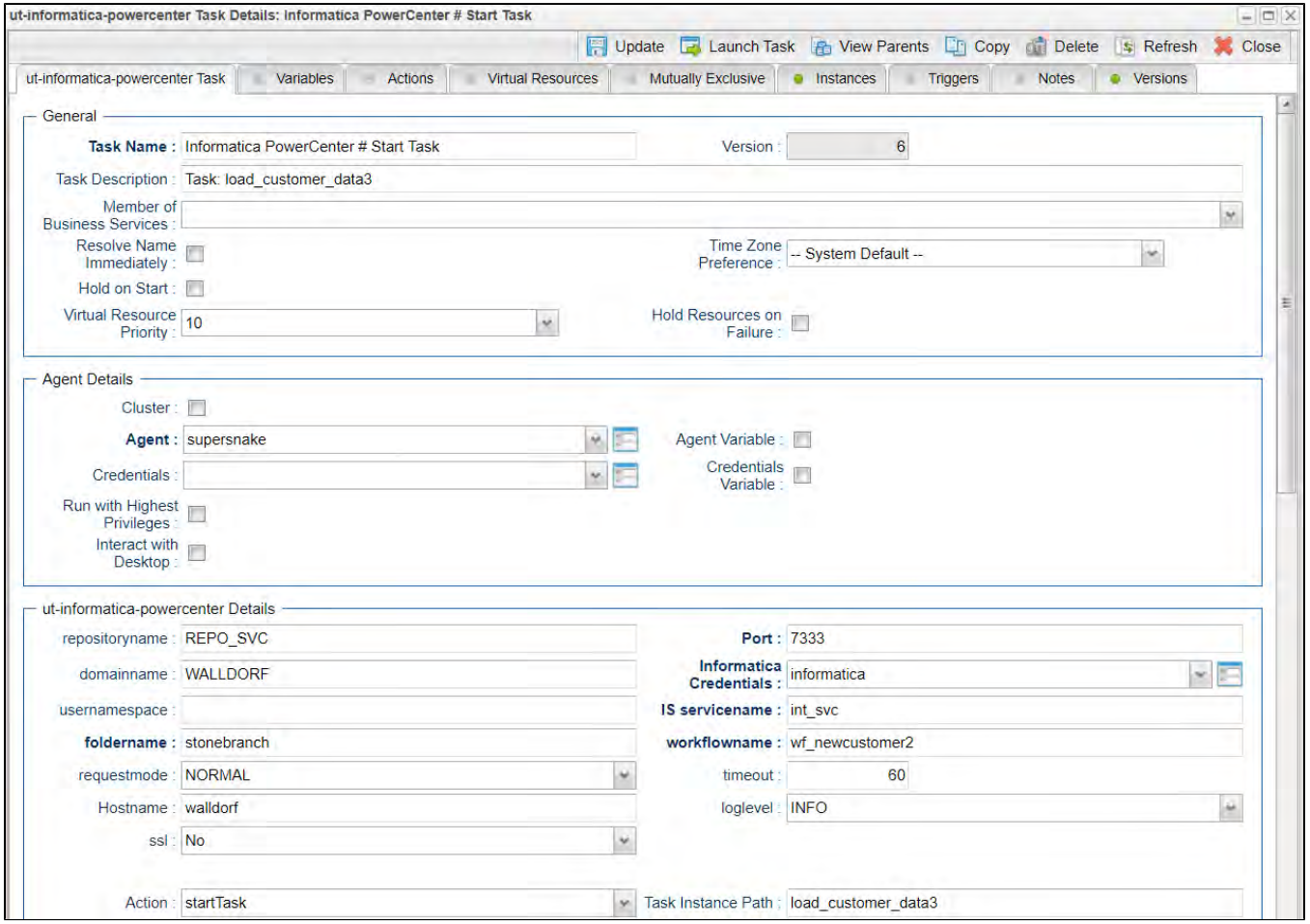
Field Descriptions for Informatica PowerCenter Universal Task

Field	Description
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL].
Agent	The Universal Agent, which runs the Python request module to call the PC “startWorkflow” SOAP Webservice. Both Linux and Windows Universal Agents are supported.
Repositoryname	Name of the Repository to log in to.
usernamespace	The security domain of the user account used to log in to the repository.

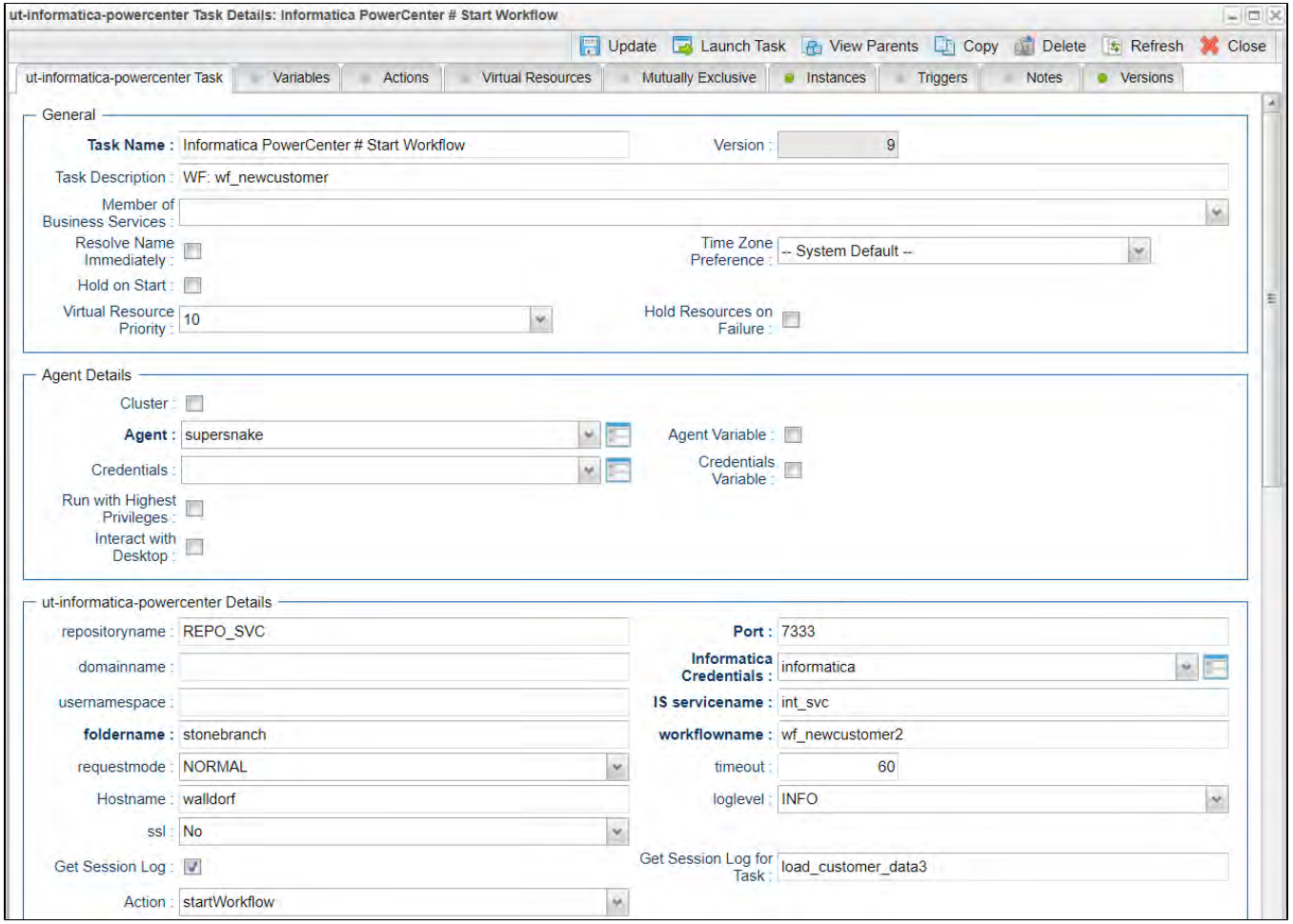
	Required if there is more than one security domain in the Informatica PowerCenter domain.
Domainname	Domain name for the Integration Service.
Foldername	Name of the folder containing the workflow.
requestmode	Indicates the recovery strategy for the session task: • NORMAL : Restarts a session without recovery. • RECOVERY : Recovers a session.
Informatica Credentials	Credentials for Informatica PowerCenter.
IS Servicename	Name of the Integration Service that runs the workflow.
workflowname	Name of the workflow to run.
Action	The following Actions can be selected: startWorkflow | startTask | startWorkflowFromTask • **startWorkflow**: Start a Task in an Informatica PowerCenter Workflow • **startTask**: Start an Informatica PowerCenter Workflow • **startWorkflowFromTask**: Start an Informatica PowerCenter Workflow from a given Task onwards Refer to Examples for an example screenshot of each action.
TaskinstancePath	Name and Path of the Task to start in the Workflow.
Get Session Log	If enabled, the Session Log of the Taskname provided in the Field ***Get Session Log for Task*** is written to the taskinstance Output. Note: In case of a Workflow with multiple task, only the session log of the task provided under the field taskinstancePath is written to the taskinstance Output.
Get Session Log for Task	The field is only visible if the choice field **Get Session Log** is enabled. Name and Path of the Task for which the session log should be retrieved.
Timeout	Maximum amount of time the Web Services Hub can take to process a request and generate a SOAP response before the request times out. If the Web Services Hub is unable to generate a response within the timeout period, it sends a fault message to the web service client and drops the connection. Default is 180 seconds. Minimum value is 60 seconds. If the operation does not complete within the timeout period, the operation fails.
Hostname	Web Services Hub host name.
Port	Web Services Hub port number.
SSL	Choose if you want to connect via *http* or *https* to your webservice hub. In the Power Center Administration GUI, you can look up the configuration by clicking on the webservice hub. For example, • *http*: *http://walldorf:7333/wsh* • *https*: *https://walldorf:10333/wsh*

Examples

Action Type: startTask

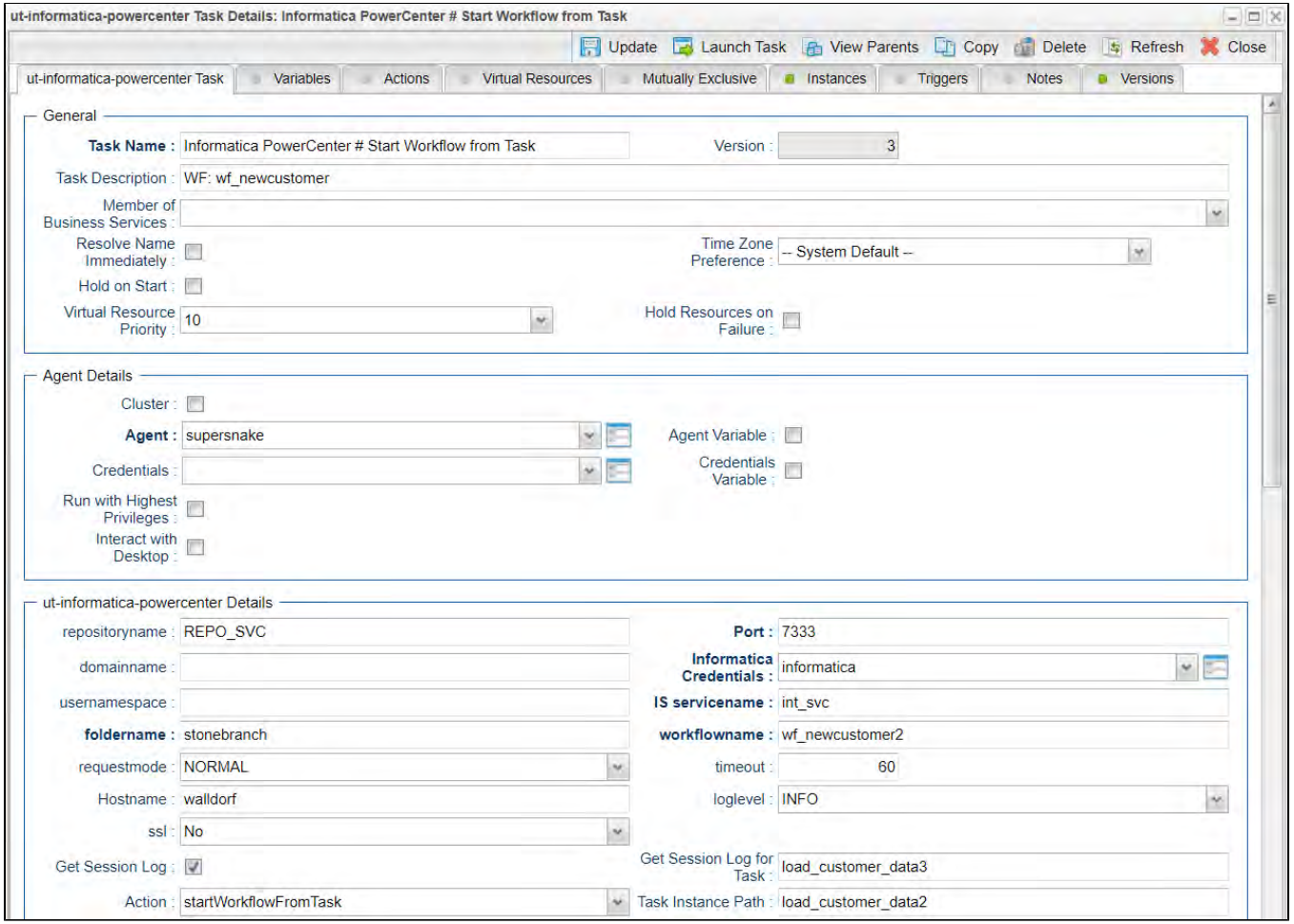


Action Type: startWorkflow



Action Type: startWorkflowFromTask

The following Task starts an Informatica PowerCenter Workflow from a specific Task onwards

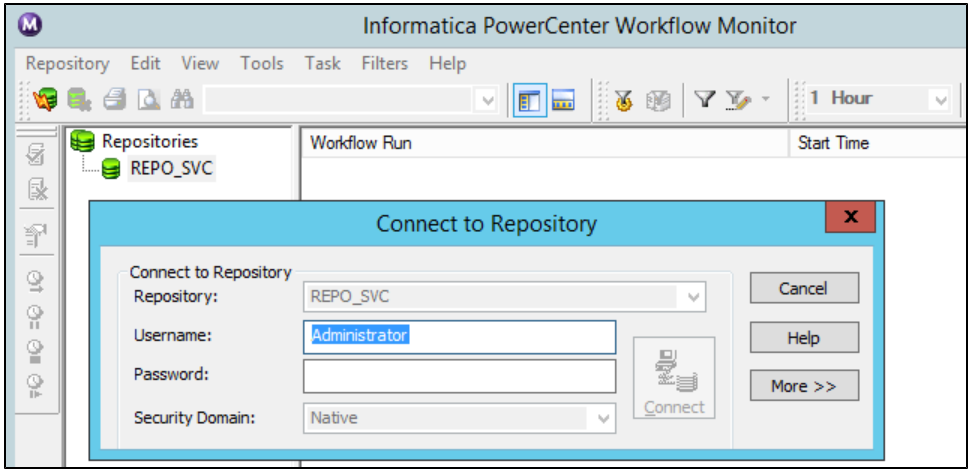


Verify Workflow execution in Informatica PowerCenter Workflow Monitor

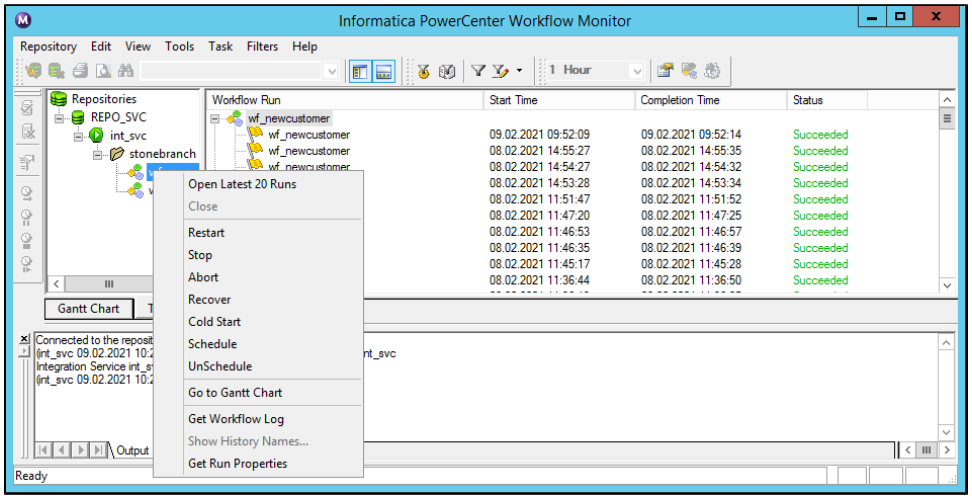
The following provides to non-Informatica PowerCenter Consultants with an Introduction on how to verify that a Workflow, which was started via the Universal Task for Informatica PowerCenter, has been successfully executed in PC.

How to Look Up a Workflow Instance in PC

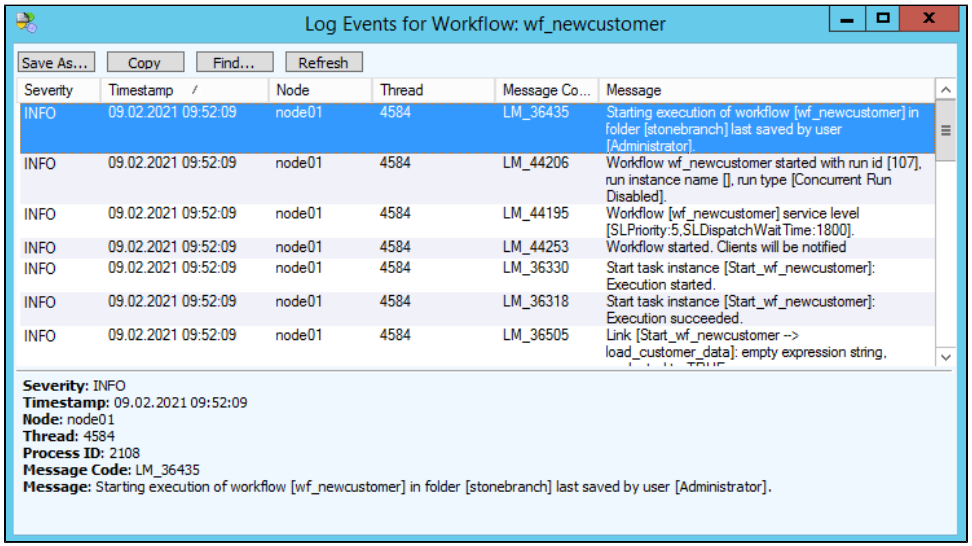
1. **Log-in to the Informatica PowerCenter Workflow Monitor.**



2. **Select your Workflow.**
   Browse to your Workflowname. In the following example, the Workflow is named: wf_newcustomer. On the right side of the screen, you can see all executed instances including their status; for example, *Succeeded*.



3. **Verify the Workflow Log**
   Right-clicking on the workflow will allow you to *Get the Workflow Log.*

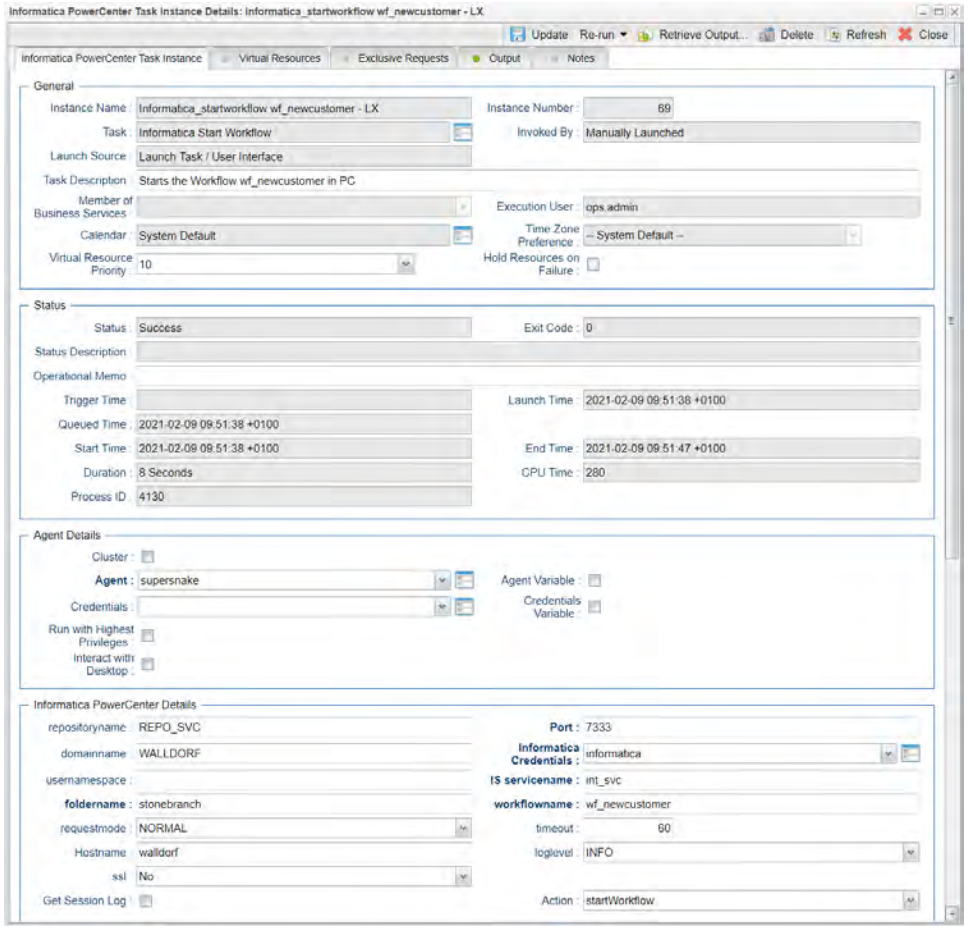


Note: The same Workflow log information is also available in the Universal Task Output.

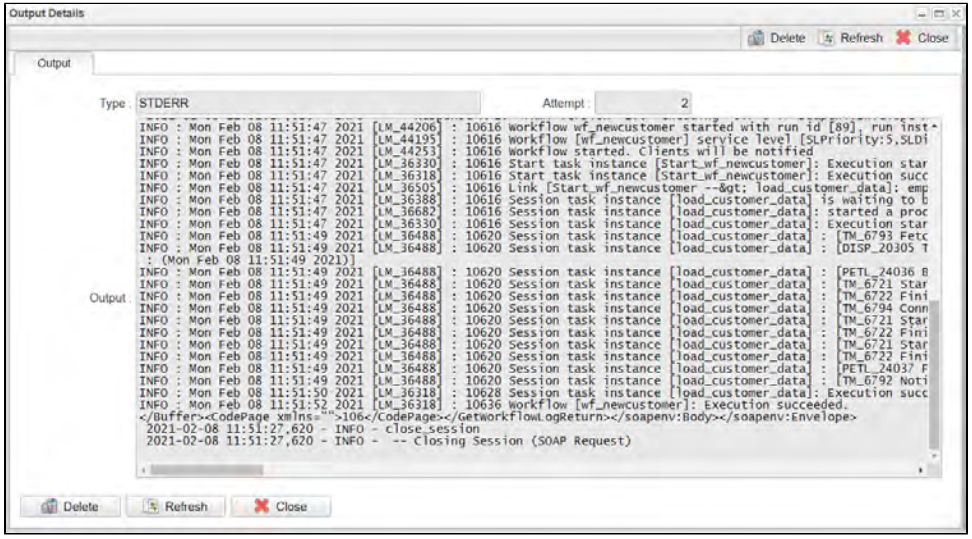
4. **Verify the Workflow Log in Universal Automation Center**

   All Log Information shown in the Informatica PowerCenter Workflow Monitor also is available in the Universal Controller Web-GUI in the Task Instance screen and Output.

   The following screenshot shows the Task Instance Screen:


Informatica PowerCenter Task Instance Details: Informatica_startworkflow wf_newcustomer - LX
Update
Re-run
Retrieve Output...
Delete
Refresh
Close
Informatica PowerCenter Task Instance
Virtual Resources
Exclusive Requests
Output
Notes
General
Instance Name : Informatica_startworkflow wf_newcustomer - LX
Instance Number : 69
Task : Informatica Start Workflow
Invoked By : Manually Launched
Launch Source : Launch Task / User Interface
Task Description : Starts the Workflow wf_newcustomer in PC
Member of Business Services :
Execution User : ops.admin
Calendar : System Default
Time Zone Preference : -- System Default --
Virtual Resource Priority : 10
Hold Resources on Failure :
Status
Status : Success
Exit Code : 0
Status Description :
Operational Memo :
Trigger Time :
Launch Time : 2021-02-09 09:51:38 +0100
Queued Time : 2021-02-09 09:51:38 +0100
Start Time : 2021-02-09 09:51:38 +0100
End Time : 2021-02-09 09:51:47 +0100
Duration : 8 Seconds
CPU Time : 280
Process ID : 4130
Agent Details
Cluster :
Agent : supersnake
Agent Variable :
Credentials :
Credentials Variable :
Run with Highest Privileges :
Interact with Desktop :
Informatica PowerCenter Details
repositoryname : REPO_SVC
Port : 7333
domainname : WALLDORF
Informatica Credentials : informatica
usernamespace :
IS servicename : int_svc
foldername : stonebranch
workflowname : wf_newcustomer
requestmode : NORMAL
timeout : 60
Hostname : walldorf
loglevel : INFO
ssl : No
Get Session Log :
Action : startWorkflow

The following screen shows the Log file in the Task Instance Output Screen:



Inter-Cloud Data Transfer

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Overview

The Inter-Cloud Data Transfer integration allows you to transfer data to, from, and between any of the major private and public cloud providers like AWS, Google Cloud, and Microsoft Azure.

It also supports the transfer of data to and from a Hadoop Distributed File System (HDFS) and to major cloud applications like OneDrive and SharePoint.

An advantage of using the Inter-Cloud Data Transfer integration over other approaches is that data is streamed from one object store to another without the need for intermediate storage.

Integrations with this solution package include:

- AWS S3
- Google Cloud
- Sharepoint
- Dropbox
- OneDrive
- Hadoop Distributed File Storage (HDFS)

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 7.0.0.0 or later is required.
- Universal Agent needs to be installed with python option (--python yes).

Software Requirements for Universal Controller

- Universal Controller 7.0.0.0 or later.

Software Requirements for the Application to be Scheduled

- Rclone: v1.55.1 or higher needs to be installed on server where the Universal Agent is installed.
- Rclone can be installed on Windows and Linux
- To install Rclone on Linux systems, run:

```
curl https://rclone.org/install.sh | sudo bash
```

Note: If the URL is not reachable from your server, the Linux installation can also be done from pre-compiled binary.

- To install Rclone on Linux system from a pre-compiled binary

Fetch and unpack

```
curl -O <https://downloads.rclone.org/rclone-current-linux-amd64.zip>
unzip rclone-current-linux-amd64.zip
cd rclone-*-linux-amd64
```

Copy binary file

```
sudo cp rclone /usr/bin/
sudo chown root:root /usr/bin/rclone
sudo chmod 755 /usr/bin/rclone
```

Install manpage

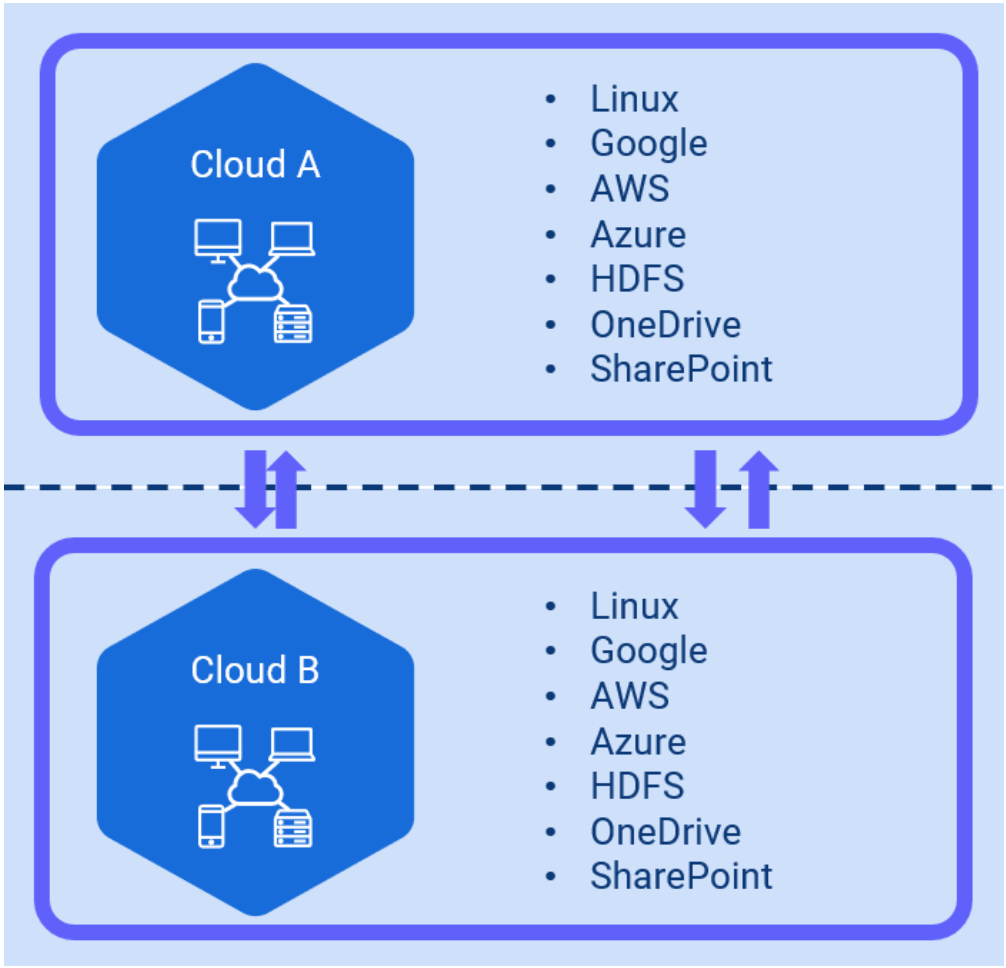
```
sudo mkdir -p /usr/local/share/man/man1
sudo cp rclone.1 /usr/local/share/man/man1/
sudo mandb
```

- To install Rclone on Windows systems:
    - Rclone is a Go program and comes as a single binary file.
    - Download the relevant binary here.
    - Extract the rclone or rclone.exe binary from the archive into a folder, which is in the windows path

Key Features

Some details about the Inter-Cloud Data Transfer Task:

- Transfer data to, from, and between any cloud provider
- Transfer between any major storage applications like SharePoint or Dropbox
- Transfer data to and from a Hadoop File System (HDFS)
- Data is streamed from one object store to another (no intermediate storage)
- Very Fast, if the object stores are in the same region
- Preserves always timestamps and verifies checksums
- Supports encryption, caching, compression, chunking
- Perform Dry-runs
- Regular Expression based include/exclude filter rules
- Supported actions are:
    - List objects, List directory,
    - Copy/ Move
    - Remove object / object store
    - Perform Dry-runs
    - Monitor object
    - Copy URL



Import Inter-Cloud Data Transfer Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Inter-Cloud Data Transfer Universal Tasks

To configure a new Inter-Cloud Data Transfer there are two steps required:

1. Configure the connection file
2. Create a new Inter-Cloud Data Transfer Task

Configure the Connection File

In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System; for example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage.

The connection file must be saved in the Universal Controller script library; for example, cloud2cloud.conf

Creation of the Connection File

The connection can be created manually by taking the sample connection file cloud2cloud.conf as template or interactively using the rclone config tool: rclone config.

If you do not want to show, in clear text, secret keys and password in the connection file, a Universal Controller credential could be used in the script. For example, if you want to encrypt the amazon s3 secret_access_key, you could set up a Universal Controller credential: AWS_SECRET_ACCESS_KEY_<D050320> and reference this credential in the script:

```
secret_access_key = ${_credentialPwd('AWS_SECRET_ACCESS_KEY_D050320')}
```

Considerations

Rclone supports connections to almost any storage system on the market:

Overview of Cloud Storage Systems

However, the current Universal Task has only been tested for the following storage types:

- LINUX
- AWS S3
- Azure Blob Storage
- Google GCS
- Microsoft One Drive incl. Share Point
- HDFS
- HTTPS URL

Note

If you want to connect to a different system, (for example, Dropbox), you should test this before taking it to production.

Create a New Inter-Cloud Data Transfer Task

For Universal Task Inter-Cloud Data Transfer, create a new task and enter the task-specific Details that were created in the Universal Template.

The following Actions are supported:

Action	Description
list directory	List directories; for example, • List object stores like S3 buckets, Azure container. • List OS directories from Linux, Windows, HDFS.
copy	Copy objects from source to target.
list objects	List objects in an OS directory or cloud object store.
move	Move objects from source to target.
remove-object	Remove objects in an OS directory or cloud object store.
remove-object-store	Remove an OS directory or cloud object store.
create-object-store	Create an OS directory or cloud object store.
copy-url	Download a URL's content and copy it to the destination without saving it in temporary storage.
monitor-object	Monitor a file or object in an OS directory or cloud object store.

In the following for each task action, the fields will be described and an example is provided.

Inter-Cloud Data Transfer Actions

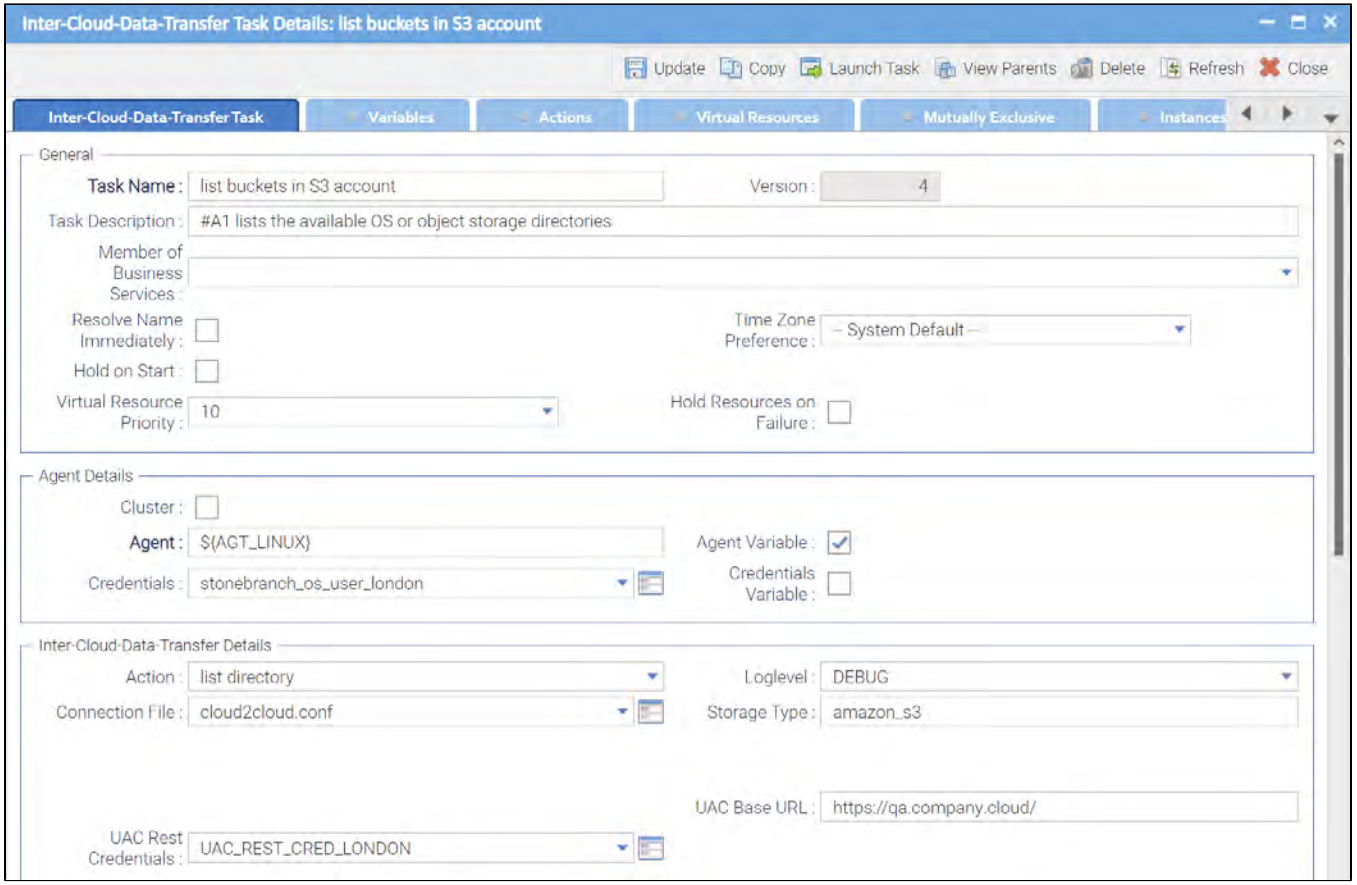
Action: list directory

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[**list directory**, copy, list objects, move, remove-object, remove-object-store, create-object-store, copy-url, monitor-object] list directories; for example, • List object stores like S3 buckets, Azure container. • List OS directories from Linux, Windows, HDFS.
Storage Type	Enter a storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Connection File	In the connection file you configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System ; for example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to section Configure the Connection File.

UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL; for example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example list all aws s3 buckets in the AWS account configured in the cloud2cloud.conf file.



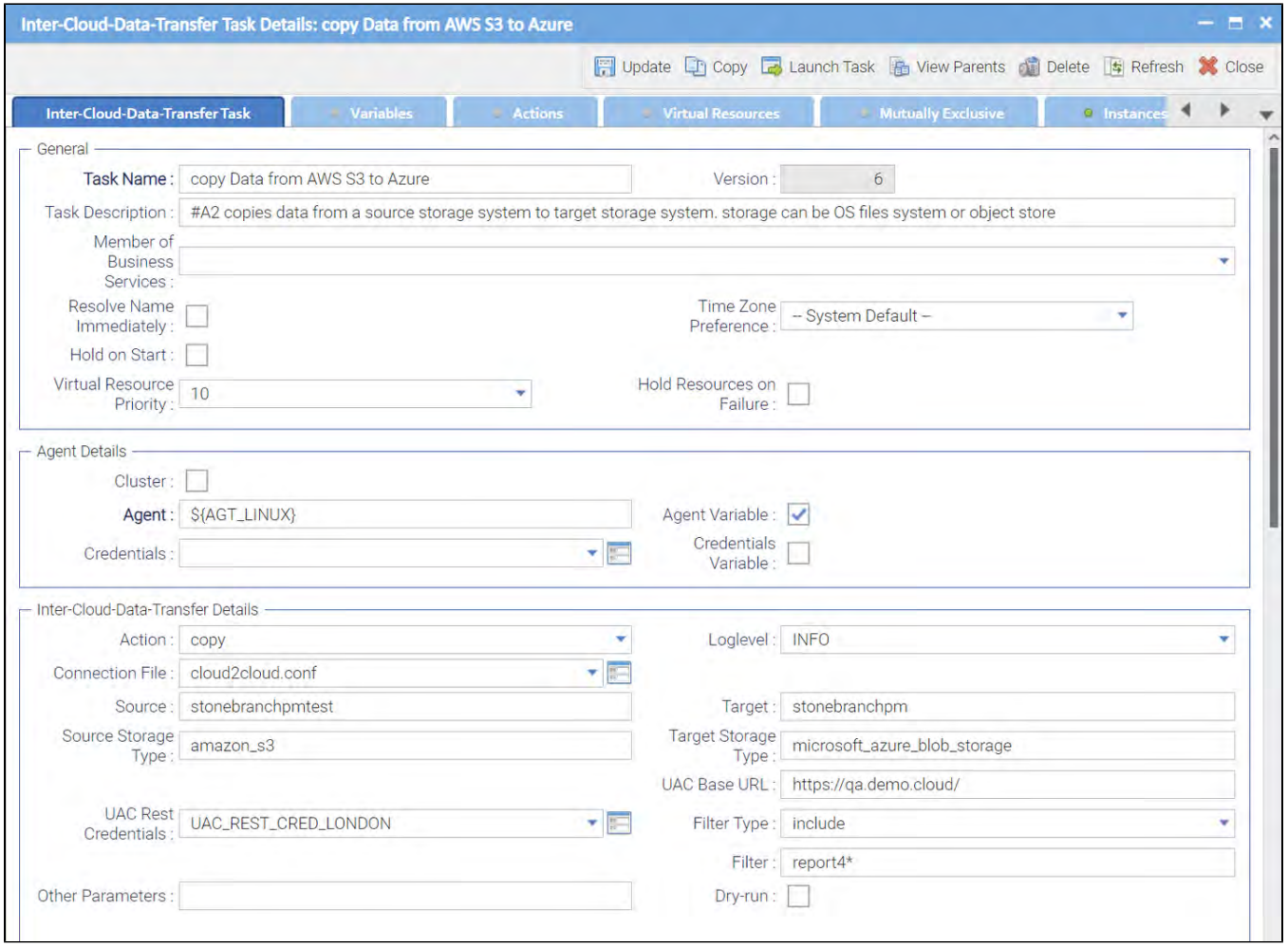
Action: copy

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.

Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, **copy**, list objects, move, remove-object, remove-object-store, create-object-store, copy-url, monitor-object] Copy objects from source to target.
Source	Enter a source storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Target	Enter a target storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Filter Type	[include, exclude, none] Define the type of filter to apply.
Filter	Provide the Patterns for matching file matching; for example, in a copy action: Filter Type: include Filter report[1-3].txt This means all reports with names matching `report1.txt` and `report2.txt` will be copied. For more examples on the filter matching pattern, refer to Rclone Filtering.
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags. Eample: To Skip files that are newer on the destination during a move or copy action, you could add the flag `--update`.
Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example copies all file starting with *report4* from the amazon s3 bucket *stonebranchpmtest* to the azure container *stonebranchpm.*



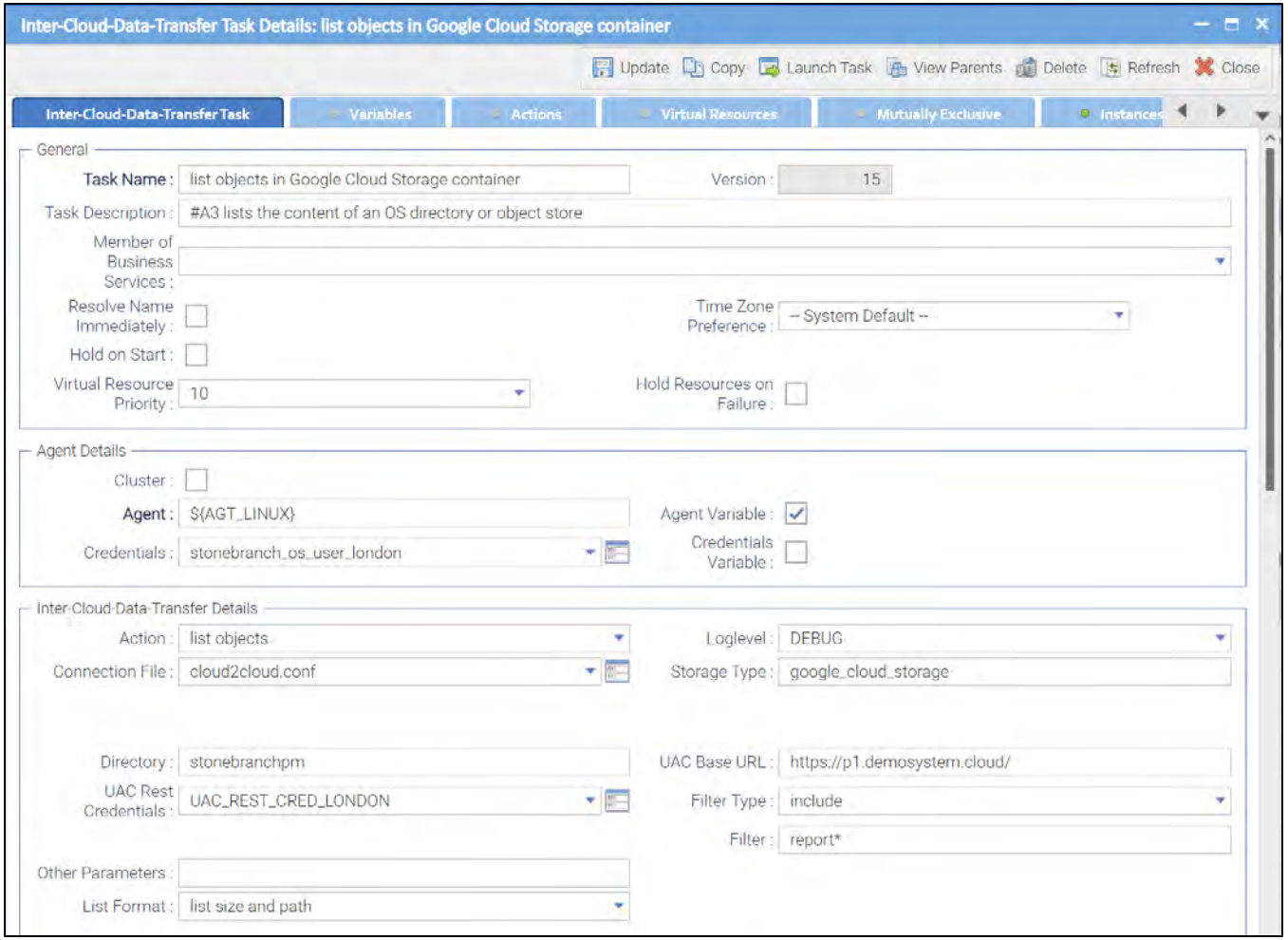
Action: list objects

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.

Action	[list directory, copy, **list objects**, move, remove-object, remove-object-store, create-object-store, copy-url, monitor-object] List objects in an OS directory or cloud object store.
Storage Type	Enter a Storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Filter Type	[include, exclude, none] Define the type of filter to apply.
Filter	Provide the Patterns for matching file matching; for example, in a copy action: Filter Type: include `Filter report[1-3].txt` means all reports with names matching `report1.txt` and `report2.txt` will be copied. For more examples on the filter matching pattern, refer to Rclone Filtering.
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags. Example: To Skip files that are newer on the destination during a move or copy action, you could add the flag `--update`.
Directory	Name of the directory you want to list the files in. For example, Directory: *stonebranchpm/out* would mean to list all objects in the bucket *stonebranchpm* folder *out.*
List Format	[list size and path, list modification time, size and path, list objects and directories, list objects and directories (Json)] The Choice box specifies how the output should be formatted.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example lists all objects starting with *report* in the s3 bucket *stonebranchpm*.



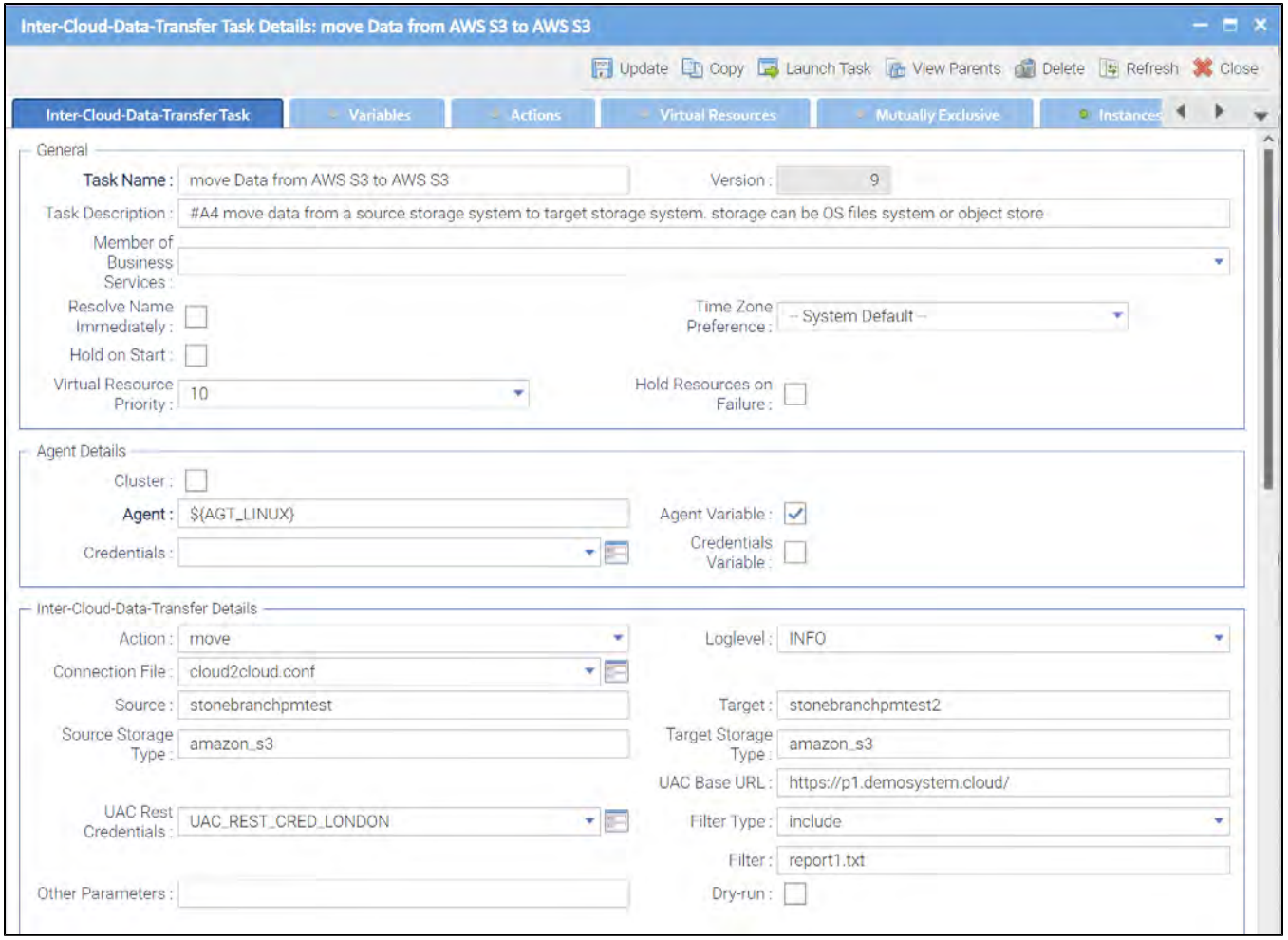
Action: move

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, copy, list objects, **move**, remove-object, remove-object-store, create-object-store, copy-url, monitor-object] Move objects from source to target.
Source	Enter a source storage Type name as defined in the Connection File; for example,

	amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Target	Enter a target storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Filter Type	[include, exclude, none] Define the type of filter to apply.
Filter	Provide the Patterns for matching file matching; for example, in a copy action: Filter Type: include `Filter report[1-3].txt` means all reports with names matching `report1.txt` and `report2.txt` will be copied. For more examples on the filter matching pattern, refer to Rclone Filtering.
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags Example: To Skip files that are newer on the destination during a move or copy action, you could add the flag `--update`.
Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example moves the object *report1.txt* from the source s3 bucket *stonebranchpmtest* to the target s3 bucket *stonebranchpmtest2*.



Action: remove-object

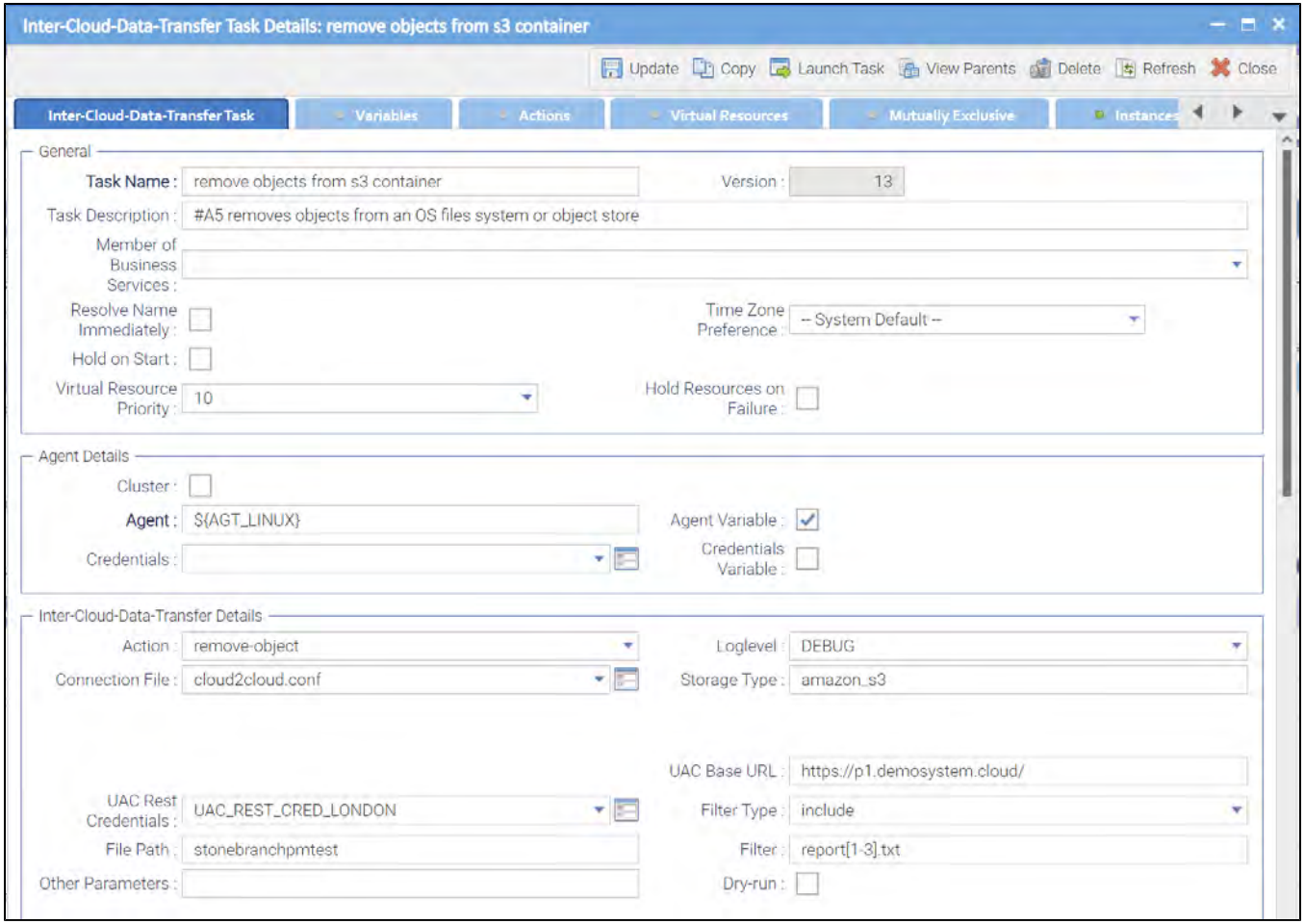
Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, copy, list objects, move, **remove-object**, remove-object-store, create-object-store, copy-url, monitor-object] Remove objects in an OS directory or cloud object store.
	Enter a storage Type name as defined in the Connection File; for example,

Storage Type	amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
File Path	Path to the directory in which you want to remove the objects. For example: File Path: *stonebranchpmtest* Filter: report[1-3].txt This removes all s3 objects matching the filter: report[1-3].txt(report1.txt, report2.txt and report3.txt) from the s3 bucket *stonebranchpmtest*.
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags.
Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example removes all s3 objects matching the filter: report[1-3].txt

(report1.txt, report2.txt and report3.txt) from the s3 bucket *stonebranchpmtest*.



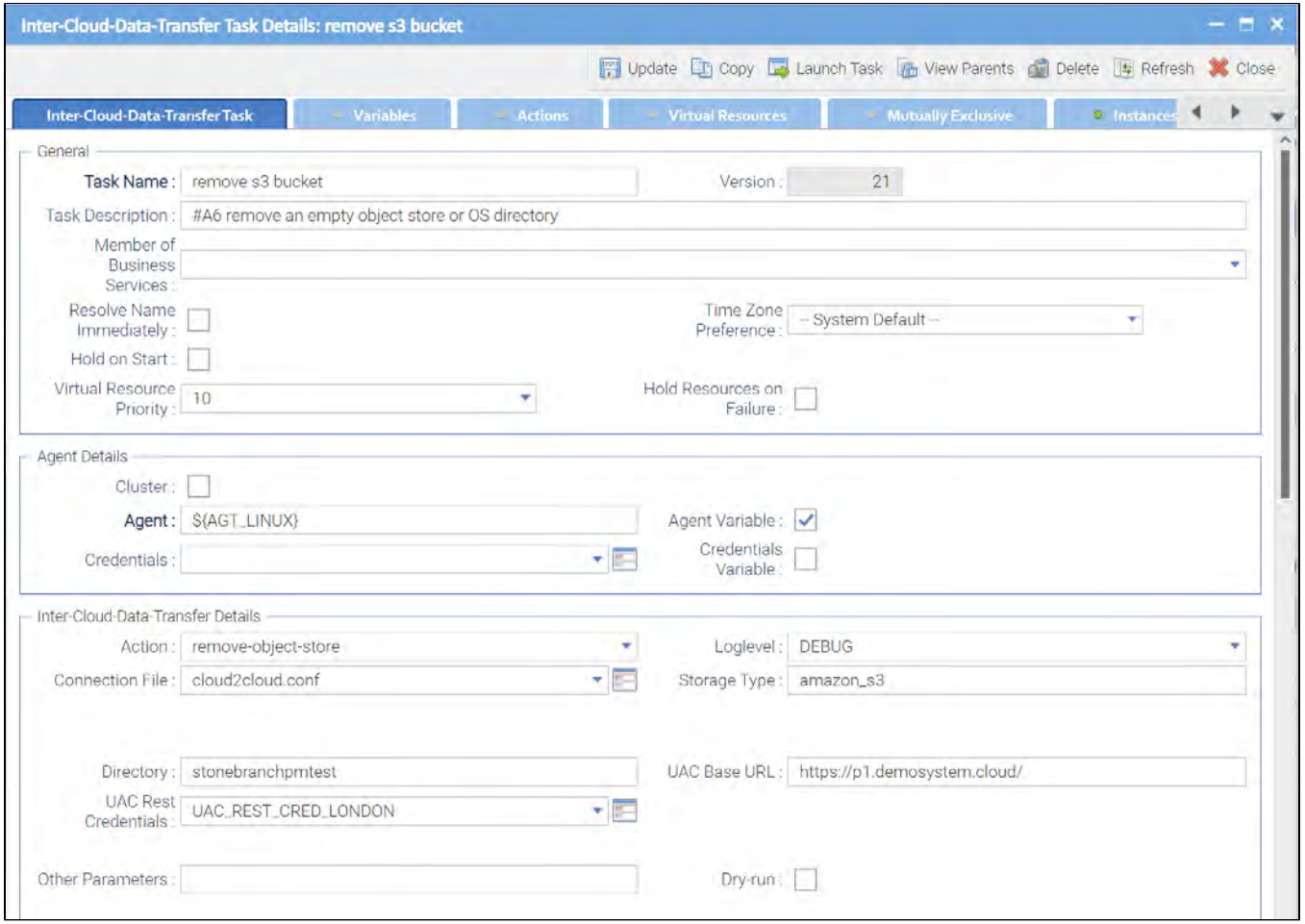
Action: remove-object-store

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, copy, list objects, move, remove-object, **remove-object-store**, create-object-store, copy-url, monitor-object] Remove an OS directory or cloud object store.
Storage Type	Enter a storage Type name as defined in the Connection File; for example,

	amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Directory	Name of the directory you want to list the files in. For example, Directory: *stonebranchpm/out* would mean to list all objects in the bucket *stonebranchpm* folder *out.*
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags.
Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example removes the s3 object store *stonebranchpmtest*.



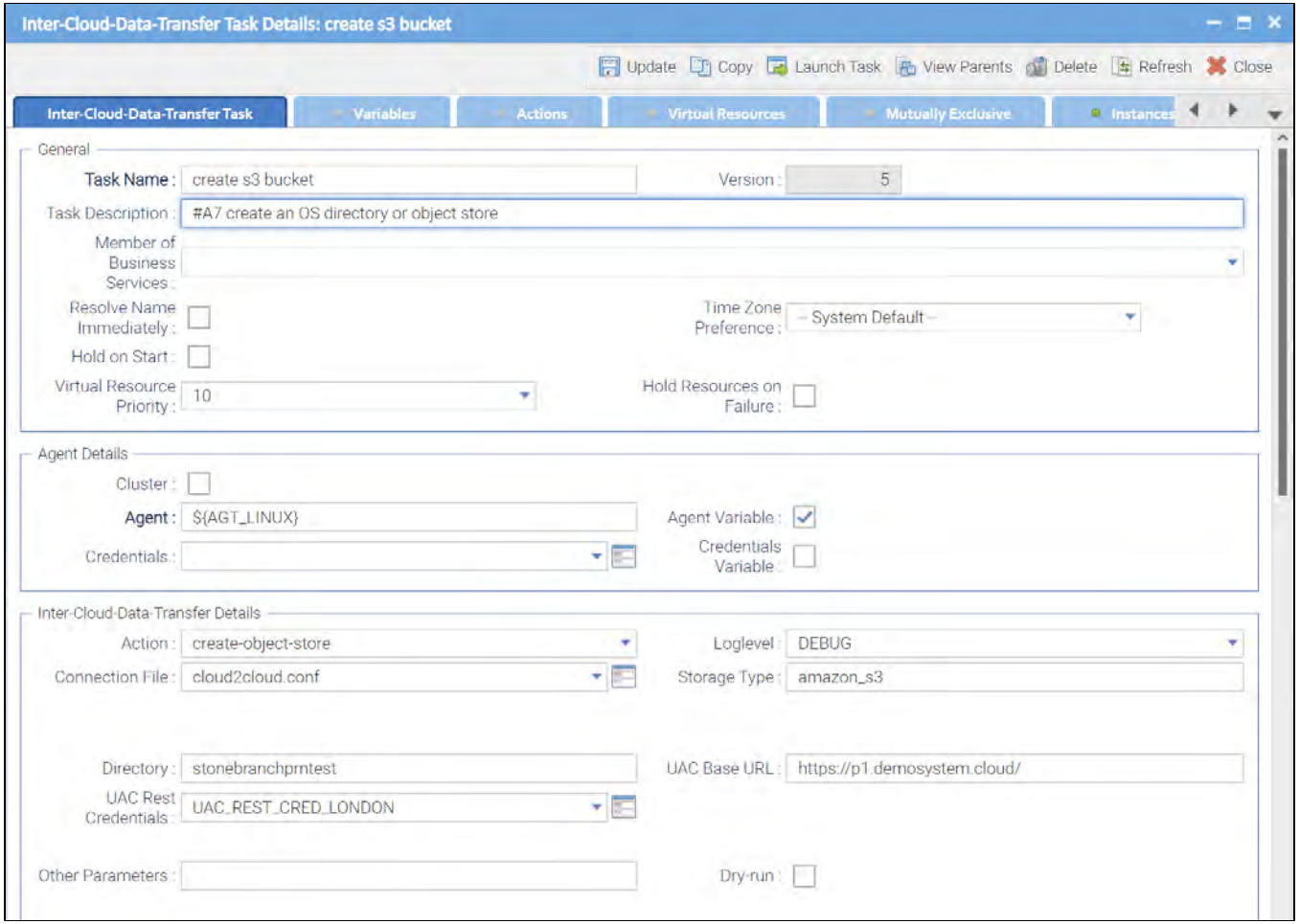
Action: create-object-store

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, copy, list objects, move, remove-object, remove-object-store, **create-object-store**, copy-url, monitor-object] Create an OS directory or cloud object store.
Storage Type	Enter a storage Type name as defined in the Connection File; for example,

	amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Directory	Name of the directory you want to create. The directory can be an object store or a file system OS directory. For example, Directory: *stonebranchpmtest* would create the bucket *stonebranchpmtest.*
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags.
Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example creates the s3 bucket stonebranchpmtest.



Action: copy-url

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, copy, list objects, move, remove-object, remove-object-store, create-object-store, **copy-url**, monitor-object] Download a URL's content and copy it to the destination without saving it in temporary storage.
Source	Enter a source storage Type name as defined in the Connection File; for example,

	amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Target	Enter a target storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux .. For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.

Other Parameters	This field can be used to apply additional flag parameters to the selected action.
	
	For a list of all possible flags, refer to Global Flags.
	
	Useful parameters for the copy-url command:
	
	`-a, --auto-filename Get the file name from the URL and use it for destination file path`
	`-h, --help help for copyurl`
	`--no-clobber Prevent overwriting file with same name`
	`-p, --print-filename Print the resulting name from --auto-filename`
	`--stdout Write the output to stdout rather than a file`

Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

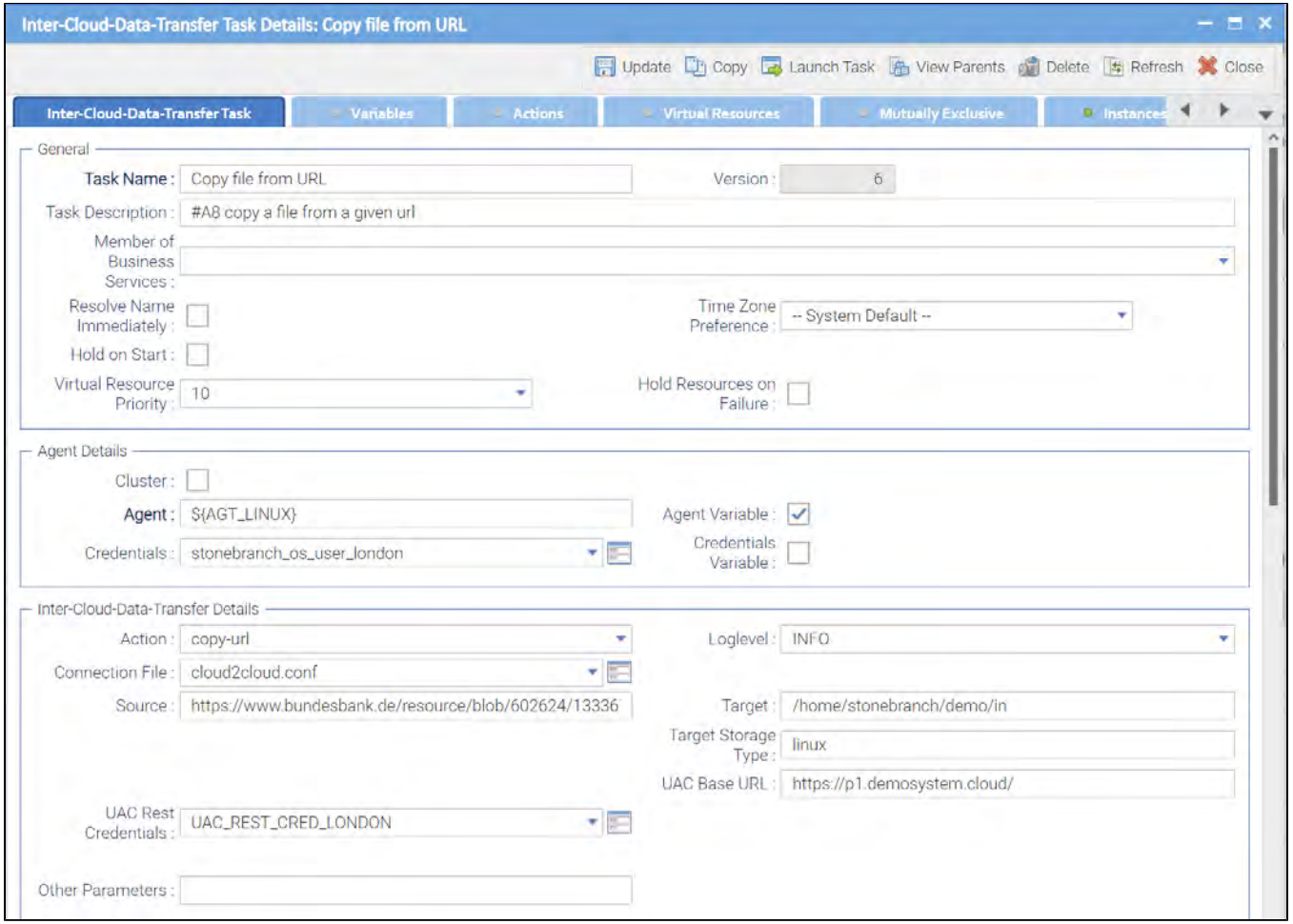
Example

The following example downloads a PDF file:

From the webaddress: https://www.bundesbank.de/resource/../blz-loeschungen-aktuell-data.pdf

Tto the linux folder: */home/stonebranch/demo/in*

The linux folder is located on the server where the *Agent${AGT_LINUX}* runs.



Action: monitor-object

Field	Description
Agent	Linux or Windows Universal Agent to execute the Rclone command line.
Agent Cluster	Optional Agent Cluster for load balancing.
Action	[list directory, copy, list objects, **move**, remove-object, remove-object-store, create-object-store, copy-url, monitor-object] Monitor a file or object in an OS directory or cloud object store.
Storage Type	Enter a storage Type name as defined in the Connection File; for example, amazon_s3, microsoft_azure_blob_storage, hdfs, onedrive, linux ..

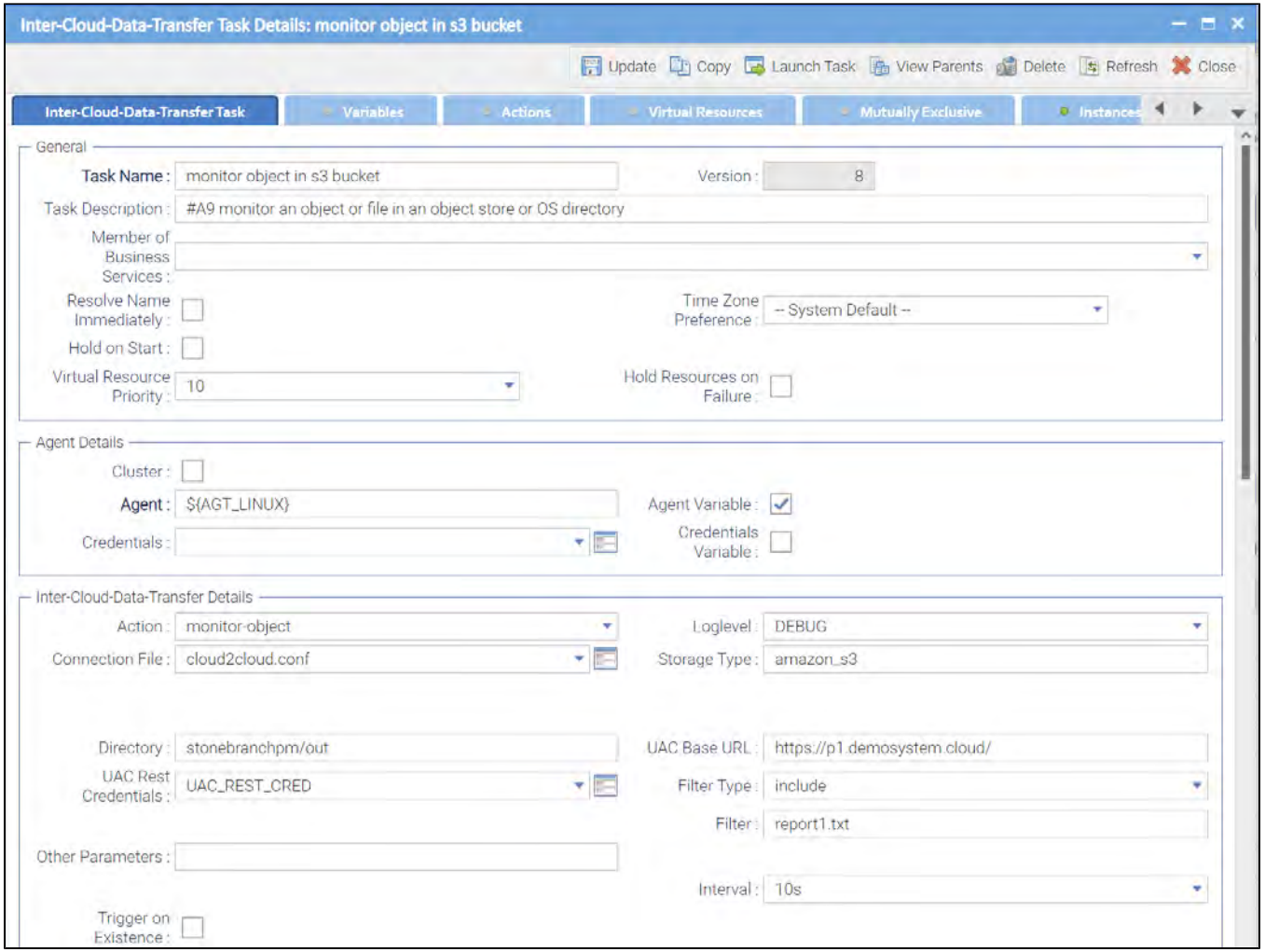
	For a list of all possible storage types, refer to Overview of Cloud Storage Systems.
Directory	Name of the directory to scan for the files to monitor. The directory can be an object store or a file system OS directory. For example: Directory: *stonebranchpm/out* Filter: *report1.txt* This would monitor in the s3 bucket folder *stonebranchpm/out* for the object *report1.txt.*
Connection File	In the connection file, configure all required Parameters and Credentials to connect to the Source and Target Cloud Storage System. For example, if you want to transfer a file from AWS S3 to Azure Blob Storage, you must configure the connection Parameters for AWS S3 and Azure Blob Storage. For details on how to configure the Connection File, refer to Configure the Connection File.
Filter Type	[include, exclude, none] Define the type of filter to apply.
Filter	Provide the Patterns for matching file matching; for example, in a copy action: Filter Type: include Filter report[1-3].txt This means all reports with names matching `report1.txt` and `report2.txt` will be copied. For more examples on the filter matching pattern, refer to Rclone Filtering.
Other Parameters	This field can be used to apply additional flag parameters to the selected action. For a list of all possible flags, refer to Global Flags. Example: To Skip files that are newer on the destination during a move or copy action, you could add the flag `--update`.
Dry-run	[checked , unchecked] Do a trial run with no permanent changes.
Trigger on Existence	[checked , unchecked] If checked, the monitor goes to success even if the file already exists when it was started.
Interval	[10s, 60s, 180s] Monitor interval to check of the file(s) in the configured directory. For example, Interval: 60s, would be mean that every 60s, the task checks if the file exits in the scan directory.
UAC Rest Credentials	Universal Controller Rest API Credentials
UAC Base URL	Universal Controller URL

	For example, https://192.168.88.40/uc
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example

The following example monitors s3 bucket folder *stonebranchpm/out* for the object *report1.txt.*

If the object is found, the monitor goes to success.



Jenkins Integration

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This integration improves the functionality of Jenkins when orchestrated from Universal Controller. It encourages collaboration by enabling the well-controlled and automated deployment of applications over to the operations side.

Overview

- UAC communicates with Jenkins through the Python Jenkins module.
- Jenkins can make REST-API calls to the Universal Controller to trigger any task or workflow.
- This task can trigger or start an existing build job in Jenkins. Universal controller will monitor the build execution in Jenkins until completion, then send the build results to the Controller. With this task, users can create a build job in Jenkins from the Controller. Any Jenkins build job definitions in XML will be stored centrally in the Controller.
- This task offers the functionality to fetch the Jenkins job build information and list running build info in Jenkins from Universal Controller.
- Enable/disable Jenkins jobs and nodes and delete/copy/rename Jenkins jobs from Universal Controller.
- When users list the Installed plugin in Jenkins, a plugin install can be triggered from the Universal Controller.
- Set the next build info for Jenkins build jobs.

Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against a Jenkins instance.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests
    - Jenkins

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.6 and later with python options installed
- Universal Agent for Linux Version 6.6 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with the following Jenkins versions:

- 2.1
- 2.2

Technical Considerations

This Universal Task uses the Python Jenkins Module functions (https://python-jenkins.readthedocs.io/en/latest/) to make REST API calls to Jenkins server.

Jenkins Integration Key Features

Feature	Description

Get Jenkins Jobs build information	Get a Jenkins job build information details
Create a Jenkins Build Job	Allows you to create a Jenkins build job by passing the XML configuration script from Universal Controller
Get Last build number	Get the latest build number for a job
Set Next build number	Allows to set the next build number for the Jenkins job
Trigger or start an existing build job	Trigger an existing Jenkins build job in Jenkins and Universal Controller will monitor the build execution in Jenkins until completion, then send the build results to the Controller
List running build info	Allows to list the build job that are running
Enable/Disable Jenkins Job	function to enable or disable a Jenkins build job
Delete/Copy/Rename Jenkins Job	function to copy or delete or rename a Jenkins Job
Get Console output for a build	Get the output of a Jenkins Build job
Enable / disable Nodes in Jenkins	Enable or Disable a Jenkins node
Get all installed plugins info	List all the plugins that are installed in a Jenkins server
Install a plugin for Jenkins environment	Allows to install a specific Jenkins plugin

Import Jenkins Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Jenkins Integration Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

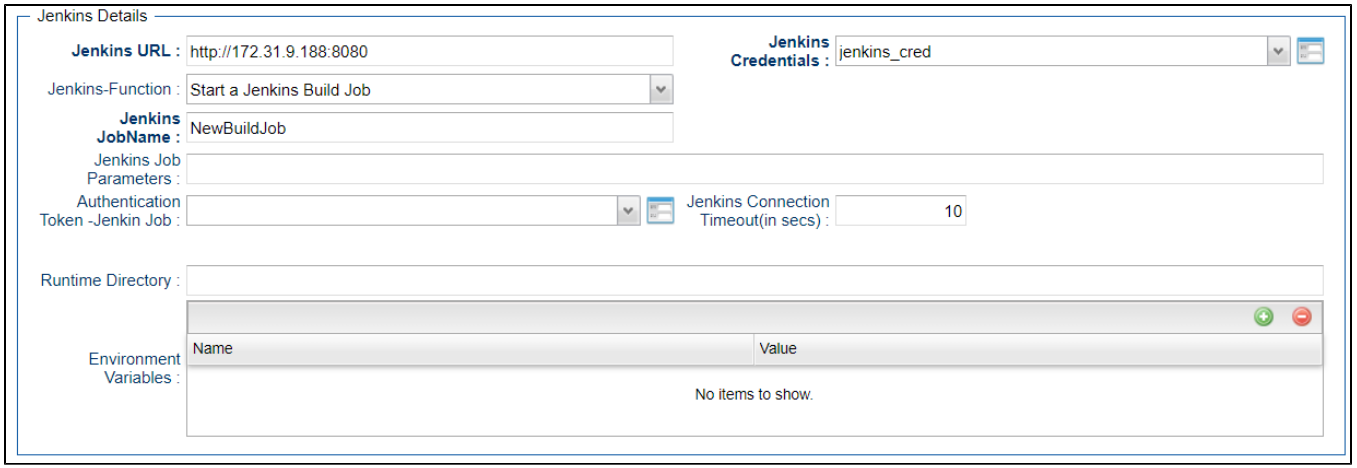
Field Descriptions for Jenkins Integration Universal Task

Field	Description
Jenkins Function	Select the desired function you would need to perform in Jenkins
Jenkins URL	URL of the Jenkins server, to make api calls
Jenkins Credentials	Jenkins User credentials or auth token to authenticate API calls
Jenkins Function	Select the desired function you would need to perform in Jenkins

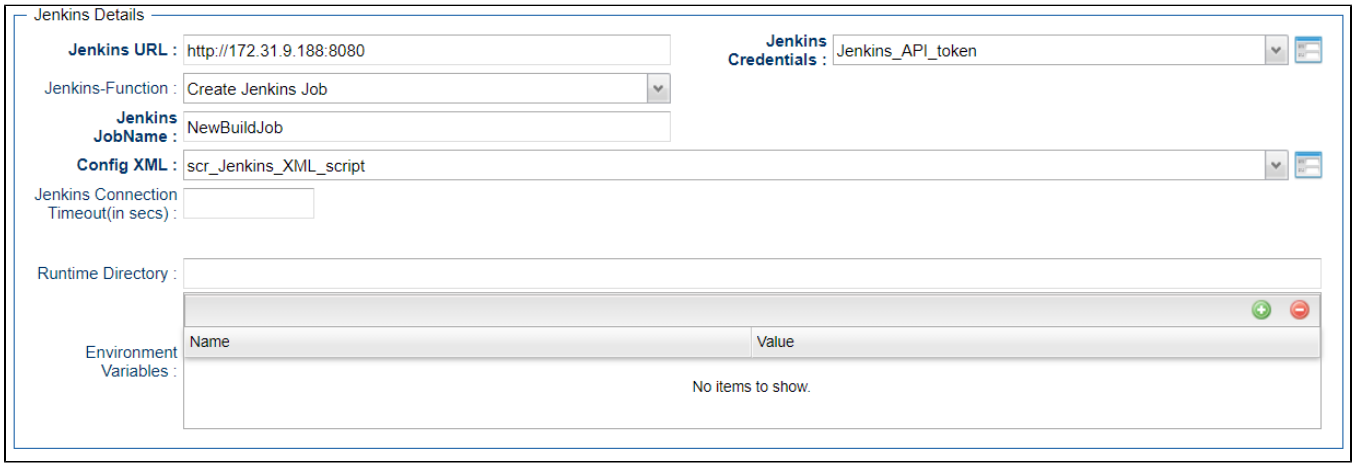
Jenkins Job Name	Name of the Jenkins Job Name
Jenkins Job Parameters	The parameters that would need to passed along for starting a Jenkins Build
Job Token	If the Jenkins can be triggered remotely by using a job token then pass on the Job token parameter
Config XML	Provide the XML script for the creation of new job in Jenkins
Rename Job	specify the job name that needs to be renamed
From jobname	give the existing from where its needs to be copied eg: demo_job or folder/demo_job
To Job Name	copy/rename to a new Job Name eg: demo_job or folder/demo_job
Enable Node	specify the node name that needs to be enabled
Disable Node	specify the node name that needs to be disabled
Delete Job	Specify the job that needs to be deleted in Jenkins
Next Build Number(+ Last Build Number)	should be greater than the last builder otherwise jenkins will ignore the request
Enable Job	Specify the job name that needs to be enabled
Disable Job	specify the job that needs to be disabled
Jenkins Build Number	Jenkins build number for the job(int)
Jenkins Connection Timeout(in secs)	Specify the connection time out interval in seconds
Jenkins Plugin Name	Provide the Jenkin Plugin short name to be installed

Examples for Jenkins Integration Universal Tasks

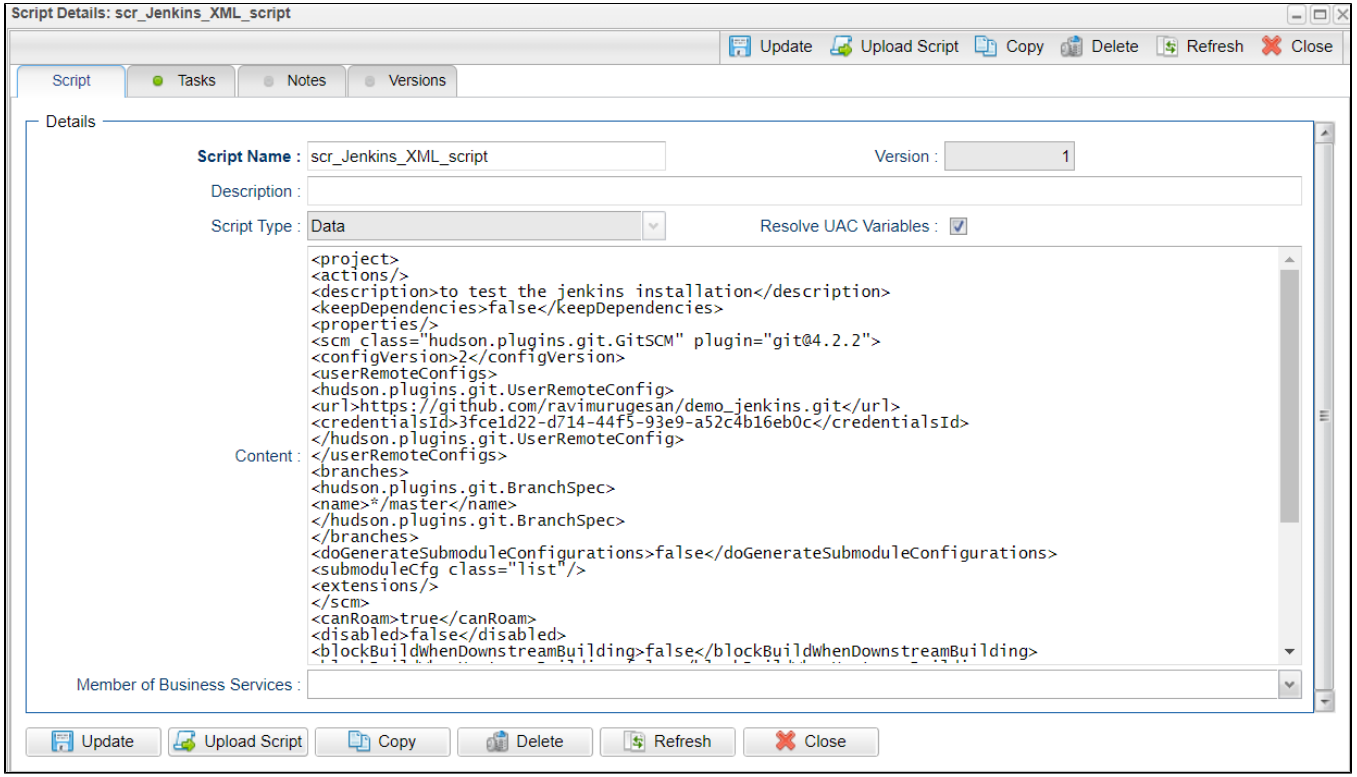
Start a Jenkins Build



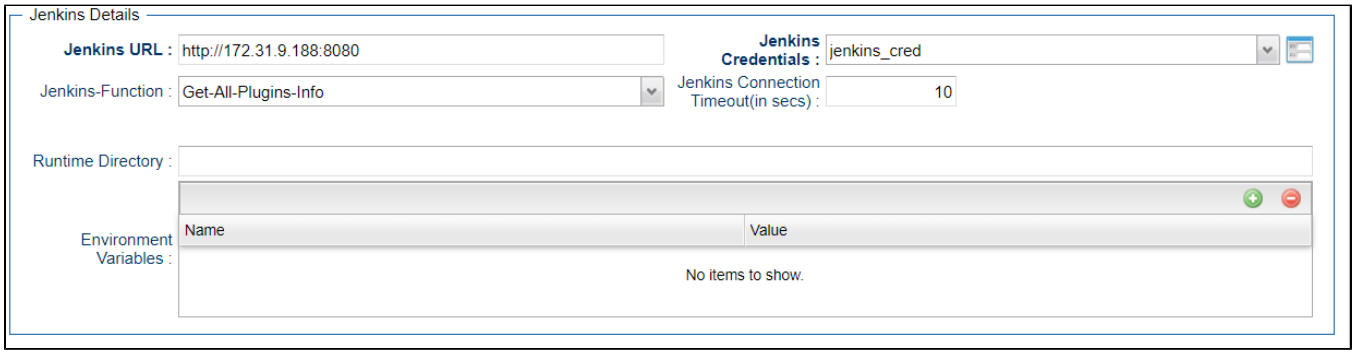
Create a Jenkins Job



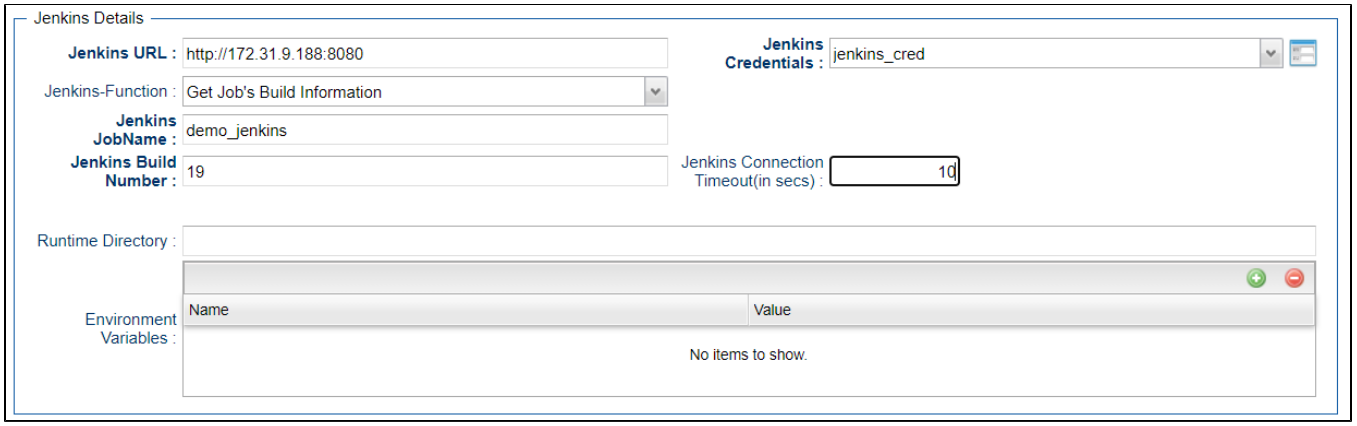
New Jenkins Job XML Configuration



Get Plugins List



Job Build Information



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

JSCAPE MFT

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This integration provides UAC customers the ability to manage and integrate their JSCAPE Managed File Transfer Server processes within their UAC automation processes and workflows.

Overview

This integration delivers a the JSCAPE MFT Universal Template that allows UAC customers to build Tasks to perform the following JSCAPE Managed File Transfer Server functions:

- PGP Encrypt
- PGP Decrypt
- Run a UDM Gateway Trigger
- SFTP File Upload
- SFTP File Download
- Trading Partner File Upload
- Trading Partner File Download
- Trading Partner File Upload using a Regex or Generic Filename Pattern
- Trading Partner File Download using a Regex or Generic Filename Pattern

Software Requirements

This integration requires a Universal Agent and a Python runtime to execute the Universal Task.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.7.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests version 2.22.0

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 7.0.0.0 and later with python options installed
- Universal Agent for Linux Version 7.0.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 7.0.0.0 and later

Software Requirements for the Application to be Scheduled

- JSCAPE Managed File Transfer Server version 12 and higher.

Integration Description

Integration Flow

The JSCAPE MFT Universal Task performs the following:

1. Login to JSCAPE MFT Server
2. Check JSCAPE MFT Server Version
3. Create Temporary JSCAPE MFT Server Trigger with a generated unique name to perform the desired action. Except Run Trigger which runs the requested Trigger definition directly.
4. Run JSCAPE MFT Server Trigger.

5. Get JSCAPE MFT Server Trigger Status
6. If Successful, delete the Temporary Trigger definition (Except Run Trigger)
7. Retrieve Trigger Log

Output Only Fields

The JSCAPE MFT Universal Task Instances will provide the following display information:

Fieldname	Description
Trigger Name	Returns the generated temporary trigger name, except for Run Trigger when the Trigger name is displayed.
Trigger Status	Returns the current status of the Trigger.
Trigger ID	Returns the JSCAPE MFT Server Trigger Process ID.
MFT Server Version	Returns the version of the JSCAPE MFT Server.

Dynamic Commands

The following Universal Task Instance specific commands are available:

Command Name	Allowed Task Instance Status'	Description
Cancel (Kill) Trigger	RUNNING	Issues the JSCAPE MFT Server Kill command against the running Trigger.
Cleanup (Delete) Temporary Trigger	FAILED	Deletes the temporary Trigger. Use this option if you do not want to rerun a failed temporary Trigger, failed triggers are not automatically deleted to facilitate a rerun of the Trigger. It is recommended to Force Finish the failed Universal Task instance after deleting the temporary trigger. This command is not effective for Run Trigger.

Task Instance Rerun Behavior

Performing a rerun against a JSCAPE MFT Universal Task instance behaves as follows.

- When the task is rerun and no JSCAPE MFT Server Trigger Process ID is available a new Trigger will be created and run.
- When the task is rerun and a JSCAPE MFT Server Trigger Process ID is available a JSCAPE MFT Server Rerun command will be issued against the existing Trigger. Note that for the Run Trigger Function the behavior of the JSCAPE MFT Server Rerun command can be impacted by the “On Rerun Restart All Actions” option in the Task instance definition.

Import Universal Template

To use the Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature, check that the Resolvable Credentials Permitted system property has been set to true. For more information about Resolvable Credentials click here.
2. Download the provided ZIP file.
3. In the Universal Controller UI, select Administration >Configuration > Universal Templates to display the current list of Universal Templates.

4. Click Import Template.
5. Select the template ZIP file and Import.

When the template has been imported successfully, the Universal Template will appear on the list, refresh your Navigation Tree to see these tasks in the Automation Center Menu.

Configure Universal Task

Create JSCAPE MFT Tasks as required.

Field Descriptions for JSCAPE MFT Universal Task

Function = PGPEncrypt

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **PGPEncrypt** Function.
Encrypted File Name	The name of the target encrypted file.
Plain Text File Name	The name of the source plaintext file.
PGP Key	Select the PGP Key to use for encryption.
Delete Source File	Specify if the source plaintext file is to be deleted.
Compress	Specify if the target encrypted file is to be compressed.

Function = PGPDecrypt

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **PGPDecrypt** Function.
Encrypted File Name	The name of the source encrypted file.
Plain Text File Name	The name of the target plaintext file
PGP Key	Select the PGP Key to use for encryption.

Delete Source File	Specify if the source encrypted file is to be deleted.

Function = RunTrigger

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **RunTrigger** Function.
Trigger Name	Select the required JSCAPE MFT Server Trigger.
On Rerun Restart All Actions	Check this option to restart Trigger form the beginning when performing a rerun of the task instance, leave unchecked to start from the failed or cancelled Trigger action.

Function = SFTPUpload

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **SFTPUpload** Function.
SFTP Host	Specify the remote SFTP server Host Name or IP address.
SFTP Port	Specify the remote SFTP server port.
SFTP Credential	Select the Credential definition to access the remote SFTP server.
Local File Name	Specify the location of the Local File.
Remote Directory	Specify the remote SFTP Server Directory.
Transfer Mode	Specify the Transfer Mode.
Overwrite if File Exists	Check to overwrite destination file(s) if it exists.
Retry Limit	Specify the Maximum Retry Attempts.
Retry Interval	Specify the Interval in Seconds Between Retries.

Function = SFTPDownload

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443

MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **SFTPDownload** Function.
SFTP Host	Specify the remote SFTP server Host Name or IP address.
SFTP Port	Specify the remote SFTP server port.
SFTP Credential	Select the Credential definition to access the remote SFTP server.
Local File Name	Specify the location of the Local File.
Remote File Name	Specify the remote SFTP File.
Transfer Mode	Specify the Transfer Mode.
Overwrite if File Exists	Check to overwrite destination file(s) if it exists.
Retry Limit	Specify the Maximum Retry Attempts.
Retry Interval	Specify the Interval in Seconds Between Retries.

Function = TradingPartnerUpload

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **TradingPartnerUpload** Function.
Trading Partner	Select the JSCAPE MFT Server Trading Partner definition.
Local File Name	Specify the location of the Local File.
Remote Directory	Specify the remote Trading Partner Directory.
Transfer Mode	Specify the Transfer Mode.
Passive Mode	Check to use passive mode.
Retry Limit	Specify the Maximum Retry Attempts.
Retry Interval	Specify the Interval in Seconds Between Retries.

Function = TradingPartnerDownload

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443

MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **TradingPartnerDownload** Function.
Trading Partner	Select the JSCAPE MFT Server Trading Partner definition.
Local File Name	Specify the location of the Local File.
Remote File Name	Specify the remote Trading Partner File.
Transfer Mode	Specify the Transfer Mode.
Passive Mode	Check to use passive mode.
Retry Limit	Specify the Maximum Retry Attempts.
Retry Interval	Specify the Interval in Seconds Between Retries.

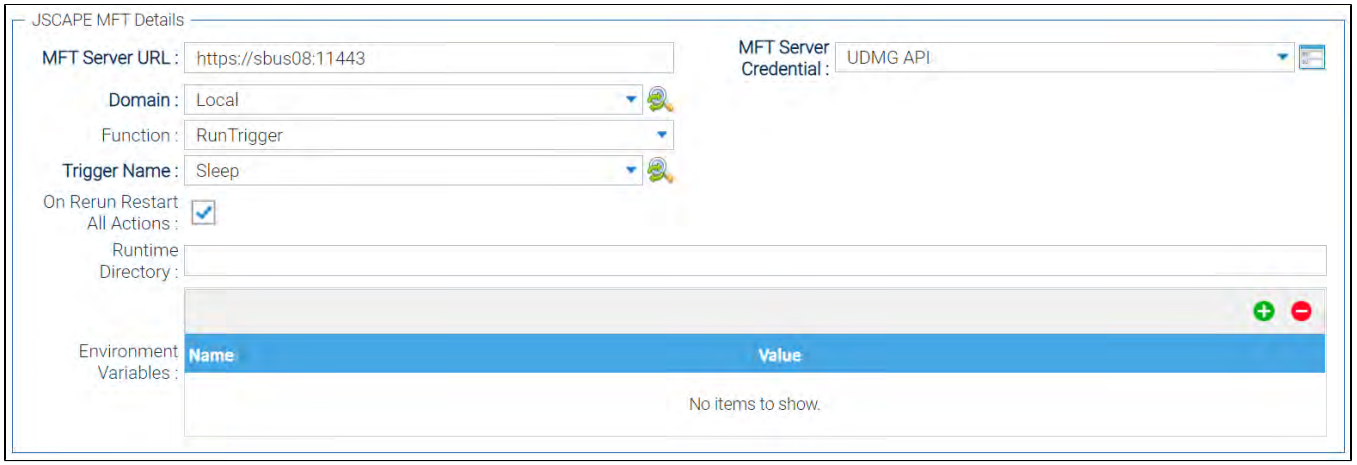
Function = TradingPartnerRegExUpload

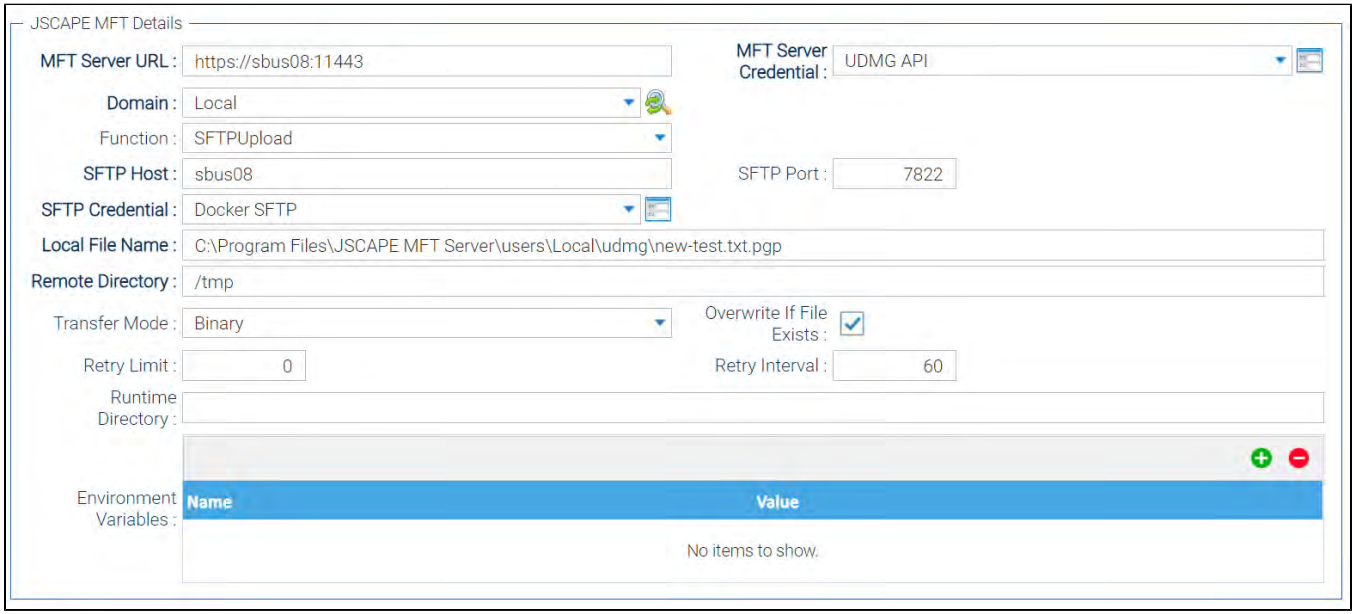
Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **TradingPartnerRegExUpload** Function.
Trading Partner	Select the JSCAPE MFT Server Trading Partner definition.
Regular Expression / Wildcard	Specify the Regular Expression or Wildcard.
Expression Type	Select either "Regular Expression" or "Widlcard".
Local Directory	Specify the location of the Local File.
Remote Directory	Specify the remote Trading Partner Directory.
Transfer Mode	Specify the Transfer Mode.
Passive Mode	Check to use passive mode.
Fail if No Files Found	Check to fail if no files match the specified expression (Regular Expression or Wildcard).
Delete on Success	Check to delete source files after successful transfer.
Retry Limit	Specify the Maximum Retry Attempts.
Retry Interval	Specify the Interval in Seconds Between Retries.

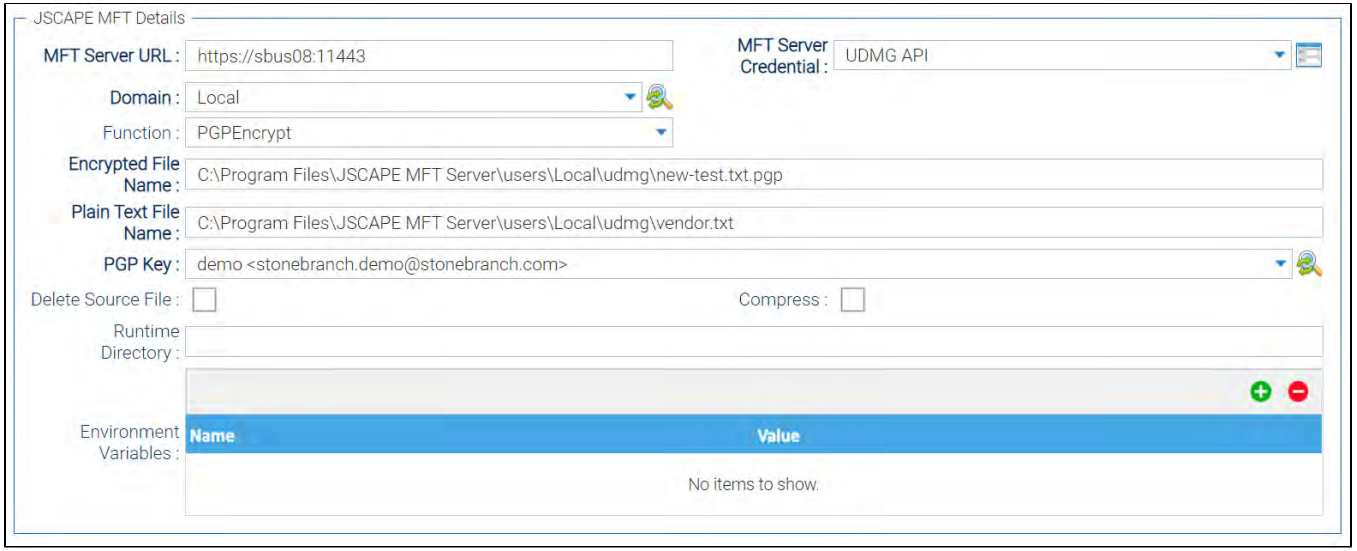
Function = TradingPartnerRegExDownload

Field	Description
MFT Server URL	MFT Server URL i.e. https://localhost:11443
MFT Server Credential	UC credential definition containing a valid JSCAPE MFT Server Administrator username and password.
Domain	Select the required JSCAPE MFT Server Domain name.
Function	Select the **TradingPartnerRegExDownload** Function.
Trading Partner	Select the JSCAPE MFT Server Trading Partner definition.
Regular Expression / Wildcard	Specify the Regular Expression or Wildcard.
Expression Type	Select either "Regular Expression" or "Widlcard".
Local Directory	Specify the location of the Local File.
Remote Directory	Specify the remote Trading Partner Directory.
Transfer Mode	Specify the Transfer Mode.
Passive Mode	Check to use passive mode.
Fail if No Files Found	Check to fail if no files match the specified expression (Regular Expression or Wildcard).
Delete on Success	Check to delete source files after successful transfer.
Retry Limit	Specify the Maximum Retry Attempts.
Retry Interval	Specify the Interval in Seconds Between Retries.

Examples for JSCAPE MFT Universal Tasks


JSCAPE MFT Details
MFT Server URL :
https://sbus08:11443
MFT Server Credential :
UDMG API
Domain :
Local
Function :
RunTrigger
Trigger Name :
Sleep
On Rerun Restart All Actions :
Runtime Directory :
Environment Variables :
Name
Value
No items to show.


JSCAPE MFT Details
MFT Server URL :
https://sbus08:11443
MFT Server Credential :
UDMG API
Domain :
Local
Function :
SFTPUpload
SFTP Host :
sbus08
SFTP Port :
7822
SFTP Credential :
Docker SFTP
Local File Name :
C:\Program Files\JSCAPE MFT Server\users\Local\udmg\new-test.txt.pgp
Remote Directory :
/tmp
Transfer Mode :
Binary
Overwrite If File Exists :
Retry Limit :
0
Retry Interval :
60
Runtime Directory :
Environment Variables :
Name
Value
No items to show.



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Microsoft Teams Integration

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

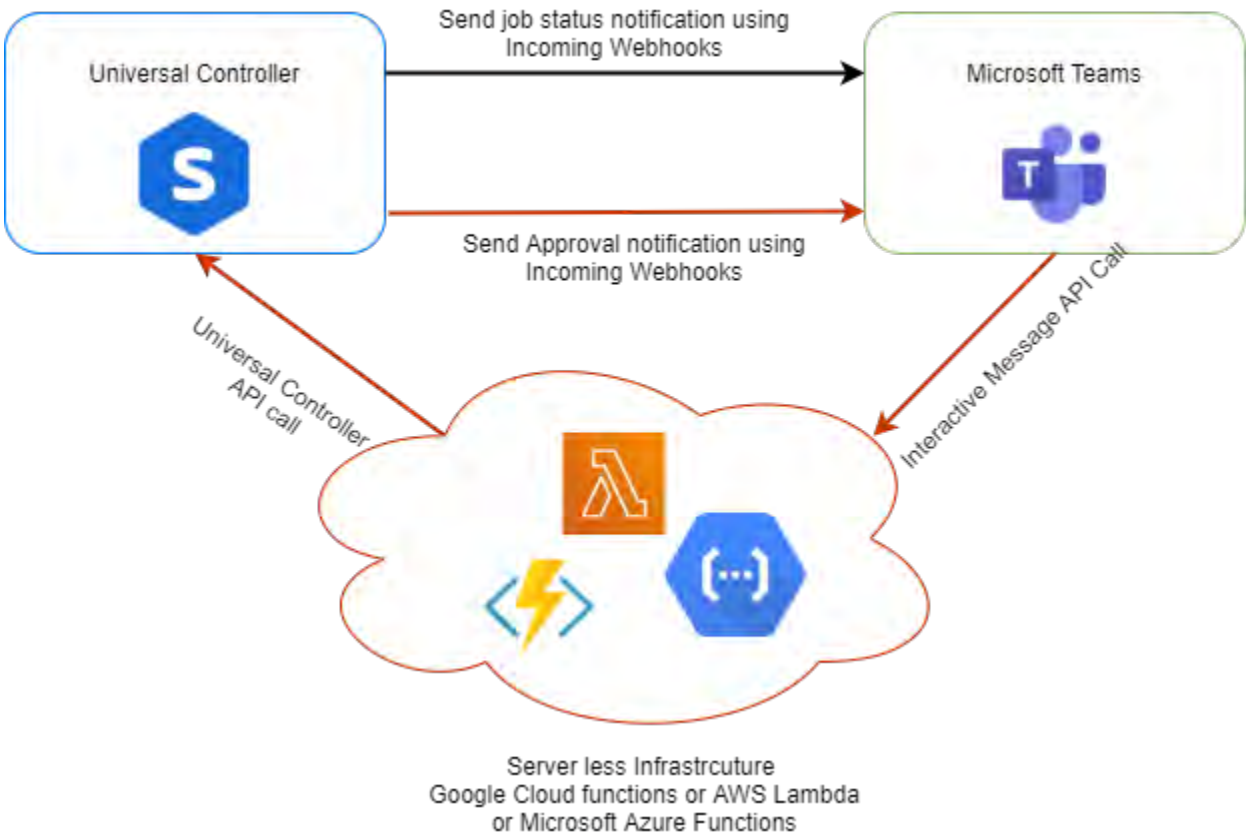
Introduction

This Universal Task allows you to send messages to an existing channel of Microsoft Teams. As a result, you can integrate this solution in UAC to notify users for UAC result or send approval notifications on Microsoft teams.

Overview

Key Features

- Quick reaction time on job failures.
- Manual task interruptions in workflows can be handled by concerned applications/business team, while workflows in Universal Controller can be resumed simply by responding to the approval notifications on Microsoft Teams.



Software Requirements

Software Requirements for Universal Template and Universal Task

This integration requires an Universal Agent and a Python runtime to execute the Universal Task.

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.6.0.0 and later with python options installed
- Universal Agent for Linux Version 6.6.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later

Software Requirements for the Application to be Scheduled

The Universal Task requires an Incoming Team channel and a Serverless Infrastructure like AWS Lambda or Google Cloud Functions or Microsoft Azure Functions.

Technical Considerations

Adding an Incoming webhook to a Microsoft Teams Channel

Note

If your MS Team's Settings => Member permissions => Allow members to create, update, and remove connectors is selected, any team member can add, modify, or delete a connector.

1. Navigate to the channel where you want to add the webhook and select (•••) More Options from the top navigation bar.
2. Choose Connectors from the drop-down menu and search for Incoming webhook.
3. Select the Configure button, provide a name, and, optionally, upload an image avatar for your webhook.
4. The dialog window will present a unique URL that will map to the channel. Make sure that you copy and save the URL you will need to provide it to the outside service.
5. Select the Done button. The webhook will be available in the team channel.

Additional Information on How to Use Approval Notification Feature

Manual tasks are typically used when there is a need for manual intervention of user in a workflow process. Traditionally the Manual Task is completed successfully in the Universal controller by clicking “Set Completed” command.

With this Universal Task for Microsoft Teams, we can provide you with a notification in the Microsoft team incoming webhook channel, when the workflow reaches the manual task with status “Action Required”.

Upon receiving the notification on the teams channel, users can click on on the “Approve” Button in the interactive message for the workflow to proceed further. This interactive message is sent from the Universal Controller.

When the “Approve” or “Reject” button is clicked in the interactive message, an API call is made to a function where it can handle the event from the Microsoft Teams. For example we use python function in AWS lambda + API gateway or Azure functions or any custom URL where the Teams messaging platform can make an API POST call to handle the user action in the message as a payload and, based on the posted payload data from the Teams, Universal Controller API call will be made to set the manual task either to set complete status or No action in the function. Please refer to the handler.py file in the serverless function folder for a sample serverless function implementation using AWS lambda.

For Approval Notification feature of the Universal Task, the “API Endpoint” provided in the task details could be an end point either in AWS lambda or Azure Function or GCP function or your custom API end point to handle the interactive message from Microsoft Teams.

Below is the sample python code that could be invoked for the Microsoft Teams Approval Notification. In the code below, update the following variables accordingly under def_handler()

- *teams_incoming_webhook* = *'XXXX'* # The incoming web hook of Microsoft Teams channel
  uname = *'XXXX'* # Universal Controller user name
  passwd = *'XXXX'* # Universal Controller user password
  uc_url = *'http:// + uname + ':' + passwd + '@XXXXXX/resources/taskinstance/setcompleted'* # URL of the Universal Controller

```
import json
import boto3
import logging
from urllib.parse import parse_qs
import requests

logger = logging.getLogger()
logger.setLevel(logging.INFO)


def lambda_handler(event, context):
    print(str(event))
    logger.info(json.dumps(event))
    payload = event['body']
    print(payload)
    jobname_split = payload.split(':')
    jobname = jobname_split[1]
    team_button = jobname_split[0]
    print(team_button)
    print(jobname)
    ###################### Teams Channel Data ################
    teams_incoming_webhook = 'XXXX'
    ###################### End of Teams Channel Data ################
    ############## Credentials for universal controller #################
    uname = 'XXXX'
    passwd = 'XXXX'
    uc_url = 'http://' + uname + ':' + passwd + \
             '@XXXXXX/resources/taskinstance/setcompleted'
    ############## End of Universal Controller Credentials #################
    # Posting request to Universal Controller
    uc_post_request(team_button, jobname, uc_url, teams_incoming_webhook)
    ################## Teams data parsed -completed ############################
    body = {
        "message": "Teams Data parsed successfully and Universal controller ""confirmed the job !",
        "input": event
    }
    response = {
        "statusCode": 200,
        "body": json.dumps(body)
    }
    return response


def uc_post_request(team_button, jobname, uc_url, teams_incoming_webhook):
    header = {'content-type': "application/json"}
    if team_button=="Approved":
        print("Intiating Request to Universal Controller")
        approval_message = {
            "name": jobname,
            "criteria": "Newest Instance"
        }
        print(uc_url)
        post_uc = requests.post(uc_url, data=json.dumps(approval_message),
            headers=header)
        # print(post_uc.text)
```

```
        if post_uc.status_code==200:
            format_response = post_uc.json()
            logger.info(format_response)
            if format_response['success'] is False:
                print("Something went wrong")
                error_message = {
                    "@type": "MessageCard",
                    "@context": "https://schema.org/extensions",
                    "summary": "This is the summary property",
                    "themeColor": "#FFFF00",
                    "sections": [
                        {
                            "activityTitle": "**Couldn't not reach**",
                            "activitySubtitle": "Something went wrong, ""action not completed"
                        }
                    ]
                }
                print("Sending error report to MS Teams Channel")
                uc_response = requests.post(teams_incoming_webhook,
                    data=json.dumps(error_message),
                    headers={'CARD-UPDATE-IN-BODY': 'True',
                             'Content-Type': 'application/json'})
            elif format_response['success'] is True:
                print("Your request is approved")
                approval_response = {
                    "@type": "MessageCard",
                    "@context": "https://schema.org/extensions",
                    "summary": "This is the summary property",
                    "themeColor": "#008000",
                    "sections": [
                        {
                            "activityTitle": "**Approved**",
                            "activitySubtitle": "Request was approved after ""review"
                        }
                    ]
                }
                header = {
                    'content-type': 'application/json'
                }
                print("Sending Notification to MS Teams Channel")
                uc_response = requests.post(teams_incoming_webhook,
                    data=json.dumps(approval_response), headers=header)
                format_response = json.loads(uc_response.text)
                print(format_response)
    elif team_button=="Rejected":
        print("Reqeust Denied")
        reject_response = {
            "@type": "MessageCard",
            "@context": "https://schema.org/extensions",
            "summary": "This is the summary property",
            "themeColor": "#FF0000",
            "sections": [
                {
                    "activityTitle": "**Rejected**",
                    "activitySubtitle": "Request was rejected after review"
                },
            ]
        }
        print("Sending Notification to MS Teams Channel")
        uc_response = requests.post(teams_incoming_webhook,
            data=json.dumps(reject_response),
            headers={'CARD-UPDATE-IN-BODY': 'True',
```

```
                'Content-Type': 'application/json'})
    print("Teams Response: ", uc_response.status_code)
```

Microsoft Teams Integration

Key Features

Feature	Description
Send Message	With Send Message function we can send a notification message to the Microsoft Teams channel with the current task instance details such as job failure, late start/run, and other important events.
Approval Notification	With Approval notifications, an interactive message is sent to the Microsoft Teams incoming channel. The user can chose to Approve or Reject the continuation of workflow execution.

Import Microsoft Teams Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Microsoft Teams IntegrationUniversal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

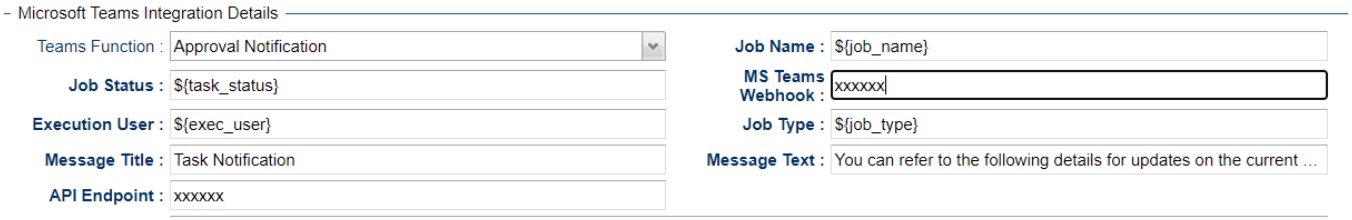
Field Descriptions for Microsoft Teams Integration Universal Task

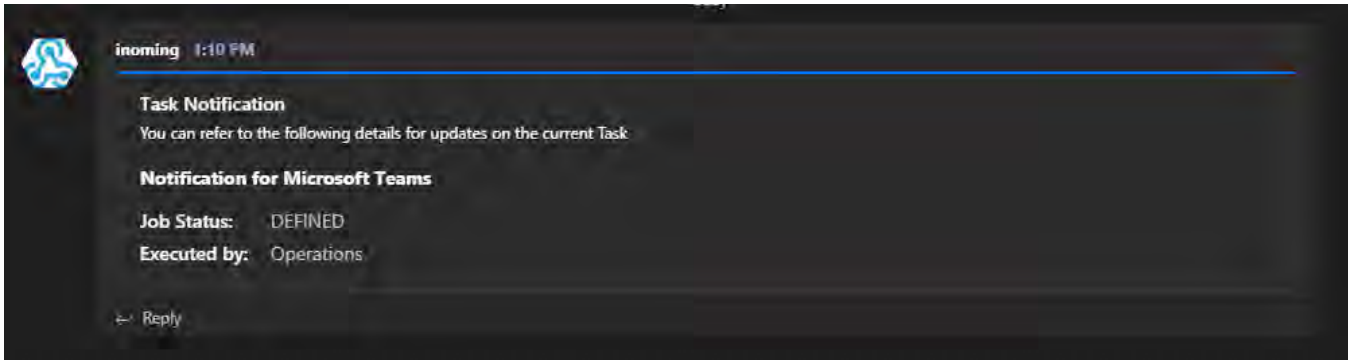
Field	Description
Send Message	
Teams Function	Send Message or Approval Notification.
Job Name	Name of the job, by default it takes the current job name ${ops_task_name}.
Job Status	Status of the job, by default it takes the current job name ${ops_status}.
MS Teams Webhook	The incoming web hook of Microsoft Teams channel.
Execution User	Details of the execution user, by default takes the current user name ${ops_execution_user}.

Job type	Task type of task instance, by default takes the current task instance type ${ops_task_type}.
Message Title	The title of the message sent to Microsoft Teams channel.
Message Text	The text of the message sent to Microsoft Teams channel.
Approval Notification	
API Endpoint	URL of the Serverless infrastructure endpoint.

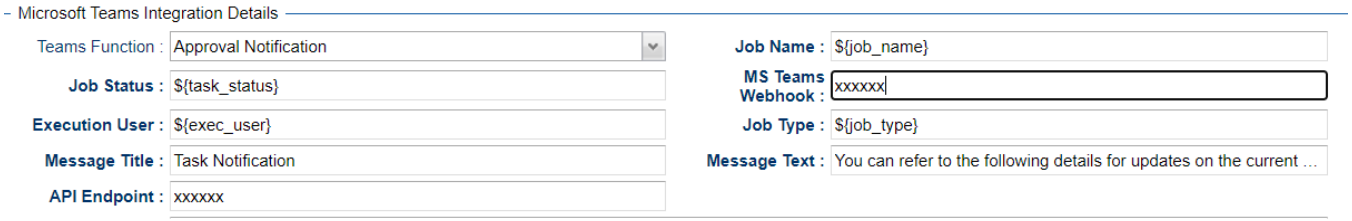
Examples for Microsoft Teams Integration Universal Tasks

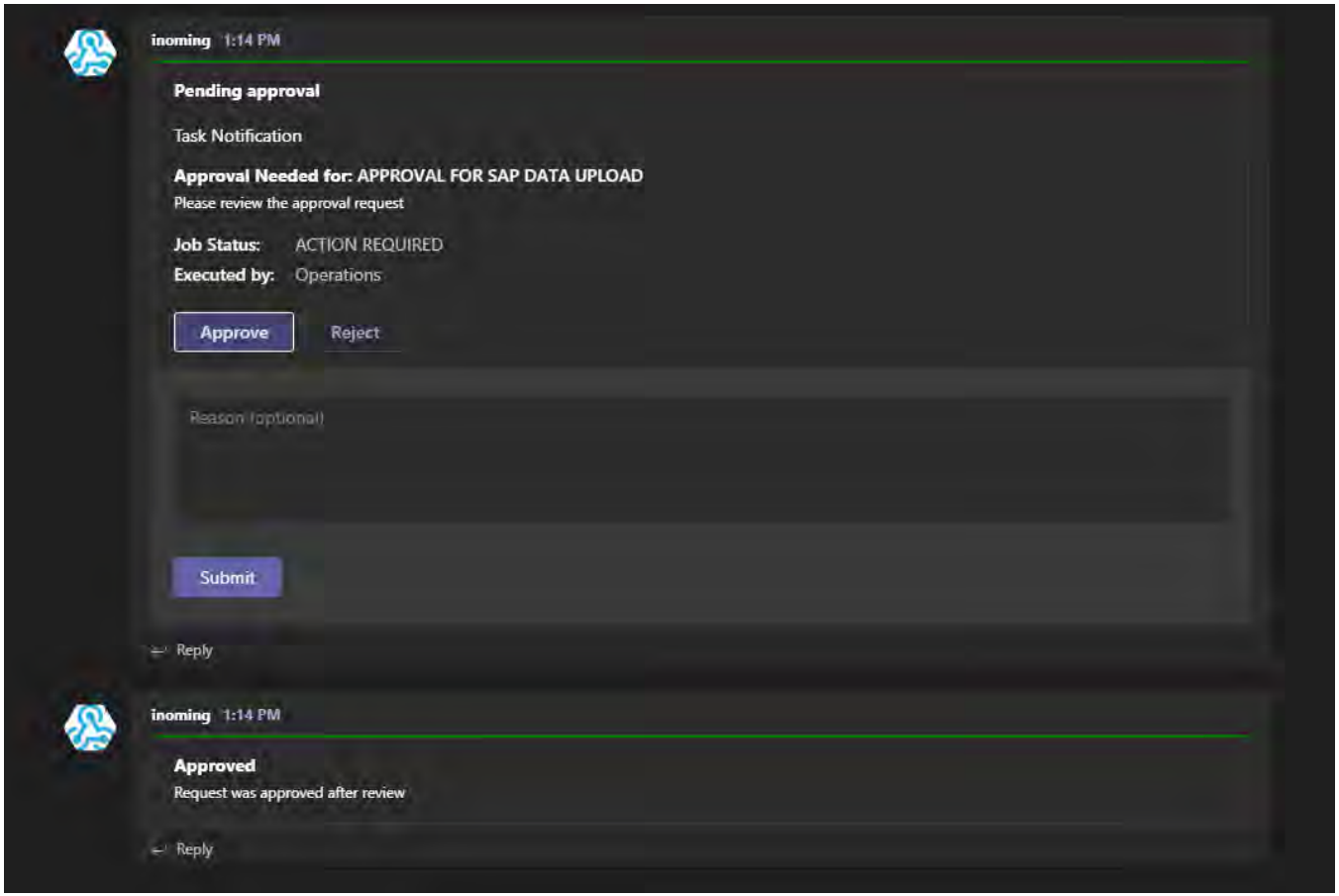
Send Message





Approval Notification





Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.
Microsoft Teams Wenhooks	https://docs.microsoft.com/en-us/microsoftteams/platform/webhooks-and-connectors/how-to/add-incoming-webhook	User documentation for creating incoming webhooks in Microsoft Teams Channel.
Requests	https://pypi.org/project/requests/#description	Documentation for python requests module.

PagerDuty

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows Stonebranch users to notify PagerDuty (Incident management platform) in the event of job Failure or long run of a job or Early finish of a job or any other event in Universal Controller. Furthermore, PagerDuty can help to aggregate alerts and group them, and provide reliable notifications, automatic escalations and on-call scheduling that could help fix support issues quickly.

Overview

Users can orchestrate the following functionalities in PagerDuty using this Universal Task:

- PagerDuty Incident

    - Create Incident
    - List Incidents
    - Update Incident Notes
- PagerDuty Event
    - Create PagerDuty Event
- PagerDuty User Management
    - Create a user in PagerDuty
    - Delete a user in PagerDuty

Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against PagerDuty.

Software Requirements for Universal Template and Universal Task

Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.

- Python modules required
    - requests

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.6 and later with Python options installed
- Universal Agent for Linux Version 6.6 and later with Python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with PagerDuty REST API V2 for incidents functionalities ,User creation & deletion and Events API V2 for event creation in PagerDuty.

Technical Considerations

- This task uses Python modules requests to make REST-API calls to the PagerDuty environment.
- PagerDuty API URL ,API token and PagerDuty user email would be required as basic input for this Universal Task.
- Refer to PagerDuty API URL: https://developer.pagerduty.com/api-reference/

Key Features

Feature	Description
Create Incident	This feature helps to create an incident in PagerDuty for an event in Universal Controller; for example: Job Failure or Job running longer.

List Incidents	List Incidents that are in PagerDuty with status triggered or resolved or acknowledged for a PagerDuty service ID.
Update Incident notes	Update a work note for an existing incident in PagerDuty.
Create Event	Create an event in PagerDuty; Specifying the details of the event in Universal Controller; severity, Job Name, Job Type, Description, Event Action.
Create User	Creates an user account in PagerDuty.
Delete User	Deletes an user account in PagerDuty.

Import PagerDuty Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure PagerDuty Integration Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

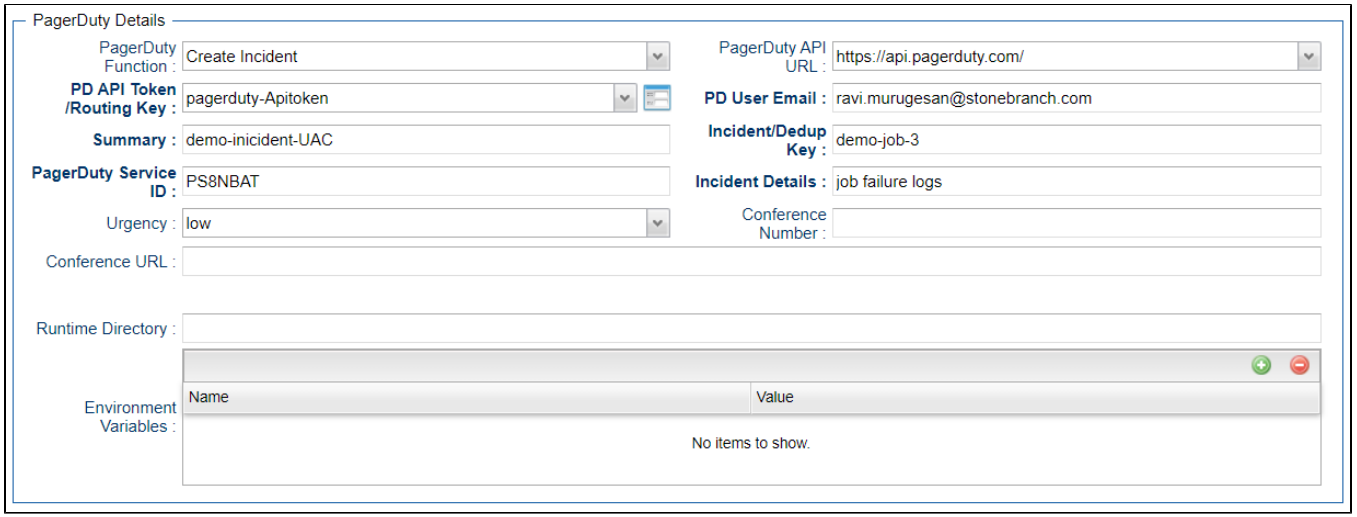
Field Descriptions for PagerDuty Integration Universal Task

Field	Description
PagerDuty Function	Select the functions that needs to be performed in PagerDuty.
PagerDuty API URL	For Event creation select : https://events.pagerduty.com/v2/enqueue and other functionality select : https://api.pagerduty.com/.
PD API Token /Routing Key	Provide the API token for incident and routing key for Event creation.
PD User Email	Provide the valid Pager Duty user Email address.
Summary	specify the incident title or an event summary.
Incident/Dedup Key	Provide an Incident key for PagerDuty incident E.g: Universal Controller job name.
PagerDuty Service ID	Service ID in PagerDuty where the incident needs to be created.
Incident Details	Provide the incidents that needs to be in pager duty.
Urgency	Select the urgency of the event in Universal Controller.
Conference Number	Conference Bridge Details for the meeting (Incase if the support needs to gets in to conference call).
Conference URL	URL for the conference meeting.

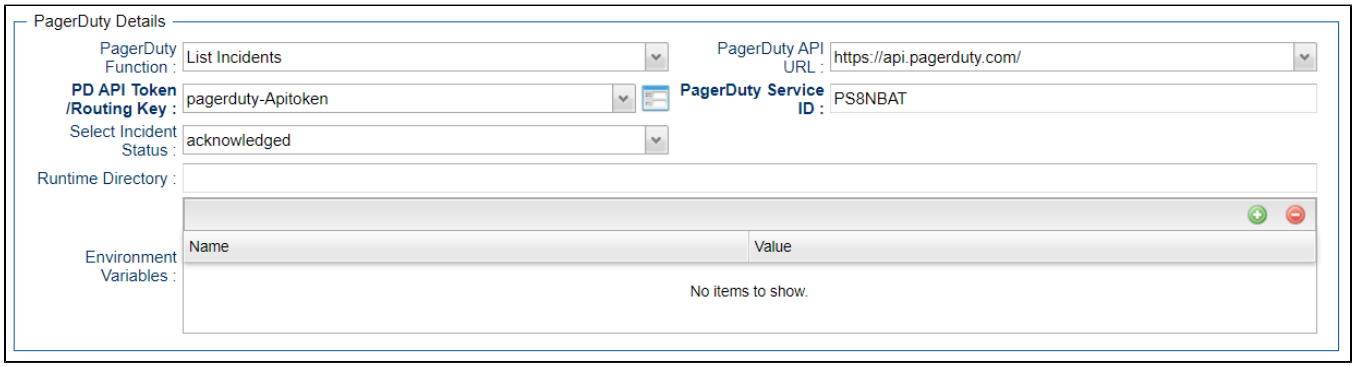
Incident ID	Provide the PagerDuty incident ID (Incase where the PagerDuty Incident note needs an update).
Incident Note	Include the note that needs to be updated for the incident.
Select Incident Status	Select the status for which the incident details needs to be retrieved from PagerDuty.
Event Action	Select the appropriate event Action trigger or acknowledge or resolve.
Severity	Select either one of the status that would be appropriate (Critical, Error, Warning, and Info).
Event Source System	The unique location of the affected system, preferably a hostname or FQDN.
Job Type	provide the job type which will fit in the component field of the pager duty event.
Job Status	Job status that will fit in to the class field for the pager duty event creation.
User ID	PagerDuty User ID.
Name	Name of the PagerDuty user.
User Email Address	User's Email Address.
Job Title	Provide the Job title of the user.
User Role	The role of the user; for example, administration.
Time Zone	Specify the user time zone.
Summary	Summary about the user creation request.

Examples for PageDuty Integration Universal Tasks

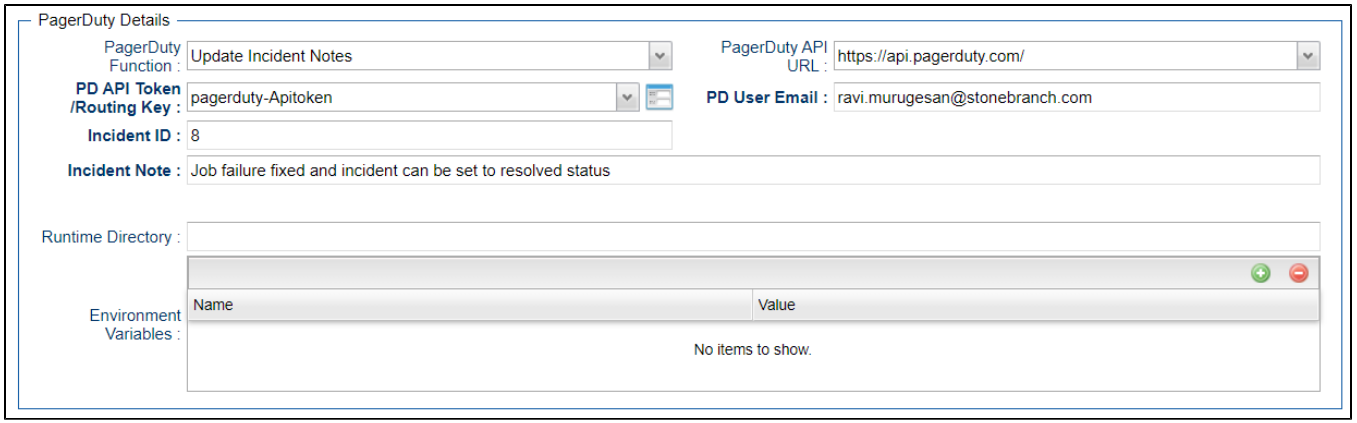
Create Incident in PagerDuty



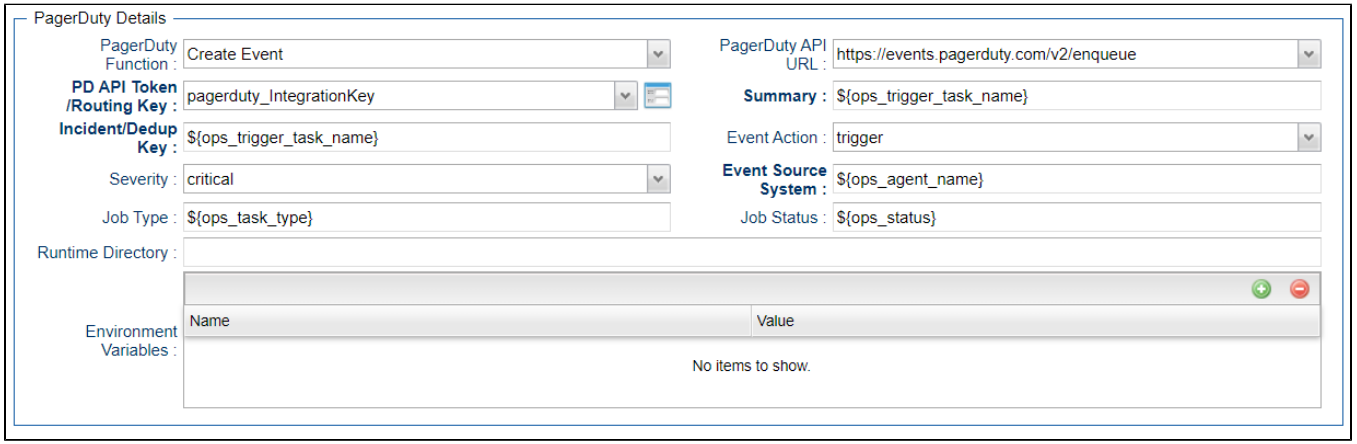
List Incidents in PagerDuty



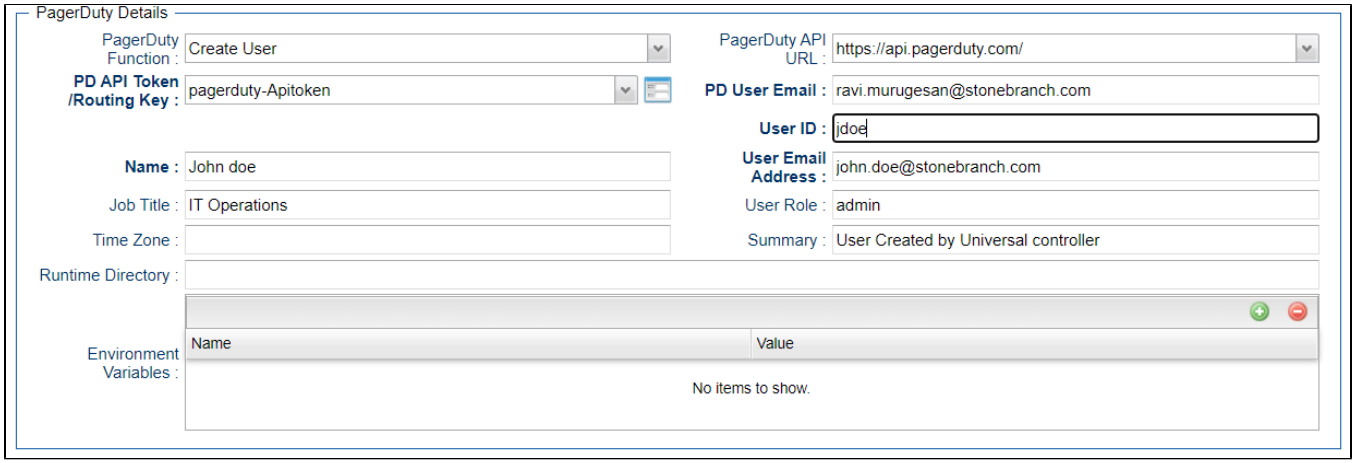
PagerDuty Update Incident Note



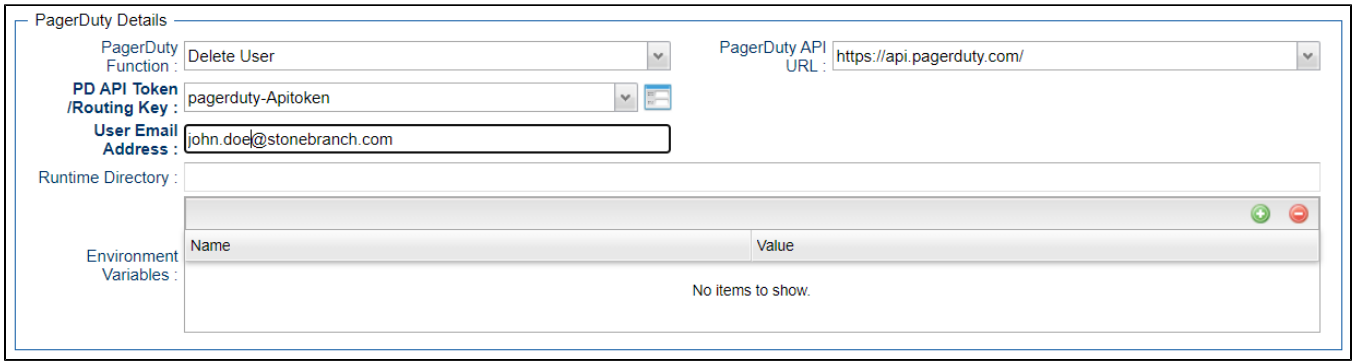
Create Event in Pager Duty



Create User in PagerDuty



Delete User in PagerDuty



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Power BI

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows Stonebranch users to perform end-to-end Orchestration and Automation of Jobs & Clusters in Databricks environment, either in AWS or Azure.

Overview

- This task will use the Databricks URL and the user bearer token to connect with the Databricks environment.
- Users can perform the following with respect to the Databricks jobs.
    - Create and list jobs

    - Get job details
    - Run now jobs
    - Run submit jobs
    - Cancel run jobs
- Also with respect to Databricks clusters, this Universal Task can perform the following operations:
    - Create, start and restart a cluster
    - Terminate a cluster
    - Get a cluster info
    - List clusters
- With respect to Databricks DBFS , this Universal Task also provides a feature to upload larger files.

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 7.0.0.0 or later is required.
- Universal Agent needs to be installed with python option (--python yes).

Software Requirements for Universal Controller

- Universal Controller Version 7.0.0.0 or later is required.

Software Requirements for the Application to be Scheduled

A Microsoft PowerBI Service User and Password are required

To use the Power BI REST API, a client application ID must be obtained by registering an application with Azure Active Directory. This registration can be completed via the four-step process at the following portal: Onboarding Embed Tool

Key Features

Some details about Universal Tasks for PowerBI:

- Refresh a dataset in a group-workspace or in my workspace.
- Refresh a dataflow in a group-workspace.
- Lookup datasets in a selected Group.
- Lookup dataflows in a selected Group.
- Connection to PowerBI Service REST API is done via the python MSAL library.
- Supports Windows and Linux Universal Agents in order to connect to the PowerBI REST API.

Current Limitations

- When a Power BI dataflow refresh fails, no detailed error description is provided by the current Power BI REST API.
- Canceling of dataflow is not provided by the current Power BI REST API.
- The task will be updated as soon as this feature is available by Microsoft.
- Canceling of a dataset refresh will be provided in the next Version.

Import the Power BI Universal Template

To use the Power BI Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature, check that the Resolvable Credentials Permitted system property has been set to true.
2. Download the provided ZIP file.
3. In the Universal Controller UI, select Administration >Configuration > Universal Templates to display the current list of Universal Templates.
4. Click Import Template.
5. Select the template ZIP file and Import.

When the template has been imported successfully, the Universal Template will appear on the list. Refresh your Navigation Tree to see these tasks in the Automation Center Menu.

Configure Power BI Universal Tasks

For Universal Task Power BI, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for Power BI Universal Task - Actions

The PowerBI Task provides two different Actions.

- refresh dataset
- refresh dataflow

For each action the specific fields are described and an example is provided.

refresh dataset - Action

The Action refreshes a dataset in a group-workspace or in the my workspace.

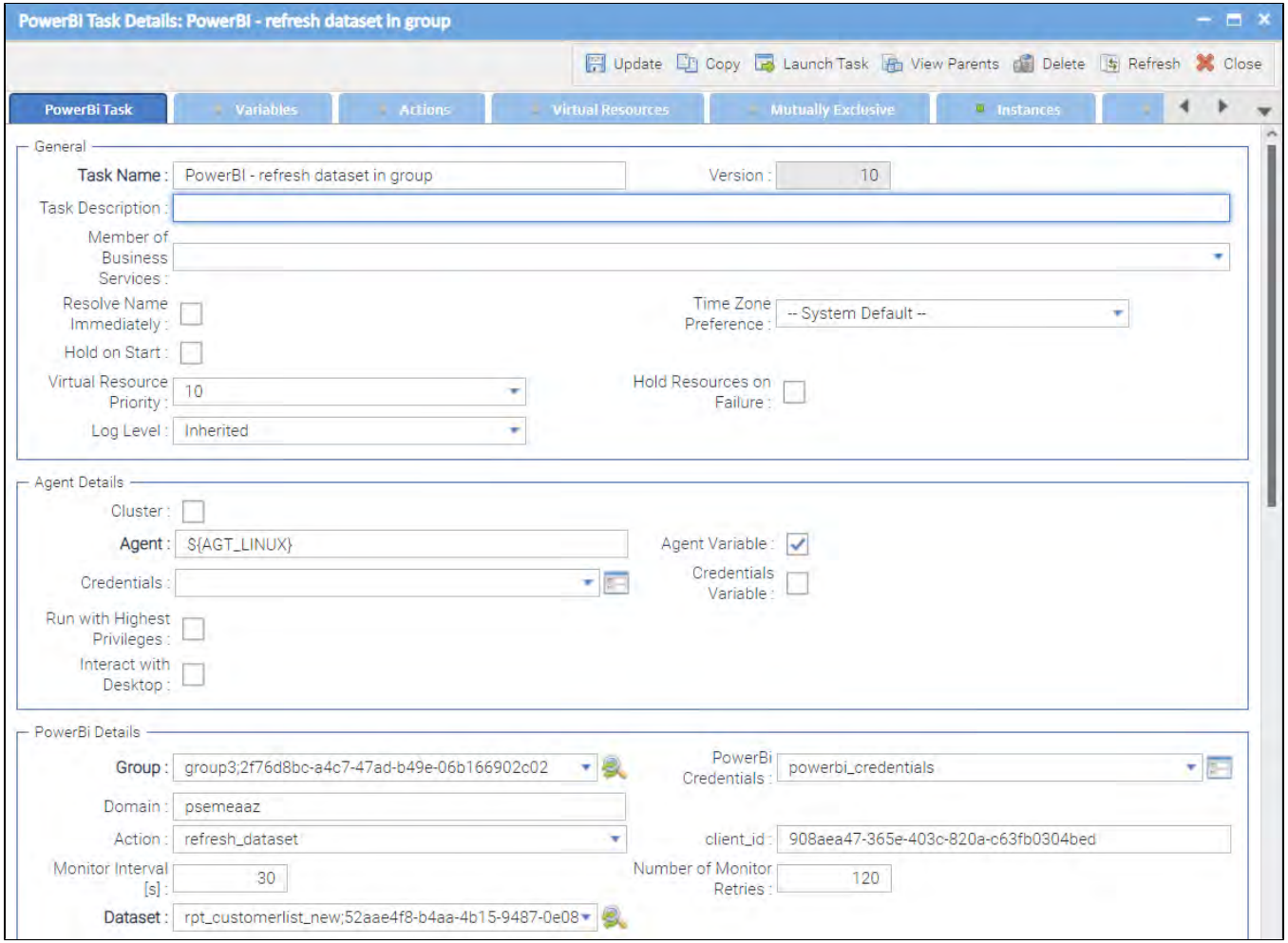
Field	Description
Power BI Credentials	Power BI Service User and Password.
client_id	Power BI Client ID. To use the Power BI REST API, a client application ID must be obtained by registering an application with Azure Active Directory. This registration can be completed via the four-step process at the following portal: Onboarding Embed Tool
Domain	Domain name. Go to Microsoft Azure> Azure Active Directory lookup: Primary domain (for example, psemeaaz.onmicrosoft.com)
Action	[refresh dataset, refresh dataflow] refresh dataset: Refreshes a dataset in a Group or my workspace. refresh dataflow: Refreshes a dataflow in a Group.
Group	In this field, the Group is selected, which contains the dataset to refresh. A Group is a workspace within Power BI.

	When pressing the magnifier you can dynamically choose the available Groups in your Power BI account (see Select the Group that Contains the dataset to Refresh). To select the *My workspace,* choose *None; None* as Group.
Dataset	Power BI dataset name. Power BI datasets represent a source of data ready for reporting and visualization When pressing the magnifier, you can dynamically choose from the datasets in the Group you pre-selected in the Group field (see Select dataset to Refresh).
Monitor Interval [s]	Default is 60s. The Monitor Interval is the frequency that the Universal Task checks the current refresh status of the dataset. The Maximum number of checks is defined by the field value *Number of Monitor Retries.* For example, Monitor Interval 60s and Number of Monitor Retries = 120 means that every 60s, the current refresh status is retrieved from Power BI, but only 120 times (120 x 60s = 2h). If after 2h, the refresh could not be completed, the Universal Task will go into status failed.
Number of Monitor Retries	Default is 120 Monitor Retries. The Number of Monitor Retries value is the the Maximum number of refresh status checks. As long as the refresh status is unknown, a retry will be performed with the frequency defined in the Monitor Interval [s] field. If the maximum number is reached the task will go into status failed. (See the Monitor Interval [s] field Description for an example.)
Log level (default is Inherited)	Universal Task logging settings: [Inherited | Trace | Debug | Info | Warn | Error | Severe]

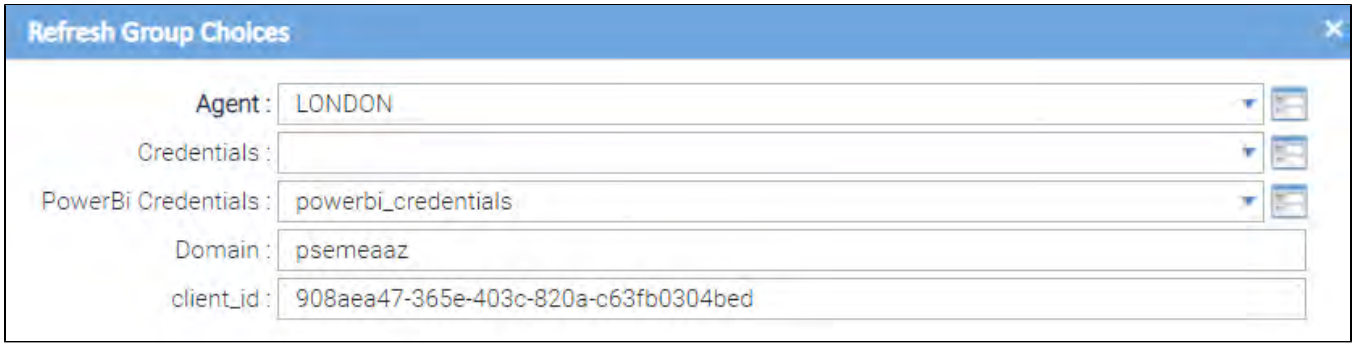
Example for Power BI Universal Task - Action: refresh dataset

The following example refreshes a dataset in group.

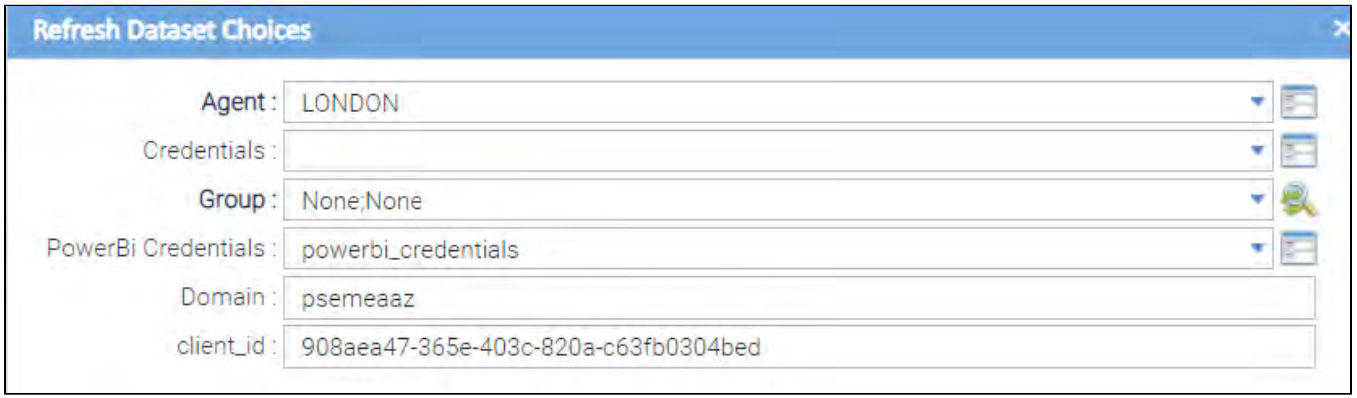
refresh dataset - Action



Select the Group that Contains the dataset to Refresh



Select dataset to Refresh



refresh dataflow - Action

This Action refreshes a dataflow in group.

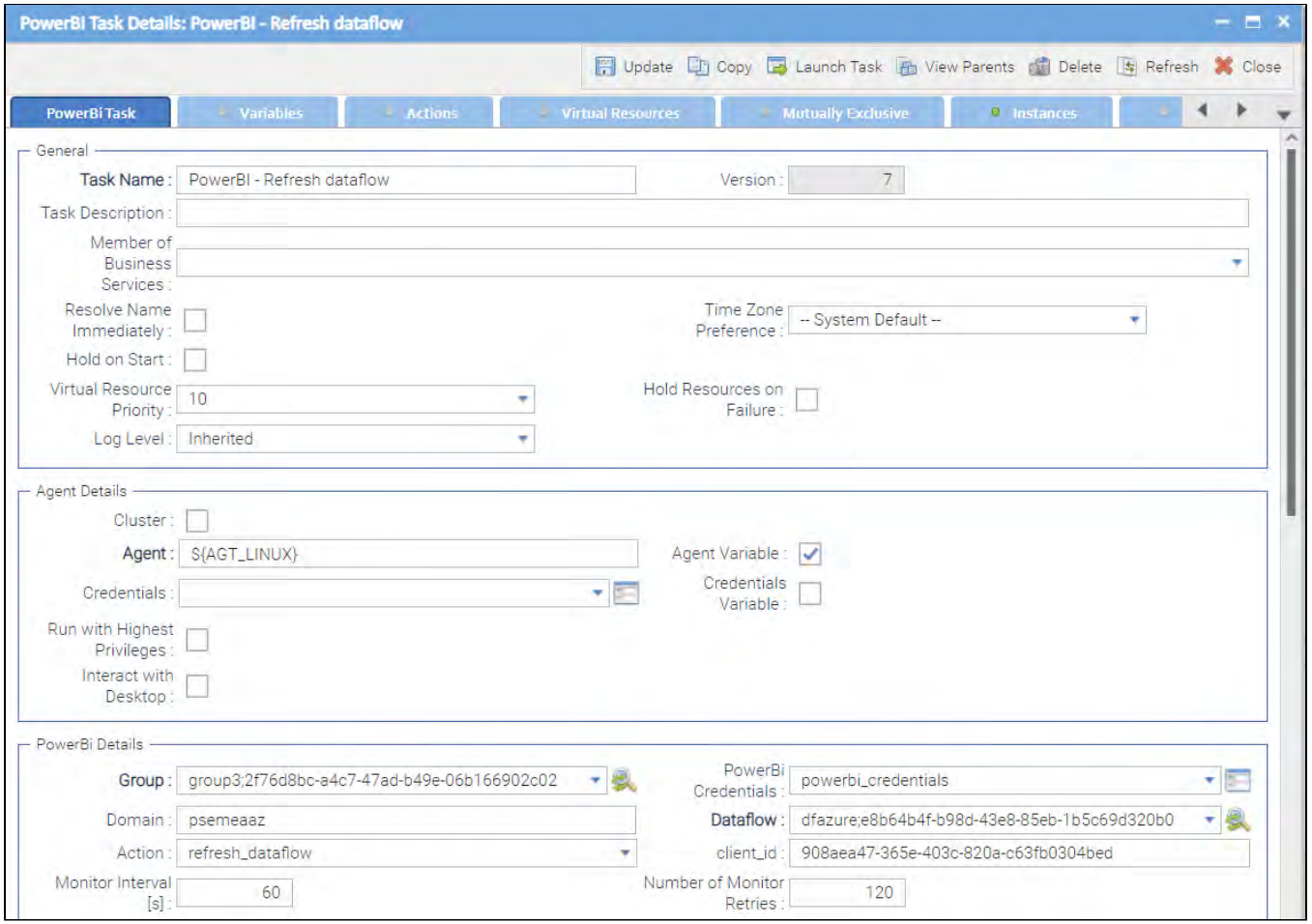
Field	Description
Power BI Credentials	Power BI Service User and Password.
client_id	Power BI Client ID. To use the Power BI REST API, a client application ID must be obtained by registering an application with Azure Active Directory. This registration can be completed via the four-step process at the following portal: Onboarding Embed Tool
Domain	Domain name. Go to Microsoft Azure> Azure Active Directory lookup: Primary domain (for example, psemeaaz.onmicrosoft.com)

Action	[refresh dataflow, refresh dataflow] refresh dataflow: Refreshes a dataflow in a Group or my workspace. refresh dataflow: Refreshes a dataflow in a Group.
Group	In this field, the Group is selected, which contains the dataflow to refresh. A Group is a workspace within Power BI. When pressing the magnifier you can dynamically choose the available Groups in your Power BI account (see Select the Group that Contains the dataflow to Refresh). To select the *My workspace,* choose *None; None* as Group.
Dataflow	Power BI dataflow name. Power BI dataflows are Power Query processes running in the cloud, with the same set of data preparation functionalities, data source connectors, gateways, and transformations. When pressing the magnifier, you can dynamically choose from the dataflows in the Group you pre-selected in the Group field (see Select dataflow to Refresh).
Monitor Interval [s]	Default is 60s. The Monitor Interval is the frequency that the Universal Task checks the current refresh status of the dataflow. The Maximum number of checks is defined by the field value *Number of Monitor Retries.* For example, Monitor Interval 60s and Number of Monitor Retries = 120 means that every 60s, the current refresh status is retrieved from Power BI, but only 120 times (120 x 60s = 2h). If after 2h, the refresh could not be completed, the Universal Task will go into status failed.
Number of Monitor Retries	Default is 120 Monitor Retries. The Number of Monitor Retries value is the the Maximum number of refresh status checks. As long as the refresh status is unknown, a retry will be performed with the frequency defined in the Monitor Interval [s] field. If the maximum number is reached the task will go into status failed. (See the Monitor Interval [s] field Description for an example.)
Log level (default is Inherited)	Universal Task logging settings: [Inherited | Trace | Debug | Info | Warn | Error | Severe]

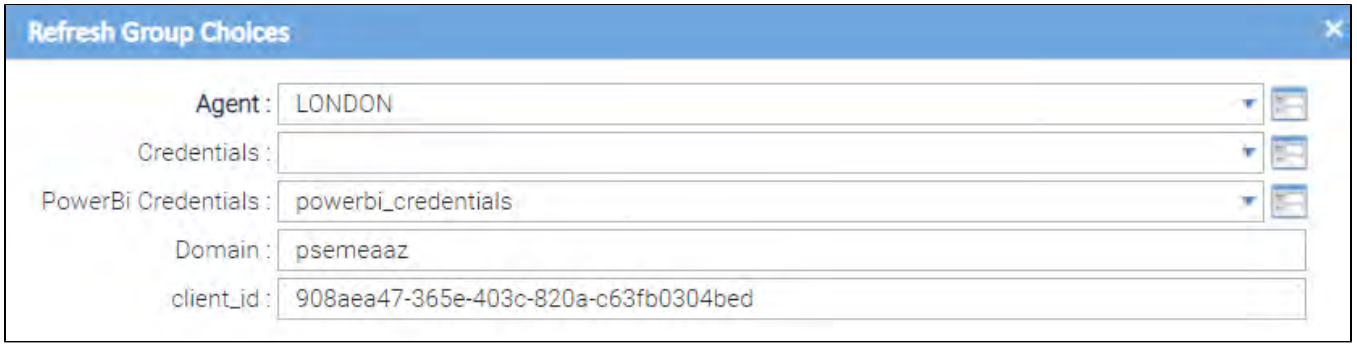
Example for Power BI Universal Task - Action: refresh dataflow

This example shows how a dataflow in a Group is refreshed,

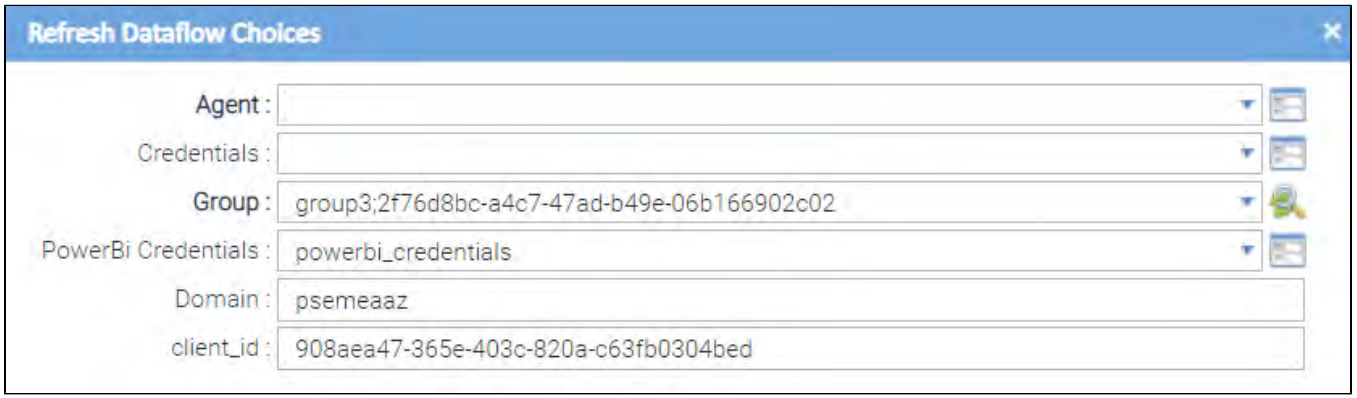
refresh dataflow - Action



Select the Group that Contains the dataflow to Refresh



Select dataflow to Refresh



Salesforce

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows users to create contact and lead objects in Salesforce, as well as execute Salesforce Object Query Language (SOQL) queries.

Overview

- Allows advanced automation by enabling creation of Salesforce objects, such as Contact and Leads, from the Universal Controller in combination with event-based triggers.
- Uses the Salesforce Object Query Language (SOQL) to search your organization’s Salesforce data for specific information.
- SOQL is similar to the SELECT statement in the widely used Structured Query Language (SQL), but is designed specifically for Salesforce data.
- With SOQL, users can construct simple but powerful query strings and execute them from within the Universal Controller.

Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against a remote ServiceNow instance.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - simple-salesforce

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.9.0.0 and later with python options installed
- Universal Agent for Linux Version 6.9.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.9.0.0 and later

Software Requirements for the Application to be Scheduled

The simple-salesforce Python module uses the Lightning Platform REST API. Salesforce supports each API version for a minimum of three years from the date of first release.

The module is also updated regularly to add features and support new API versions.

Universal Task Description

Key Features

Feature	Description
Create Contact, Lead	Create Contact and Lead objects in Salesforce. Combine with event-based triggers or use within workflows to create these objects as a result of events or predecessors.
Execute SOQL query	Query the Salesforce Database using the flexible querying language SOQL. Uses the Universal Controller script library to store and maintain SOQL scripts.

Please note: More functions can easily be added to this Universal Task, based on the available functionality of the simple-salesforce Python module. More fields to hold more information on; for example, Leads or Contacts can also be added to the Universal Template.

Import Salesforce Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Salesforce Universal Task

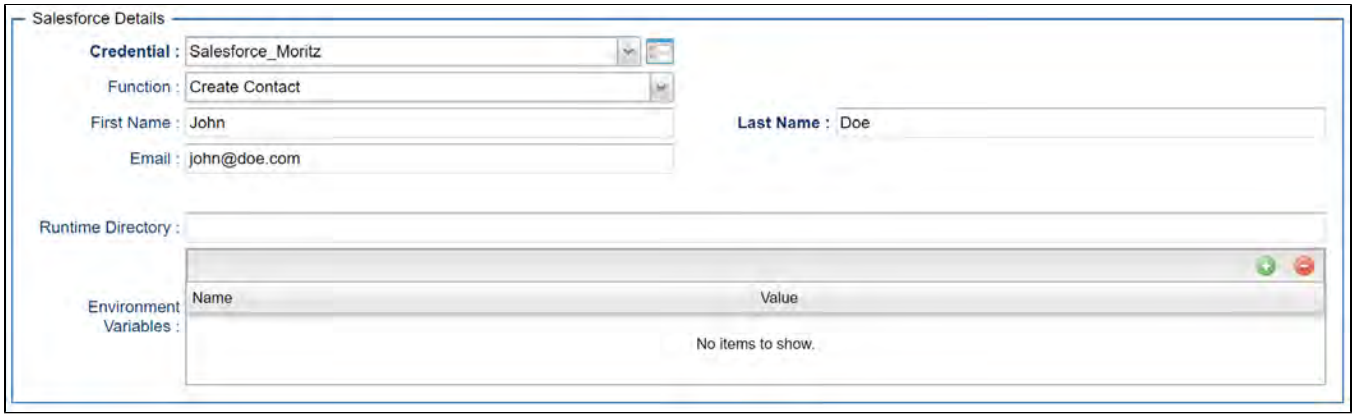
For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Salesforce Universal Task

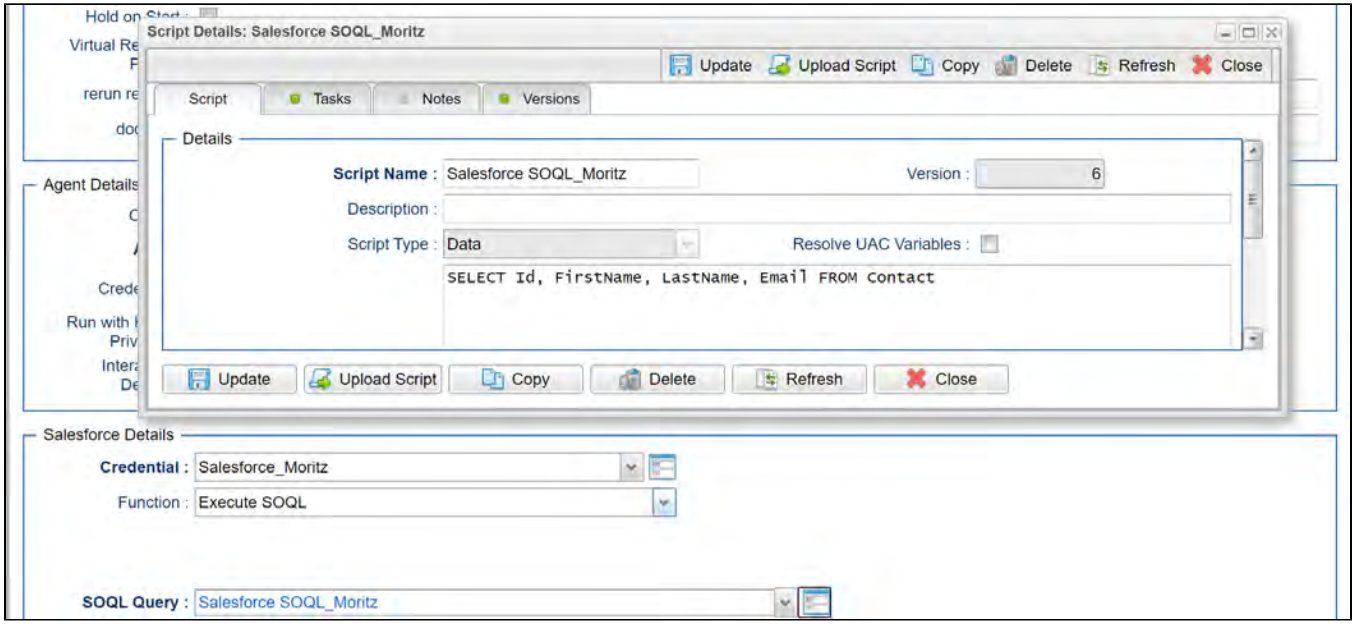
Field	Description
Credential	Username, Password and Security Token for your Salesforce user. Please use the "Passphrase" field for the Security Token.
Function	Select between the available functions of this Universal Task.
First Name	First name of the contact to be created.
Last Name	Last name of the contact to be created.
Email	Email address of the contact to be created.
SOQL Query	Script field. Stores the SOQL Query to be executed as a script in the internal script library.

Examples for Salesforce Universal Tasks

Create Contact



Execute SOQL



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

SAP Batch Input

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

Batch input sessions enter data non-interactively into an SAP system. Batch input is typically used to transfer data from non-SAP systems to SAP systems or to transfer data between SAP systems.

This SAP Task for SAP batch input allows you schedule and execute batch input sessions in SAP.

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 6.x or later is required, including SAP Connector license (USAP).

Software Requirements for Universal Controller

- Universal Controller 6.x or later is required.

Software Requirements for the Application to be Scheduled

This task allows you to run Batch Input session on the following SAP Systems:

- SAP ERP (SAP ECC 6.0)
- SAP S/4 HANA On Premise (not tested)

Key Features

Some details about the Universal Tasks for SAP Batch Input:

The SAP Task for SAP batch input allows you to schedule and execute batch input sessions in SAP

The following functionalities can be performed in UAC with respect to SAP Batch Input:

- Run a batch input session.
- You only need to provide the batch input session name in the task variable.
- It is possible to use wild cards “*” to run multiple Batch Input Sessions.
- There is no need to created manually a variant for the batch input session in SAP.
- The SAP Task uses the feature inline variants of USAP to create a temporary Variant for the ABAP RSBDCSUB with the batch input session name.

Import SAP Task for Batch Input

To use the SAP Task for Batch Input, you first must perform the following:

In the Universal Controller UI, select Automation Center > Tasks > All Tasks and load via the list Import functionality the Task XML files into the Controller (see Task List Import).

When the Task has been imported successfully, the Task will appear on the list of SAP Tasks in the Automation Center Menu under: Automation Center > Tasks > SAP Task

Configure SAP Task for Batch Input

1. Copy the imported “template” Task RUN SAP BATCH INPUT Session.
2. Provide a name according to your naming standards.
3. Adjust the field values to your requirements.

Field Descriptions for SAP Task for Batch Input

The SAP Task for Batch Input contains the following fields.

Field	Value	Description
Utility Agent	SAP Utility Agent	Universal Agent with SAP Connector (USAP)
SAP Connection	SAP Connection	SAP Connection to the SAP System in scope SAP Connections provide all the SAP server information necessary for Universal Controller to execute an SAP Task on an SAP system (see SAP Connection for details).

SAP Credentials	SAP Credentials	Login credentials that the Controller will use to access the SAP system.
Command Group	Submit	Submit the SAP Job to SAP.
start	checked	Starts the Submitted Job in SAP.
Wait	checked	Waits until the started Job in SAP finished or fails.
Definition or Model:	USAP Definition File	USAP Definition File.
Script or File System	Script	Script.
Script	new_variant	Script to run the Batch Input ABAP RSDBDCSUB with the temporary Variant containing the batch input session name.
SAP Command Options	-rawspool no	Output Parameter, to show the result of the ABAP RSDBDCSUB in the Task Output.

Task Variable

In the task Variable, you define the Batch Input Session to schedule. It is possible to use wild cards “*” to run multiple Batch Input Sessions.

Task Variable Name	Description
session_name	Name of the Batch Input Session(s) to schedule. Wild cards “*” are supported to run multiple Batch Input Sessions.

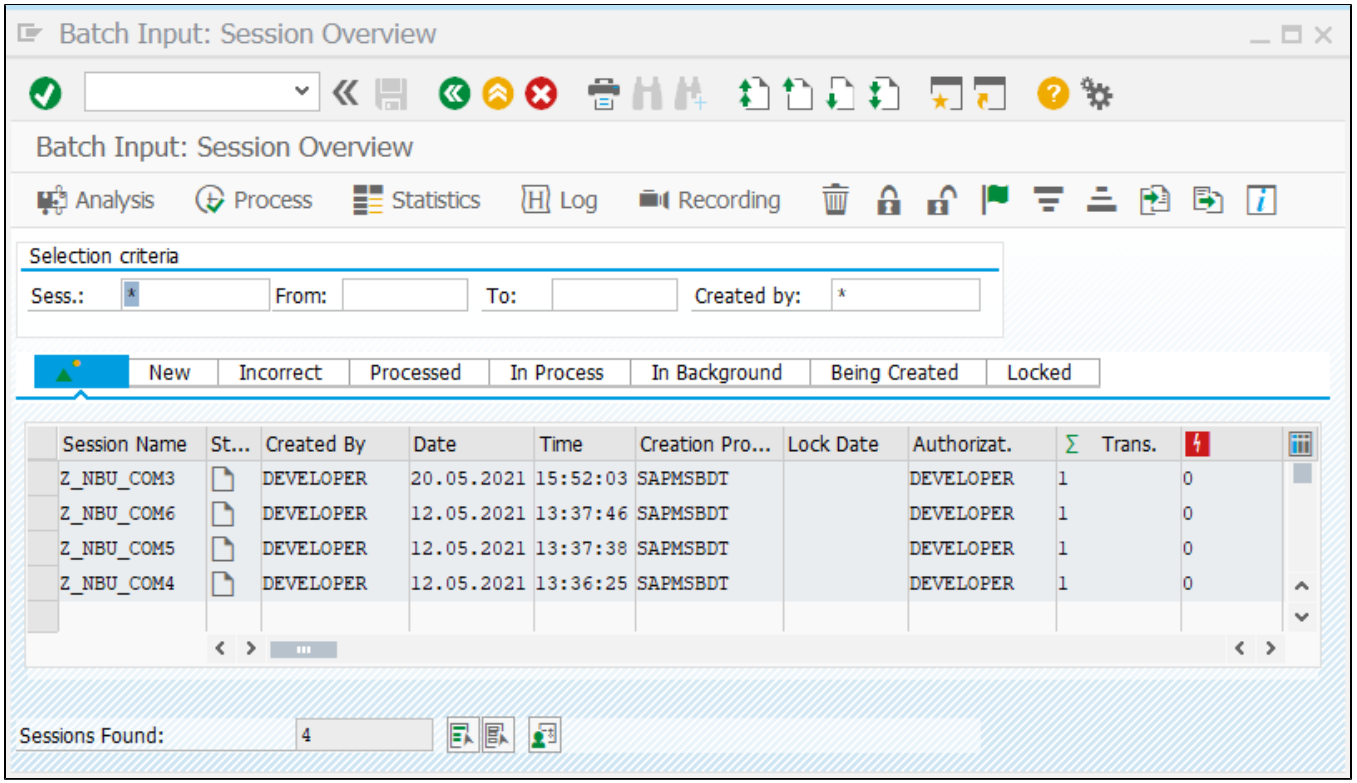
Example for SAP Batch Input Task

The following example runs a Batch Input Session.

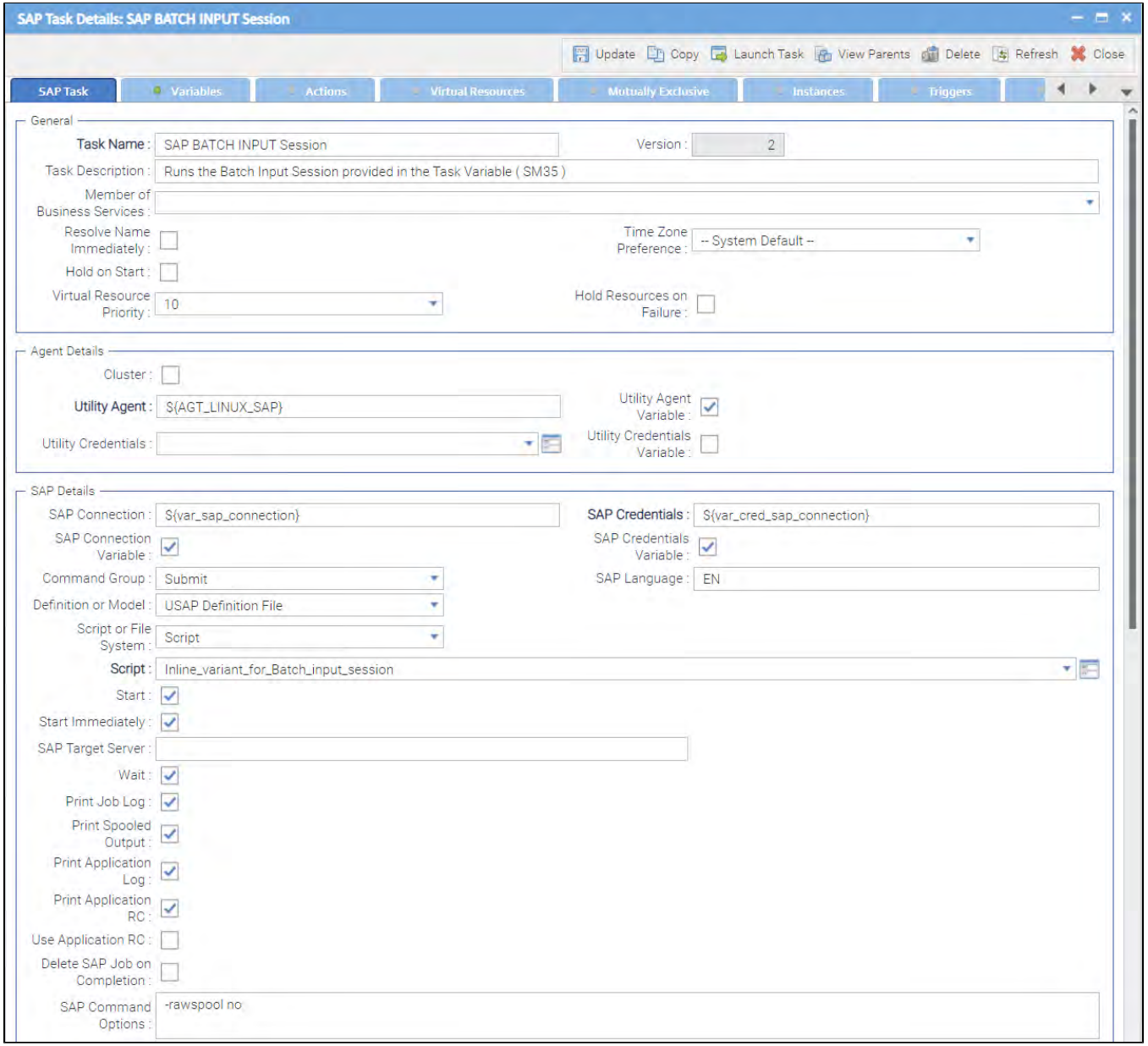
SAP Batch Input: Session Overview

The following image shows the currently configured Batch Input Sessions in SAP.

In the following example, the Batch Input Session: Z_NBU_COM3 will be executed via the SAP Batch Input Task.

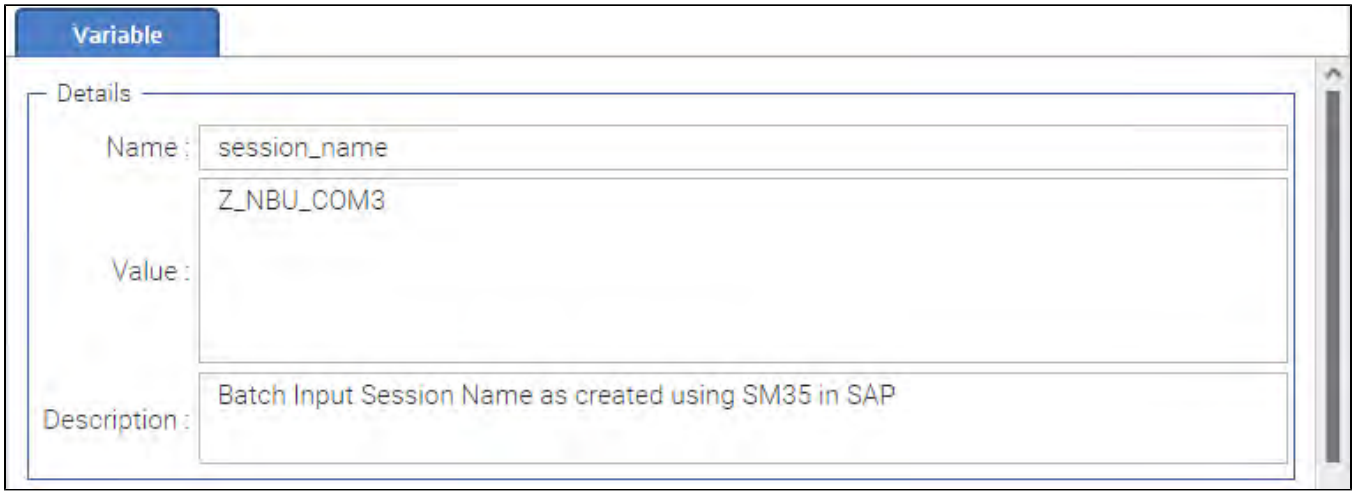


SAP Batch Input Task



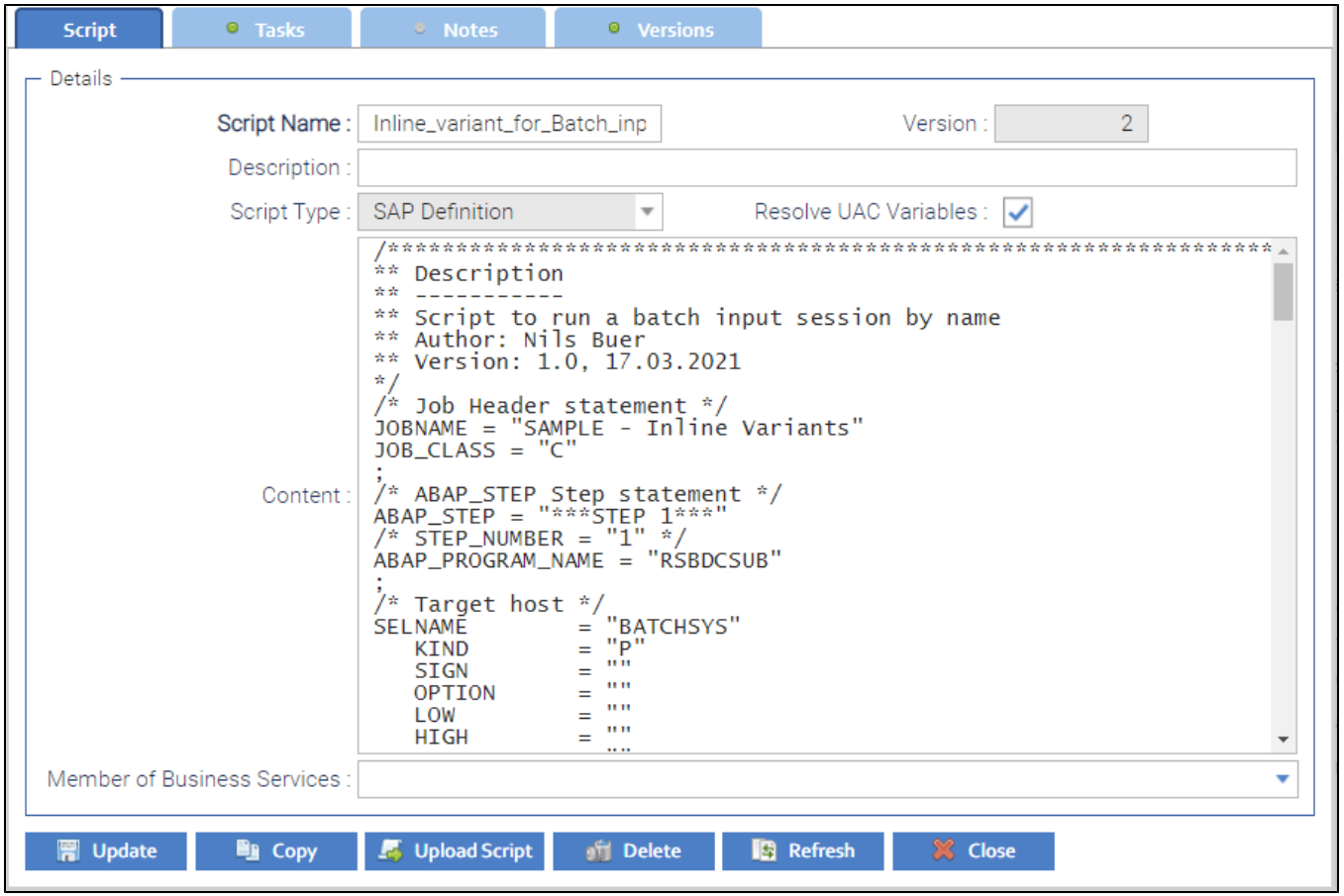
Task Variable

In the task Variable, you define the Batch Input Session to schedule. It is possible to use wild cards “*” to run multiple Batch Input Sessions.



Script

The USAP script dynamically creates a temporary Variant for the Batch Input session name.


Script
Tasks
Notes
Versions
Details
Script Name :
Inline_variant_for_Batch_inp
Version :
2
Description :
Script Type :
SAP Definition
Resolve UAC Variables :
Content :
/***
** Description
** -----------
** Script to run a batch input session by name
** Author: Nils Buer
** Version: 1.0, 17.03.2021
*/
/* Job Header statement */
JOBNAME = "SAMPLE - Inline Variants"
JOB_CLASS = "C"
;
/* ABAP_STEP Step statement */
ABAP_STEP = "***STEP 1***"
/* STEP_NUMBER = "1" */
ABAP_PROGRAM_NAME = "RSBDCSUB"
;
/* Target host */
SELNAME = "BATCHSYS"
KIND = "P"
SIGN = ""
OPTION = ""
LOW = ""
HIGH = ""
Member of Business Services :
Update
Copy
Upload Script
Delete
Refresh
Close

SAP Calendar Import

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Overview

This Universal Task allows you to import the SAP Factory Calendar and the related Holiday Calendar into the Universal Controller.

You can either import a user-defined list of SAP calendar using a csv file or all valid SAP calendar. The Calendar Import can be scheduled to be always in sync with SAP; for example, import all Calendar every day. It is also possible to import Calendars from different SAP System.

Software Requirements

Software Requirements Universal Agents and Controller

- Universal Agent for Linux or Windows Version 6.9.0.0 or later are required

Software Requirements Universal Controller

- Universal Controller 6.9.0.0. or later is required
- A Universal Controller license key with support for SAP connector is required

Software Requirements for the Application to be scheduled

In order to connect to the SAP System the SAP NetWeaver RFC SDK 7.50 libraries are required from SAP.

Those can be downloaded from the SAP Software Download: SAP NetWeaver RFC SDK 7.50

Universal Task for SAP Calendar Import Key Features

Some details about the Universal Tasks for SAP Calendar Import:

- The Universal Task imports either a user defined csv-list or all SAP Factory- and related Holiday Calendar, which match the configured selection criteria into the Universal Controller.
- The Calendar Import can be scheduled to be always in sync with SAP e.g. import all Calendar every day
- The csv list is saved incl. Versioning into the Universal Controller script library
- The Universal Task runs on Linux as well as Windows Universal Agents
- The calendar export is done using the SAP certified XBP-RFC interface
- SAP is always considered as the Master; Calendars are always export from SAP, never import to SAP.
- All Calendar functionalities are support incl. Factory, Holiday and “Special Rules”
- Calendar can be imported from different SAP System - automatically the SID and Client of the SAP, where the import was done is added as a Postfix to the imported Calendar
- You can set different log-levels for the Universal task, providing you more information in case of issues
- Support for Application Server Connection and Destination Connection (nwrfc.ini) e.g. Load Balancer connections, SAP SNC, etc.

Import SAP Calendar Import Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure SAP Calendar Definitions Task

For the new Universal Task type, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for the SAP Calendar Import Definitions Task

Field	Description
Universal Controller URL	Universal Controller URL; for example, Local Universal Controller: http://192.168.88.10:8080/uc/ Stonebranch SaaS Cloud Universal Controller:

	https://superstore.stonebranchdev.cloud/
Universal Controller Credentials	Credentials of the Universal Controller Webservice API
Import All Calendar	Yes | No If “Yes” is selected all SAP Calendar based on the selection criteria entered in the fields: Start Year and Years are imported. If “No” is selected the Calendar provided in the Calendar Input File will be imported to Universal Controller
Start Year	From this year onwards the Calendar will be imported from SAP. Example: Start Year = 2020, Years = 2 will import all calendar starting from 2020 until 2022
Years	Number of years to import starting from the year provide in the field Start Year.
First Day of Week	[Monday,Tuesday, Wednesday, Thursday, Friday, Sunday] Set the first day of the week for the calendar

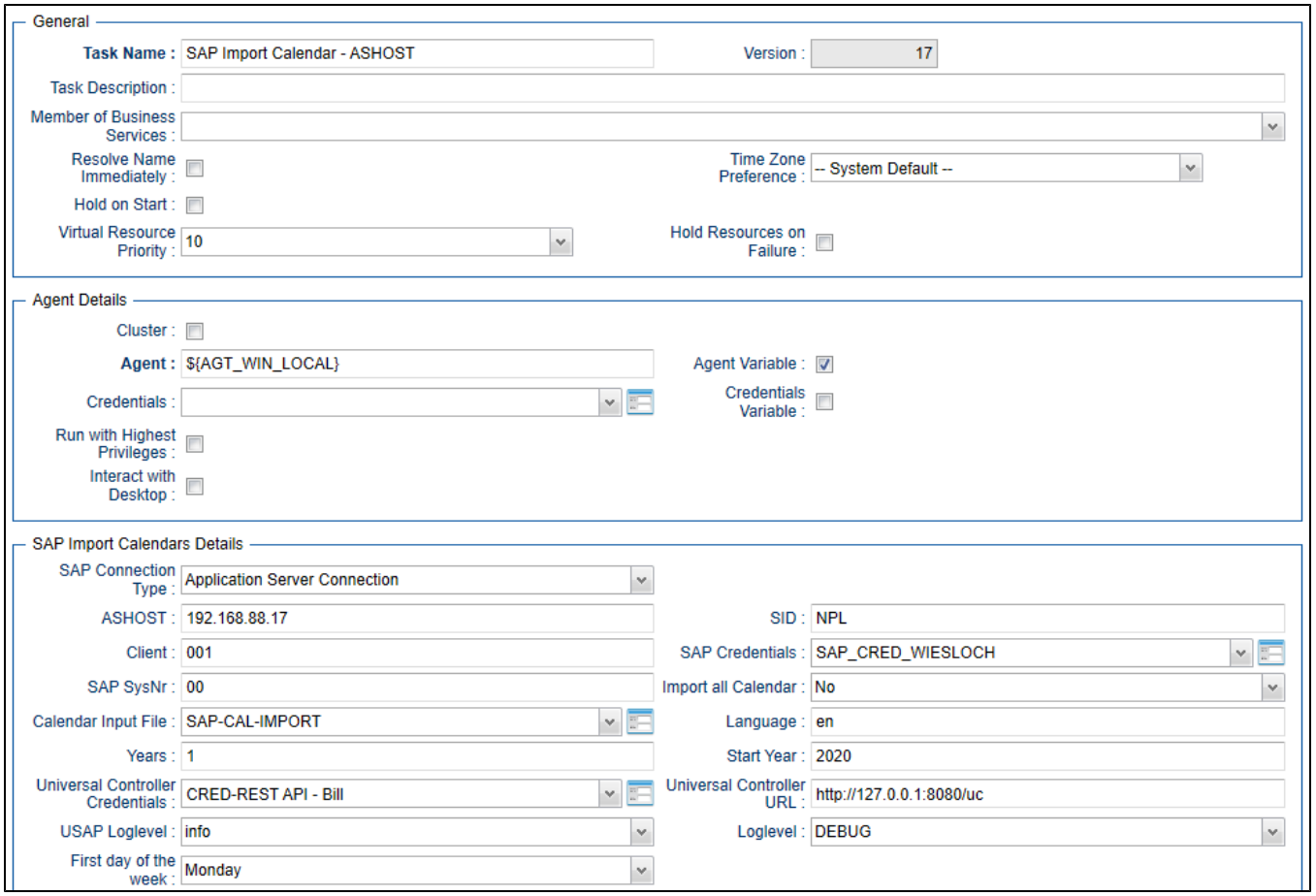
Calendar Input File	This field is only used when the choice field import all calendar is set to No.
	
	The Calendar Input File file has the following structure:
	
	NOTE: If a factory calendar has no holiday calendar, then leave the filed empty; for example, 02, means extract factory calendar 02 without a holiday calendar.
	
	The format is:
	
	<Factory Calendar ID>,<Holiday Calendar ID>
	
	Example: calendar_import.csv
	
	`factory_calid,holiday_calid`
	`01,08`
	`02,`
	`AJ,04`

Language	SAP Language e.g. EN for English
SAP Connection Type	[Application Server Connection | NRFW.ini Connection)] Select the SAP Connection Type: Application Server Connection or Destination Connection using the nwrfc.ini file. Default location for the nwrfc.ini file is: • Linux: /opt/universal/uagsrv • Windows: C:\Program Files\Universal\UAGSrv
SAP Credentials	Credentials of the SAP System
SAP ASHOST	SAP Application to connect to. This field is only visible in case of SAP Connection Type = “Application Server Connection”
SID	SAP System Identifier (3 digits); for example, NPL

	The Imported calendar will have the SID as prefix Imported Calendar Name: <**SID**>_<SAP CLIENT>_<SAP CALENDAR ID>
SAP CLIENT	SAP Client (3 digits); for examaple, 100 The Imported calendar will have the SID as prefix Imported Calendar Name: <SID>_<**SAP CLIENT**>_<SAP CALENDAR ID>
SAP SysNr	SAP System Number (2 digits) e.g. 00
SAP Destination	SAP Destination in the nwrfc.ini. This field is only visible in case of SAP Connection Type = "*NRFW.ini Connection*"
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]
USAP loglevel	Loglevel of the Universal Connector for SAP trace | audit | info | warn | error

Calendar Input File	This field is only used when the choice field import all calendar is set to No.
	The Calendar Input File file has the following structure:
	NOTE: If a factory calendar has no holiday calendar, then leave the filed empty; for example, 02, means extract factory calendar 02 without a holiday calendar.
	The format is:
	<Factory Calendar ID>,<Holiday Calendar ID>
	Example: calendar_import.csv
	`factory_calid,holiday_calid`
	`01,08`
	`02,`
	`AJ,04`

Example: SAP Import SAP Calendar - Application Server Connection



SAP Calendar Import Name

The Imported calendar will have the following naming in Universal Controller:

<SID>_<**SAP CLIENT**>_<SAP CALENDAR ID>

Example

NPL_100_01

SID = NPL, CLIENT = 100, SAP Factory Calendar ID = 01

SAP Data Services

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

Using analytics tools to collect massive amounts of Big Data from your organization is one thing. Extracting meaning from that data and using it to drive real growth is another. Business Objects analytics from SAP can help you unleash the power of collective insight by delivering enterprise business intelligence, agile visualizations, and advanced predictive analytics to all users.

Leverage the capabilities of SAP® Business Objects and schedule any SAP Business Object Data Services ETL job in Stonebranch's Universal Automation Center by using the “AL_RWJobLauncher.exe” utility, which comes with the SAP Data Services installation.

This Universal Task allows you to execute an SAP Data Services “ETL” Job using the “AL_RWJobLauncher.exe”.

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 6.9.0.0 or later is required.
- Universal Agent needs to be installed with python option (--python yes).

Software Requirements for Universal Controller

- Universal Controller 6.9.x or later is required.

Software Requirements for the Application to be Scheduled

- This Universal Task was tested against SAP Data Services 4.2 SP7.
- The Data Service Server needs to have a Universal Agent installed to call the *AL_RWJobLauncher.exe* utility.

Key Features

- It is based on the "*AL_RWJobLauncher.exe*". which is part of the Data Services Install.
- The Task runs on Data Services for Windows and Linux.
- The Universal Task provides the same error and trace information as the SAP Data Services Mgt. Console.
- You can select different log-levels ; for example, Info and Debug.
- You can configure all connection Parameters via the Universal Task.
- For all Parameters an exception handling has been implemented.

Current limitations

- Currently,only "Enterprise Security" is supported.

Import SAP Task for Batch Input

To use the SAP Data Services Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Download the provided ZIP file.
3. In the Universal Controller UI, select Administration >Configuration > Universal Templates to display the current list of Universal Templates.
4. Click Import Template.
5. Select the template ZIP file and Import.

When the template has been imported successfully, the Universal Template will appear on the list. Refresh your Navigation Tree to see these tasks in the Automation Center Menu.

Note

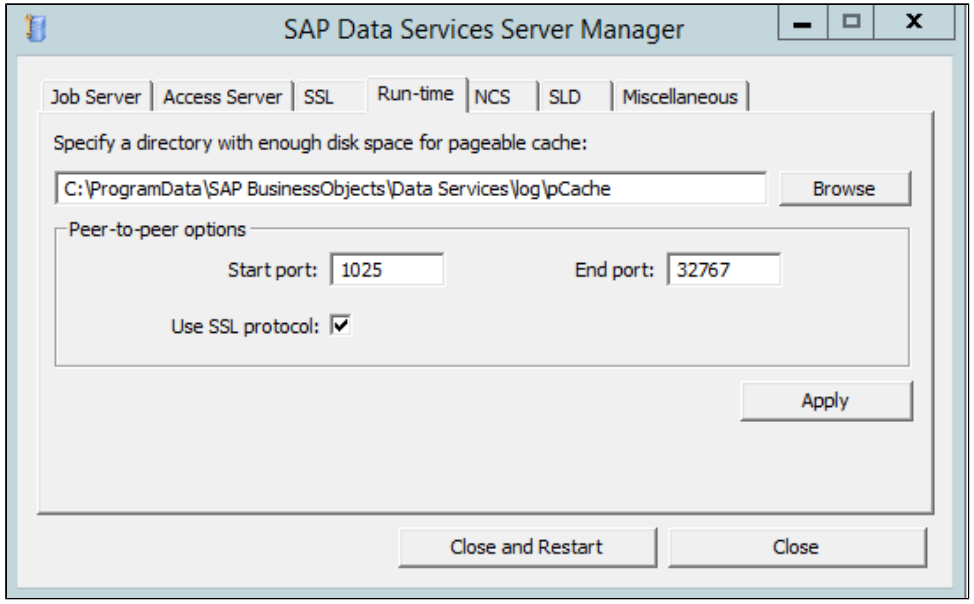
If you have Universal Controller 6.9., you must import the Universal Template via an XML list import. In Universal Controller, go to "All Tasks" and load the Universal Template XML file, via the Import function, into the Controller.

Configure SAP Data Services Tasks

For Universal Task SAP Data Services, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for SAP Data Services Tasks

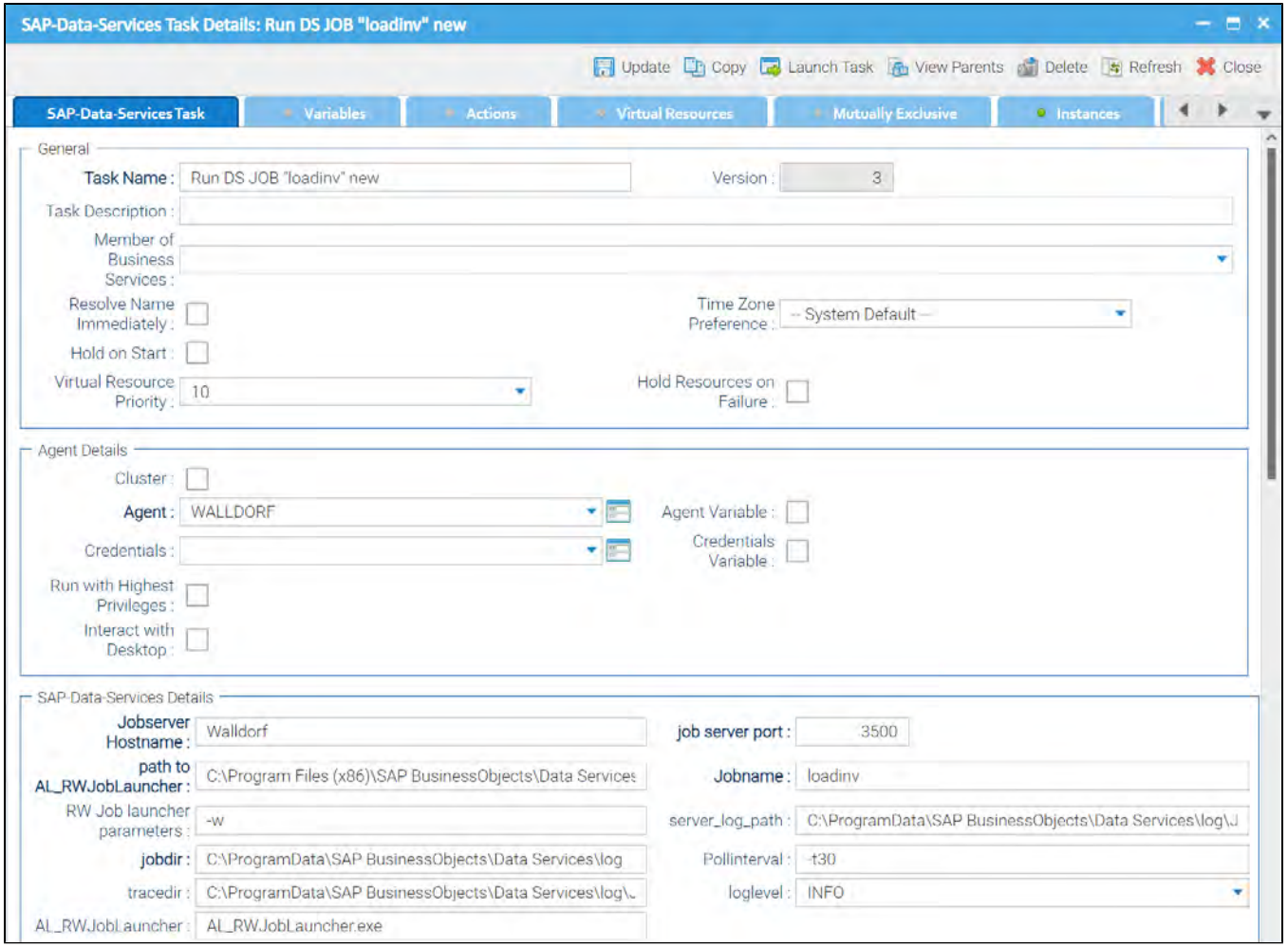
The following fields must be configured for the SAP Data Services Task.

Field	Description
Agent	Universal Agent installed on the SAP Data Services Server. This Agent runs the AL_RWJobLauncher.exe CLI.
Jobserver Hostname	Name of the Jobserver, which runs the Data Services Job. The Jobserver can be looked up in the SAP Data Service Management Console under: Administrator -> Batch
Job server port	Port of the Jobserver, which runs the Data Services Job. The Jobserver Port can be looked up in the SAP Data Service Management Console under: Administrator -> Batch
Path to AL_RWJobLauncher.exe	Location, where the AL_RWJobLauncher.exe is installed on the SAP Data Services Server.
Jobdir	Directory, where the Job execution files are exported to. The directory must contain the following files: <Jobname>.txt and <jobname>.bat Note: Those files are available only if the job Export job execution command has been performed in advance in the SAP Data Service Management Console. In the example above, the files have the names: • *loadinv.txt* • *loadinv.bat* To find the Location, open the data services Manager Tab: Run-time. The jobdir is the directory without pCache. 

tracedir	Directory where the trace and log files are located. The directory should contain trace/error files, which look as follows: *trace_MM_DD_YYYY_HH_MM_SS_...* *error_MM_DD_YYYY_HH_MM_SS_...*
Server_log_path	Path to the server log. It should contain the file: *server_eventlog_YYYY-MM-DD.txt*
loglevel	Logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Poll interval	Interval, in seconds, for how often the Data Services Server is polled to get the current Job status.
Jobname	Name of the Data Services Job. It can be looked up in the SAP Data Service Management Console under: Administrator -> Batch
Log level (default is Inherited)	Universal Task logging settings: [Inherited | Trace | Debug | Info | Warn | Error | Severe]

Example forSAP Data Services Task

The following Task start the Data Service Job: loadinv on the server Walldorf



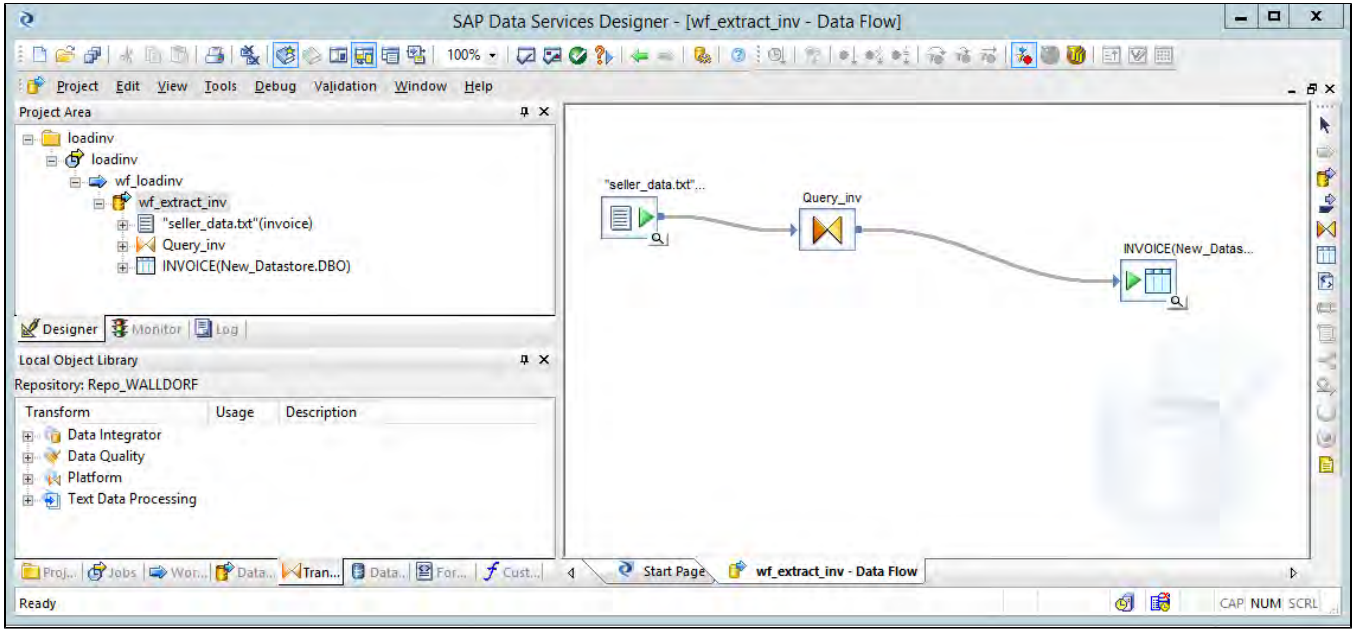
SAP Data Services Processing Steps

The following describes how to run a Data Services Task.

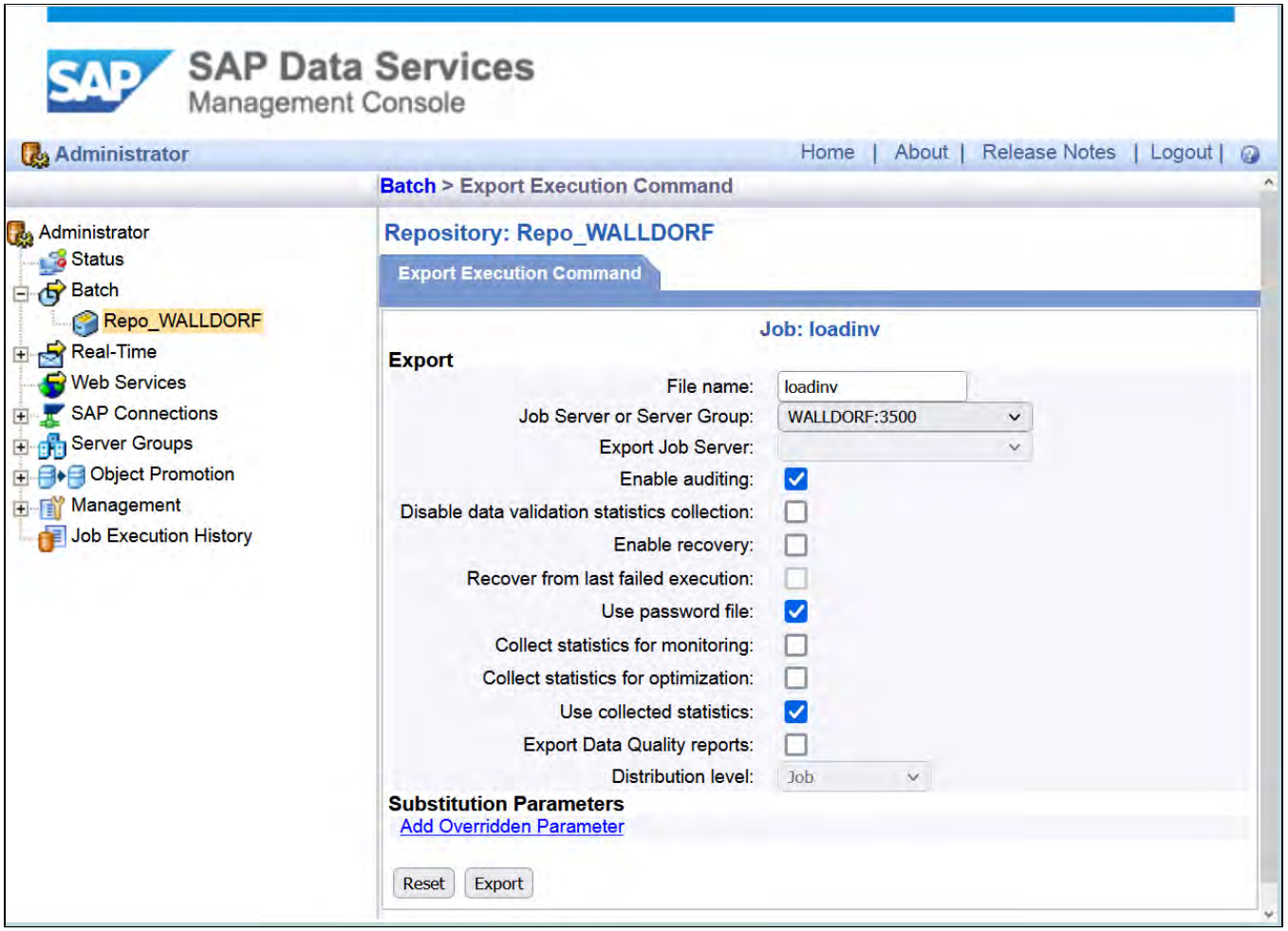
The Data Services Data Flow (Step 1) is only shown for completeness so that that Scheduling Operations team members with limited SAP Data Services Knowledge understand the concept.

1. Job Configuration in Data Service Designer

Configure the SAP Data Services ETL job in the Data Service Designer.

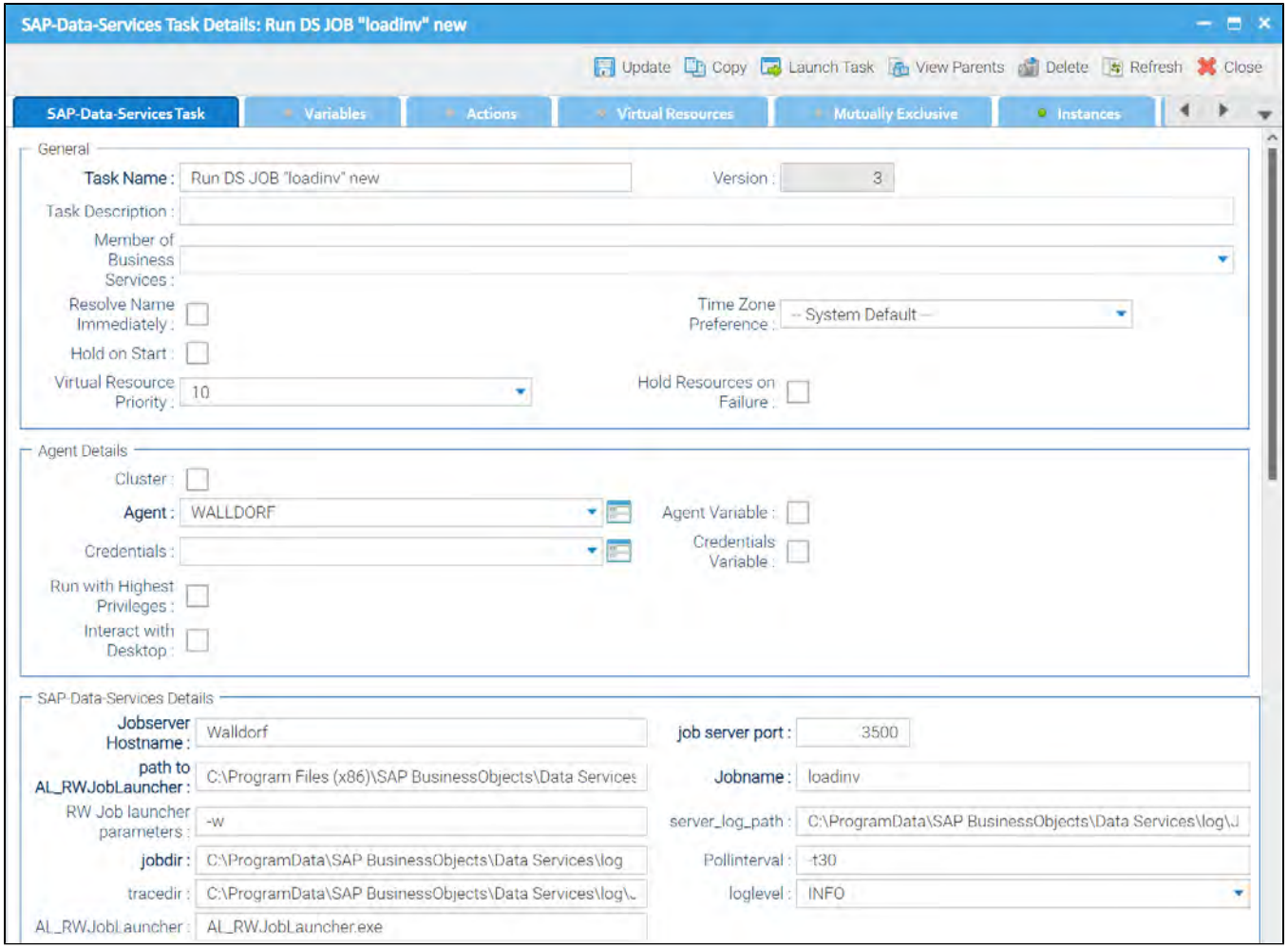


2. Export job execution command in SAP Data Service Management Console



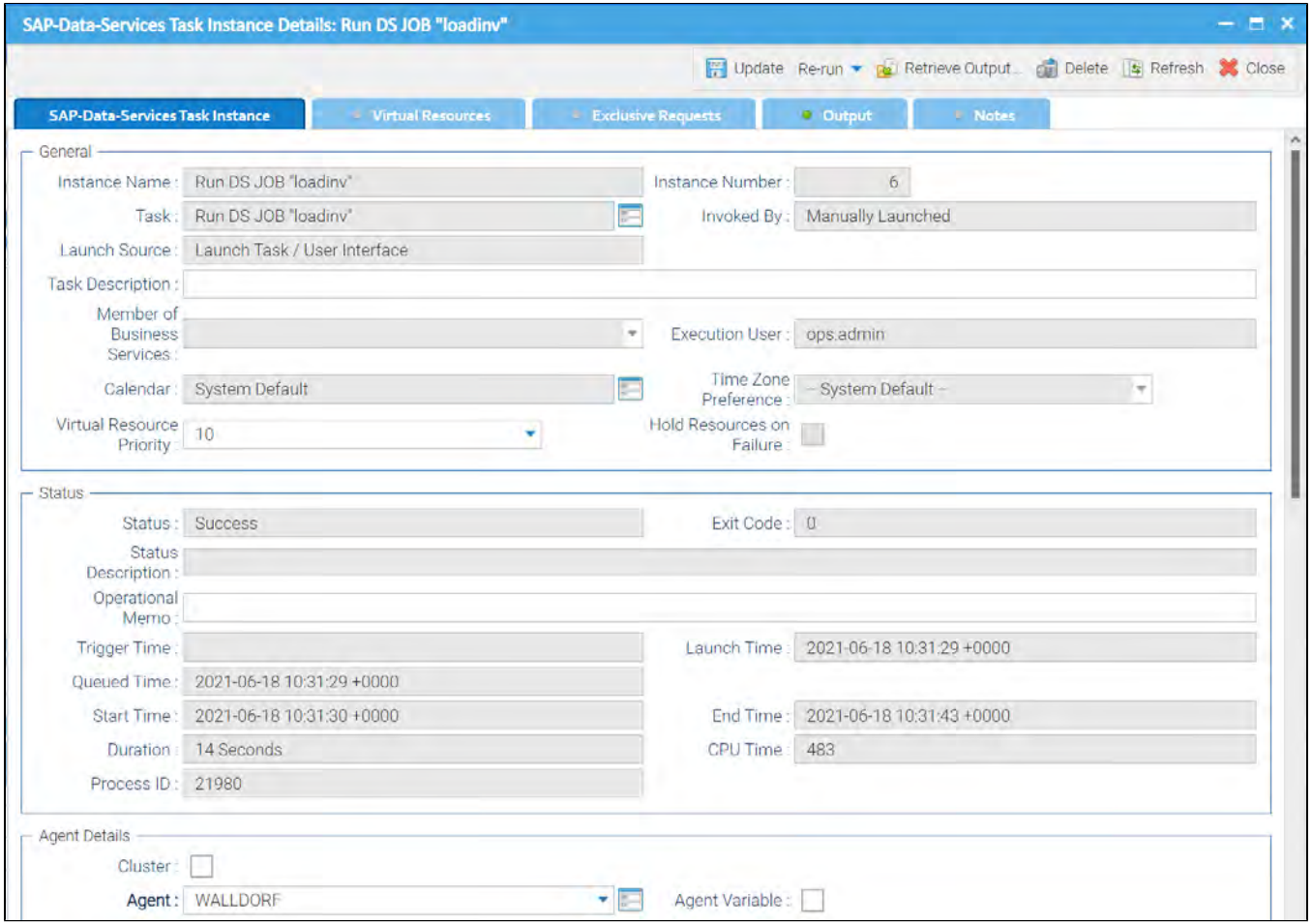
3. Configure the SAP Data Services Task to run the exported job in Universal Controller

Universal Task configuration:

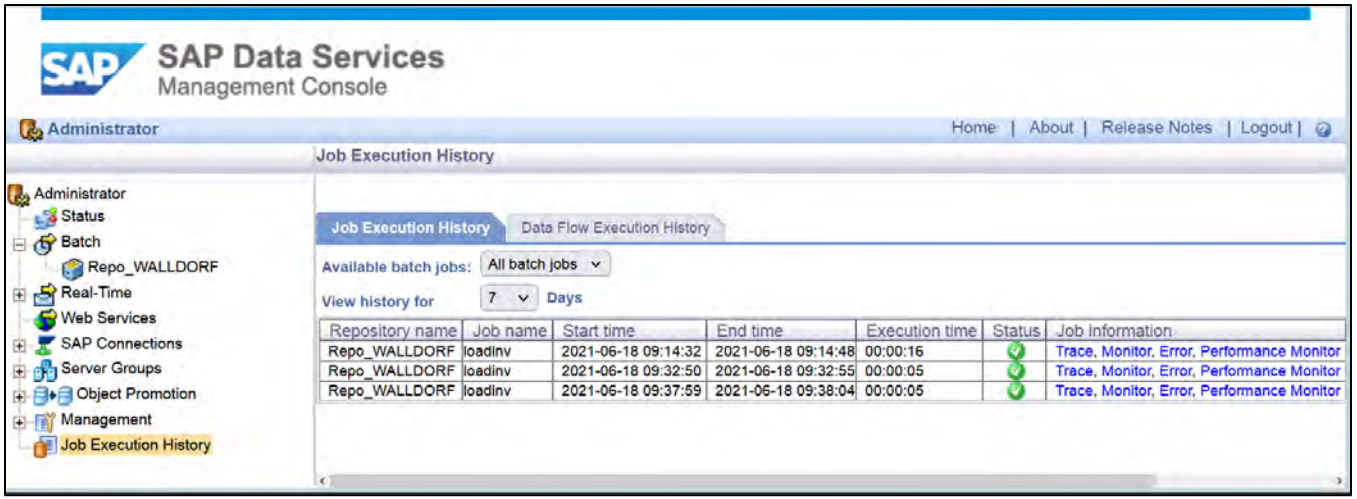


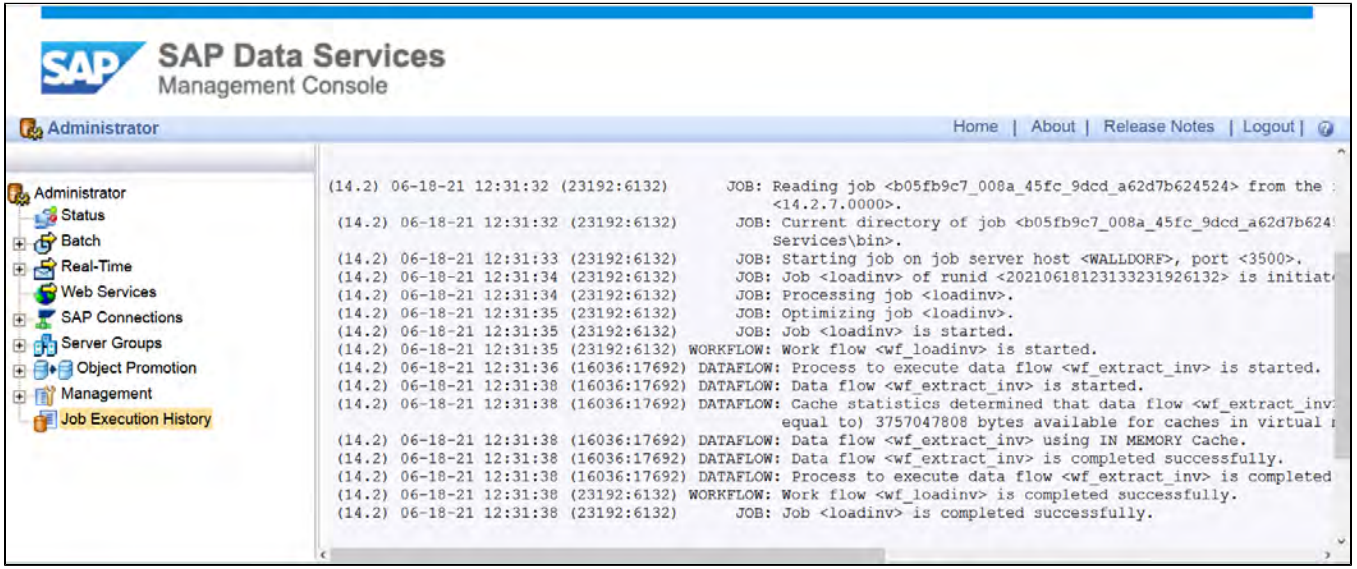
4. Launch the Universal Task

Universal Task Instance after the launch of the task:



5. Verify job execution in SAP Data Service Management Console

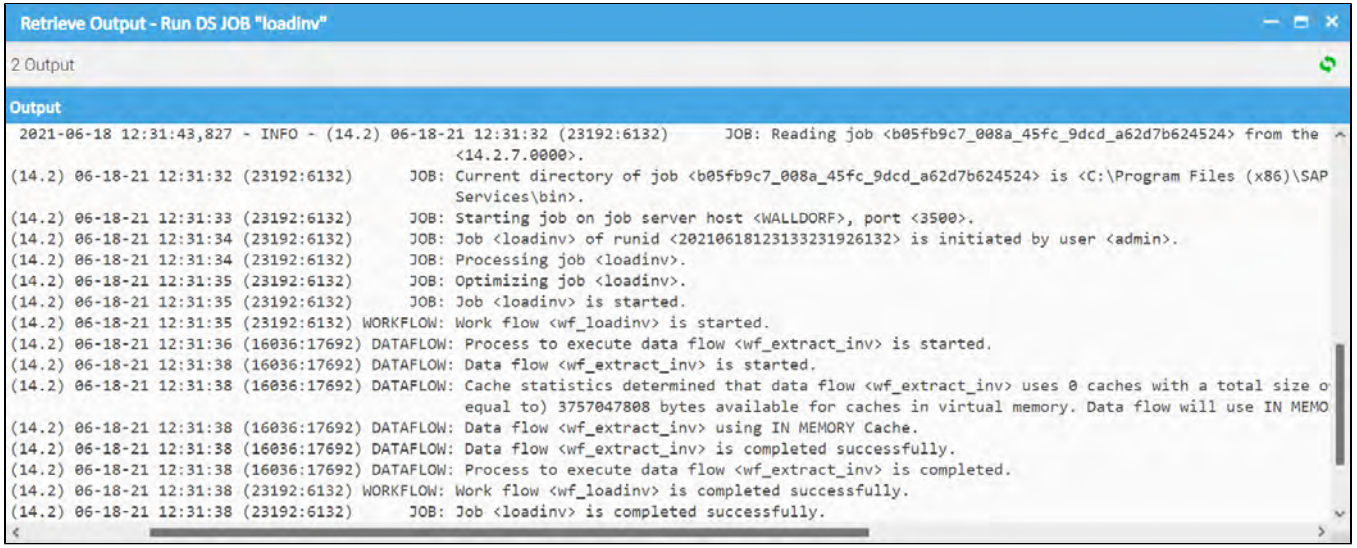




Note

The Log and trace file in the SAP Data Service Management Console are the same as in the output of the Universal Task.

The following shows the Universal Task Output:


Retrieve Output - Run DS JOB "loadinv"
2 Output
Output
2021-06-18 12:31:43,827 - INFO - (14.2) 06-18-21 12:31:32 (23192:6132) JOB: Reading job <b05fb9c7_008a_45fc_9dcd_a62d7b624524> from the <14.2.7.0000>.
(14.2) 06-18-21 12:31:32 (23192:6132) JOB: Current directory of job <b05fb9c7_008a_45fc_9dcd_a62d7b624524> is <C:\Program Files (x86)\SAP Services\bin>.
(14.2) 06-18-21 12:31:33 (23192:6132) JOB: Starting job on job server host <WALLDORF>, port <3500>.
(14.2) 06-18-21 12:31:34 (23192:6132) JOB: Job <loadinv> of runid <20210618123133231926132> is initiated by user <admin>.
(14.2) 06-18-21 12:31:34 (23192:6132) JOB: Processing job <loadinv>.
(14.2) 06-18-21 12:31:35 (23192:6132) JOB: Optimizing job <loadinv>.
(14.2) 06-18-21 12:31:35 (23192:6132) JOB: Job <loadinv> is started.
(14.2) 06-18-21 12:31:35 (23192:6132) WORKFLOW: Work flow <wf_loadinv> is started.
(14.2) 06-18-21 12:31:36 (16036:17692) DATAFLOW: Process to execute data flow <wf_extract_inv> is started.
(14.2) 06-18-21 12:31:38 (16036:17692) DATAFLOW: Data flow <wf_extract_inv> is started.
(14.2) 06-18-21 12:31:38 (16036:17692) DATAFLOW: Cache statistics determined that data flow <wf_extract_inv> uses 0 caches with a total size o
equal to) 3757047808 bytes available for caches in virtual memory. Data flow will use IN MEMO
(14.2) 06-18-21 12:31:38 (16036:17692) DATAFLOW: Data flow <wf_extract_inv> using IN MEMORY Cache.
(14.2) 06-18-21 12:31:38 (16036:17692) DATAFLOW: Data flow <wf_extract_inv> is completed successfully.
(14.2) 06-18-21 12:31:38 (16036:17692) DATAFLOW: Process to execute data flow <wf_extract_inv> is completed.
(14.2) 06-18-21 12:31:38 (23192:6132) WORKFLOW: Work flow <wf_loadinv> is completed successfully.
(14.2) 06-18-21 12:31:38 (23192:6132) JOB: Job <loadinv> is completed successfully.

SAP Event History Monitor

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task queries the SAP Event history table for a selected SAP Event & Parameter. If the Event is found, it gets confirmed, so that it is not triggered again. Optionally, a task can be launched based on the occurrence of an Event & Parameter.

Software Requirements

Software Requirements for Universal Agent

- Universal Agent for Linux or Windows Version 7.0.0.0 or later is required.
- Universal Agent needs to be installed with python option (--python yes).

Software Requirements for Universal Controller

- Universal Controller 7.0.0.0 or later is required.

Software Requirements for the Application to be Scheduled

SAP Credentials (Username and Password) to connect via the XBP-RFC Interface to SAP ERP System is required.

Key Features

Some details about Universal Tasks to monitor the SAP Event History:

- Query the SAP Event history table for a selected SAP Event & Parameter.
- For the Parameters, wildcards "*" are supported.
- Confirm Events in the SAP History Table (SM62). When the Event and Parameter have been identified, the Event is confirmed in SAP so that the monitor will not trigger a second time (optional setting).
- Launch a task in Universal Controller based on the occurrence of an Event & Parameter.
- The Universal Task is supported on Linux and Windows Agents.
- An automatic re-start of the task can be achieved by adding an action to the Universal Task, which restarts the task in the case of status success.
- This Universal Task supports SAP Application Server and SAPNWRFC.INI Destination connections.
- You can set different log-levels for the Universal task, which provides you more information in case of issues.

Import the SAP Event History Monitor Universal Template

To use theSAP Event History Monitor Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to true. For more information about Resolvable Credentials click here.
2. Download the provided ZIP file.
3. In the Universal Controller UI, select Administration >Configuration > Universal Templates to display the current list of Universal Templates.
4. Click Import Template.
5. Select the template ZIP file and Import.

When the template has been imported successfully, the Universal Template will appear on the list. Refresh your Navigation Tree to see these tasks in the Automation Center Menu.

Configure SAP Task for Batch Input

For Universal Task SAP Event History Monitor, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for SAP Task for Batch Input

Fill Out the Universal Task for each SAP Event and Parameter to monitor.

Field	Description
Agent	Linux or Windows Universal Agent to run the USAP commands.
Agent Cluster	Optional Agent Cluster for load balancing.
SAP Connection Type	[Application Server Connection, SAPNWRFC.INI Connection] The Universal Task support SAP Application Server and SAPNWRFC.INI Destination connections. **Application Server Connection**:

	• **Client**: SAP Client to connect to; for example, 100 • **SAP SysNr**: SAP System Number; for example, 00 • **ASHOST:** SAP Application Server hostname or IP; for example, walldorf **SAPNWRFC.INI Connection:** • **Client:** SAP Client to connect to; for example, 100 • **Destination:** name of the destination entry in the file sapnwrfc.ini; for example, walldorf Note The file sapnwrfc.ini needs to be in the home directory of the user executing the task. If no user credentials are provided, this is: • Linux: /opt/universal/uagsrv • Windows: C:\Program Files\Universal\UAGSrv
SAP ASHOST	SAP Application to connect to. This field is visible only if SAP Connection Type = “Application Server Connection”
SAP CLIENT	SAP Client (3 digits); for example, 100 The Imported calendar will have the SID as prefix Imported Calendar Name: <SID>_<**SAP CLIENT**>_<SAP CALENDAR ID>
SAP SysNr	SAP System Number (2 digits) e.g. 00 This field is visible only if SAP Connection Type = “*Application Server Connection*”
SAP Destination	SAP Destination in the nwrfc.ini. This field is visible only if SAP Connection Type = “*NRFW.ini Connection*”
USAP Dir	Directory where the USAP binary is stored Linux: /opt/universal/usap/bin Windows: C:\Program Files\Universal\USap\bin
Event ID	Name of the Event to Scan for in the SAP Event History Table.
Event Parameter	Name of the parameter to scan for. Note: wildcard “*” *is supported ; for example, ua** searches for all event parameters beginning with ua. If no Event Parameters is provided, any Event Parameter will match.
Event Status to Select	[New, Confirmed, Any Status] • **NEW**: scan only for new Events (default). • **Confirmed**: scan only for confirmed Events. • **Any Status**: scan for any status of Events.

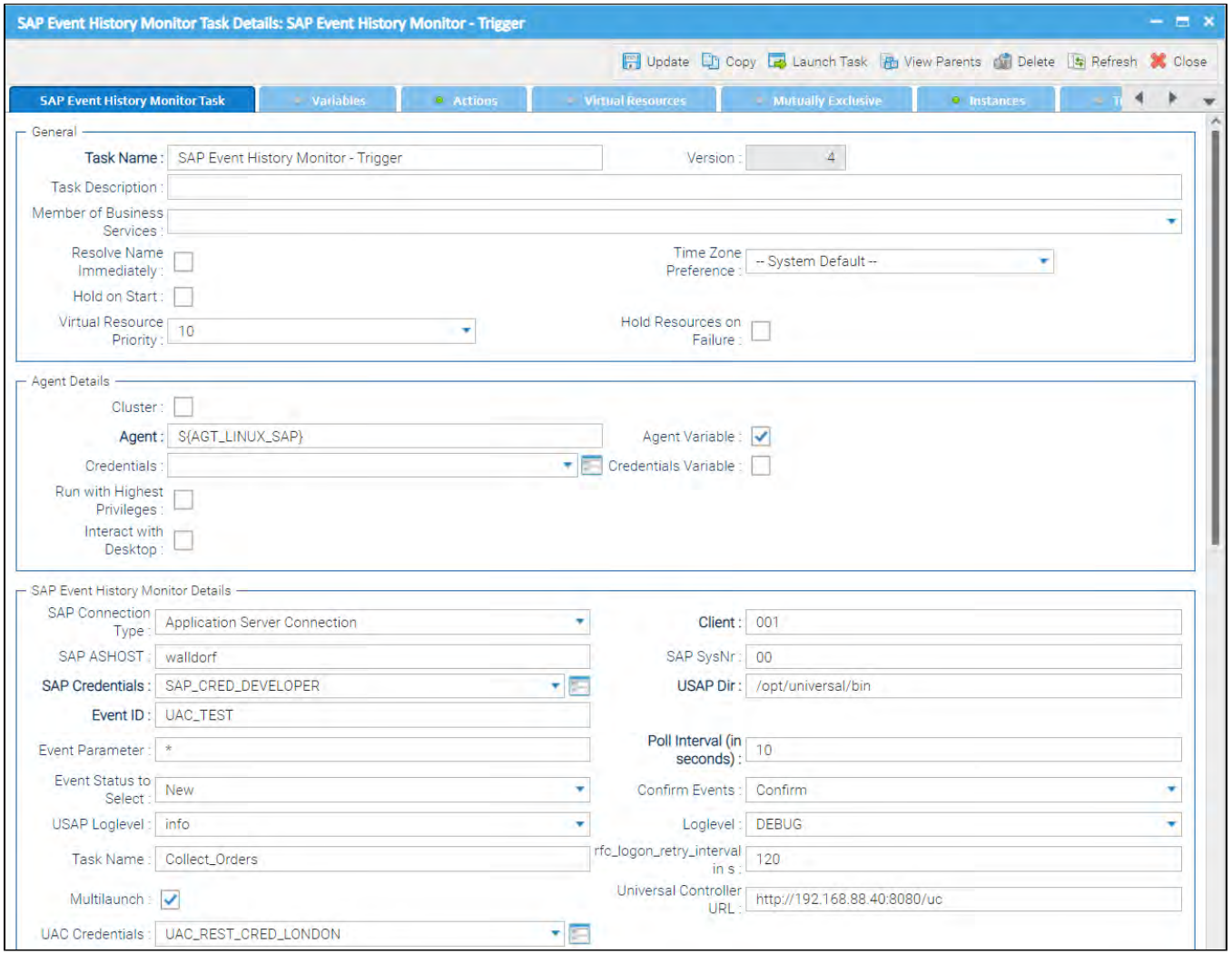
Universal Controller URL	Universal Controller URL. For example, https://192.168.88.40/uc
Taskname	Name of the task to start in case a new event has been identified. If no task is specified, the UT runs in Monitoring mode and goes to success in case an Event has been identified; for example, you can add the Event Monitor to a Workflow (Note: in that case remove the action, which automatically re-starts the UT in case of status “success”)
Confirm Events	[Confirm , New] Default is “Confirm”. **Confirm**: This ensures that the same event only triggers the event monitor once. **New**: Leaves the event in status “New”, mainly for testing purpose.
USAP loglevel	Log level of the Universal Connector for SAP trace | audit | info | warn | error
rfc_logon_retry_interval in s	interval in seconds between an new logon attempt to SAP. Per default, 10 retries will be performed; for example, 120s means every 120s a new logon attempt to SAP is performed. If after 10 retries no logon to SAP was possible the Task fails.
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example for SAP Event History Monitor Task

The following example monitors the SAP Event: UAC_TEST with Parameter: UAC in the SAP Event History.

If the Event occurs in the SAP Event history table with status “NEW”, the Task “Collect_Orders” is launched and the Event is confirmed in SAP.

In the following example, the Parameter “Multilaunch” is enabled. This means that if several events are found with the status “New”, the task “Collect_Orders” will be launched for each confirmed event.



Event Configuration in SAP

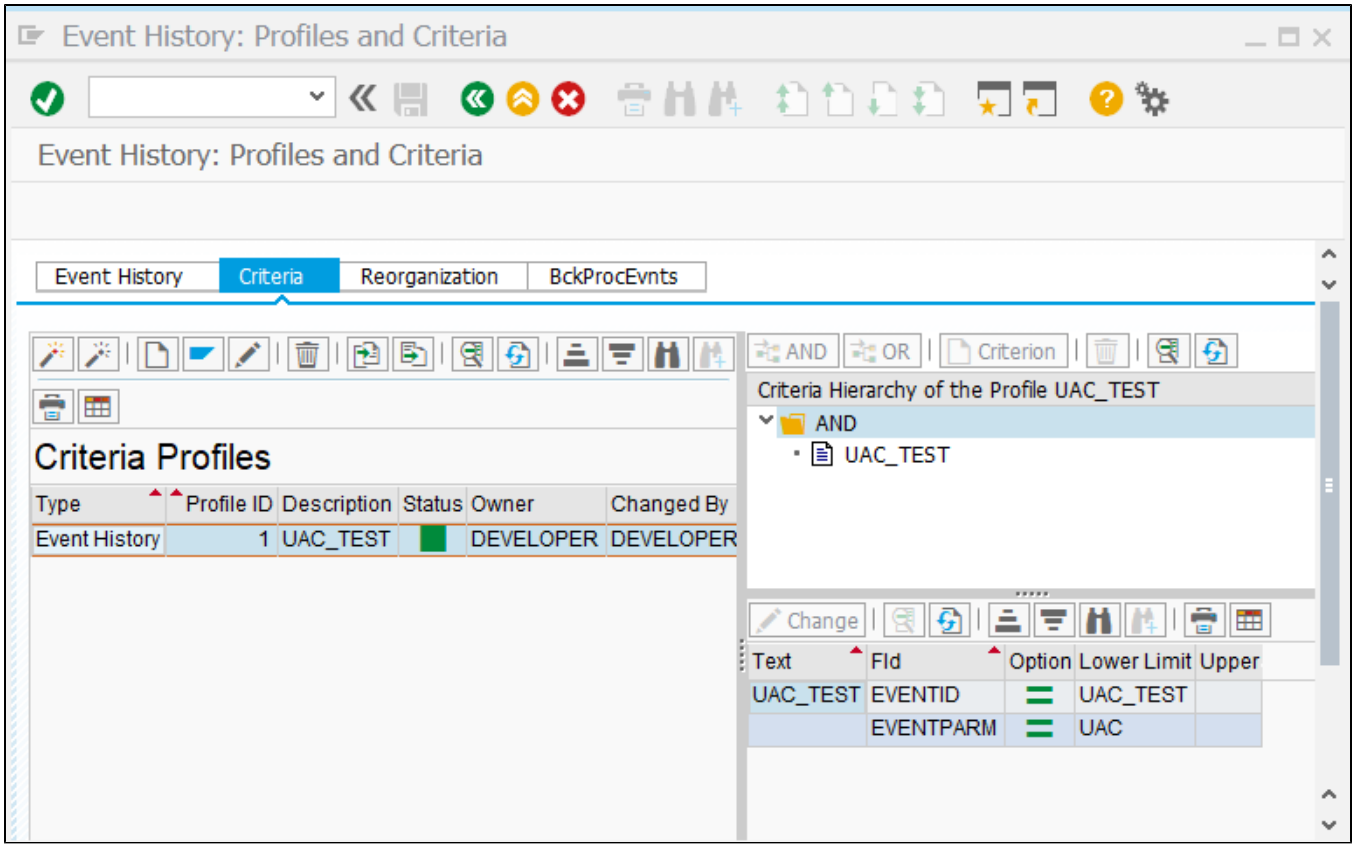
The SAP History Event Task Monitor scans for Events in the SAP Event history.

An Event only shows up in the Event history if an appropriate event criteria profile has been set-up in SAP by using transaction SM62.

Note

Optionally, a criteria profile can also be set-up via an SAP Task of command group “Set CM Profile” in Universal Controller.

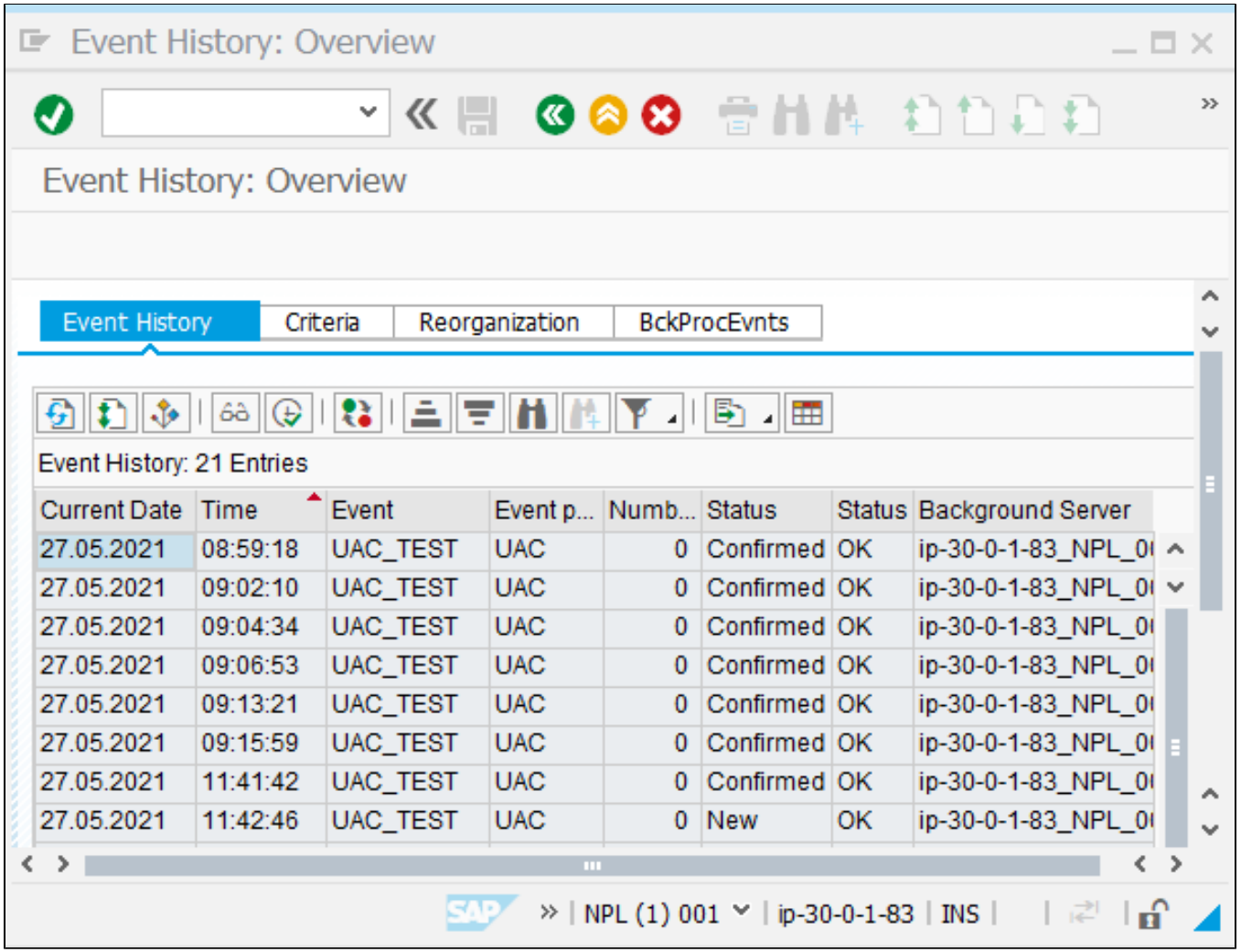
The following screen shows an example of the set-up in SAP using SM62.



Event History Table

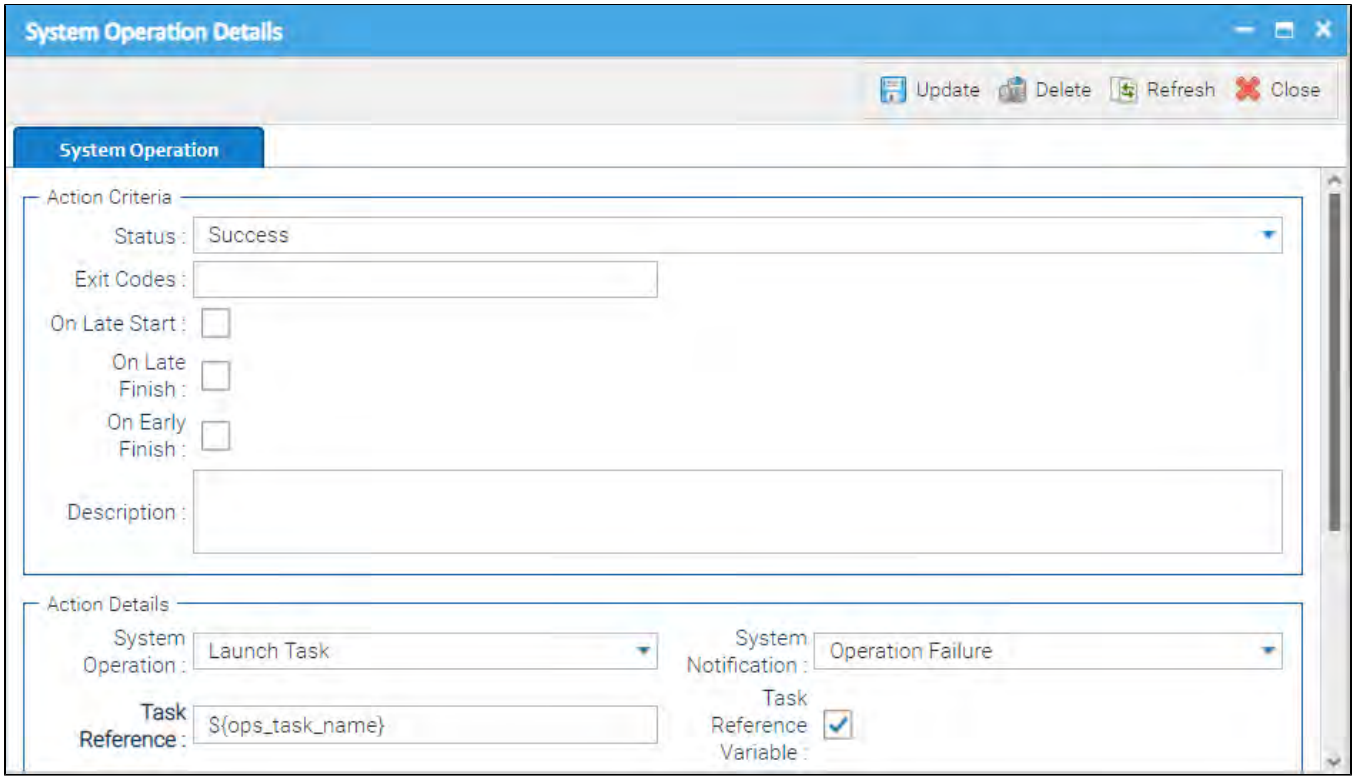
The following provides and example of the Event history table in SAP (SM62).

Only Events showing up here can trigger the SAP Event History Monitor UT.



Task Monitor Trigger mode

You can create a Task Monitor trigger from this task by adding an action to the Universal Task, which restarts the task in the case of status success.


System Operation Details
Update
Delete
Refresh
Close
System Operation
Action Criteria
Status :
Success
Exit Codes :
On Late Start :
On Late Finish :
On Early Finish :
Description :
Action Details
System Operation :
Launch Task
System Notification :
Operation Failure
Task Reference :
${ops_task_name}
Task Reference Variable :

SAP Extract Job Definitions

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Overview

This Universal Task allows you to export SAP Job definitions from SAP into one flat file for each Job selected for extraction. The Jobs definitions to extract from SAP are provided in a CSV input file saved in the Universal Controller script library. For each Job to extract the SAP Jobname and SAP Job count ID needs to be provided in the input file.

The extracted SAP Job definition files can then by read by the Stonebranch Conversion Tool (CTK), which transitions each read Job definition file into an SAP Task. If the SAP Job was in status Released, meaning Start conditions had been defined in SAP for that Job, than automatically a time trigger with the scheduling criteria will be created by the transition tool. As result the transitioned SAP Task are ready to be scheduled in the same way as in SAP.

This document focuses on the Universal Task, which extracts the Job definitions from SAP. The Stonebranch Conversion Tool (CTK) to read the extracted Job definitions files and create SAP Tasks and related trigger from it is described in the CTK documentation.

Software Requirements

Software Requirements Universal Agents and Controller

- Universal Agent for Linux or Windows Version 6.9.0.0 or later are required

Software Requirements Universal Controller

- Universal Controller 6.9.0.0. or later is required
- A Universal Controller license key with support for SAP connector is required

Software Requirements for the Application to be Scheduled

In order to connect to the SAP System the SAP NetWeaver RFC SDK 7.50 libraries are required from SAP.

Those can be downloaded from the SAP Software Download: SAP NetWeaver RFC SDK 7.50

SAP Extract Job Definitions Key Features

- EDxport SAP Job definitions from SAP into one flat file for each SAP Job
- Also any defined start criteria will be exported for SAP each Job
- Jobs to be extracted can be in any Status (for example, SCHEDULED, RELEASED, CANCELED, FINISHED)
- The Jobs to be extracted are provide via list saved in the Universal Controller script library
- A detailed log file will be provided after each extraction process to identify Jobs which could not be found in SAP; for example, a Jobname has been provided in the input file, which does not exists in the SAP System
- The extracted SAP Job definition files can than by read by the Stonebranch Conversion Tool (CTK), which transitions each read Job definition file into an SAP Task, including defined Start criteria
- Support for Application Server Connection and Destination Connection (nwrfc.ini); for example, Load Balancer connections, SAP SNC)

Import SAP Extract Job Definitions Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure SAP Extract Job Definitions Task

For the new Universal Task type, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for the SAP Extract Job Definitions Task

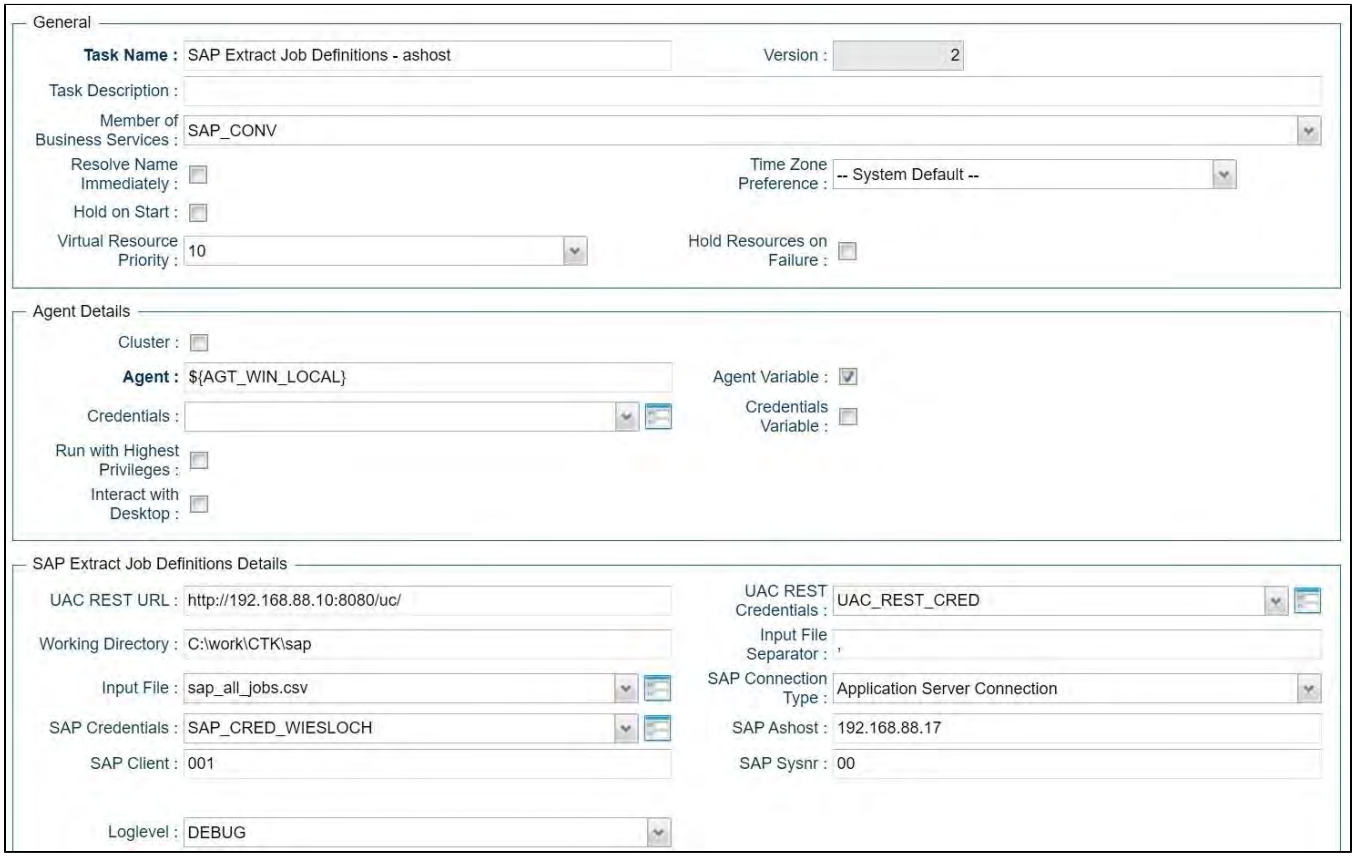
Field	Description
UAC REST URL	Universal Controller URL; for example: Local Universal Controller: http://192.168.88.10:8080/uc/ Stonebranch SaaS Cloud Universal Controller: https://superstore.stonebranchdev.cloud/
UAC REST Credentials	Credentials of the Universal Controller Webservice API
Working Directory	Any Linux or Windows directory, which will be used to store the extracted Job definitions in. The directory must exist on the server where the Universal SAP Connector Agent is installed. Examples: • Windows: C:\work\CTK\sap • Linux: /home/stone/work/CTK/sap
Input File Separator	This field contains the Input File Separator used in the Input File; for example, “,” or “;”

Input File	Input file containing the SAP Jobs to extract.
	
	The format is:
	
	<SAP Jobname >,<SAP Jobcount ID>
	
	Example: sap_jobs.csv
	
	`SAP-CTK-01_EVERYDAY_1000,15444900`
	`SAP-CTK-02_FDOM_1000,16483700`
	`SAP-CTK-03-WORDAY-1000,09063600`
	`SAP-CTK-04-EVERYDAY,11290200`
	`SAP-CTK-04_MULTISTEP_EVERYHOUR,17402500`

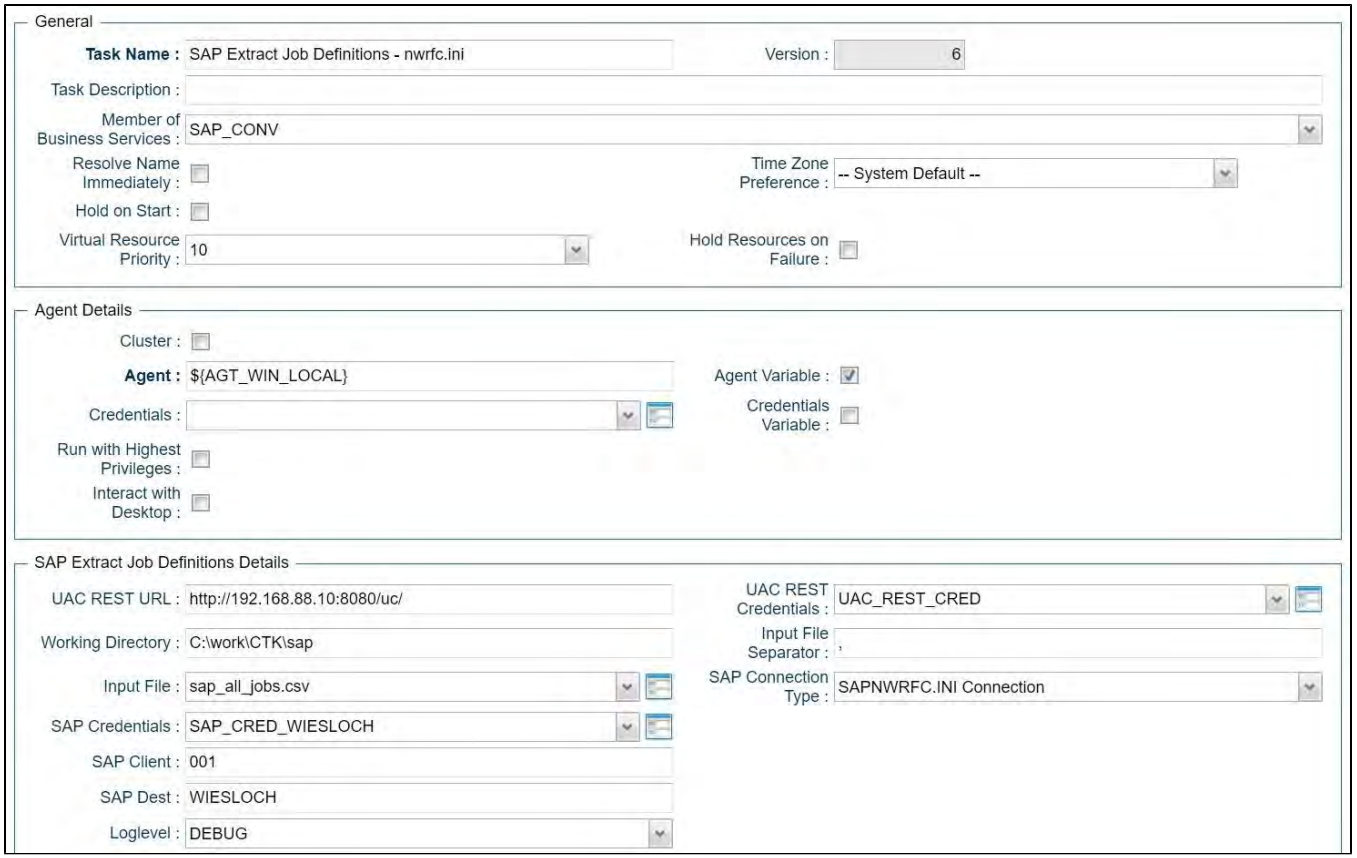
SAP Connection Type	[Application Server Connection | NRFW.ini Connection)] Select the SAP Connection Type: Application Server Connection or Destination Connection using the nwrfc.ini file. Default location for the nwrfc.ini file is: • Linux: /opt/universal/uagsrv • Windows: C:\Program Files\Universal\UAGSrv
SAP Credentials	
SAP ASHOST	SAP Application to connect to. This field is only visible in case of SAP Connection Type = “Application Server Connection”

SAP CLIENT	SAP Client e.g. 100
SAP Sysnr	SAP System Number; for example, 00
SAP Dest	SAP Destination in the nwrfc.ini. This field is only visible in case of SAP Connection Type = "*NRFW.ini Connection*"
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example: SAP Extract Job Definition - Application Server Connection



Example: SAP Extract Job Definition - Destination Connection (nwrfc.ini)



Execute SAP Extract Job Definitions Task

When you launch the SAP Extract Job Definitions Task it will connect to the provide SAP System an extract the Job definition to one flat file per SAP Job. In addition the a log-file will be generated, showing you if all SAP Jobs could be extracted or if some jobs could not be found.

The Log-files and Job definition flat files will be written to the following directories:

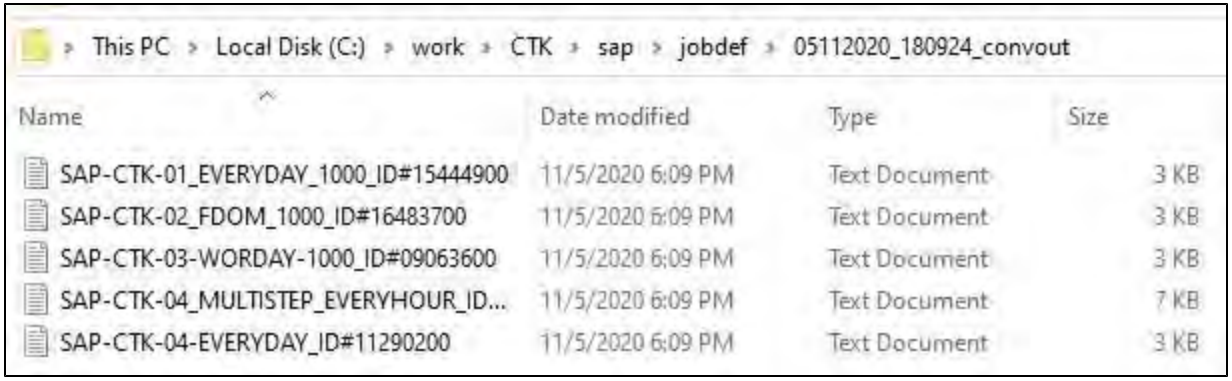
- <Working Directory>jobdef<DDMMYYY_HHMMSS_convout>
- <Working Directory>log

<Working Directory>: is the directory set in the Task Field: Working Directory e.g. *C:\work\CTK\sap*

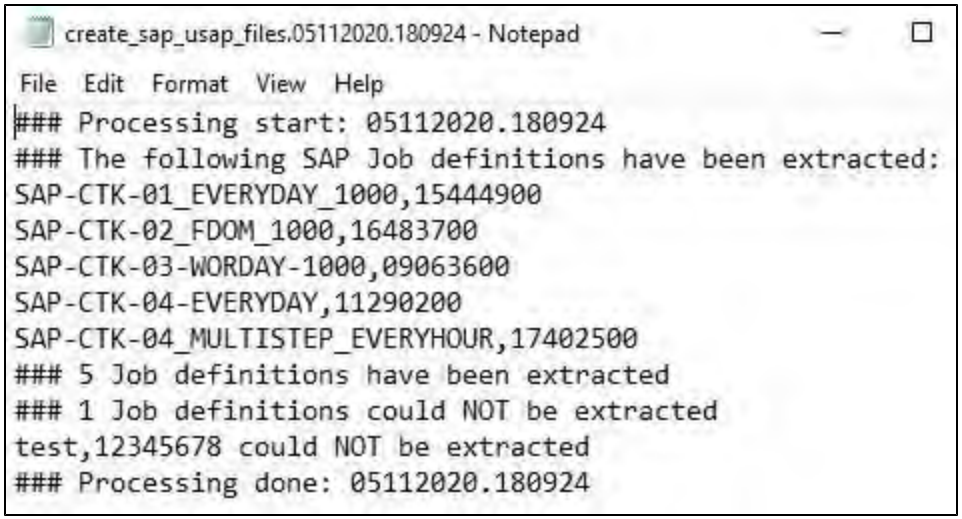
Example Output:

Working Directory = *C:\work\CTK\sap*

Extracted Job Definitions



Log-file Example



ServiceNow Integration

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows customers to create incident tickets, problem tickets, and change requests in ServiceNow straight from the Universal Controller. It can then be combined with the event- or time-based automation capabilities of the Controller to, for example., automatically create an incident ticket in case a task execution fails.

Creating or modifying other objects in ServiceNow can easily be achieved by adding respective fields to the Universal Template and extending the script.

Overview

- This task enables customers to create an incident ticket in ServiceNow and optionally attach the output of predecessor tasks to the ticket.

- It can create problem tickets in ServiceNow.
- It can create change requests in ServiceNow.
- This task combines with the event- or time-based automation capabilities of the Controller to automate the IT service management process.

Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against a remote ServiceNow instance.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests

Software Requirements for Universal Agent

- Universal Agent for Windows x64 Version 6.9.0.0 and later with python options installed, or
- Universal Agent for Linux Version 6.9.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.9.0.0 and later

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with the following ServiceNow versions:

- Paris
- Orlando
- New York
- Madrid

ServiceNow Key Features

Feature	Description
Create an incident ticket	Create an incident ticket in your ServiceNow instance and assign it to certain users or groups in order to raise awareness of, for example, failed tasks in UAC. Optionally, attach the STDOUT and STDERR of a sibling task (task in the same workflow) or any task in UAC directly to the incident so that the assigned users can immediately start investigations on the error and act accordingly.
Create problem	Create a problem in ServiceNow to report on problems that could have arisen in your IT environment.
Create change request	Create a change request in ServiceNow to start the change management procedure depending on the configuration item. Note: Changes of UAC environments (e.g. promotion of a new workflow from DEV to PROD) can be enforced to follow your ServiceNow Change management procedure by automating the promotion of bundles (see Bundles & Promotion) from ServiceNow (for example, when the Change Request on the UAC configuration item moves to the “implement” phase) to UAC via the Web Service API. This requires some configuration in ServiceNow via Flows or Workflows.

Import ServiceNow Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure ServiceNow Universal Task

For the new Universal Task type, create a new task and enter the task-specific details that were created in the Universal Template.

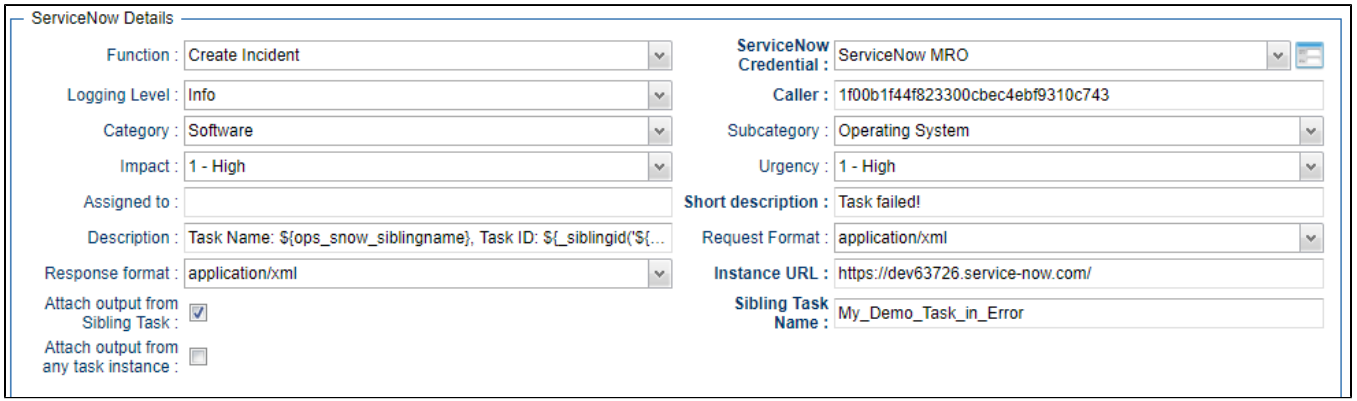
Field Descriptions for ServiceNow Universal Task

Field	Description
Function	Select ServiceNow function to be used (Create Incident, Create Problem, Create Change).
ServiceNow Credential	Credential object, save Username/Password for accessing ServiceNow via API.
Logging Level	Select level of logging for the task execution.
Caller	Person who reported or is affected by this incident (sys_id).
Category	Select pre-defined categories for incident creation.
Subcategory	Select pre-defined subcategories for incident creation.
Impact	Select impact of incident (high, medium, low).
Urgency	Select urgency of incident (high, medium, low).
Assigned to	Person primarily responsible for working this task in ServiceNow (sys_id).
Short Description	Short description of the incident, Universal Controller Variables can be used.
Description	Description of the Incident Ticket, Universal Controller Variables can be used.
Request Format	Select request format of API call (json, xml, text).
Response Format	Select response format of API call (json, xml, text).
Instance URL	URL of the ServiceNow instance; for example, https://dev53724.service-now.com.
Attach output from Sibling Task	Check if output from a sibling task (task within the same workflow) should be attached to the incident ticket.
Sibling Task Name	Name of the sibling task to which the output will be attached.

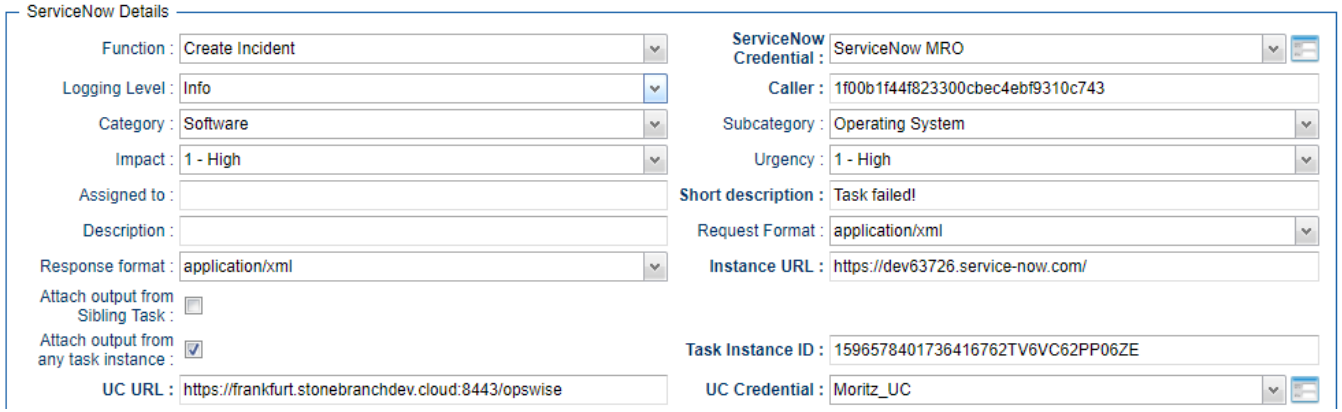
Attach output from any task instance	Check if output from any task instance available from any Controller should be attached to the incident ticket. Uses the REST API.
Task Instance ID	UUID of the task instance of which the output shall be attached (can be a variable).
UC URL	URL of the target Universal Controller; for example, https://frankfurt.stonebranchdev.cloud:8443/uc or http://localhost:8080/uc.
UC Credential	Credential for accessing the Controller. Must have API access on the target system.

Examples for ServiceNow Universal Tasks

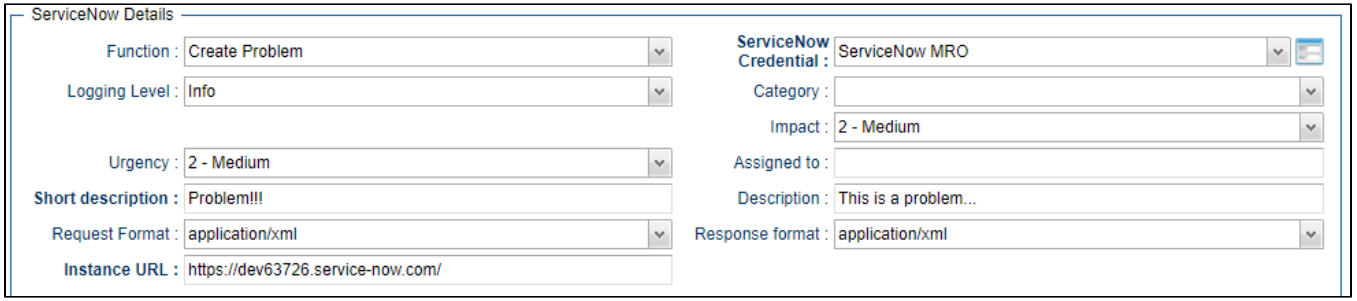
Create ServiceNow Incident Ticket - Attach Output from Sibling Task



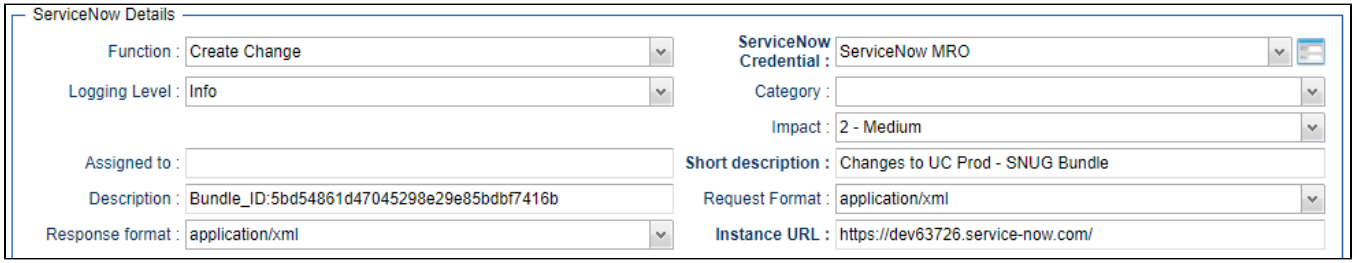
Create ServiceNow Incident Ticket - Attach Output from Task Instance



Create Problem Ticket



Create Change Request



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Slack Notifications

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

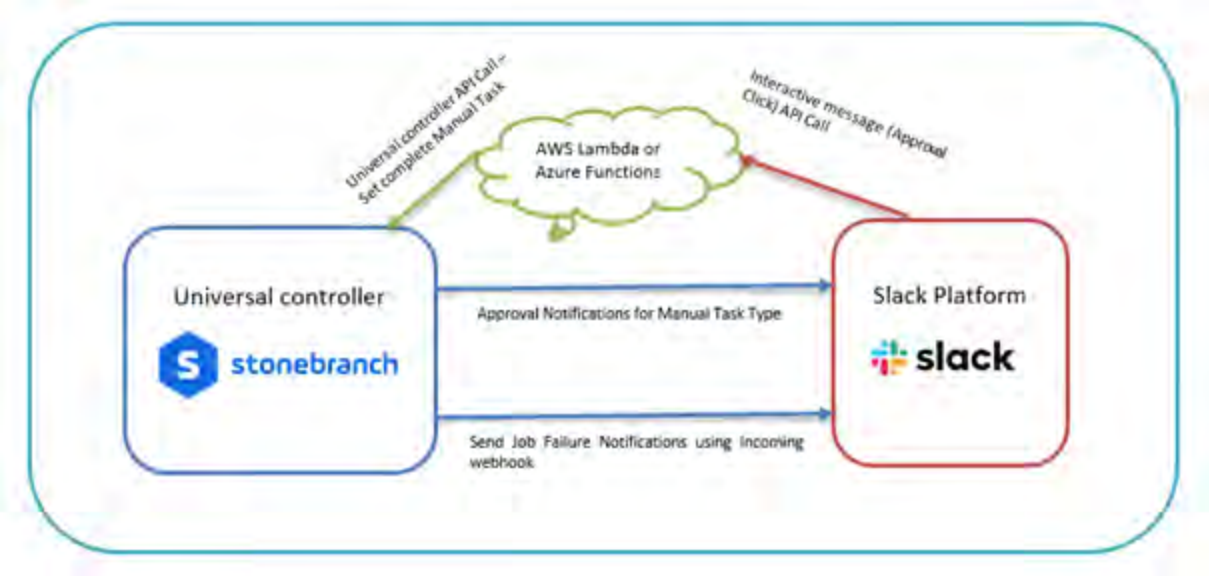
Introduction

This Universal Task sends job status notifications to a Slack channel. It also enables users to send interactive messages in Slack for Universal Controller manual task approvals.

Overview

- Notifies users of job failure, late start/run, and other important events via a Slack channel.
- Approval in Slack for Manual task type – users simply click on an approval button in Slack message to run manual tasks to success in Universal Controller, triggering continuation of workflow execution.

- Quick reaction time on job failures.
- Manual task interruptions in workflows can be handled by concerned applications/business team, while workflows in Universal Controller can be resumed simply by responding to the approval message in Slack.



Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against a Slack account with Incoming webbook enabled for a job notification. Also, in order to cater the approval functionality from slack for an manual task type in Universal Controller, you will need to have an Interactivity enabled in slack with the request URL which will be used to send the http POST request for interactive messages by slack.

Software Requirements for Universal Template and Universal Task

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against .

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.8.0.0 and later with python options installed
- Universal Agent for Linux Version 6.8.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.8.0.0 and later

Software Requirements for the Application to be Scheduled

The task can be used against any of your slack account or workspace that is of either free or standard or plus or enterprise grid

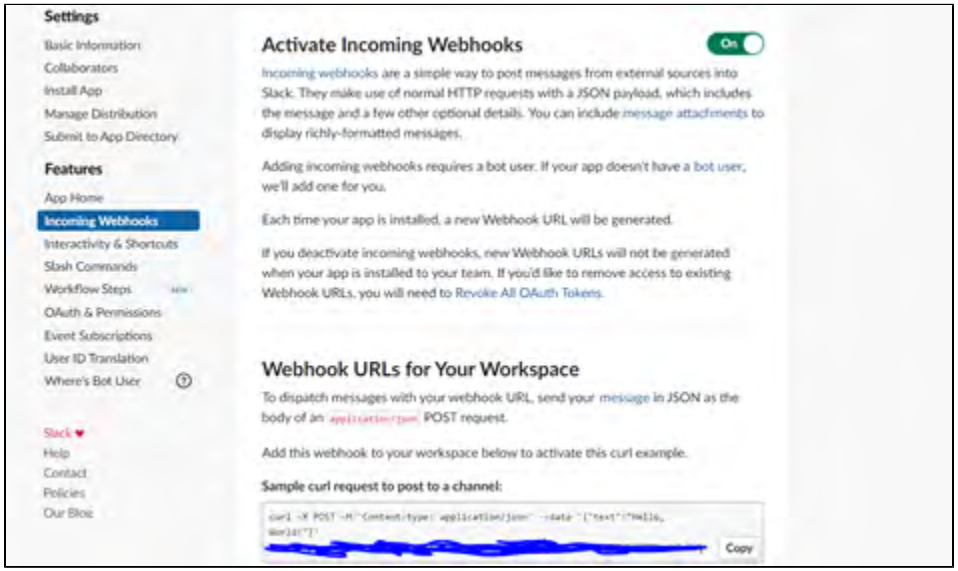
Technical Considerations

The request URL provided in Slack could be an end point either in AWS lambda or Azure Function, GCP function, or your custom API end point to handle the interactive message from slack and advise universal controller on approval or rejection of the manual task.

Below is the sample python code that could be invoked for the slack interactive message handling

Steps to Activate Incoming Webbooks:

- Go to your Browser and provide the URL: https://api.slack.com/apps/.
- Select or create an application that would be appropriate for sending Universal Controller notifications.
- Click on Incoming webhooks on the left menu and activate Incoming webhooks as below.



The URL generated here will be used in the Universal Task for posting message to slack platform

Slack Notification Key Features

Feature	Description
Job Notification	This feature can be used to send any job notification to a slack channel; for example, a job failure, Job long running, or Job held.
Approval Notification	Typically, Manual task types in universal controller are used, when there is a user manual intervention needed in a workflow task type and the Manual Task is completed successfully in universal controller by clicking on to the manual task command “Set Completed” traditionally . Now this task for slack can notify slack channel when the manual task reaches status: “Action Required” and User in slack can simply click on the approve/Reject Button in the slack interactive message that was sent by universal controller and then the manual task in the workflow can either go to success if approved or wait in the same status if rejected

Import Slack Notification Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Slack Notification Universal Task

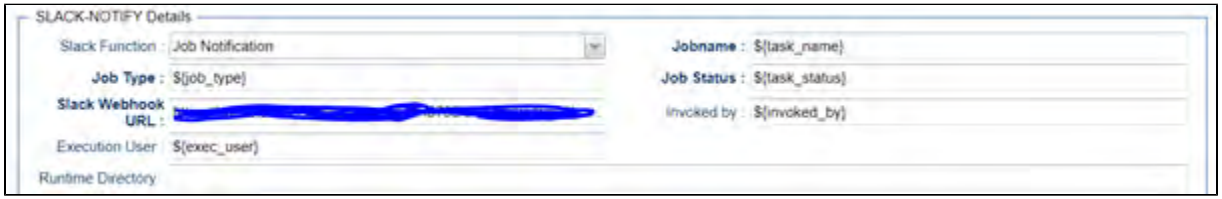
For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

Field Descriptions for Slack Notification Universal Task

Field	Description
Slack Function	Approval Notification (ideal to associate with manual tasks)
Job Name	Name of the job :${ops_task_name}
Job Status	Status of the job:${ops_status}
Slack Incoming Webhook	Incoming webhook URL for your slack account
Execution User	Execution User of the Manual Task:${ops_execution_user}
Job type	Task type of task instance: ${ops_task_type}

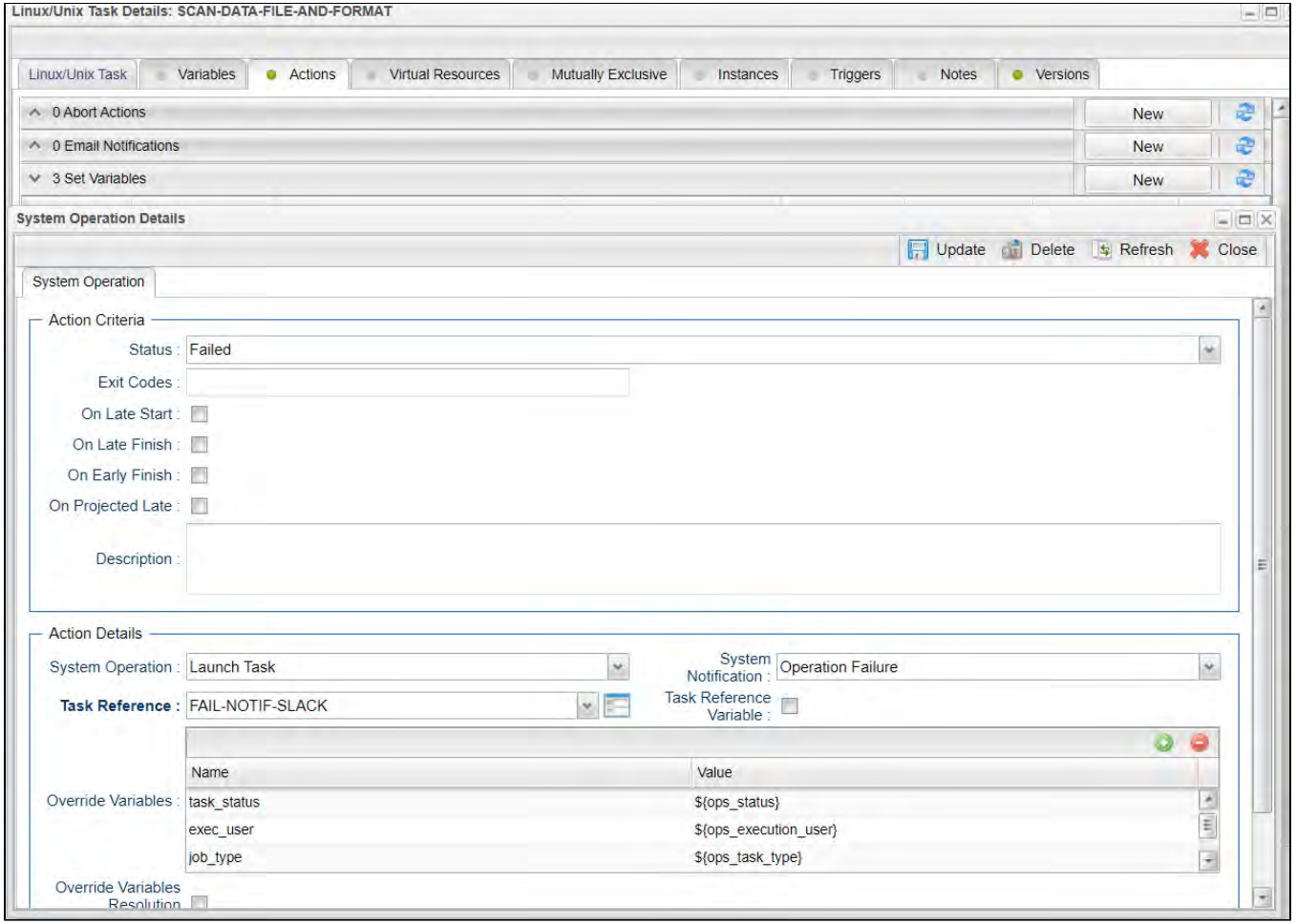
Examples for Slack Notification Universal Tasks

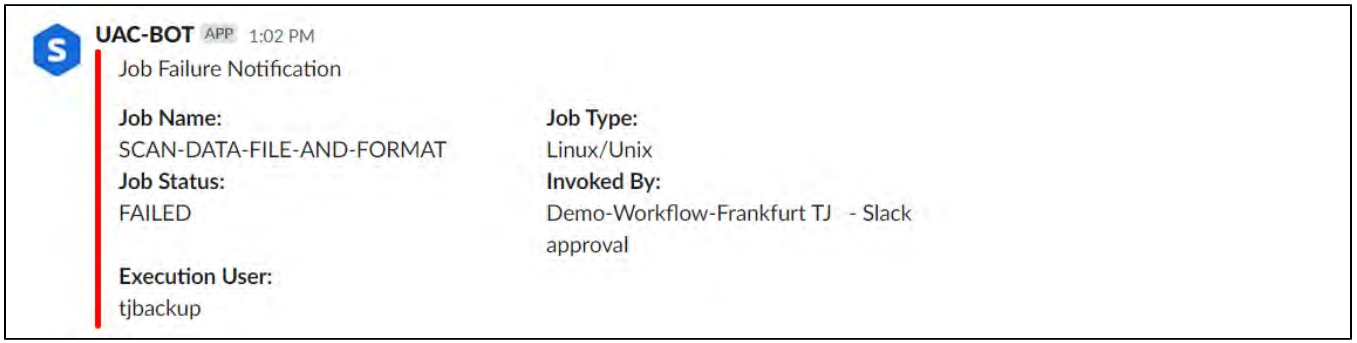
Send a Job Status Notification to slack



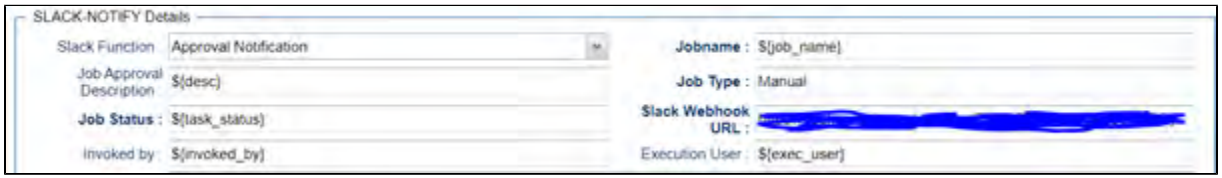
Calling a slack Job Status Notification through Action -->System Operations and Launch Task

(Also this could be made generic for a group of jobs or all jobs by calling the slack notification job from Task monitor job.)

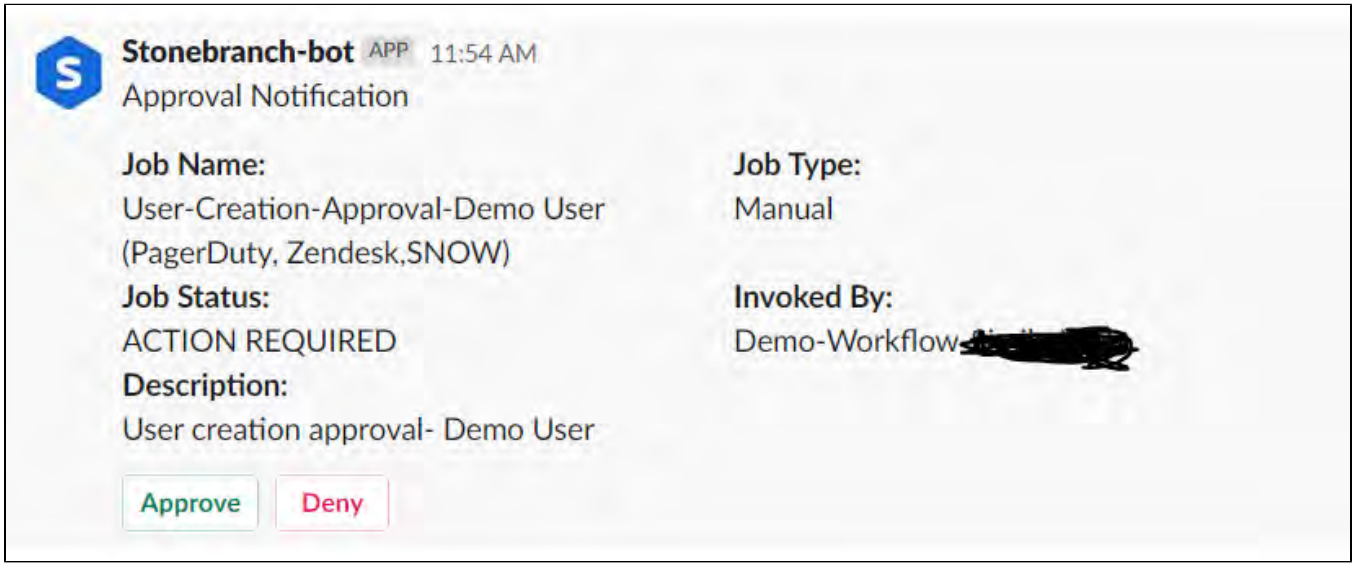

Linux/Unix Task Details: SCAN-DATA-FILE-AND-FORMAT
Linux/Unix Task
Variables
Actions
Virtual Resources
Mutually Exclusive
Instances
Triggers
Notes
Versions
0 Abort Actions
New
0 Email Notifications
New
3 Set Variables
New
System Operation Details
Update
Delete
Refresh
Close
System Operation
Action Criteria
Status : Failed
Exit Codes :
On Late Start :
On Late Finish :
On Early Finish :
On Projected Late :
Description :
Action Details
System Operation : Launch Task
System Notification : Operation Failure
Task Reference : FAIL-NOTIF-SLACK
Task Reference Variable :
Name
Value
Override Variables :
task_status
${ops_status}
exec_user
${ops_execution_user}
job_type
${ops_task_type}
Override Variables Resolution


UAC-BOT APP 1:02 PM
Job Failure Notification
Job Name:
SCAN-DATA-FILE-AND-FORMAT
Job Status:
FAILED
Execution User:
tjbackup
Job Type:
Linux/Unix
Invoked By:
Demo-Workflow-Frankfurt TJ - Slack approval

Send a Approval Notification to slack



Approval Message in Slack



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

Snowflake

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows Stonebranch users to orchestrate, schedule, trigger, and monitor the Snowflake load and unload process from different data sources (cloud storage or local VM’s) directly from Universal Controller. It uses Python libraries to perform all functions listed in the following sections. Alternatively, you also can perform all these operations using the snowflake JDBC driver which you can add to the Universal Controller libraries, and use SQL Task to perform any operations with Snowflake (https://docs.snowflake.com/en/user-guide/jdbc-download.html)

Overview

Users can orchestrate the Snowflake functionalities using the following features available in the Universal Task

- UAC functionalities (Snowflake -Loading)
    - Load data from AWS S3 to Snowflake.
    - Load data from Azure Storage to Snowflake.
    - Load data from Google storage to Snowflake.
    - Load Internal stage file to Snowflake Table.
    - Copy from local server to Internal staging.
- UAC functionalities (Snowflake - Unloading)
    - Unload Snowflake data to AWS S3.
    - Unload Snowflake data to Azure Storage.
    - Unload Snowflake data to Google Storage.
    - Unload Snowflake data to Internal stage.
    - Unload from internal stage to local server.
- UAC functionalities (Snowflake – Execute Commands)
    - Execute a Snowflake command.

Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against Snowflake.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - https://raw.githubusercontent.com/snowflakedb/snowflake-connector-python/v2.3.9/tested_requirements/requirements_36.reqs
    - snowflake-connector-python
    - snowflake-ingest
    - Please refer to Snowflake URL : https://docs.snowflake.com/en/user-guide/python-connector-install.html for the latest Python connector details.

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.6 and later with Python options installed.
- Universal Agent for Linux Version 6.6 and later with Python options installed.

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later.

Software Requirements for the Application to be Scheduled

This Universal Task has been tested with the snowflake-connector-python=2.3.9 and snowflake-ingest=1.0.3.

Technical Considerations

- This task uses Python modules snowflake-connector-python and snowflake-ingest to make REST-API calls to Snowflake, Additionally, as a prerequisite, users might need to install other packages listed in https://raw.githubusercontent.com/snowflakedb/snowflake-connector-python/v2.3.9/tested_requirements/requirements_36.reqs.
- Snowflake login credential, Snowflake Account name, and URL would be needed in UAC for this Universal Task.
- In the case of data ingest from internal staging to Snowflake table through a pipeline, you would need to supply a private key file from local server and public key for the same to be loaded against the Snowflake user used in Universal Task.
- Use the Snowflake instructions is in URL : https://docs.snowflake.com/en/user-guide/key-pair-auth.html for Key pair authentication and rotation process.

Key Features

Feature	Description
Load data from AWS S3 to Snowflake	Load the S3 bucket file(s) into a Snowflake table, You may specify the file format options and copy options appropriately.
Load data from Azure Storage to Snowflake	Load the Azure container blobs into a Snowflake table, You may specify the file format options and copy options appropriately.
Load data from Google storage to Snowflake	Load the google storage bucket files into a Snowflake table, You may specify the file format options and copy options appropriately. Please note that you will need to define the storage integration in Snowflake and provide this in the Universal Task.
Load Internal stage file to Snowflake Table	Files that are available in Snowflake internal storage to be loaded into a Snowflake table using the pipe name and authentication using private public key files. The data can be loaded in Snowflake using patterns.
Copy from local server to Internal staging	Copies files from local Windows or Linux server to Snowflake internal staging area.
Unload Snowflake data to AWS S3	This feature helps to unload the data from a Snowflake table to a AWS S3 bucket, file format options and copy options can be provided appropriately.
Unload Snowflake data to Azure Storage	This feature helps to unload the data from a Snowflake table to an Azure container; file format options and copy options can be provided appropriately.
Unload Snowflake data to Google Storage	This feature helps to unload the data from a Snowflake table to a Google cloud Storage; file format options and copy options can be provided appropriately. Also, this would need an storage integration name from Snowflake.
Unload Snowflake data to Internal stage	Unloads the Snowflake table into an internal staging area in Snowflake.
Unload from internal stage to local server	This feature helps to copy the files in staging area to a local windows or a Linux server.
Execute a Snowflake command	Users can use this feature to execute snowflake commands; for example: Copy, Remove, Select, Delete, etc.

Import Snowflake Integration Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure Snowflake Integration Universal Task

For the new Universal Task type, create a new task, and enter the task-specific details that were created in the Universal Template.

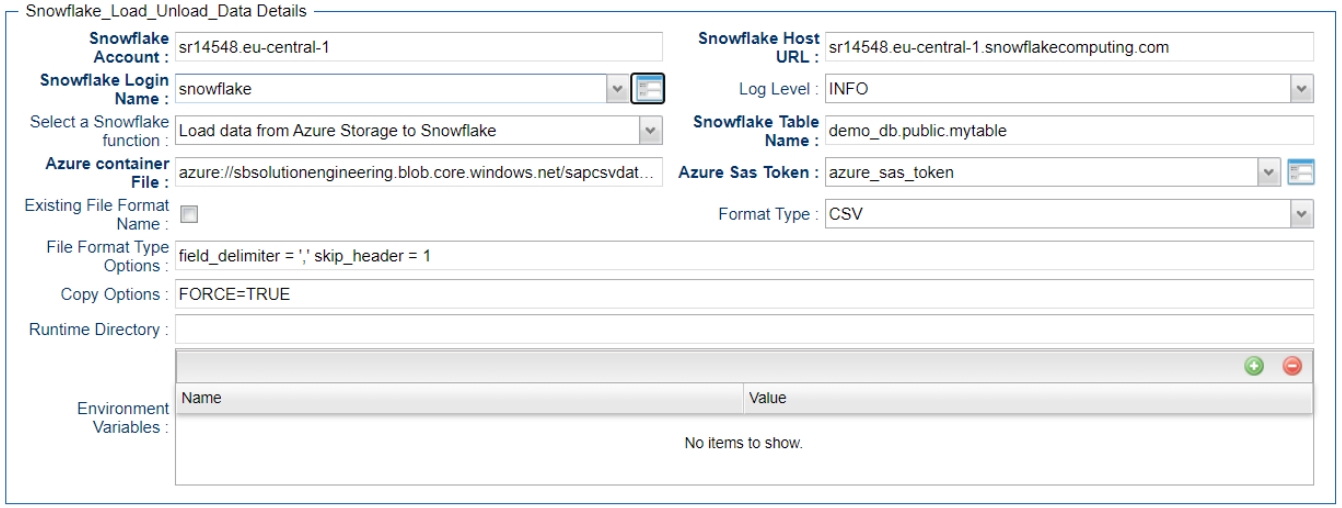
Field Descriptions for Snowflake Integration Universal Task

Field	Description
Snowflake Account	Provide the Snowflake user account; for example, sr14548.eu-central-1.
Snowflake Host URL	Specify your host information in the form of a URL; for example, sr14548.eu-central-1.snowflakecomputing.com.
Snowflake Login Name	Specify your Snowflake login name..
Log Level	Select a log level.
Select a Snowflake function	Select the required Snowflake function (if the required function not available, check the SQL task to invoke snowflake DB).
Snowflake Command	Provide a Snowflake command either in SQL or put / get commands etc.
Local file Name & Path	Provide the local file name that need to be copied to stage.
Stage Name	Provide the stage(internal) name in Snowflake.
Encrypted Private Key	Select only if you have a encrypted private key.
Stage File Name(s)	Provide the stage file names; if there are multiple files, separate by comma.
Pipe Name	Specify the fully-qualified name of the pipe to use to load the data.
Private Key File Path	Provide the private key file path to establish connection to Snowflake for data ingest.
Private Key Password	Provide the password for private if it is encrypted.
Snowflake Table Name	Provide the full path and the Snowflake table name where the table to be loaded.
Use AWS Credentials	Check this if you need to supply AWS access key credentials.
AWS Key ID & Secret Access Key	Provide the AWS secret Access Key (runtime user AWS Key id and secret key in the password section).
AWS Storage Integration	Provide the name of the AWS storage integration created in Snowflake.

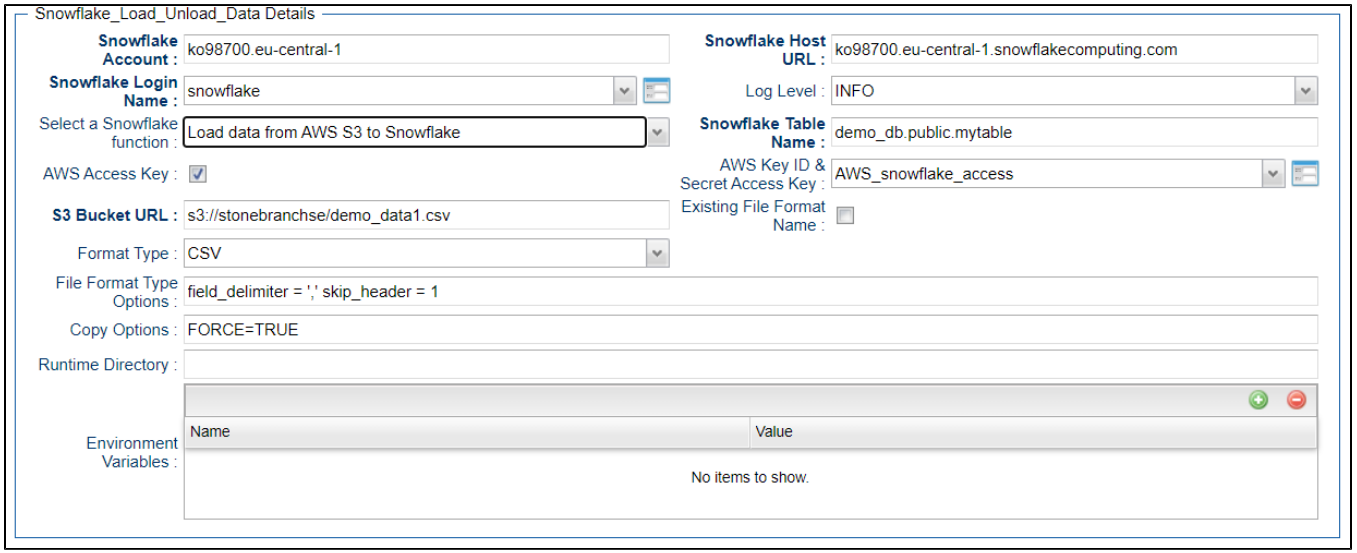
Load using Pattern	If you need load the data using pattern, check this option.
S3 Bucket URL	Provide your S3 bucket URL s3://<your_s3_bucket>/data/.
Azure container File	Provide the Azure container file URL starting with azure://
Azure Storage Integration	Provide the Azure storage integration created in Snowflake.
GCP storage URL	Provide the bucket name and the file name; for example, gcs://mybucket/data/files.
Azure Sas Token	Provide the Azure Sas Token.
Existing File Format Name	Specify an existing named file format to use for loading data into the table.
Storage Integration	Provide the Snowflake storage integration details.
Pattern	Regular expression pattern string - specifying the file names and/or paths to match.
File Format Name	Specifies an existing named file format to use for loading data into the table.
Format Type	Specify the format type: CSV | JSON | AVRO | ORC | PARQUET | XML.
File Format Type Options	Specify the file format type options; for example, FIELD_DELIMITER = '|' if CSV type -- Refer to https://docs.snowflake.com/en/sql-reference/sql/copy-into-table.html
Copy Options	Include other copy options; for example, ON_ERROR = CONTINUE or FORCE=TRUE- Refer to https://docs.snowflake.com/en/sql-reference/sql/copy-into-table.html
Polling Interval (Secs)	Provide the Polling Interval time in Seconds, in the case of data ingestion from internal staging to Snowflake Table.
Number of times to Poll	Specify the number of times to poll; otherwise, default is set to 50, in the case of data ingestion from internal staging to Snowflake Table.

Examples for Snowflake Integration Universal Tasks

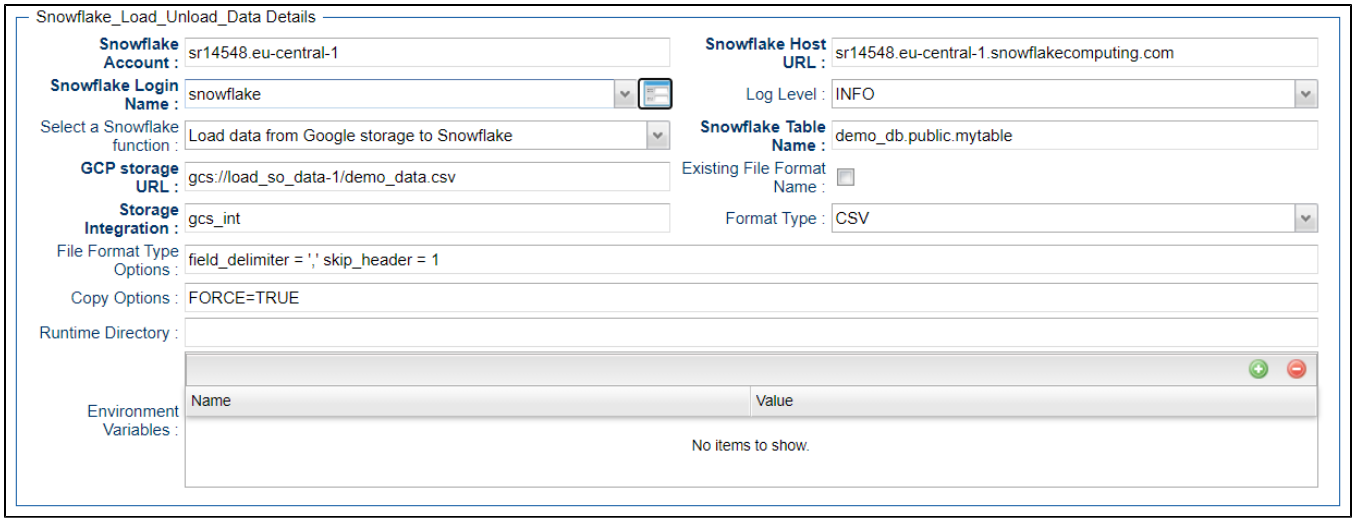
Load Data from Azure Storage to Snowflake Table



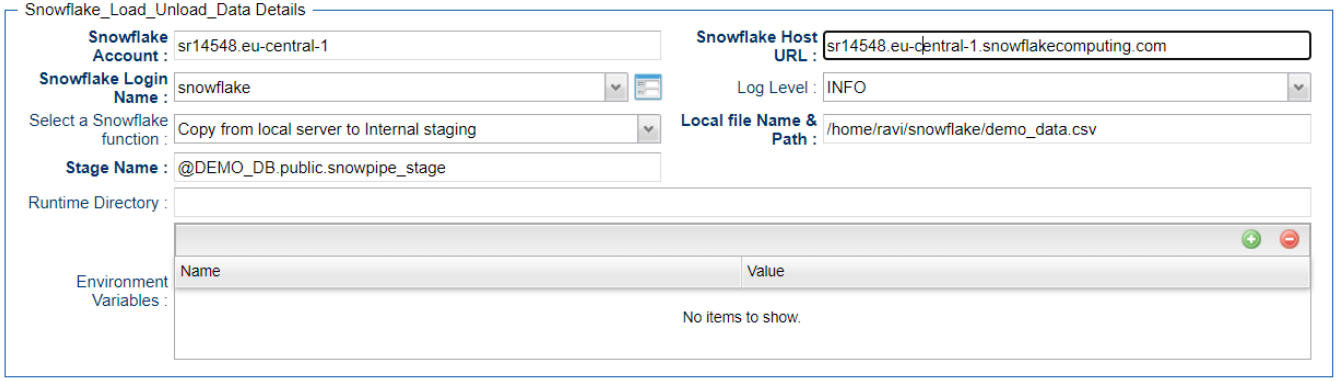
Load Data from AWS S3 to Snowflake Table



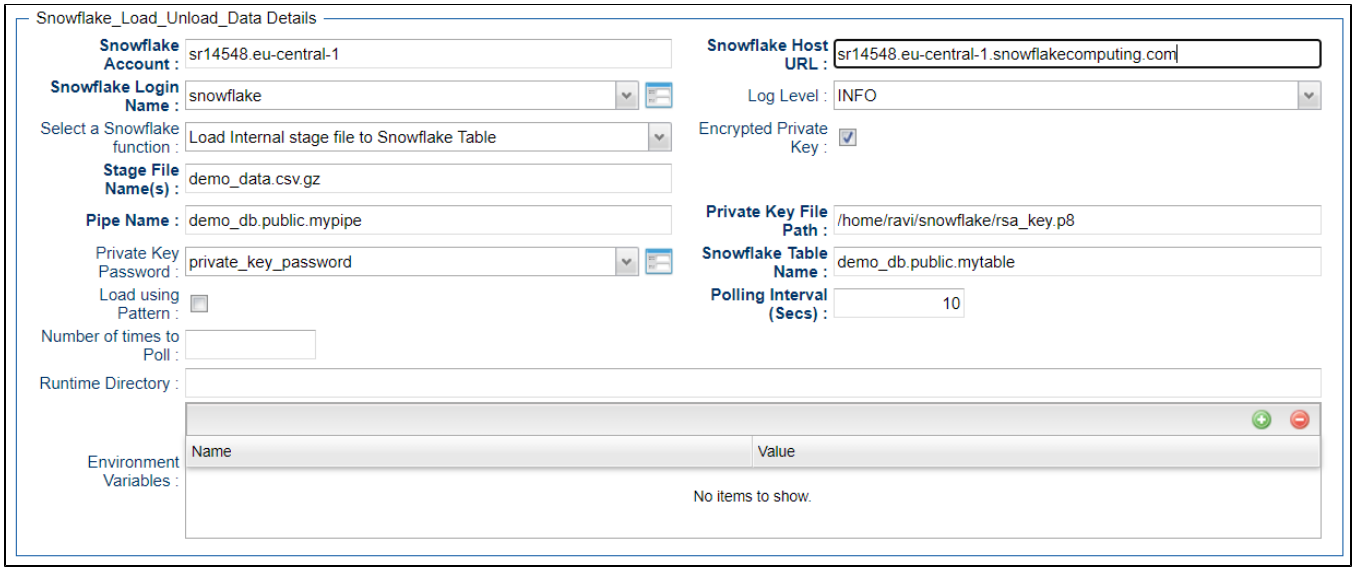
Load Data from Google Cloud Storage to Snowflake Table



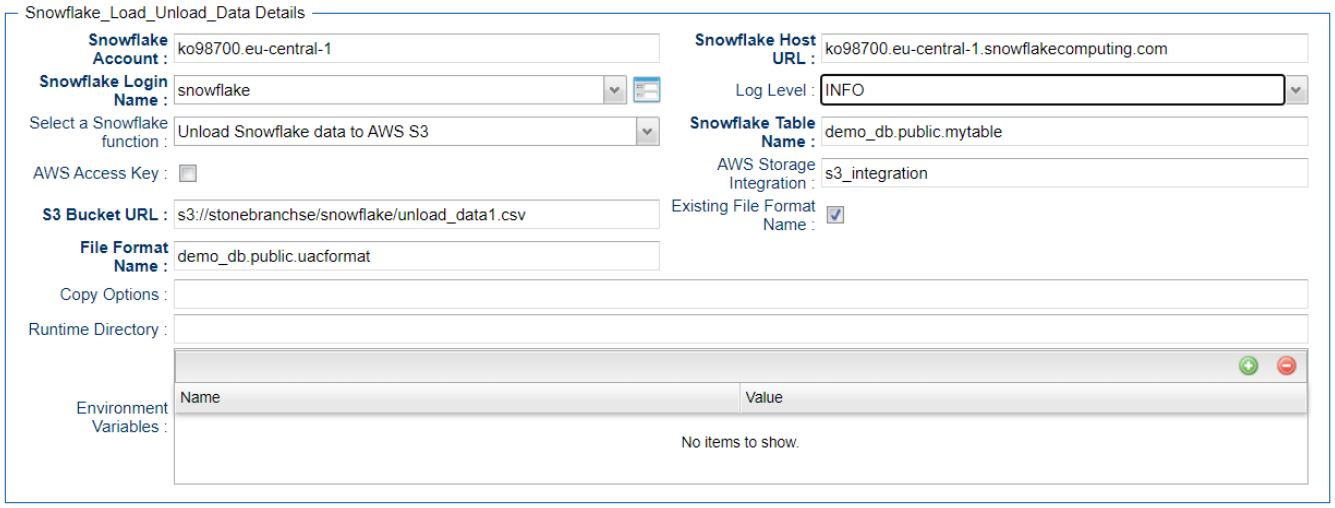
Copy Local File to Snowflake Staging



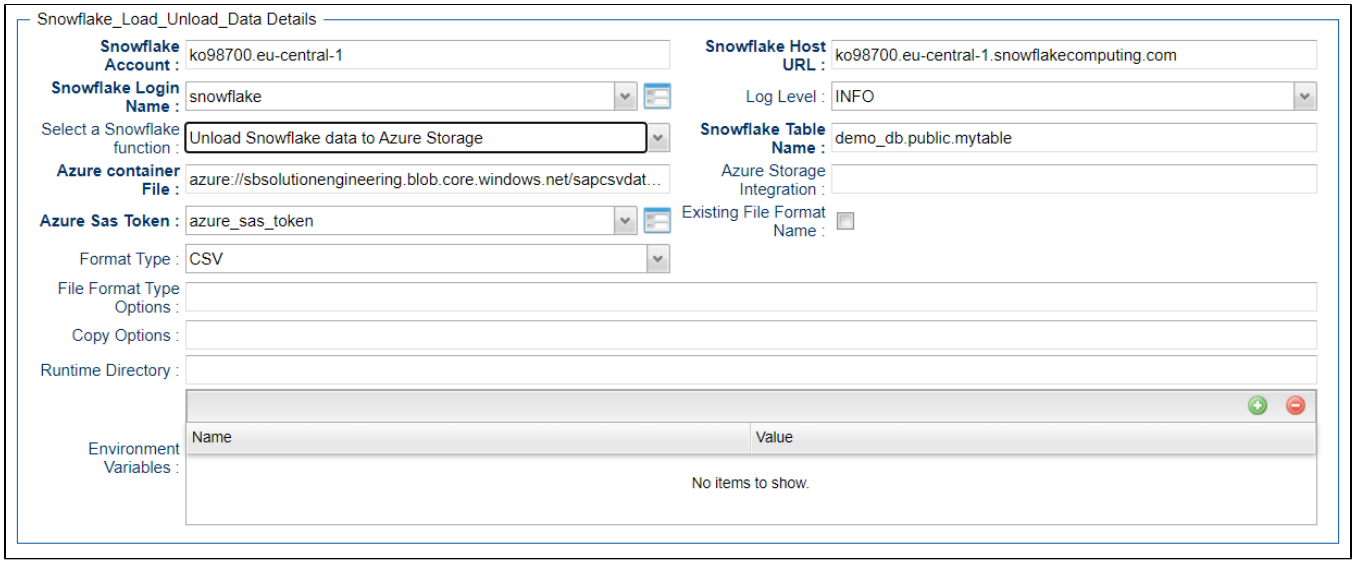
Load Snowflake Staging File to Table



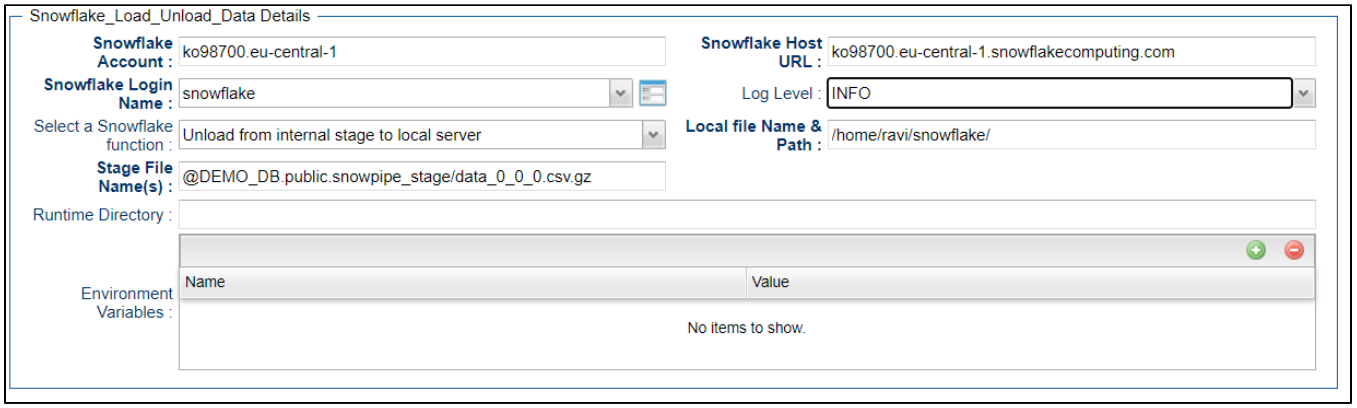
Unload Snowflake Table to AWS S3 Storage



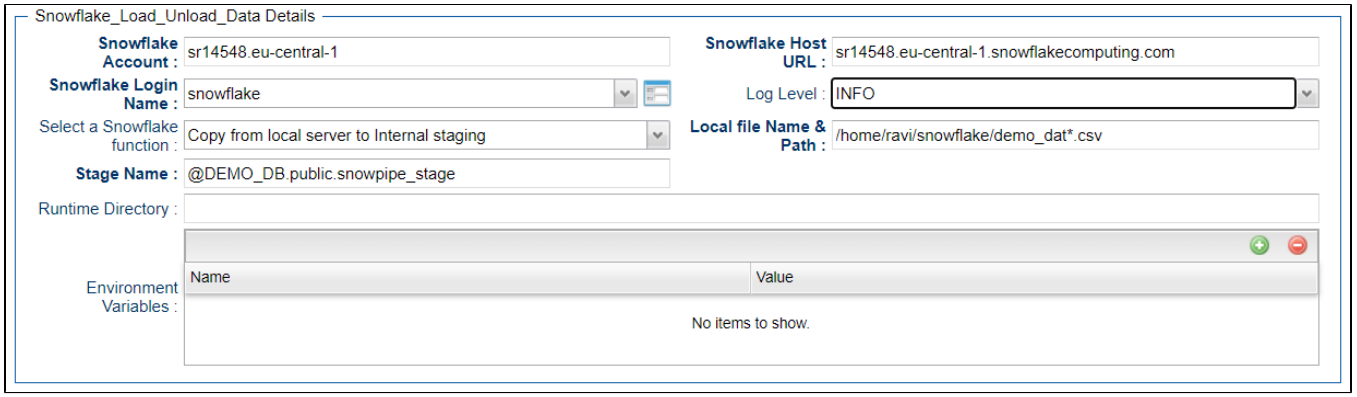
Unload Snowflake Table to Azure Storage



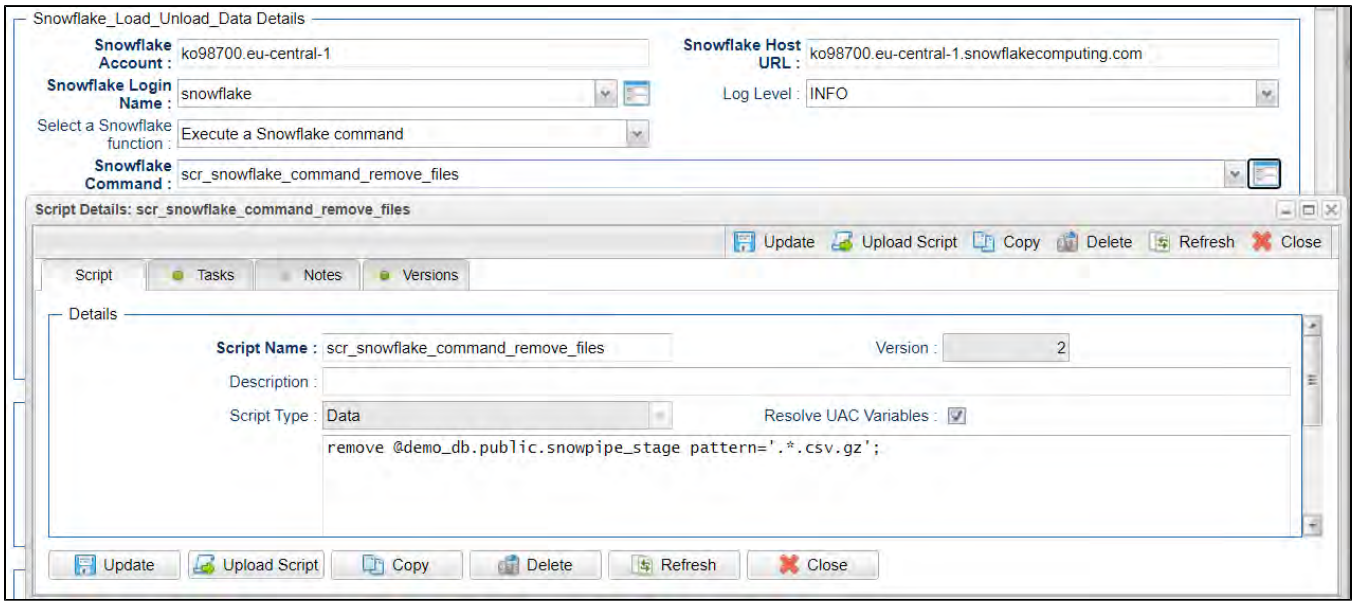
Download Snowflake Stage File to Local Linux Server



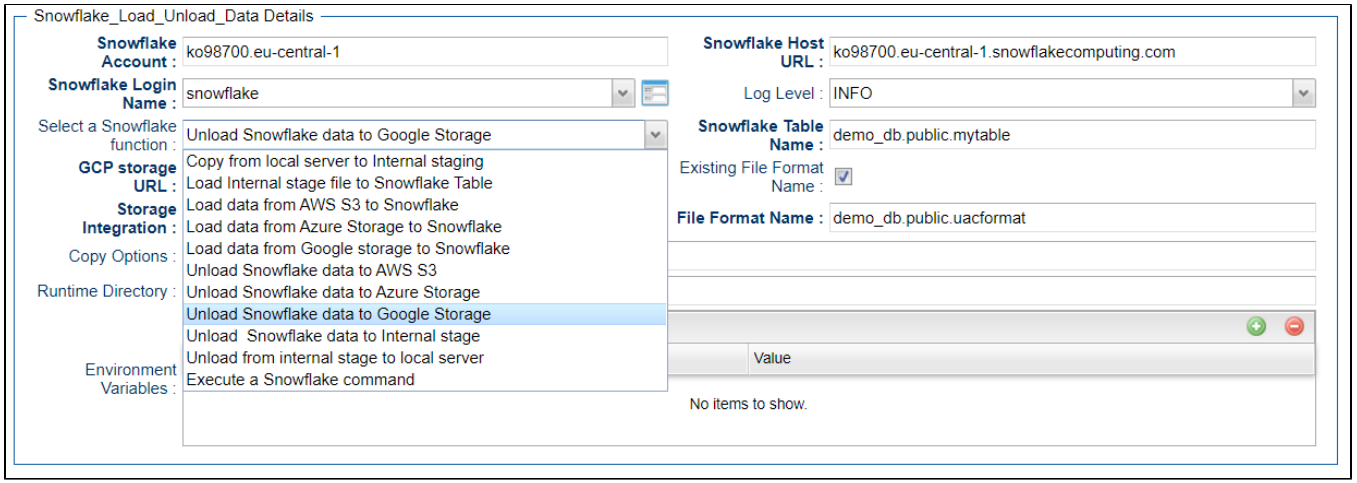
Copy Multiple Files (Linux Server) to Snowflake Staging Area



Executing a Snowflake Command



Snowflake Universal Task Functions



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

SQL

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows users to execute SQL scripts and functions against a MySQL, PostgreSQL, Microsoft SQL Server, Oracle and SAP HANA database.

It uses an agentless connection via ODBC towards SQLSERVER, MySQL and PostgreSQL and the oracle basic instant client to connect to an Oracle database.

Software Requirements

Software Requirements for Universal Template and Universal Task

- Universal Agent for Linux or Windows Version 6.9.0.0 or later is required.
- Universal Agent needs to be installed with python option (--python yes).
- The python ODBC module pyodbc v4.0.30 needs to be installed for MySQL, PostgreSQL and Microsoft SQL Server connections.
- The python SAP HANA module hdbcli v2.6.58 needs to be installed for SAP HANA connections.
- The python Oracle module cx_oracle 8.0.1needs to be installed for Oracle Database connections.

Software Requirements Universal Controller

- Universal Controller 6.9.0.0. or later is required

Software Requirements for the Application to be Scheduled

The Universal Task has been tested for the following databases, Versions and Connector. Please make sure that the connector is installed for your database.

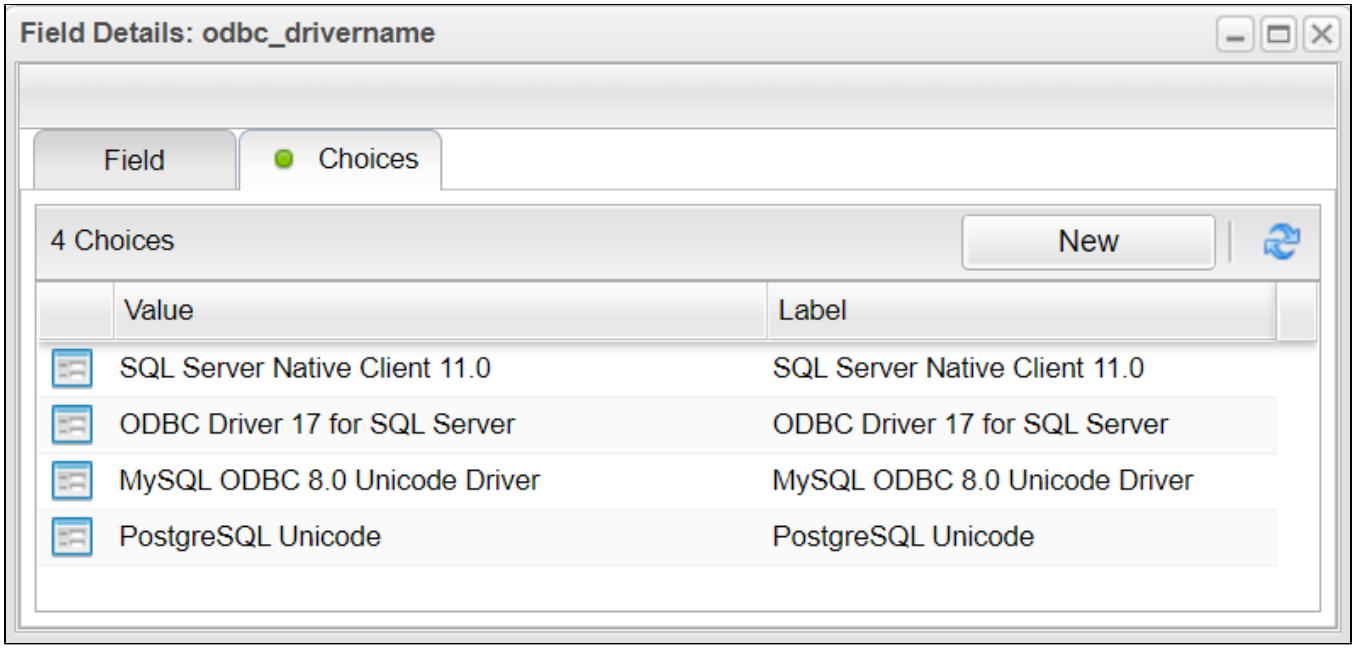
Name	Version	Database Connector
MySQL	8	*MySQL ODBC 8.0 Unicode Driver*
PostgreSQL	13	*PostgreSQL Unicode*
Microsoft SQL Server	15	either one of them: • *SQL Server Native Client 11.0,* • *ODBC Driver 17 for SQL Server*
Oracle	18XE	*Oracle Instant Client v19.6.0.0.0*
SAP HANA		not required - part of the SAP Python hdbcli module

Note

If you have a different Database Connector than mentioned in the table above. you can add a new connector to the Universal Task by adding it to the Universal Template of the Universal Task with name SQL.

In Universal Controller

Administration Universal Templates SQL Fields odbc_drivername New - button



SQL Universal Task Key Features

The solution supports the following file transfer scenarios:

- The Universal Task supports execution of SQL scripts for Oracle, MySQL, PostgreSQL, Microsoft SQL Server and SAP HANA.
- For Oracle the execution of SQL scripts and oracle PLSQL blocks are supported.
- All connections are agentless via ODBC for SQLSERVER, MySQL and PostgreSQL.
- Oracle connections are performed agentless using the oracle basic instant client.
- SAP Hana Database connections are performed agentless using the SAP HANA client for Python.
- The Universal Task supports both Universal Agent for Linux/Unix and Windows.
- You can select different log-levels e.g. Info and debug.
- You can decide whether or not the SQL-output is provided in the standard out.
- All Passwords are encrypted using Controller Credentials.
- For SQLSERVER *Windows Authentication* and *SQLSERVER Authentication* is supported.

Import SQL Universal Task Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.
5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure SQL Universal Tasks

For the new Universal Task type, create a new task and enter the task-specific Details that were created in the Universal Template.

Field Descriptions for SQL Universal Task

The following will provide a configuration Example for each of the supported Databases:

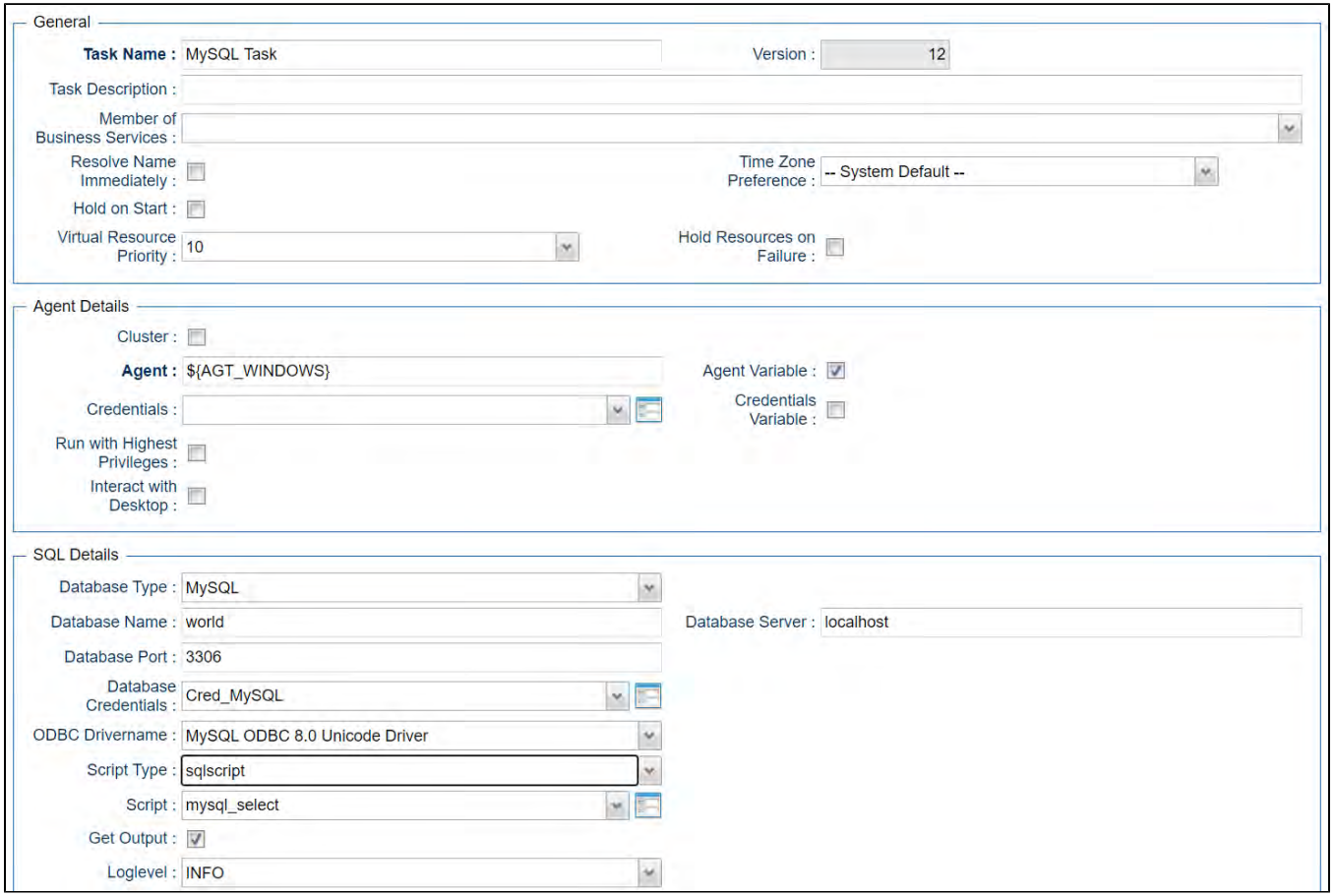
- Database Type: MySQL
- Database Type: Oracle - SQL
- Database Type: Oracle - PLSQL Block
- Database Type: PostgreSQL
- Database Type: Microsoft SQL Server
- Database Type: SAP HANA

Database Type: MySQL

Field	Description
Database Type	Type of database to connect: • **MySQL** • Oracle • PostgreSQL • Microsoft SQL Server • SAP HANA
Database Name	**Database name** or oracle service name
Database Server	Database Servername e.g. localhost
Database Port	Port of the Database e.g. • **MySQL: 3306** • Oracle: 1541 • PostgreSQL: 5432 • Microsoft SQL Server: 1433 • SAP HANA: 39013
Database Credentials	Database Connection Credentials
ODBC Drivername	Name of the ODBC driver - the field is only relevant for MySQL, PostgreSQL and Microsoft SQL Server. The following driver are available for selection: • ***MySQL: MySQL ODBC 8.0 Unicode Driver*** • *MS SQL Server: SQL Server Native Client 11.0,* • *MS SQL Server: ODBC Driver 17 for SQL Server*

	• *PostgreSQL: PostgreSQL Unicode* Additional driver can be added in the SQL Task Universal Template under: *Administration Universal Templates SQL Fields odbc_drivername New*
Script	The database script to execute Note: The name should not contain a Universal Controller Variable e.g. script name: *sb-${**ENV**}-proc01* will **not** work, because it contains a variable (${ENV}) in the name.
Get Output	*prints the SQL output to STDOUT* *In case of an Oracle Stored procedure print the dbms_output to STDOUT.*
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

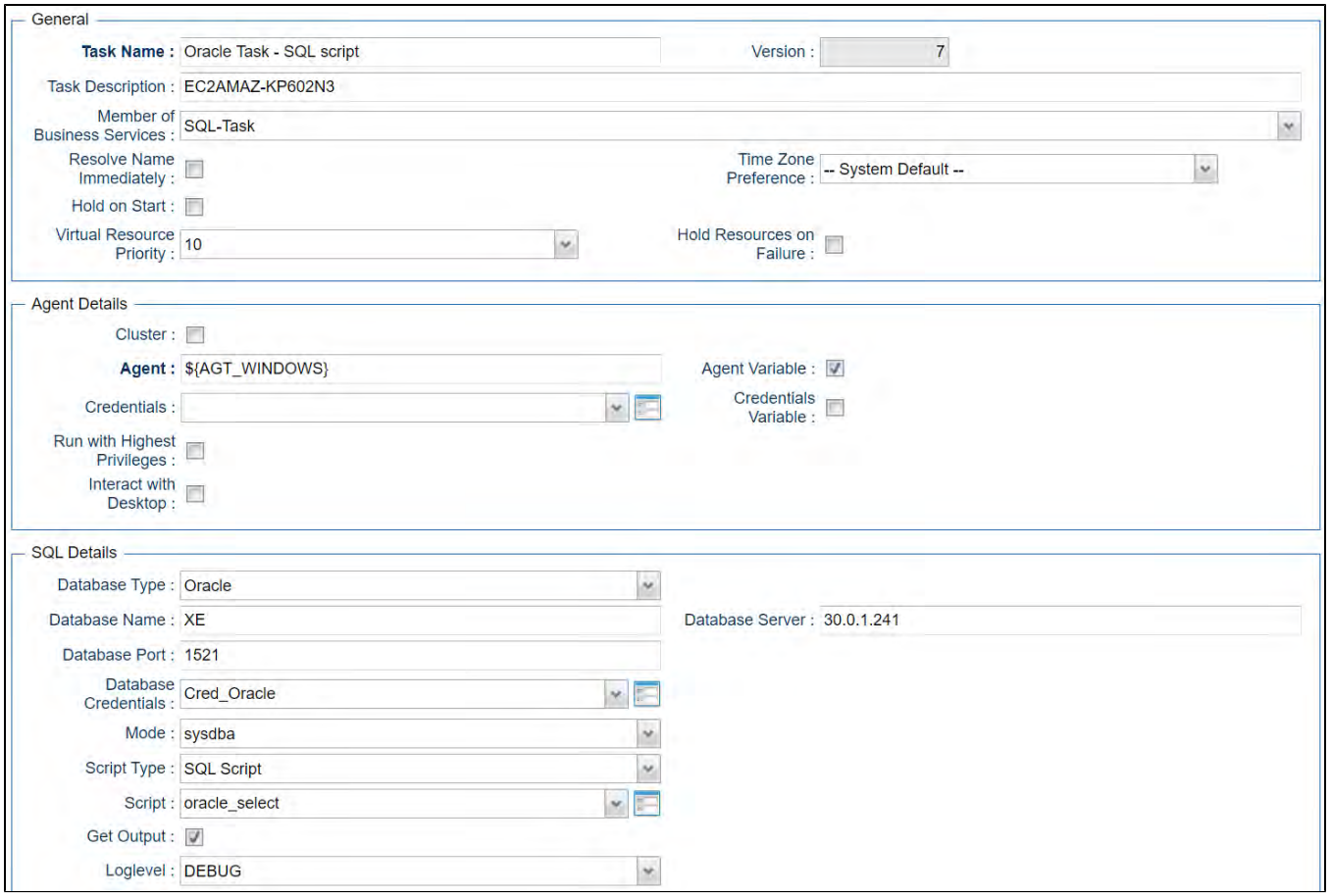
Example: Run a SQL script on MySQL



Database Type: Oracle - SQL

Field	Description
Database Type	Type of database to connect: • MySQL • **Oracle** • PostgreSQL • Microsoft SQL Server • SAP HANA
Database Name	Database name or **oracle service name**
Database Server	Database Servername e.g. localhost
Database Port	Port of the Database e.g. • MySQL: 3306 • **Oracle: 1541** • PostgreSQL: 5432 • Microsoft SQL Server: 1433 • SAP HANA: 39013
Database Credentials	Database Connection Credentials
Mode	Connection Authorization Mode: [sysdba | sysasm | sysoper | sysbkp | sysdgd | syskmt | None] The field is only relevant oracle connections
Script Type	The field is only relevant for oracle connections [**SQL Script** | PL/SQL Block] • **SQL Script - select to execute an SQL Script** • PL/SQL Block - select to run an PL/SQL Block
Script	The database script to execute Note: The name should not contain a Universal Controller Variable e.g. script name: *sb-${**ENV**}-proc01* will **not** work, because it contains a variable (${ENV}) in the name.
Get Output	*Prints the SQL output to STDOUT* *In case of an Oracle Stored procedure print the dbms_output to STDOUT.*
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example: Run a SQL Script on Oracle - SQL

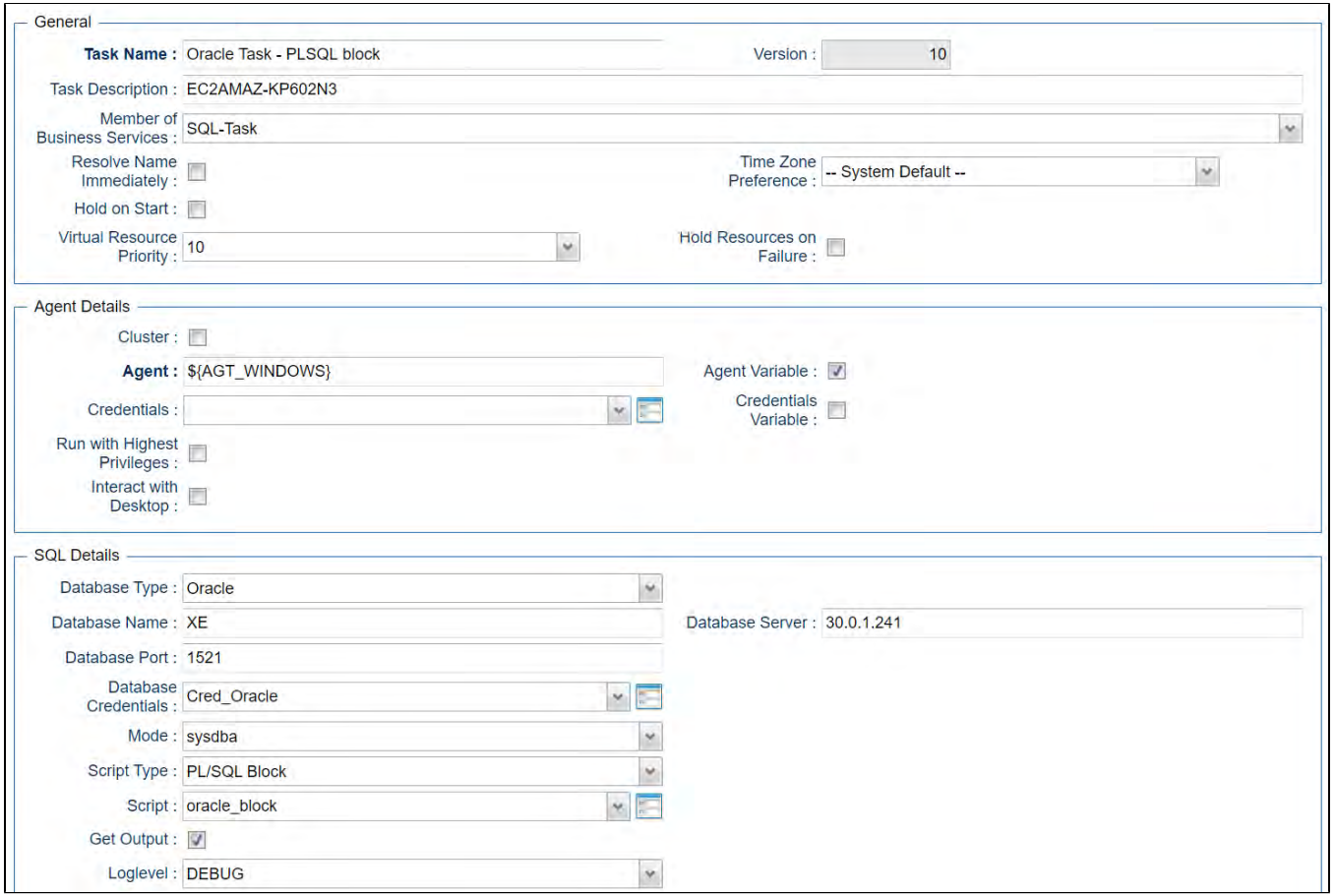


Database Type: Oracle - PLSQL Block

Field	Description
Database Type	Type of database to connect: • MySQL • **Oracle** • PostgreSQL • Microsoft SQL Server • SAP HANA
Database Name	Database name or **oracle service name**
Database Server	Database Servername e.g. localhost

Database Port	Port of the Database e.g. • MySQL: 3306 • **Oracle: 1541** • PostgreSQL: 5432 • Microsoft SQL Server: 1433 • SAP HANA: 39013
Database Credentials	Database Connection Credentials
Mode	Connection Authorization Mode: [sysdba | sysasm | sysoper | sysbkp | sysdgd | syskmt | None] The field is only relevant oracle connections
Script Type	The field is only relevant for oracle connections [SQL Script | **PL/SQL Block**] • SQL Script - select to execute an SQL Script • **PL/SQL Block - select to run an PL/SQL Block**
Script	The database script to execute Note: The name should not contain a Universal Controller Variable e.g. script name: *sb-${**ENV**}-proc01* will **not** work, because it contains a variable (${ENV}) in the name.
Get Output	*prints the SQL output to STDOUT* ***In case of an Oracle Stored procedure print the dbms_output to STDOUT.***
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example: Run an Oracle - PLSQL Block

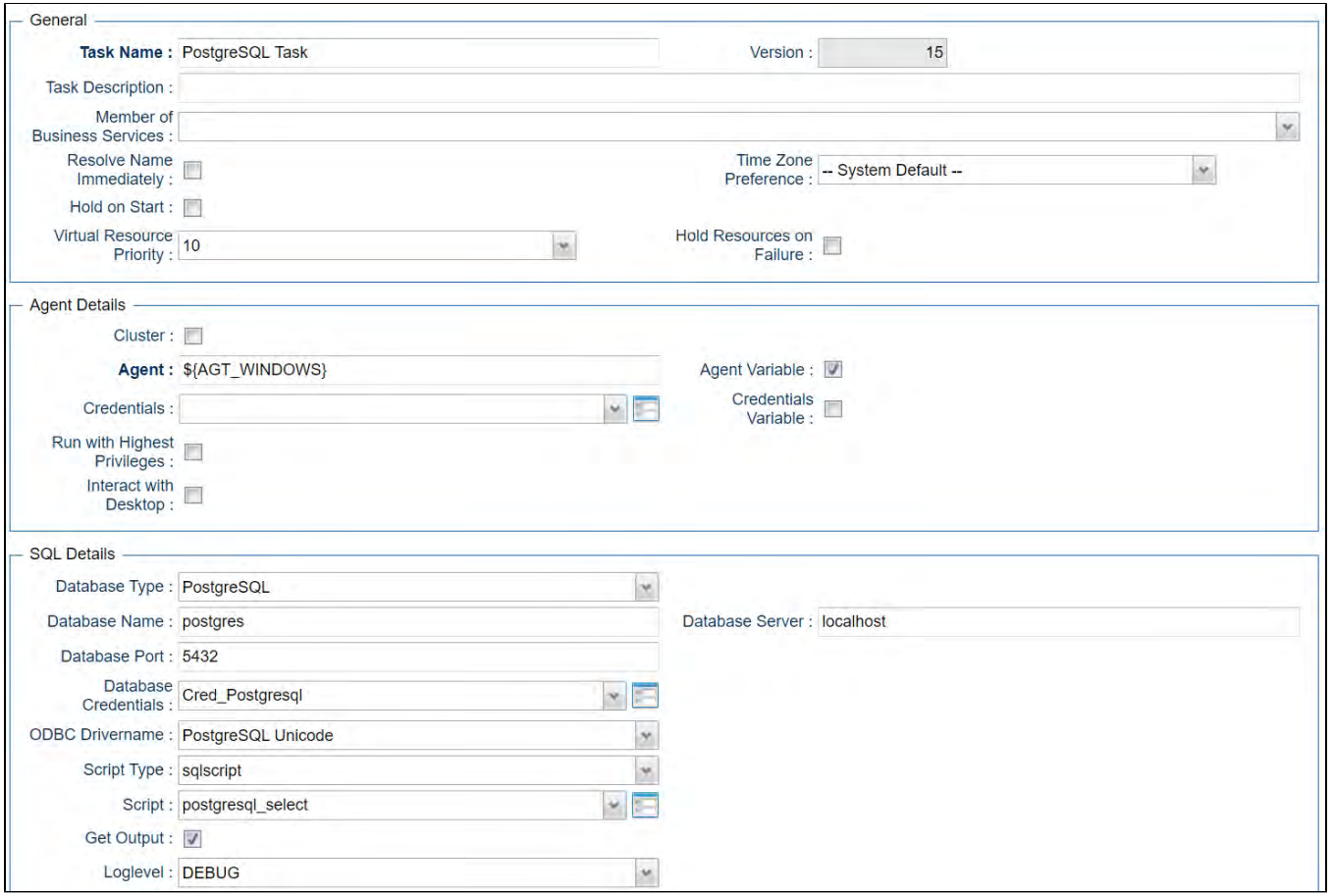


Database Type: PostgreSQL

Field	Description
Database Type	Type of database to connect: • MySQL • Oracle • **PostgreSQL** • Microsoft SQL Server • SAP HANA
Database Name	Database name or oracle service name
Database Server	Database Servername e.g. localhost

Database Port	Port of the Database e.g. • MySQL: 3306 • Oracle: 1541 • **PostgreSQL: 5432** • Microsoft SQL Server: 1433 • SAP HANA: 39013
Database Credentials	Database Connection Credentials
ODBC Drivername	Name of the ODBC driver - the field is only relevant for MySQL, PostgreSQL and Microsoft SQL Server. The following driver are available for selection: • *MySQL: MySQL ODBC 8.0 Unicode Driver* • *MS SQL Server: SQL Server Native Client 11.0,* • *MS SQL Server: ODBC Driver 17 for SQL Server* • ***PostgreSQL: PostgreSQL Unicode*** Additional driver can be added in the SQL Task Universal Template under: *Administration Universal Templates SQL Fields odbc_drivername New*
Script	The database script to execute Note: The name should not contain a Universal Controller Variable e.g. script name: *sb-**${ENV}**-proc01* will **not** work, because it contains a variable (${ENV}) in the name.
Get Output	*prints the SQL output to STDOUT* *In case of an Oracle Stored procedure print the dbms_output to STDOUT.*
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example: Run a SQL Script on PostgreSQL

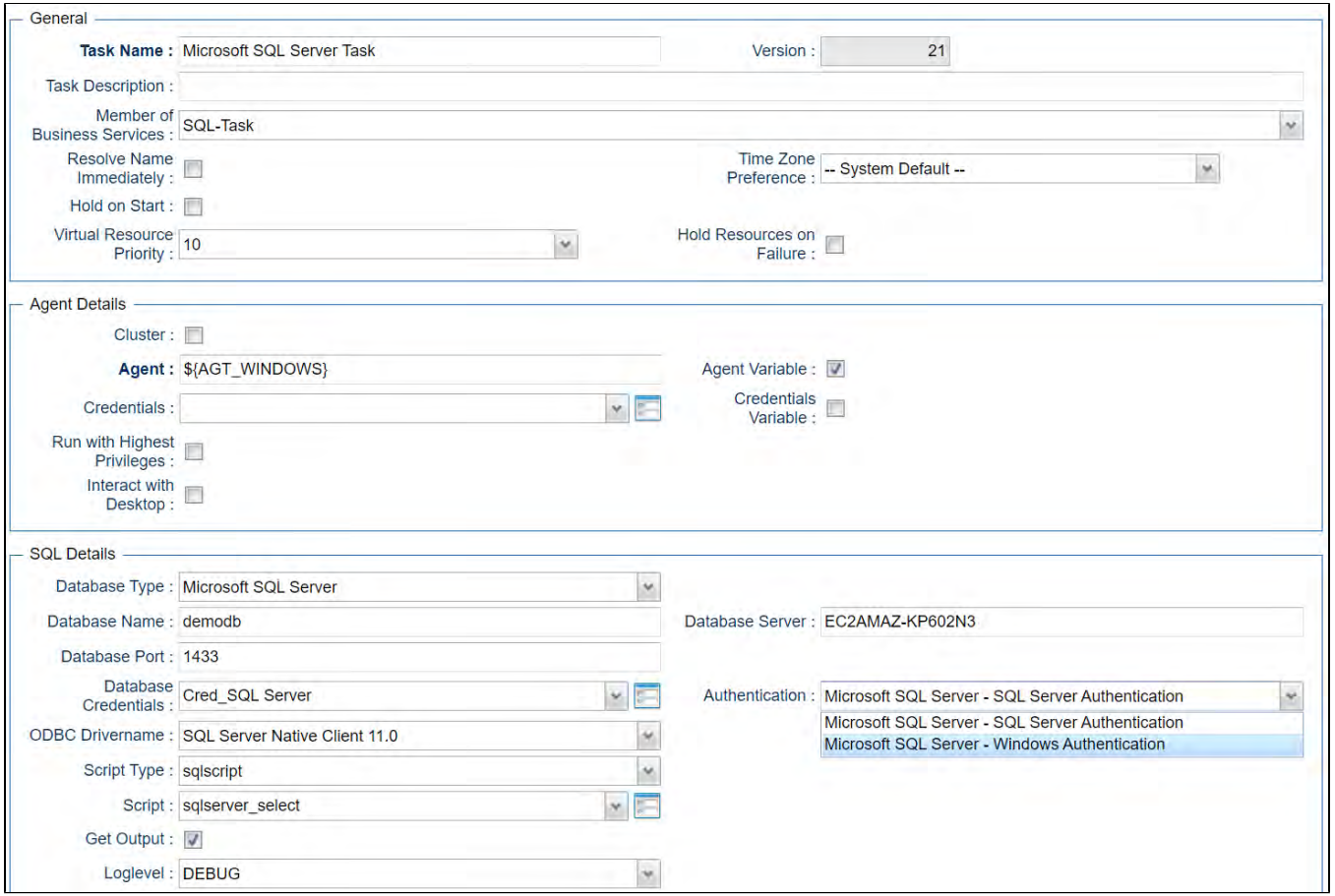


Database Type: Microsoft SQL Server

Field	Description
Database Type	Type of database to connect: • MySQL • Oracle • PostgreSQL • **Microsoft SQL Server** • SAP HANA
Database Name	**Database name** or oracle service name
	Database Servername e.g. localhost

Database Server	
Database Port	Port of the Database e.g. • MySQL: 3306 • Oracle: 1541 • PostgreSQL: 5432 • **Microsoft SQL Server: 1433** • SAP HANA: 39013
Database Credentials	Database Connection Credentials
Authentication	*Supported Methods:* • *Microsoft SQL Server - Windows Authentication* • *Microsoft SQL Server - SQL Server Authentication* Note: If "*Microsoft SQL Server - Windows Authentication*" is chosen you should select under the Agent "Credentials" the Windows user, who should execute the script. The database credentials are not used for SQLSERVER -Windows Authentication.
ODBC Drivername	Name of the ODBC driver - the field is only relevant for MySQL, PostgreSQL and Microsoft SQL Server. The following driver are available for selection: • *MySQL: MySQL ODBC 8.0 Unicode Driver* • ***MS SQL Server: SQL Server Native Client 11.0,*** • ***MS SQL Server: ODBC Driver 17 for SQL Server*** • *PostgreSQL: PostgreSQL Unicode* Additional driver can be added in the SQL Task Universal Template under: *Administration Universal Templates SQL Fields odbc_drivername New*
Script	The database script to execute Note: The name should not contain a Universal Controller Variable e.g. script name: *sb-${**ENV**}-proc01* will **not** work, because it contains a variable (${ENV}) in the name.
Get Output	*prints the SQL output to STDOUT* *In case of an Oracle Stored procedure print the dbms_output to STDOUT.*
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example: Run a SQL Script on Microsoft SQL Server

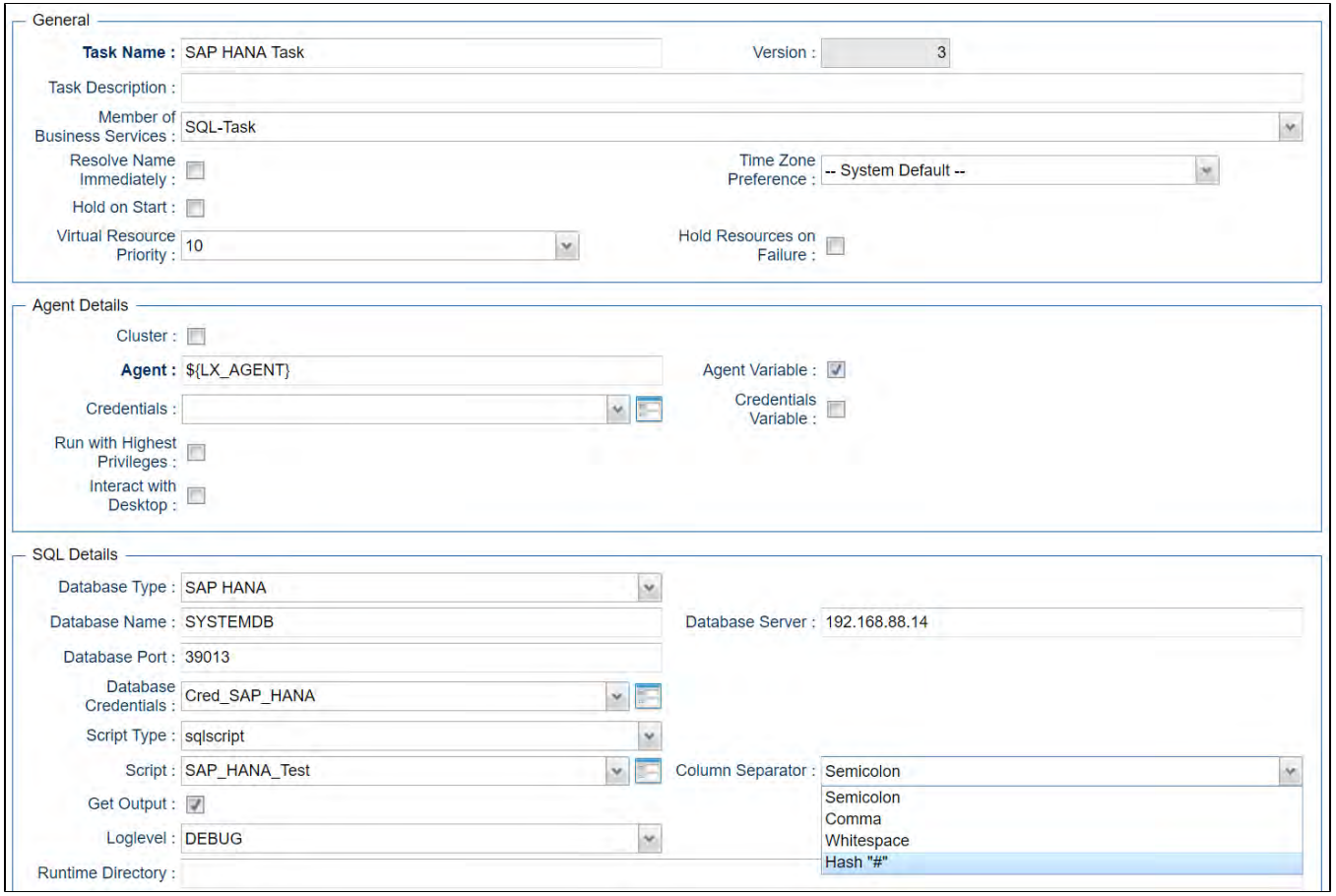


Database Type: SAP HANA

Field	Description
Database Type	Type of database to connect: • MySQL • Oracle • PostgreSQL • Microsoft SQL Server • **SAP HANA**
Database Name	Database name or oracle service name
Database Server	Database Servername e.g. localhost

Database Port	Port of the Database e.g. • MySQL: 3306 • Oracle: 1541 • PostgreSQL: 5432 • Microsoft SQL Server: 1433 • **SAP HANA: 39013**
Database Credentials	Database Connection Credentials
Script	The database script to execute Note: The name should not contain a Universal Controller Variable e.g. script name: *sb-${**ENV**}-proc01* will **not** work, because it contains a variable (${ENV}) in the name.
Column Separator	The field is only relevant for SAP HANA connections Output separator [Semicolon | Comma | Hash | Whitespace]
Get Output	*prints the SQL output to STDOUT* *In case of an Oracle Stored procedure print the dbms_output to STDOUT*.
Loglevel	Universal Task logging settings [DEBUG | INFO| WARNING | ERROR | CRITICAL]

Example: Run a SQL Script on SAP HANA



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.

UiPath for UAC Integration

Disclaimer

This download is designed as a template to be adapted to your environment. In some cases, templates will need to be changed to work with your current Universal Automation Center (UAC) setup. This download is free to use. However, the download is not supported, and no warranty is provided by Stonebranch for this document and the related download. The use of this document and the related download is at your own risk. Before using this download in a production system, please perform testing.

Stonebranch assumes no liability for any issues caused by the performance of this download.

Request Support:

Stonebranch offers paid support, by request, for select Universal Tasks. Paid support provides installation and on-going technical support. Universal Tasks that are eligible for paid support will be noted as “support eligible” within the Universal Task listing page. To learn more, please contact Stonebranch.

Introduction

This Universal Task allows Stonebranch users to schedule, trigger, and monitor the UiPath (RPA) process directly from the Universal Controller.

Overview

- This task uses a Python request module to make REST-API calls to the UiPath orchestrator.
- It can trigger the RPA process in UiPath using just UiPath process name, orchestrator base URL, UiPath account name and service instance.
- The task triggers the UiPath process for execution, monitors until process completion, and populates the results in Universal Controller.
- It also features a tight integration with ITSM tools, meaning that it can auto-create incidents in case of UiPath RPA process execution failure.

Software Requirements

This integration requires an Universal Agent and a Python runtime to execute the Universal Task against a UiPath orchestrator instance.

Software Requirements for Universal Template and Universal Task

- Requires Python 3.6 or higher. Tested with the Universal Agent bundled Python distribution.
- Python modules required:
    - requests

Software Requirements for Universal Agent

Either:

- Universal Agent for Windows x64 Version 6.7.0.0 and later with python options installed
- Universal Agent for Linux Version 6.7.0.0 and later with python options installed

Software Requirements for Universal Controller

- Universal Controller Version 6.6.0.0 and later

Software Requirements for the Application to be Scheduled

This Universal Task can work with the following UiPath Orchestrator versions:

- 2020.10.1
- 2020.4.1
- 2019.10.14
- 2018.4.1

Technical Considerations

Please note that the UiPath access token is consumed by this Universal Task from the Global Variable named: Uipath_access_token. You can update the access token periodically through a web services task.

UiPath Key Features

Feature	Description
Trigger a UiPath Job	Trigger a UiPath job/process and monitor the UiPath (RPA) process directly from the Universal Controller.

Import UiPath Built-In Universal Template

To use the built-in Universal Template, you first must perform the following steps:

1. This Universal Task requires the Resolvable Credentials feature. Check that the Resolvable Credentials Permitted system property has been set to **true**.
2. Copy or Transfer the Universal Template file to a directory that can be accessed by the Universal Controller Tomcat user.
3. In the Universal Controller UI, select Configuration > Universal Templates to display the current list of Universal Templates.
4. Right-click any column header on the list to display an Action menu.

5. Select Import from the menu, enter the directory containing the Universal Template file(s) that you want to import, and click OK.

When the files have been imported successfully, the Universal Template will appear on the list.

Configure UiPath Universal Task

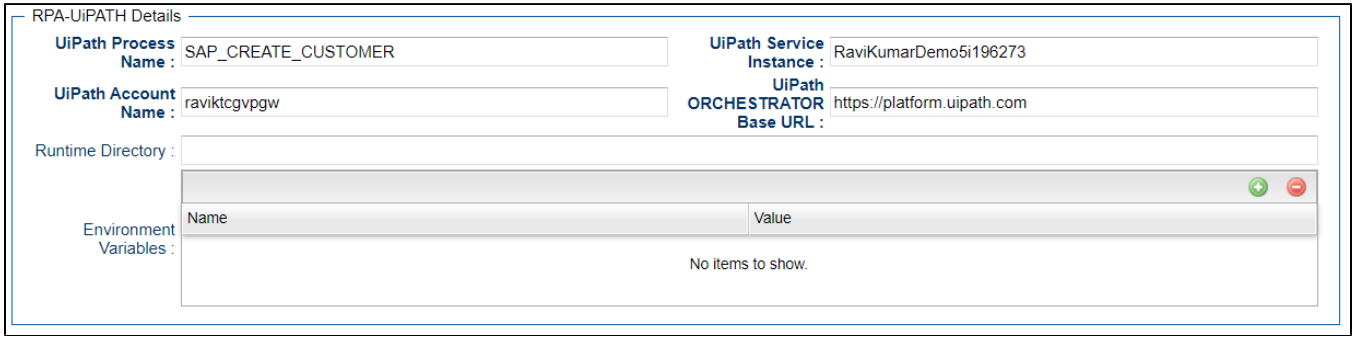
For the new Universal Task type, create a new task and enter the task-specific details that were created in the Universal Template.

Field Descriptions for UiPath Universal Task

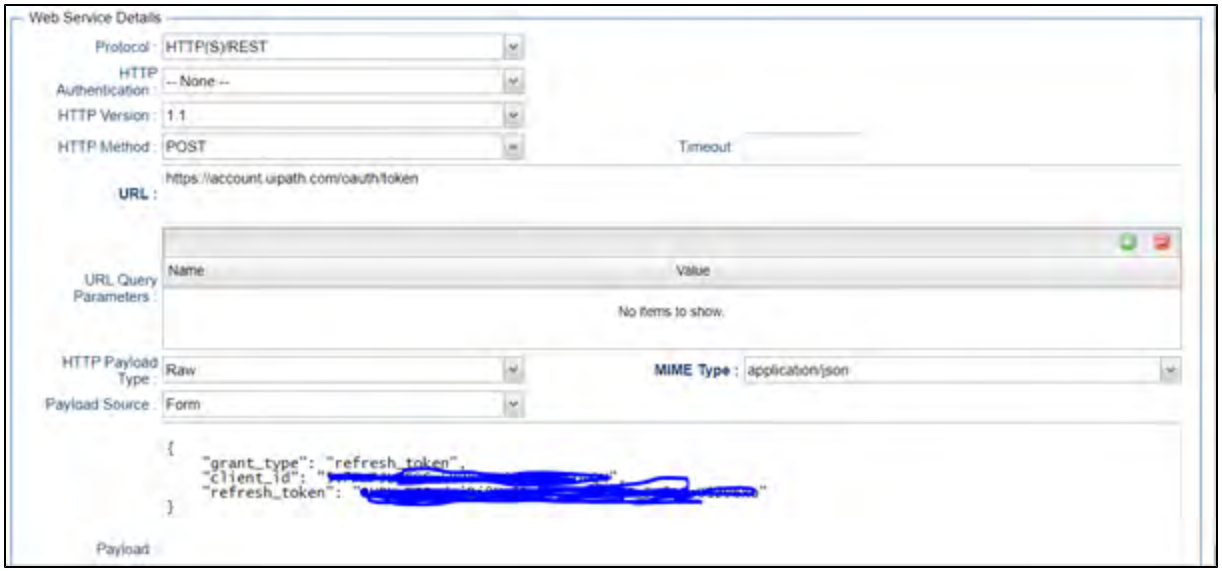
Field	Description
UiPath Process Name	Actual RPA workflow name that needs to be executed by UiPath Bot
UiPath service Instance	Service instance name can retrieve using the Api get: https://platform.uipath.com/cloudrpa/api/account/[account_logical_name]/getAllServiceInstances
UiPath Orchestrator URL	URL for the UiPath Orchestrator where its installed. Normally the cloud orchestrator/Community URL is https://platform.uipath.com:
UiPath Account Name	This is the UiPath Orchestrator Tenant Name: https://platform.uipath.com/cloudrpa/api/getAccountsForUser

Examples for UiPath Universal Tasks

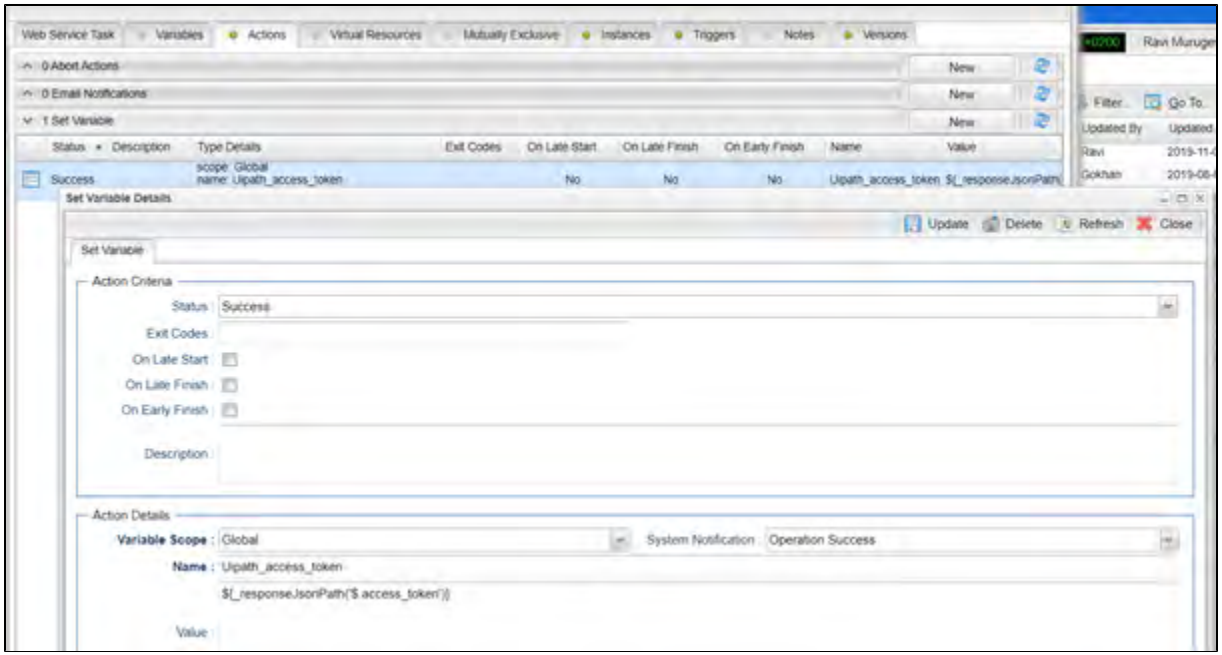
UiPath Sample Universal Task



Sample webservices Task for Access Token Generation



Generated Access Token can be Stored in a Global Variable by Using the UAC function for the Above webservices Task in Action Set Variable



Document References

This document references the following documents:

Name	Location	Description
Universal Templates	https://docs.stonebranch.com/confluence/display/UC70/Universal+Templates	User documentation for creating Universal Templates in the Universal Controller user interface.
Universal Tasks	https://docs.stonebranch.com/confluence/display/UC70/Universal+Tasks	User documentation for creating Universal Tasks in the Universal Controller user interface.