

SDA: Post Processing Script Quick Start Tutorial

A post processing script can be used as a mechanism for collecting output from a particular process step and storing it as a property to be referenced in another step. A typical use case for post processing scripts would be to detect errors or abnormalities in the process step output.

In this tutorial will focus on another use of post processing scripts to collect specific information from the process step output and store it as properties for use in a later step. A generic process will be created that is designed to run on an agent hosted on a Windows system.

Table of Contents

SDA: Post Processing Script Quick Start Tutorial	1
Step 1: Create a new global process	2
Step 2: Add a shell step	3
Step 3: Create a Post Processing Script	4
Step 4: Update Global processs with Post Processing Script.....	5
Step 5: Run the process and view the output properties	5
Further tasks	6

Step 1: Create a new global process


Note: Global processes are a good option for prototyping and testing new functionality in SDA.

- Login to SDA under **Management | Global Processes** click on the **Create Process Button** 
- Specify a Name: **Get OS versions and free space** and click **Build Out**


Create Process ✕

Name *

Description

Default Working Directory * 

Step 2: Add a shell step

- Click the Design Button  to invoke the process designer
- On the left hand side, under **Tools** select **Scripting | Shell | Shell** and drag a new Shell step to the beginning of the process flow.
- Under the properties tab on the left hand side specify the name as: **GetOSV and Freespace**
- Paste the following batch file code into the **Shell Script** field:

```
@echo off

echo Obtaining OS version and free space

ver

echo Getting free space

for /F "usebackq tokens=3" %%s IN (`DIR C:\ /-C /-O /W`) DO (

    set FREE_SPACE=%%s

)

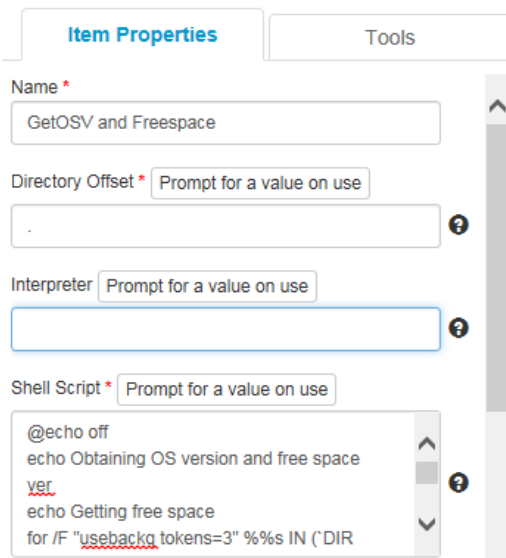
echo Free: %FREE_SPACE%

echo Complete
```

Note: The batch script uses the ver command to report on the Windows version currently being run. It also runs some batch code in order to extract number of bytes free on c:

Note: To achieve the same functionality in Linux or UNIX you could replace the ver command with `uname -a`, and use the command `df -k` to return the free space on all the mounted partitions.

- Join the shell step to the start and end steps



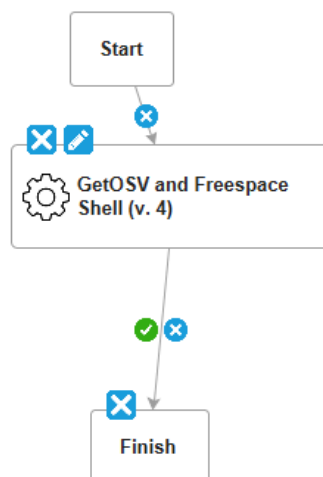
Item Properties | Tools

Name *
GetOSV and Freespace

Directory Offset * Prompt for a value on use
.


Interpreter Prompt for a value on use

Shell Script * Prompt for a value on use
@echo off
echo Obtaining OS version and free space
ver
echo Getting free space
for /F "usebackq tokens=3" %%s IN (`DIR



- Click the **Save Icon** 

Step 3: Create a Post Processing Script

- Click the **Home** menu button 
- Then click **Administration | Automation**
- From the drop down select **Post Processing Scripts**
- Click the add script button 

Edit Script

Name *

Description

Script Body *

```
properties.put("Status", "Success");
var process = [];

scanner.register("(?)Microsoft Windows ", function(lineNumber, line)
{
var WinVersion = line.replace("Microsoft Windows ", "");
properties.put("OS_VERSION", WinVersion);
});

scanner.register("(?)Free: ", function(lineNumber, line)
```

- Within the script body, paste the following Javascript code


```
properties.put("Status", "Success");
var process = [];



scanner.register("(?)Microsoft Windows ", function(lineNumber, line)
{
var WinVersion = line.replace("Microsoft Windows ", "");
properties.put("OS_VERSION", WinVersion);
});

scanner.register("(?)Free: ", function(lineNumber, line)
{
var FreeSpace= line.replace("Free: ", "");
properties.put("C_FREE", FreeSpace);
});

scanner.scan();
```

Step 4: Update Global process with Post Processing Script

- Click the **Home** menu button 
- Then click **Management | Global Processes**
- Click **Get OS versions and free space**

- In the process within the **Get OS and free space** step on the pen icon 
- On the left hand pane Click the **drop down** under Post Processing Script and select **Get OS and free space**
- Click the **Save Icon** 

Step 5: Run the process and view the output properties




- Exit the process designer by click in the X icon
- In the top right corner click the **Run** button, select an agent and click **Next**
- On the next screen click **Submit**

- Wait till the process completes, and click the **output** icon
- Output similar to the following should appear, notice the Windows version and free output are both highlighted

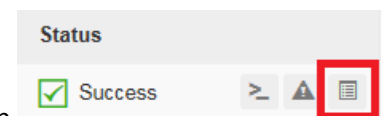
```
-----
16
17 command output:
18 Obtaining OS version and free space
19
20 Microsoft Windows [Version 6.3.9600]
21 Getting free space
22 Free: 3465392128
23 Complete
```

- Click Close to dismiss the dialog

- Click the **properties icon** to view the input and output properties to the step
- From the drop down select **Output properties**
- Notice the properties OS_VERSION and C_FREE with their respective values

Output properties 

Name	Value
Status	Success
logPath	g:\Program Files (x86)\s Agent\core\var\temp\78dde116d228fc\logs6984
LOI	20,22
OS_VERSION	[Version 6.3.9600]
C_FREE	3465392128



Further tasks

The output properties can be referenced subsequent process steps. For example, if we wanted to reference the `OS_VERSION` property we could use the syntax: `${p:<step-name>/<property-name>}` or `${p:prevStep/<property-name>}`

An optional task would be to expand the process **GetOSV and Freespace** to run a shell script step that reports the operating system of the agent. To do this add a second Shell step to run the following:

```
echo Agent Operating System: ${p:prevStep/OS_VERSION}
```

Save and re-test the process.