

## Automatic Rollback Tutorial with SDA 6

### Introduction

Automatic Rollbacks are an important part of any deployment activity. A typical use case scenario might be that part of the deployment has failed or a certain environmental setting is incorrect, which would require the system to automatically be rolled back to the previous state.

### An example scenario

The following logic would be representative of a typical rollback scenario:

- Release 4 of the software is currently deployed.
- Release 5 is then deployed and is dependent on Artefact X being in place, so during post deployment a check is made.
- If Artefact X is not in place, rollback to Release 4.

### Pre-requisite set up

This scenario assumes that the following have been defined:

- A simple application called **App1** and its associated component **Ap1Comp1**
- **App1** has: an environment mapped to a resource and a process defined
- **Ap1Comp1** is of type File Versioned
- The component **Ap1Comp1** contains two versions **V1** and **V2**
- **V1** - contains the file: **hello1.txt**
- **V2** - contains the files: **hello1.txt hello2.txt**

### Steps to define and test the rollback scenario

The following four steps will give an overview of how an existing application can be configured.

Step 1: Ensure that the Inventory Management is set to Advanced .....	2
Step 2: Define two processes for the component.....	3
Step 3: Define the Application Process Logic .....	5
Step 4: Test the rollback and success scenarios.....	9

## Step 1: Ensure that the Inventory Management is set to Advanced

In order to use rollback steps in an application process, the inventory management needs to be changed from the default of Automatic to Advanced.

To do this

1. Click on the Application
2. Click on the **Processes** button in the application toolbar
3. Click on the Application Process
4. Click on X to dismiss the process designer
5. Click on the **Edit** button
6. Set the **Inventory Management** to **Advanced**

The screenshot shows the 'RunAp1: App1' interface. At the top, there are 'Design' and 'Version' buttons. Below them, the process name 'RunAp1' and application 'App1' are displayed. The 'Inventory Management' is set to 'Advanced'. A dialog box titled 'Edit Application Process' is open, showing the following fields:

- Name: RunAp1
- Description: (empty text area)
- Required Application Role: none
- Inventory Management: Advanced (highlighted with an orange box)
- Offline Agent Handling: Check Before Execution
- Version for Ap1Comp1: None

Buttons for 'Cancel' and 'Save' are visible at the bottom right of the dialog box.

## Step 2: Define two processes for the component

Two processes should be defined for the component

1. A process performs the deployment and a diagnostic test (**Deploy and Test App 1**)
2. A process that simply deploys the artefacts. (**Deploy App 1**)

Ap1Comp1

Activity Details Inventory Versions **Processes** Tasks Templates History Security

Inactive

Search 10 per page 1 / 1

Name	Description	
Run Deploy and Test		
Run Deploy Only		

### Component process definition: Run and Deploy Test

Create the **Run Deploy and Test** process which contains the download and diagnostic step



**Step definition: Check if Directory exists**

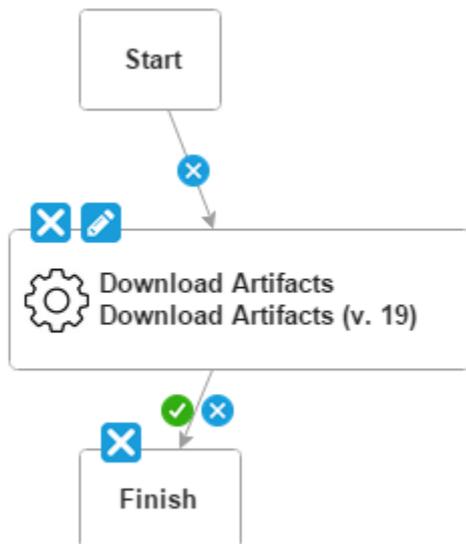
The **Check if Directory exists** step, checks for the presence of a directory, e.g. `c:\work\sda\comp1\test`

The screenshot shows a dialog box with two tabs: 'Item Properties' (selected) and 'Tools'. Under 'Item Properties', there is a 'Name' field with the text 'Check if Directory exists'. Below it is a 'Directories' field with a dropdown menu set to 'Prompt for a value on use' and a text area containing 'c:\work\sda\test'. At the bottom, there is an unchecked checkbox labeled 'Ignore Failure'.

Note: Please ensure that when specifying the path, ensure the default setting containing the dot value: `.` is removed.

**Component process definition: Run Deploy Only**

The **Run Deploy Only** process that performs just the download step



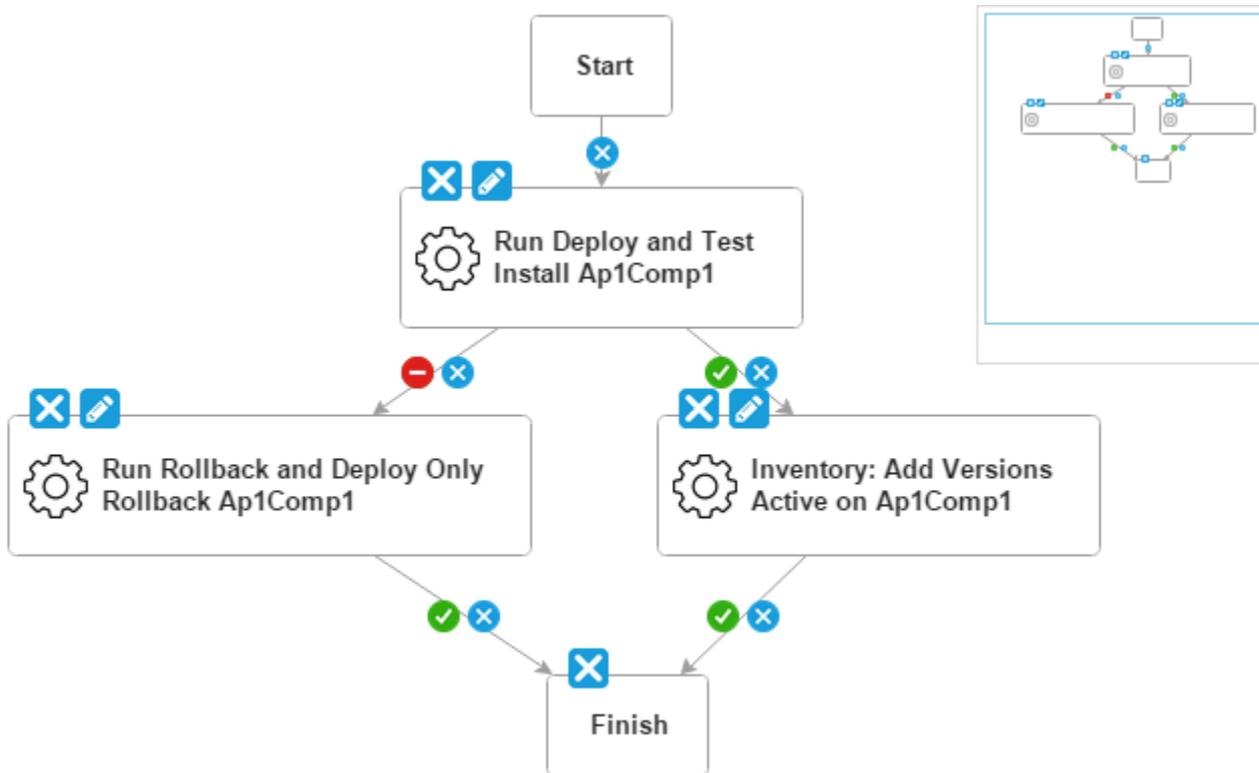
### Step 3: Define the Application Process Logic

1. Click on the Application App1
2. Click on the **Processes** button in the application toolbar
3. Add a new Application Process called **Deploy and Test App1** with an **Inventory Management** set to **Advanced** then click on **Save**.

4. Click on the newly defined process to invoke the process designer and create the following process

#### Application Process Deploy and Test App1 definition

The Application Process should contain the following logic



The process flow above describes the following logic:

1. Attempt a deployment
2. If it fails rollback the deployed files
3. If successful add to the inventory

Note: The Inventory Add Versions step ensures that the version is added to the request inventory if the component installs correctly.

### Step definition: Run Deploy and Test step

Contains the following and is defined from

<b>Item Properties</b>	Tools
Name *	
<input type="text" value="Run Deploy and Test"/>	
Component *	
<input type="text" value="Ap1Comp1"/>	
Use Versions Without Status *	
<input type="text" value="Active"/>	
Component Process *	
<input type="text" value="Run Deploy and Test"/>	
<input type="checkbox"/> Ignore Failure	

### Step definition: Inventory: Add Versions

The **Inventory: Add Versions** step contains the following and is obtained from **Utility Steps | Inventory Update**

<b>Item Properties</b>	Tools
Component *	
<input type="text" value="Ap1Comp1"/>	
Limit to Resource Role	
<input type="text" value="Any"/>	
Action *	
<input type="text" value="Add Desired Inventory"/>	
Status *	
<input type="text" value="Active"/>	

**Step Definition: Rollback and Deploy Only step**

1. Calls the **Run Deploy Only** component process
2. On the Ap1Comp1 component
3. Set the Component Process to be **Run Deploy Only**
4. The **Rollback Type** must set to **Replace with Last Deployed**.

Item Properties Tools

Name \*  
Run Rollback and Deploy Only 1

Component \*  
Ap1Comp1 2

Remove Versions With Status \*  
Active

Component Process \*  
Run Deploy Only 3

Ignore Failure ?

Limit to Resource Role  
Any

Rollback Type \*  
Replace with Last Deployed 4

Max # of concurrent jobs \*

### Modify the Deploy Only Download Artifacts step

Currently a rollback will not remove files that were created in the version being rolled back from.

E.g.

V1: hello1.txt

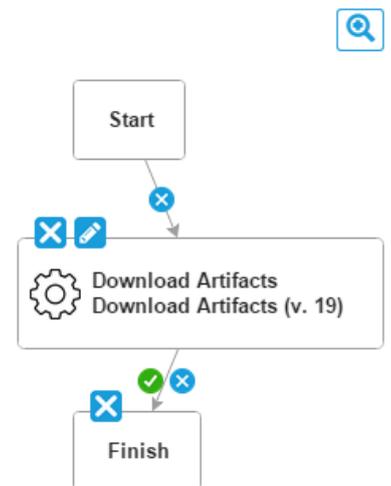
V2: hello1.txt hello2.txt

A rollback from V2 to V1 would leave **hello2.txt** intact

Note: In order to remove files that were created in the version to be rolled back the following change needs to be made.

1. Modify the component process **Run Deploy Only**,
2. Edit the Download Artifacts step
3. Modify the value **Sync Mode** to **Sync and Clean**, as shown below and save the modification

The screenshot shows the 'Item Properties' configuration panel for a 'Download Artifacts' step. The 'Sync Mode' dropdown menu is highlighted with an orange border and shows 'Sync and Clean' selected. Other fields include Name (Download Artifacts), Directory Offset (Prompt for a value on use), Includes (Prompt for a value on use, with a text area containing '\*\*/\*'), Excludes (Prompt for a value on use), and Sync Mode (Prompt for a value on use).



## Step 4: Test the rollback and success scenarios

Note: The component **Ap1Comp1** contains two versions **V1** and **V2**

### Rollback Test

1. Ensure directory (e.g. `c:\work\sda\comp1\test`) exists for the second diagnostic step
2. Deploy **V1** using **Deploy and Test App1**
3. **V1** is deployed and second diagnostic step is successful
4. Remove the directory<sup>1</sup> checked by the second diagnostic step
5. Deploy **V2** using **Deploy and Test App1**
6. **V2** is deployed and second step FAILS
7. Rollback step is executed
8. **V1** is deployed
  - e.g. file **hello1.txt** exists in the Agent Deployment area

### Rollback not required test

1. Ensure directory (e.g. `c:\work\sda\comp1\test`) exists for the second diagnostic step
2. Deploy **V1** using **Deploy and Test App1**
3. **V1** is deployed and second diagnostic step is successful
4. Deploy **V2** using **Deploy and Test App1**
5. **V2** is deployed and second step is successful
  - e.g. files: **hello1.txt hello2.txt** exist in the Agent Deployment area

## Further Tasks

The scenario above is a simple example, more complex scenarios can also be managed by SDA. Examples of more complex scenarios would include:

- Rolling back multiple components in the correct order
- Rollback components that use incremental versions – E.g. a database
- Pausing before automated rollback (with manual tasks) to see if environment issues could be resolved

*End of Document*

---

<sup>1</sup> E.g. delete the directory `c:\work\sda\comp1\test` from the local disc on the machine the agent is installed on