

IBM UrbanCode Deploy



Android Plug-in

Version 1.0 This information is under development and can change at any time; the released version will be available on ibm.com.

Note

Before using this information and the product it supports, read the information in "Notices," on page 27.

This edition applies to version 1.0 of Android plug-in for IBM UrbanCode Deploy and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Android SDK plug-in	1
Software requirements and compatibility for the Android SDK plug-in	1
Configuring the agent computer	2
Configuring the Android SDK	3
Configuring the emulator	4
Configuring the physical device	4
Building and deploying mobile applications	5
Examples for the Android SDK plug-in	6
Example: Testing an Android application on an emulator.	6
Example: Testing an Android application on a physical device	8
Example: Concurrent testing of an Android app	11
Process steps in the Android SDK plug-in	12
Install Application	12

Remove Application	14
Create Emulator.	16
Remove Emulator.	17
Start Emulator	17
Stop Emulator	19
Unit Test	20
UI Test.	21
Random UI Test	23
ADB	25
Android.	26
Appendix. Notices	27
Privacy policy considerations	29
Trademarks	29

Android SDK plug-in

The Android SDK plug-in includes steps to achieve continuous delivery of Android applications by interacting with emulators and physical devices.

The plug-in includes process steps that are related to the following categories:

Table 1. Categories and related process steps

Category	Process Steps
Application installation and removal	<ul style="list-style-type: none">• “Install Application” on page 12• “Remove Application” on page 14
Emulator management	<ul style="list-style-type: none">• “Create Emulator” on page 16• “Remove Emulator” on page 17• “Start Emulator” on page 17• “Stop Emulator” on page 19
Application testing	<ul style="list-style-type: none">• “Unit Test” on page 20• “UI Test” on page 21• “Random UI Test” on page 23
Interfacing with tools	<ul style="list-style-type: none">• “ADB” on page 25• “Android” on page 26

To install this plug-in, see Installing plug-ins.

Software requirements and compatibility for the Android SDK plug-in

The Android SDK plug-in has specific software requirements and compatible software.

IBM® UrbanCode Deploy requirements

The following table lists the required versions of IBM UrbanCode Deploy for the Android SDK plug-in.

Plug-in version	Required UrbanCode Deploy version
Android SDK plug-in Version 1.0	IBM UrbanCode Deploy Version 6.0.1 or later.

Agent computer requirements

In UrbanCode Deploy, agents are processes that do the actual work of deploying components. You can install agents on the same computer where you installed UrbanCode Deploy, or you can install agents on separate computers.

Restriction: The agent must run on a computer that is configured to use English as the system language.

Compatible Android SDK version	Java™ requirements
Android SDK Tools 22.3 (ADB 1.0.31). Note: To run the emulator on 64-bit Linux computers, install the 32-bit package. Restriction: To use the uiautomator testing framework, you must have the following versions of the Android development tools installed: <ul style="list-style-type: none"> • Android SDK Tools 21 or later • Android SDK Platform API 16 or later 	<ul style="list-style-type: none"> • Oracle JDK 6 or later. • IBM SDK for Java 6 or later. Note: You must install the JDK or SDK, not a just a runtime environment.

Build computer requirements

The computer that builds the Android application must have the following software installed. Depending on your environment, you might need to install extra resources on the build computer.

Compatible Android SDK version	Java requirements
Android SDK Tools 22.3 (ADB 1.0.31).	<ul style="list-style-type: none"> • Oracle JDK 6 or later. • IBM SDK for Java 6 or later. Note: You must install the JDK or SDK, not a just a runtime environment.

Related reference:

“**UI Test**” on page 21

UI testing simulates a user interacting with an application. For example, this step might include moving between windows and verifying the information that is displayed. This step is the Android SDK-based uiautomator testing framework. When this process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Configuring the agent computer

The agent is the process that deploys the Android application. You must install the Android SDK on the computer that runs the agent.

Procedure

To install the Android SDK on the agent computer:

1. Install the Android SDK tools. Download and install either the **Android ADT bundle** or the **Android SDK tools** only from <http://developer.android.com/sdk/index.html#download>.
2. Using the SDK Manager, you might have to install the following packages:
 - SDK Tools.
 - Platform Tools.
 - One or more SDK Platforms. The SDK Platform is the version of Android that your application runs on. (It is the Target ID when you use the “**Create Emulator**” on page 16 plug-in step.)
 - If you are deploying to an emulator, ensure that you install the System Image for the SDK Platform.

Note: The first time that you run the emulator or work with the other SDK tools, a dialog box opens asking if you want to send usage statistics to Google. You must respond to the dialog box one time before you can work with any of the tools.

Because this dialog box can interfere with testing on an emulator, you can use the following steps to trigger and respond to the dialog box:

- a. Run the Android Device Monitor to prompt the dialog box. In the *Android SDK installation location/tools* directory, run `monitor.bat`.
- b. When the usage statistics dialog box opens, select a response and click **Proceed**.
- c. Close the Android Device Monitor.

For more information about the Android SDK Manager and the SDK packages, see <http://developer.android.com/tools/help/sdk-manager.html>

What to do next

If you are attaching a physical device to the agent computer, you must install drivers to connect the device. See “Configuring the physical device” on page 4.

Configuring the Android SDK

You must configure the agent with the location of the Android SDK.

Before you begin

Install the agent and add the agent as a resource:

1. Install an agent on the target computer. For information on installing agents, see http://pic.dhe.ibm.com/infocenter/ucdeploy/v6r0m1/topic/com.ibm.udeploy.doc/topics/resources_agents.html.
2. On the IBM UrbanCode Deploy server, verify that the agent is installed by clicking **Resources > Agents**. Your agent must be included in the list of agents.
3. Add the agent as a resource.
 - a. To the left of the Agents tab, click the **Resources** tab.
 - b. Click **Create Top-Level Group**, enter a **Name** for the new group, and then click **Save**. This group is a container for resources, such as agents and components, and is mapped to the deployment environment. On the same line as the resource group, click **Actions > Add Agent**.
 - c. In the **Create New Resource** window, select the agent in the Agent list, and then click **Save**. The agent displays as a child of the resource group.

For information about resources, see http://pic.dhe.ibm.com/infocenter/ucdeploy/v6r0m1/topic/com.ibm.udeploy.doc/topics/resources_ch.html

About this task

By defining the path to the Android SDK as a property of the Resource, you can reuse the property across steps.

Procedure

Add the path to the Android SDK to the resource:

1. In IBM UrbanCode Deploy, click **Resources** and then click the resource that you created in the previous step.
2. Click the **Configuration** tab and then click **Resource Properties**.
3. Click **Add Property**.
4. In the **Name** field, enter `AndroidSDKPath`. In the **Value** field, enter the path to the location of the Android SDK on the agent computer. For example, enter `C:\mobile\android-sdk-windows`.

Configuring the emulator

You can improve emulator performance by using snapshots and hardware acceleration. Also, you might want to automate updating the emulator.

About this task

You might need to perform several tasks to configure the emulator:

- Ensure that the emulator supports the Android application that you are deploying.
- If a firewall is installed on the agent computer, it might prevent communication with the emulator.
- To improve performance of the emulator, you can use the following methods:
 - In the Android AVD Manager, use the **Save to Snapshot** and **Launch from Snapshot** options to decrease the amount of time it takes to start the emulator. Snapshots save the state of the virtual device.

Tip: You can also use the steps in this plug-in to work with Snapshots.

- Use hardware acceleration. For details on using hardware acceleration, see <http://developer.android.com/tools/devices/emulator.html#acceleration>

For information about working with the emulator, see <http://developer.android.com/tools/devices/emulator.html>.

Configuring the physical device

If you are deploying to a physical device, configure the device and the agent computer so that they can communicate.

About this task

In general, the physical device must be attached to the agent computer and must be visible to the Android Debug Bridge (ADB). If you are attaching a physical device to the agent computer, you might need to perform several configuration tasks:

- Ensure that the physical device is configured so that the device is unlocked and does not lock after a time if the device is not used.
- Install the required drivers on the agent computer so that the physical device can be connected by USB. If you are working with Google Nexus devices, in most cases you must install the Google USB driver. For other devices, you must install the corresponding OEM USB driver.

For detailed instructions on connecting and configuring physical devices, see [Using Hardware Devices](#).

To verify that the device is successfully connected, run the ADB command **adb devices**. The command outputs a list of serial numbers and current states of the connected devices and emulators. For more information, see <http://developer.android.com/tools/help/adb.html#devicestatus>.

Building and deploying mobile applications

You can set up your development environment so that you can build your Android applications and, by using the Android SDK plug-in for IBM UrbanCode Deploy, deploy the build results to a target emulator or physical device for testing.

Before you begin

Ensure that IBM UrbanCode Deploy is installed and running.

Extra software might be required, such as:

- Source control management (SCM) system.
- Build engine.

About this task

Before you can build and deploy Android applications to a target physical device or emulator, you must complete the following configuration steps:

1. Configure the build system.
2. Configure the Android SDK.
3. Configure the agent computer.
4. Configure the physical device or configure the emulator for deployment.
5. Configure UrbanCode Deploy, including the following tasks:
 - Create components.
 - Create component processes or application processes that include steps from the Android SDK plug-in to deploy the Android application.

Procedure

After you set up the build and UrbanCode Deploy, you can build and deploy Android applications by using the following high-level steps:

1. Check in (commit) changes from your development environment into a source control management (SCM) system.
2. Build the application and add a new version to UrbanCode Deploy.

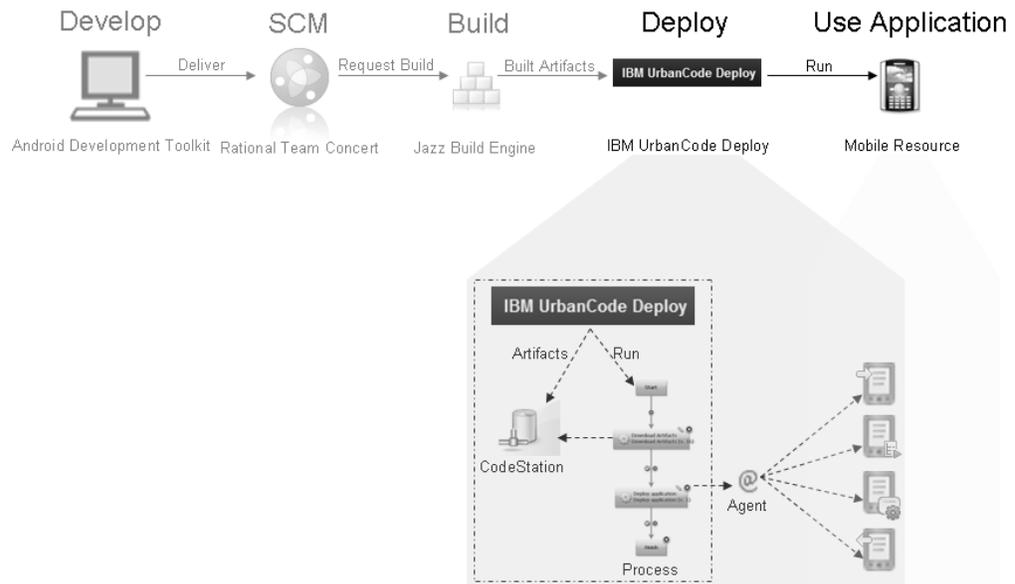
Tip: Assign a version of the Android application to the component version in UrbanCode Deploy. For example, if the Android application has a commercial version of 1.0 in the AndroidManifest.xml and the internal version from the latest build is 16, assign version 1.0.16 to the component version in UrbanCode Deploy. Keeping the version numbers synchronized helps you to recover if you encounter a problem. For example, if the latest version of the Android application was not deployed successfully to the emulator or device.

3. Request deployment in UrbanCode Deploy.

Results

The mobile application artifacts are deployed to the target device or emulator.

The following image shows the role of the Android SDK plug-in in the end-to-end process in an example development, build and deployment environment:



Examples for the Android SDK plug-in

The following examples show how to use the Android SDK plug-in. You must customize processes to your application and environment.

Example: Testing an Android application on an emulator

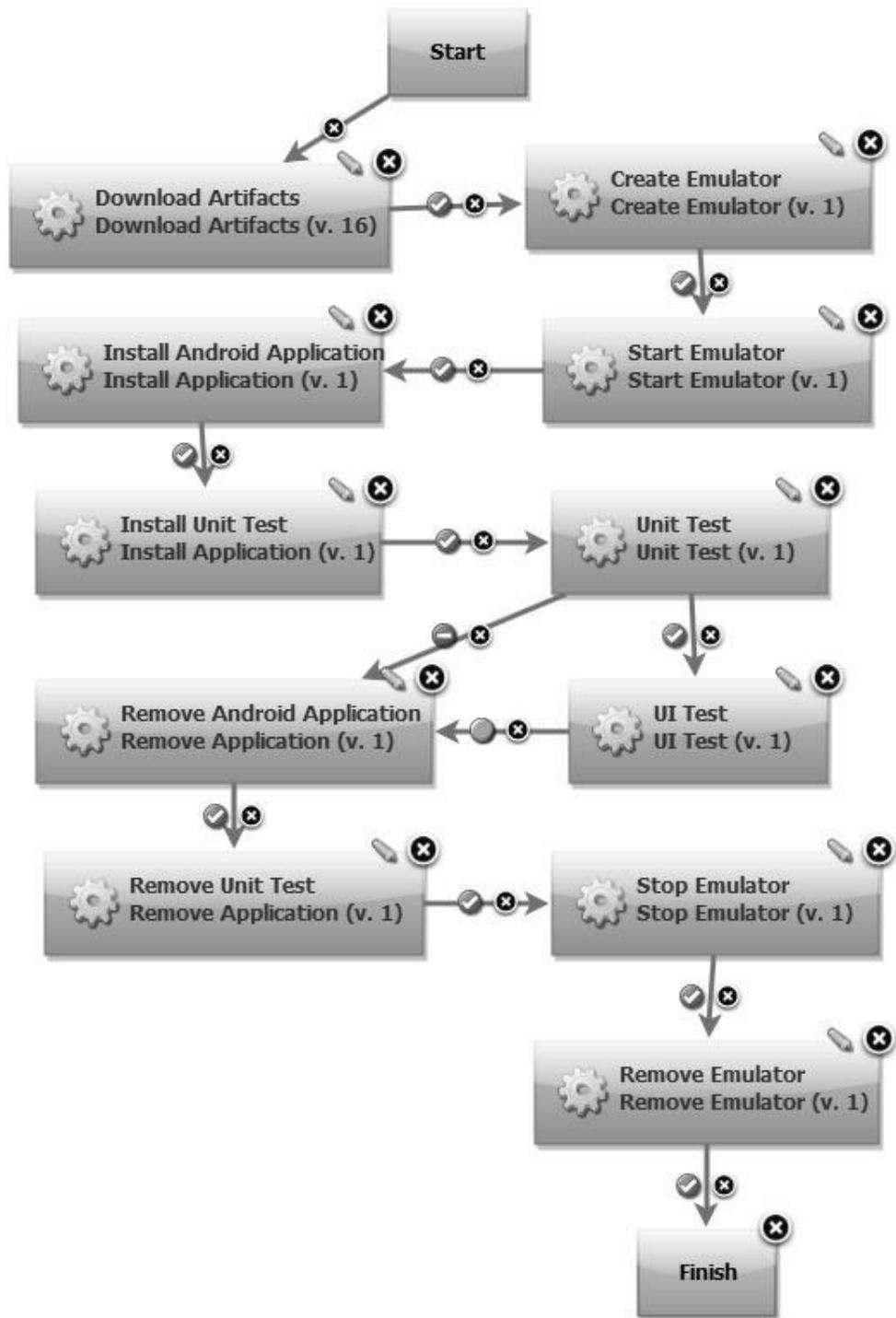
The following example component process tests an Android application on an emulator.

This process assumes that the application binary files are available in CodeStation in the component version. It also assumes that the agent and environment have the required software and configuration for this plug-in.

1. The **Download Artifacts** step (from IBM UrbanCode Deploy) retrieves the binary files from CodeStation.
2. The **“Create Emulator”** on page 16 step creates new emulator target. The step defines an emulator by name with the target Android version, processor type, and other details for the emulator.
3. The **“Start Emulator”** on page 17 step starts the emulator.
4. The **“Install Application”** on page 12 plug-in step that is named **Install Android Application** installs the specified Android application (.apk file) onto the emulator.
5. The **Install Application** plug-in step that is named **Install Unit Test** installs the specified unit test onto the emulator.
6. The **“Unit Test”** on page 20 step tests a single Activity in the application. In the step, you provide the name of the test package to run and the class name for the instrumented runner to use when it runs the test. If the **Unit Test** step completes successfully, the **“UI Test”** on page 21 step runs. If the **Unit Test** step fails, the **UI Test** step is skipped.
7. The **UI Test** step simulates a user interacting with the application, such as moving between screens and verifying the displayed information. In this step, you specify a list of JAR files that contain the test cases to be pushed to the

target and the class or package names of the tests to run. If you do not enter the path to the JAR files, then the files are not copied to the target.

8. The “**Remove Application**” on page 14 plug-in step that is named **Remove Android Application** removes the specified package from the target. You can select whether to keep the application data.
9. The **Remove Application** plug-in step that is named **Remove Unit Test** removes the unit test.
10. The “**Stop Emulator**” on page 19 step stops the target emulator. Stop any emulator that you start. Stopping the emulator frees up resources on the agent computer.
11. The “**Remove Emulator**” on page 17 step removes the emulator that you created.



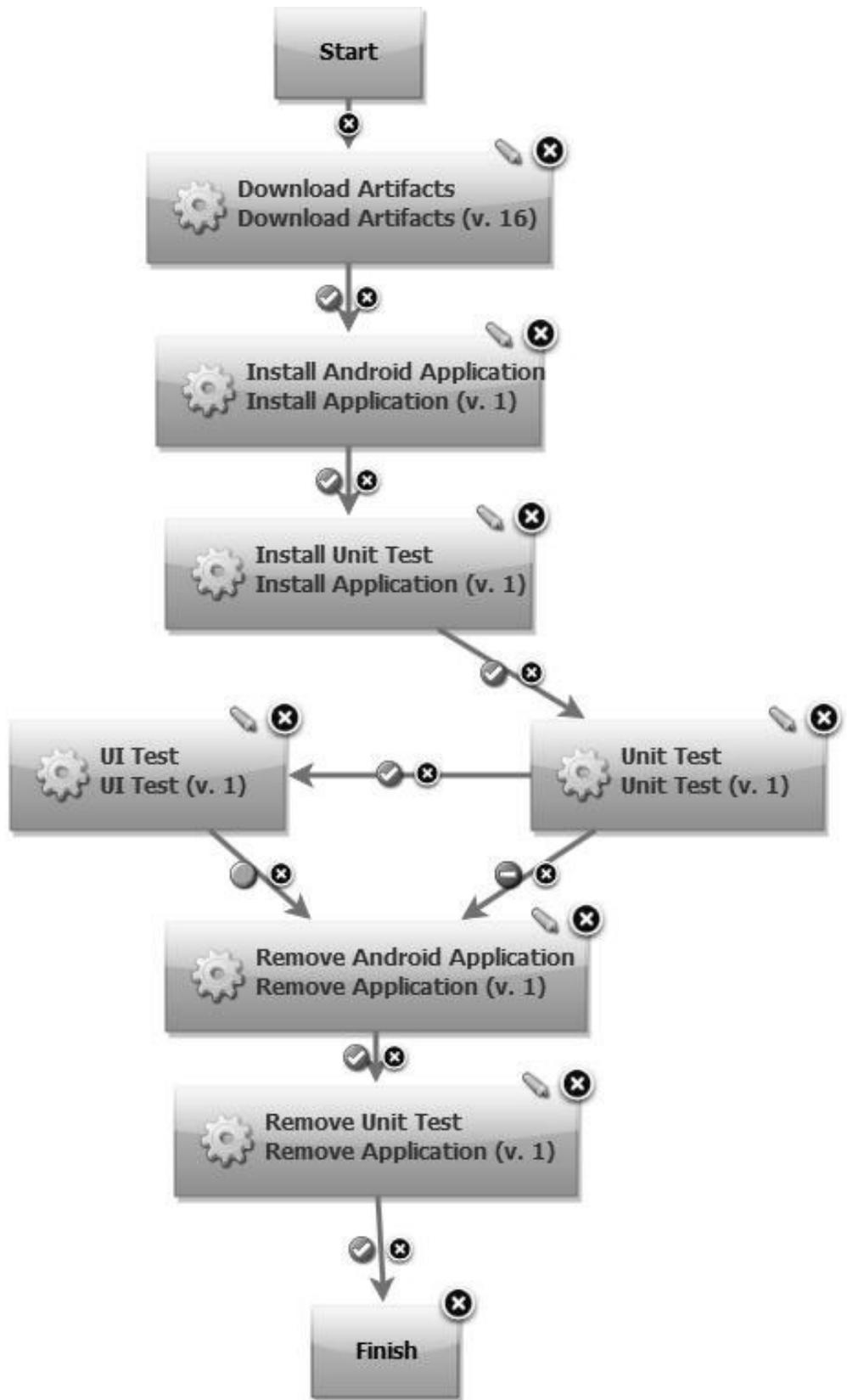
Example: Testing an Android application on a physical device

The following example component process tests an Android app on a physical device that is attached to the agent computer.

This process assumes that the application binary files are available in CodeStation in the component version. It also assumes that the agent and environment have the required software and configuration for this plug-in.

This process is similar to the example for testing an Android application on an emulator. In this example you do not create, start, stop, or remove emulators.

1. The **Download Artifacts** step (from IBM UrbanCode Deploy) retrieves the binary files.
2. The “**Install Application**” on page 12 plug-in step that is named **Install Android Application** installs the specified Android application (.apk file) onto the physical device.
3. The **Install Application** plug-in step that is named **Install Unit Test** installs the specified unit test onto the physical device.
4. The “**Unit Test**” on page 20 step tests a single Activity in the application. In the step, you provide the name of the test package to run and the class name for the instrumented runner to use when it runs the test. If the **Unit Test** step completes successfully, the “**UI Test**” on page 21 step runs. If the **Unit Test** step fails, the **UI Test** step is skipped.
5. The **UI Test** step simulates a user interacting with the application, such as moving between screens and verifying the displayed information. In this step, you specify a list of JAR files that contain the test cases to be pushed to the target and the class or package names of the tests to run. If you do not enter the path to the JAR files, then the files are not copied to the target.
6. The “**Remove Application**” on page 14 plug-in step that is named **Remove Android Application** removes the specified package from the target. You can select whether to keep the application data.
7. The **Remove Application** plug-in step that is named **Remove Unit Test** removes the unit test.



Example: Concurrent testing of an Android app

The following example component process uses two existing emulators, one a Nexus One and the other a Nexus S, to verify an application on different configurations by using random user interface (or monkey) tests.

This process assumes that the application binary files are available in CodeStation in the component version. It also assumes that the agent and environment have the required software and configuration for this plug-in.

1. The “**Start Emulator**” on page 17 steps start the emulators by using the emulator name. This step outputs the serial number of the emulator that started so that subsequent process steps can refer to the correct target in the case where multiple emulators are running.

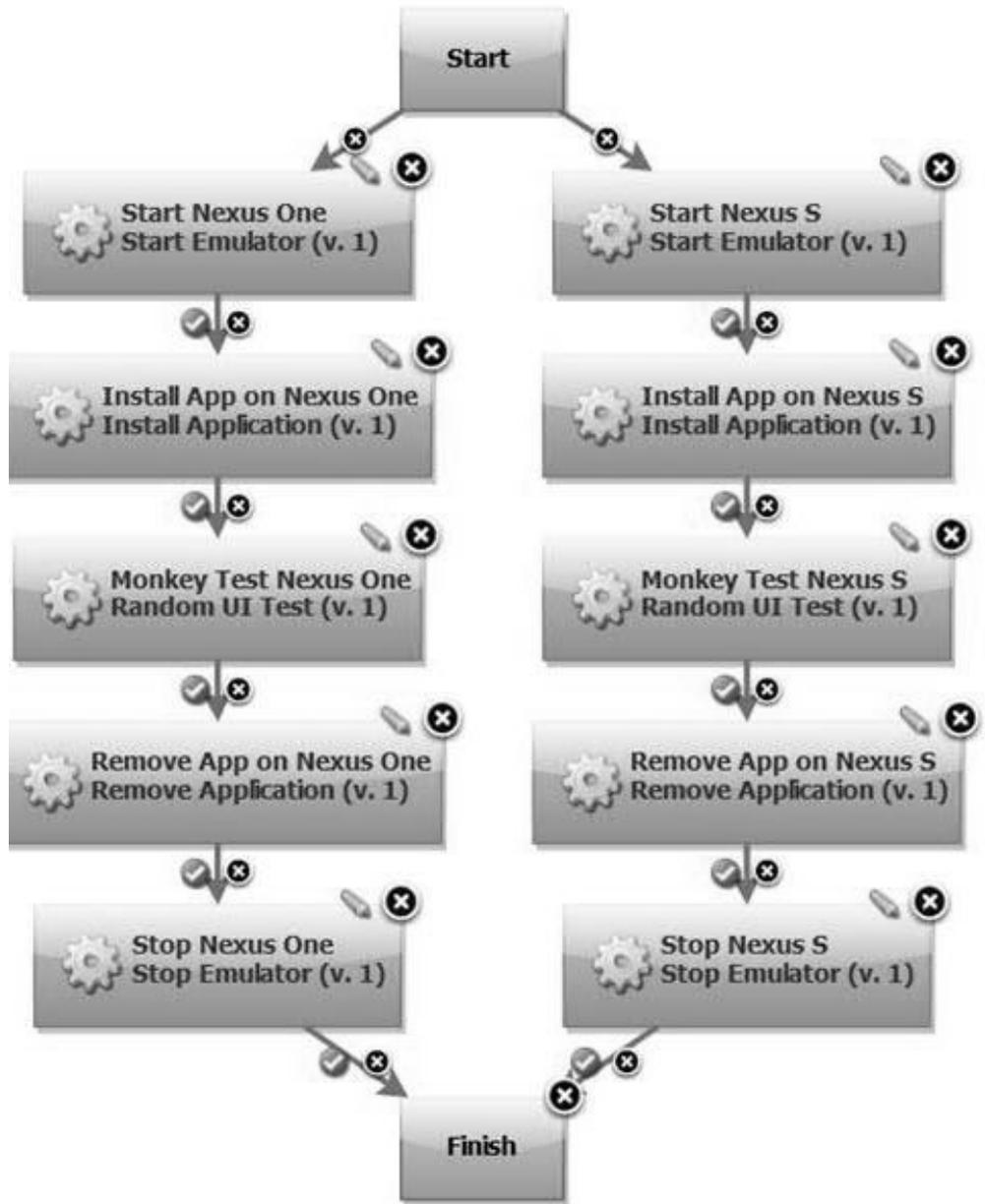
Note: Each step in the process must be assigned a unique name. For example, the two **Start Emulator** steps are named **Start Nexus One** and **Start Nexus S**.

2. The “**Install Application**” on page 12 step installs the specified Android application (.apk file) onto the target emulator.

Note: Because multiple emulators are running in this example, you must use the **Serial Number** property to identify the correct target. You do not need to know the serial number; instead, you must modify the default value for the **Serial Number** property (`${p?:Start Emulator/serialNumber}`) so that the Start Emulator step name matches the name that was given to the step that started the emulator.

In this example, the value for Serial Number for the step **Install App on Nexus One** is `${p?:Start Nexus One/serialNumber}` and for **Install App on Nexus S**, the Serial Number is `${p?:Start Nexus S/serialNumber}`.

3. The “**Random UI Test**” on page 23 runs the monkey tool in the Android Debug Bridge (ADB). The monkey tool generates a specified number of user interface (UI) events against the application.
4. The “**Remove Application**” on page 14 step removes the specified package from the target. You can select whether to keep the application data.
5. The “**Stop Emulator**” on page 19 step stops the emulator that is used during deployment. Stop any emulator that you start. Stopping the emulator frees up resources on the agent computer.



Process steps in the Android SDK plug-in

The Android SDK plug-in includes steps that support application installation and removal, emulator management, application testing, and interfacing with tools.

Install Application

Install an Android application on the target emulator or physical device. The application is verified to ensure that it is a valid .apk file. When this process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Table 2. Properties for the Install Application step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Application Path	String	The path to the Android application package (.apk) to install on the target. The default value is the application property that is configured in the component properties. For example: \${p:component/application}	Yes
Install Into	Select Box	<p>The target to apply the process step to. You can select one of the following targets:</p> <p>Default The default target. Select this option when only one emulator or physical device is running.</p> <p>Emulator Select this option if you want to target an emulator and only one emulator is running. There can be one or more physical devices that are running.</p> <p>Device Select this option when only one physical device is attached. Select this option if you want to target a physical device and only one physical device is running. There can be one or more emulators that are running.</p> <p>Serial Number Select this option if there are multiple emulators or physical devices that are running. Tip: You must use the Serial Number property to specify the serial number.</p>	No

Table 2. Properties for the Install Application step (continued)

Name	Type	Description	Required
Serial Number	String	<p>The serial number of the running target. For emulators, the default value is obtained from the Start Emulator process step. If the target is not available, the process step fails with an error.</p> <p>Note: You must modify the Serial Number property to identify the correct target.</p> <p>For example, if the name of the Start Emulator step that started the target emulator is Start Nexus One, then the value for Serial Number for that emulator is <code>\${p?:Start Nexus One/serialNumber}</code></p> <p>Tip: To use this property, you must select the Serial Number option as the target for the Install Into property.</p>	No
Reinstall	Check Box	Specifies whether to reinstall the application on the target and preserve the data.	No
Additional arguments	String	<p>Used to customize the Install Application process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices.</p> <p>Use either of the following formats:</p> <ul style="list-style-type: none"> Specify the path to a file. For example, <code>C:\test.properties</code> Enter a space separated list of arguments. For example, <code>-l -s</code> 	No

Remove Application

Remove the Android application from the target emulator or physical device by using the package name. When this process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Table 3. Properties for the Remove Application step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: <code>C:\android-sdk</code>	Yes

Table 3. Properties for the Remove Application step (continued)

Name	Type	Description	Required
Package Name	String	The package name of the application to remove from the target. The default value is the application property that is configured in the component properties. For example: \${p:component/package}	Yes
Uninstall From	Select Box	<p>The target to apply the process step to. You can select one of the following targets:</p> <p>Default The default target. Select this option when only one emulator or physical device is running.</p> <p>Emulator Select this option if you want to target an emulator and only one emulator is running. There can be one or more physical devices that are running.</p> <p>Device Select this option when only one physical device is attached. Select this option if you want to target a physical device and only one physical device is running. There can be one or more emulators that are running.</p> <p>Serial Number Select this option if there are multiple emulators or physical devices that are running. Tip: You must use the Serial Number property to specify the serial number.</p>	No

Table 3. Properties for the Remove Application step (continued)

Name	Type	Description	Required
Serial Number	String	<p>The serial number of the running target. For emulators, the default value is obtained from the Start Emulator process step. If the target is not available, the process step fails with an error.</p> <p>Note: You must modify the Serial Number property to identify the correct target.</p> <p>For example, if the name of the Start Emulator step that started the target emulator is Start Nexus One, then the value for Serial Number for that emulator is <code>\${p?:Start Nexus One/serialNumber}</code></p> <p>Tip: To use this property, you must select the Serial Number option as the target for the Install Into property.</p>	No
Keep Data	Check Box	Select this option to save application data during the removal of the application.	No

Create Emulator

Create an emulator target. You can specify the target Android version and specifications for the new emulator. You can optimize emulator performance by enabling snapshot creation.

Table 4. Properties for the Create Emulator step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: <code>C:\android-sdk</code>	Yes
Emulator Name	String	The name of the emulator that you want to create.	Yes
Target ID	String	The ID of the target to use when the emulator is created. The Target ID specifies properties such as the Android version and processor type (ABI). To determine the available target identifiers on the agent machine, run the Android process step with the <code>list targets</code> argument.	Yes
CPU	String	The CPU or ABI of the emulator to create. If multiple CPUs or ABIs are specified on the target, you must specify the CPU property. The Android <code>list targets</code> command displays the available CPU values.	No
Force Creation	Check Box	Select this option to force the creation of the emulator.	No

Table 4. Properties for the Create Emulator step (continued)

Name	Type	Description	Required
Create Snapshot	Check Box	Select this option to enable snapshot creation on the emulator to optimize performance.	No
Additional Arguments	String	Use to customize the Create Emulator process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices. Use either of the following formats: <ul style="list-style-type: none"> Specify the path to a file. For example, C:\test.properties Enter a space separated list of arguments. For example, -l -s 	No

Remove Emulator

Removes the emulator target to clean up the system at the end of your process run. The Remove Emulator process step does not stop the emulator.

Important: Run the Stop Emulator process step before you remove the emulator.

Table 5. Properties for the Remove Emulator step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Emulator Name	String	The name of the emulator that you want to remove.	Yes

Start Emulator

Start an emulator to target it for use during a process.

Tip: Ensure that you stop the emulator in the same process in which it was started to support the removal of that emulator.

You can limit the number of concurrent emulators that can run on an agent machine. However, this limit might cause multiple requests to occur at the same time if many processes attempt to start the emulator. Use the acquire lock step from IBM UrbanCode Deploy to gate the Start Emulator process step to prevent concurrency issues.

After the target emulator starts, the serial number is output for use in subsequent process steps.

Table 6. Properties for the Start Emulator step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Emulator Name	String	The name of the emulator that you want to start.	Yes
Load from Snapshot	Check Box	Select this option to start the emulator from a snapshot. Although this option improves emulator start up, saving the emulator in memory to disk might slow down the emulator shutdown.	No
Additional Arguments	String	Use to customize the Start Emulator process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices. Use either of the following formats: <ul style="list-style-type: none"> Specify the path to a file. For example, C:\test.properties Enter a space separated list of arguments. For example, -l -s 	No
Snapshot Name	String	The name of the snapshot to use when the emulator starts. By default, the default-boot value is used.	No
Output File	String	The location of file to which the output from the emulator startup is logged. Note: The output from the emulator startup does not appear in the console due to the long running process of the emulator.	No
Maximum Concurrent Emulators	String	The maximum number of emulators that can run simultaneously on the agent machine. Emulator startup waits until the timeout expires and then the step fails.	Yes
Concurrent Emulator Maximum Retries	String	The maximum number of times to try starting the emulator when the limit of concurrent emulators is reached. The process step polls every 20 seconds. The default value is 4.	Yes
Startup Polling Maximum Retries	String	The maximum time (in seconds) to wait between checking for emulator startup. The process step attempts to verify startup every 20 seconds and then indicates that the startup failed. The default value is 15.	Yes

Stop Emulator

Stop the target emulator that is used during deployment. The Stop Emulator process step cleans up the environment by freeing up resources on the agent machine. By freeing up resources on the agent machine, other tests can run on emulators in the same or different deployment processes. When this process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Table 7. Properties for the Stop Emulator step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Target	Select Box	<p>The target to apply the process step to. You can select one of the following targets:</p> <p>Default The default target. Select this option when only one emulator or physical device is running.</p> <p>Emulator The single emulator that is running.</p> <p>Device Select this option when only one physical device is attached.</p> <p>Serial Number Select this option when you want to use the serial number of the attached target. Tip: You must use the Serial Number property to specify the serial number.</p>	No
Serial Number	String	<p>The serial number of the running target. The default value is obtained from the Start Emulator step. If the target is not available, the process step fails with an error.</p> <p>Tip: To use this property, you must specify the Serial Number option for the Target property.</p>	No
Save to Snapshot	Check Box	Select this option to save the emulator to a snapshot when the emulator shuts down. Although this option improves emulator start up, saving the emulator in memory to disk might slow down the emulator shutdown.	No
Snapshot Name	String	The name of the snapshot to save to on shutdown. By default, the default-boot value is used.	No

Unit Test

Unit test a single activity in your application. You can install the unit test as another application on the target and run all tests or a subset of tests. When the Unit Test process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Table 8. Properties for the Unit Test step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Package Name	String	The name of the test package to run.	Yes
Runner Class	String	The class name of the instrumented runner to use when the test is run.	Yes
Test Against	Select Box	<p>The target to apply the process step to. You can select one of the following targets:</p> <p>Default The default target. Select this option when only one emulator or physical device is running.</p> <p>Emulator Select this option if you want to target an emulator and only one emulator is running. There can be one or more physical devices that are running.</p> <p>Device Select this option when only one physical device is attached. Select this option if you want to target a physical device and only one physical device is running. There can be one or more emulators that are running.</p> <p>Serial Number Select this option if there are multiple emulators or physical devices that are running. Tip: You must use the Serial Number property to specify the serial number.</p>	No

Table 8. Properties for the Unit Test step (continued)

Name	Type	Description	Required
Serial Number	String	<p>The serial number of the running target. For emulators, the default value is obtained from the Start Emulator process step. If the target is not available, the process step fails with an error.</p> <p>Note: You must modify the Serial Number property to identify the correct target.</p> <p>For example, if the name of the Start Emulator step that started the target emulator is Start Nexus One, then the value for Serial Number for that emulator is <code>\${p?:Start Nexus One/serialNumber}</code></p> <p>Tip: To use this property, you must select the Serial Number option as the target for the Install Into property.</p>	No
Raw Format	Check Box	Select this option to output results in raw format.	No
Additional Arguments	String	<p>Use to customize the Unit Test process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices.</p> <p>Use either of the following formats:</p> <ul style="list-style-type: none"> • The path to a file. For example, <code>C:\test.properties</code> • A space separate list of arguments. For example, <code>-l -s</code> 	No
Timeout	String	The timeout (in milliseconds) after which the Android process command is stopped.	No

UI Test

UI testing simulates a user interacting with an application. For example, this step might include moving between windows and verifying the information that is displayed. This step is the Android SDK-based uiautomator testing framework. When this process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Table 9. Properties for the UI Test step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
JAR Files	String	A space separated list of JAR files that contain the test cases to use when this process step is run.	Yes
Push to Target Path	String	The path to push all of the listed JAR files to on the target. If no path is specified, the JAR files are not updated on the target.	No
Test Classes	String	The class or package names of the tests to run. Use either of the following formats: <ul style="list-style-type: none"> • The path to a property file that contains the list of classes to run. For example, C:\test.properties • A space separated list of class or package names. 	Yes
Test Against	Select Box	The target to apply the process step to. You can select one of the following targets: <p>Default The default target. Select this option when only one emulator or physical device is running.</p> <p>Emulator Select this option if you want to target an emulator and only one emulator is running. There can be one or more physical devices that are running.</p> <p>Device Select this option when only one physical device is attached. Select this option if you want to target a physical device and only one physical device is running. There can be one or more emulators that are running.</p> <p>Serial Number Select this option if there are multiple emulators or physical devices that are running. Tip: You must use the Serial Number property to specify the serial number.</p>	No

Table 9. Properties for the UI Test step (continued)

Name	Type	Description	Required
Serial Number	String	<p>The serial number of the running target. For emulators, the default value is obtained from the Start Emulator process step. If the target is not available, the process step fails with an error.</p> <p>Note: You must modify the Serial Number property to identify the correct target.</p> <p>For example, if the name of the Start Emulator step that started the target emulator is Start Nexus One, then the value for Serial Number for that emulator is <code>#{p?:Start Nexus One/serialNumber}</code></p> <p>Tip: To use this property, you must select the Serial Number option as the target for the Install Into property.</p>	No
Additional Arguments	String	<p>Use to customize the UI Test process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices.</p> <p>Use either of the following formats:</p> <ul style="list-style-type: none"> • The path to a file. For example, <code>C:\test.properties</code> • A space separated list of arguments. For example, <code>-l -s</code> 	No
Clean Up JAR Files	Check Box	Specifies whether to clean up and remove the JAR files after the UI Test process step completes. By default this property is selected.	No
Timeout	String	The timeout (in milliseconds) after which the UI Test process command is stopped.	No

Random UI Test

Use Random UI Test to test applications without writing test cases. This monkey tool is included in the Android SDK to perform random UI testing. When the Random UI Test process step runs, the Android Debug Bridge (ADB) version is output for debugging.

Table 10. Properties for the Random UI Test step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Test Against	Select Box	<p>The target to apply the process step to. You can select one of the following targets:</p> <p>Default The default target. Select this option when only one emulator or physical device is running.</p> <p>Emulator Select this option if you want to target an emulator and only one emulator is running. There can be one or more physical devices that are running.</p> <p>Device Select this option when only one physical device is attached. Select this option if you want to target a physical device and only one physical device is running. There can be one or more emulators that are running.</p> <p>Serial Number Select this option if there are multiple emulators or physical devices that are running. Tip: You must use the Serial Number property to specify the serial number.</p>	No
Serial Number	String	<p>The serial number of the running target. For emulators, the default value is obtained from the Start Emulator process step. If the target is not available, the process step fails with an error.</p> <p>Note: You must modify the Serial Number property to identify the correct target.</p> <p>For example, if the name of the Start Emulator step that started the target emulator is Start Nexus One, then the value for Serial Number for that emulator is <code>\${p?:Start Nexus One/serialNumber}</code></p> <p>Tip: To use this property, you must select the Serial Number option as the target for the Install Into property.</p>	No

Table 10. Properties for the Random UI Test step (continued)

Name	Type	Description	Required
Event Count	String	The number of events to send to the target during testing.	Yes
Additional Arguments	String	Use to customize the Random UI Test process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices. Use either of the following formats: <ul style="list-style-type: none"> • The path to a file. For example, C:\test.properties • A space separated list of arguments. For example, -l -s 	No
Timeout	String	The timeout (in milliseconds) after which the Android process command is stopped.	No

ADB

Android Debug Bridge (ADB) is the main command that is used to interact with the target emulator or physical device. This command starts a server on the agent machine to interact with emulators and devices. You can use the ADB command to add the UI test JAR files to the target, list connected devices, and list the installed packages.

Table 11. Properties for the ADB step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Additional Arguments	String	Use to customize the ADB process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. Use either of the following formats: <ul style="list-style-type: none"> • The path to a file. For example, C:\test.properties • A space separated list of arguments. For example, -l -s 	No

Table 11. Properties for the ADB step (continued)

Name	Type	Description	Required
Timeout	String	The timeout (in milliseconds) after which the Android process command is stopped.	No

Android

Android is the command that is used to interact with Android Virtual Devices (AVDs).

For example, you can use this command to perform the following actions:

- Discover information about potential target configurations. For example, to create an emulator with a specific Android version.
- List the available emulators for startup.
- Update the target version with new targets.

Table 12. Properties for the Android step

Name	Type	Description	Required
SDK Installation Path	String	The path to the Android SDK that is used to run the process step. The default value is the path from the resource. For example: C:\android-sdk	Yes
Additional Arguments	String	Use to customize the Android process step by providing access to options that are not included in the properties themselves. You can provide these options directly in the process step or within a property file that is referenced by the step. For example, the Additional Arguments property can provide access to the option to use arguments in the Android Debug Bridge (ADB) step to list the attached devices. Use either of the following formats: <ul style="list-style-type: none"> • The path to a file. For example, C:\test.properties • A space separated list of arguments. For example, -l -s 	Yes
Timeout	String	The timeout (in milliseconds) after which the Android process command is stopped.	No

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational® Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2014.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, or to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.