



C5050-062^{Q&As}

Rational UrbanCode Deploy

Pass IBM C5050-062 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/c5050-062.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by IBM Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



**QUESTION 1**

When integrating with WebLogic (WLDeploy plugin), one required field is the location of the "WLDeploy Ant Task Jar." Where can you obtain this important file that is required for integration?

- A. You can download it from <http://oracle.com>.
- B. You can find it in the WebLogic\\'s server directory.
- C. You can build the jar file manually in the WebLogic\\'s server/lib directory.
- D. You can download it from the WebLogic interface.

Correct Answer: C

Basic Steps for Using wldesploy To use the wldesploy Ant task:

1.

Set your environment.

On Windows NT, execute the setWLSEnv.cmd command, located in the directory WL_HOME\server\bin, where WL_HOME is the top-level directory of your WebLogic Server installation. On UNIX, execute the setWLSEnv.sh command, located in the directory WL_HOME/server/bin, where WL_HOME is the top-level directory of your WebLogic Server installation.

2.

In the staging directory, create the Ant build file (build.xml by default). If you want to use an Ant installation that is different from the one installed with WebLogic Server, start by defining the wldesploy Ant task definition:

3.

If necessary, add task definitions and calls to the wlserver and wlconfig tasks in the build script to create and start a new WebLogic Server domain.

4.

Add a call to wldesploy to deploy your application to one or more WebLogic Server instances or clusters.

See Sample build.xml Files for wldesploy and wldesploy Ant Task Attribute Reference.

5.

Execute the Ant task or tasks specified in the build.xml file by typing ant in the staging directory,



optionally passing the command a target argument:

prompt> ant

References: https://docs.oracle.com/cd/E13222_01/wls/docs92/programming/wldeploy.html

QUESTION 2

What do Source Configuration plugins enable the user to do?

- A. Explicitly define an IBM UrbanCode Deploy agent that will download a component's files
- B. Explicitly define a sub-process that will perform selective configuration of a component's files
- C. Explicitly define a procedure for interactively selecting which component files to download when the deployment process runs
- D. Explicitly define a sub-process for downloading a component's files from multiple repositories

Correct Answer: A

References: <https://developer.ibm.com/urbandcode/videos/source-configuration-plugins-ibm-urbancodedeploy/>

QUESTION 3

You have multiple application environments sharing the same resource. What will you see in the inventory of this resource?

- A. Only the versions of components deployed from one application
- B. All versions of all components deployed from the sharing applications



- C. Last versions of all components deployed from the sharing applications
- D. Only the last version of the component deployed from the application that was used last

Correct Answer: B

The intention is that you could look at the resource tree on the environment detail page to see all versions currently deployed to resources in that environment.

References: <https://developer.ibm.com/answers/questions/8129/where-to-find-what-is-really-deployed-in-an-environment.html>

QUESTION 4

What is the correct order of steps for a typical deployment configuration?

- A. Define Application Process; Create component Snapshot; Map Component to Environment; Identify Resource
- B. Define Component and Component Process; Define Application; Configure Environment; Identify Resource
- C. Define Component; Import Version to Component; Define Application Process; Configure Environment; Map Component to Resource
- D. Define Application; Map Component to Recourse; Identify Agent and Environment; Configure Component Process

Correct Answer: B

In outline, deployments are done by performing the following steps:

1.

Define Components

2.

Define Component Processes

3.

Define Application

4.

Configure Environment

5.

Identify Agent

References: ftp://public.dhe.ibm.com/software/rationalsdp/documentation/product_doc/UrbanCode/uDeploy/version_4-6/uDeploy_WebHelp_4-6/content/quickstart_chapter.html

QUESTION 5



A development team wants to capture all the versions of the components that were deployed in the QA environment and the want to deploy them on the Production environment. What is the best approach for doing this in IBM UrbanCode Deploy?

- A. They should create a tag to identify which revisions were deployed in the QA, and when executing the deployment to Production they should select all the versions that have the defines tag.
- B. They should create an application baseline that includes all the component versions deployed on the QA.
- C. They should compare the QA environment with the Production environment, and from the differences create a delta deployment that should be applied to Production.
- D. They should create a snapshot of the QA environment and deploy this snapshot to Production.

Correct Answer: A

Using Tagging to Implement Blue/Green Deployments Many clients using IBM UrbanCode Deploy are interested in implementing Blue/Green deployments. A Blue/Green deployment is a term that refers to a method of deploying the same Components to multiple servers or server clusters in a way that eliminates the need to bring them all offline at the same time. Whether the goal is to maintain availability to a service or application during an upgrade or your internal system requires constant server uptime, simply taking all resources offline simultaneously just isn't an option.

References: <https://developer.ibm.com/urbandcode/docs/blue-green-deployments-using-tags-ibmurbancode-deploy/>

[C5050-062 PDF Dumps](#)

[C5050-062 Practice Test](#)

[C5050-062 Braindumps](#)