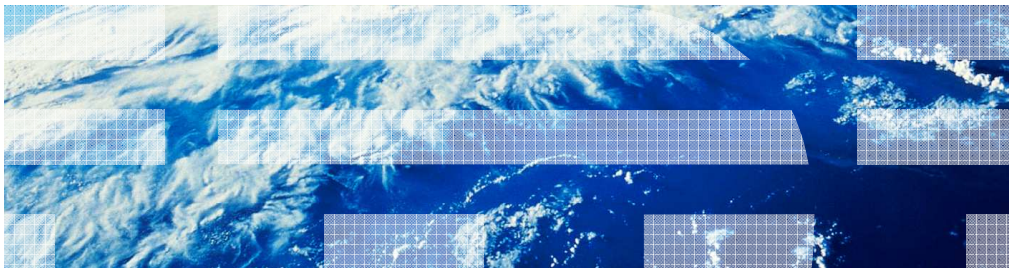

IBM WebSphere CloudBurst Appliance V2.0

Script packages



This presentation will cover script packages.

Agenda

- Script package creation
- Predefined environment variables
- Script package format
- Script package deployment

You will cover all life cycle events of a script package from creation to deployment. In addition, you are shown the format of a typical script package and a best practice which involves including the script package definition directly inside of the archive.

Script package creation

This section will cover the creation of a script package.

Creating a script package

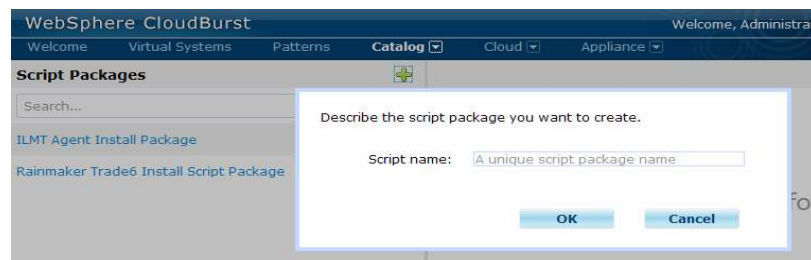
- Two ways to get to the script package creation page

The screenshot shows the WebSphere CloudBurst console interface. At the top, there is a navigation bar with tabs for 'Welcome', 'Virtual Systems', 'Patterns', 'Catalog', 'Cloud', and 'Appliance'. The 'Catalog' tab is selected, and a dropdown menu is open, showing options: 'Virtual Images', 'Product Licenses', 'Script Packages', and 'Emergency Fixes'. The 'Script Packages' option is highlighted with a mouse cursor. Below the navigation bar, the main content area is titled 'Welcome to WebSphere Cloud' and contains several sections: 'Deployment made easier' with four steps (Set up the appliance, Set up the cloud, Create a virtual system, View virtual systems), and 'Additional tasks' with four items (Add virtual images, Add script packages, Create reusable patterns, Use command line tools). The 'Add script packages' link in the 'Additional tasks' section is highlighted with a red box. At the bottom left, the page number '4' is visible, and at the bottom right, the copyright notice '© 2010 IBM Corporation' is present.

There are two ways in which to start the process of creating a script package. You can navigate to the welcome page and click the “Add script package” link or you can navigate to Catalog > Script Packages at the top of the page.

Adding a script package

- Navigate to **Catalog > Script Packages > +**
- Only the name of the script is required to start the process of creating a script package



Script packages contain settings and other artifacts that you want run against a target virtual machine. A script package is started after the environment has been deployed and initialized. This means that the operating system and WebSphere® Application Server environment have been completely started and configured before script package execution.

Script package attributes

- Attribute definitions
 1. Upload your script package
 2. Create environment variables
 3. Specify a temporary directory
 4. Specify where the logs will end up

Script package

Script package files:
Your file was uploaded successfully.

Environment: DB2_HOSTNAME = (no default value) [remove]
DB2_PORT = (no default value) [remove]
Add variable =

Working directory:

Logging directory:

Executable:

Arguments:

Executes:

Included in patterns:

Access granted to:

Script package creation at a minimum requires that you specify a unique name. Beyond specifying a name all other parameters are optional and are dependent on your scenario. The next few slides will cover the various parameters and their meanings.

One, upload an archive that contains any needed artifacts. These artifacts can be, but not limited to, scripts and executables. The uploaded archive has two limitations. First, it must be either of format .zip or .tar. Second, the archive can not exceed 512MB.

Two, define any variables that you want made available to your scripts while they are started on the target virtual machine. You can define values now or leave them blank and require that they be filled in when the pattern is deployed. A simple example of this can be seen in the screen capture. The variable DB2_HOSTNAME is defined and no value is provided. During deployment the value needs to be specified. As you can see, this one script package can then be modified for each deployment by passing in a different value.

Three, specify a working directory. This is a temporary directory on the target virtual machine in which WebSphere CloudBurst™ will expand the archive.

Four, tell WebSphere CloudBurst where it can find the logs after the script package completes its execution.

Script package attributes (continued)

- Attribute definitions
 1. Specify an executable
 2. Supply arguments to the executable
 3. Grant and remove permissions

7

Script packages

© 2010 IBM Corporation

One, specify the command that you want run on the target virtual machine. In this case the “wsadmin” command is going to be run. It should be noted that this does not have to be the “wsadmin” command. You can start an existing command on the virtual image or a command that you have uploaded in your script package’s archive. You might have noticed the use of variables, first in the “Logging directory” field and now in the “Executable” field. These are WebSphere CloudBurst predefined variables that are covered in a follow-up slide.

Two, specify the arguments to the executable. You can specify the arguments along with the executable file in the “Executable” field, but for clarity they are separated into separate fields.

Three, the person that creates the script package owns the script package. This means that no one, but the owner can see, modify or deploy the script package. If other users or user groups require access to your script package you will need to explicitly grant them permission by clicking the “Add more...” text field. The users that you choose here have no impact on the user WebSphere CloudBurst will execute the script package as. WebSphere CloudBurst will execute the script package on the target virtual machine as the “root” user account. If you want to change this you will need to explicitly call the “su” command in your script to change the user prior to execution.

Script package attributes (continued)

- Attribute definitions
 1. Specify point in time you want script to execute

Script package

Script package files:

Your file was uploaded successfully.

Environment: DB2_HOSTNAME = (no default value) [remove]
DB2_PORT = (no default value) [remove]
Add variable =

Working directory:

Logging directory:

Executable:

Arguments:

Executes: at virtual system creation
at virtual system deletion
when I initiate it

Included in patterns:

Access granted to:

Script packages can be instructed to start at virtual system creation, deletion or when you explicitly initiate it. By default, “at virtual system creation” is chosen. If you choose “when I initiate it” you are required to explicitly initiate the execution of the script package. You can do this by navigating to the virtual systems page. This is covered in a follow-up slide.

One thing to note about “when I initiate it” is that the script package is not uploaded to the target virtual machine until you initiate it. Any changes that you have made up to that point are uploaded to the target virtual machine. There is no limit on the number of times that you can initiate your script package for execution on the target virtual machine. Each time you initiate the execution, the latest edition of the script package is uploaded and executed.

Predefined environment variables

This section will cover some of the predefined variables that WebSphere CloudBurst makes available to you.

Predefined environment variables

- WebSphere CloudBurst contains predefined environment variables that you can use in your script packages

Subset of predefined environment variables	
CELL_NAME	WebSphere cell name
NODE_NAME	WebSphere node name
WAS_INSTALL_ROOT	WebSphere Application Server installation root (example: C:/WebSphere/AppServer)
WAS_PROFILE_ROOT	WebSphere Application Server profile root (example: C:/WebSphere/AppServer/profiles/DefaultProfile)
PROFILE_NAME	WebSphere Application Server profile short name (example: DefaultProfile)
HOSTNAME	Host name of virtual machine hosting the WebSphere Application Server instance
WAS_USERNAME	WebSphere Application Server user name
WAS_PASSWORD	WebSphere Application Server password

There are many predefined variables that can be used by your scripts. The values are specific to each deployment. Some values do not change such as WAS_INSTALL_ROOT. This slide shows you a subset of the available variables.

Script package format

This section will cover the script package format.

Script package format

- Archived into a single file
- Example contents of a script package

```

AdminTask.createApplicationServer('MyLabNode1', '{-name server1 -
templateName default -genUniquePorts true }')

AdminApp.install('dgreeting.ear', '{ -cell MyLabCell -node MyLabNode1 -
server: server1 -nopreCompileJSPs -distributeApp -
nouseMetaDataFromBinary -nodeploy:jb -appName greeting -
createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
warn -noprocessEmbeddedConfig -filepermission
.\.dll=755#.*.so=755#.*.a=755#.*.sl=755 -
noallowDispatchRemoteInclude -noallowServiceRemoteInclude -
asynchRequestDispatchType DISABLED -nouseAutoLink}')

AdminConfig.save()
    
```

createJDBCResources.jy

createCluster.jy

appInstall.jy

dgreeting.ear

12

Script packages

© 2010 IBM Corporation


Script packages can be used for any setup and not just application installation. You are also not locked in to calling `wsadmin.sh` as your executable file. You can make a call to any executable file residing on the system.

The example script package in this slide contains four scripts and an EAR file. There are scripts to create JDBC™ resources and clusters, and scripts to install the application. There is nothing special about these scripts. The scripts are standard `wsadmin` Jython scripts. The scripts are extracted into the virtual machine and are run under root authority like any other script. Your scripts do not need to be aware that WebSphere CloudBurst is involved.


Script package – single file upload

- Upload a single file when creating a script package
- Avoids having to create a .zip or .tgz to wrapper your file before uploading

Test Script - Single File

Current status:  Draft

Script package files:

The script package is in [configureProduct.jy](#)  [Download](#)

If your script package consists only of a single file you are now able to upload that single file. Before V2.0, you were required to create a .zip or .tgz to wrapper the single file before uploading.

cbscript.json

- Define and include script package definition within the archive
- File called “cbscript.json” with defined parameters should be placed in the root of the archive


```
[
{
  "name": "My Demo Script Package",
  "version": "1.0.0",
  "description": "Demo to show off script packages.",
  "command": "${WAS_PROFILE_ROOT}/bin/wsadmin.sh",
  "location": "/tmp/myLabApp",
  "log": "${WAS_PROFILE_ROOT}/logs",
  "timeout": "0",
  "commandargs": "-user ${WAS_USERNAME} -password ${WAS_PASSWORD} -lang jython -f /tmp/myLabApp/appInstall.jy",
  "keys":
  [
    {
      "scriptkey": "DB2_HOSTNAME",
      "scriptvalue": "",
      "scriptdefaultvalue": ""
    }
  ]
}
]
```

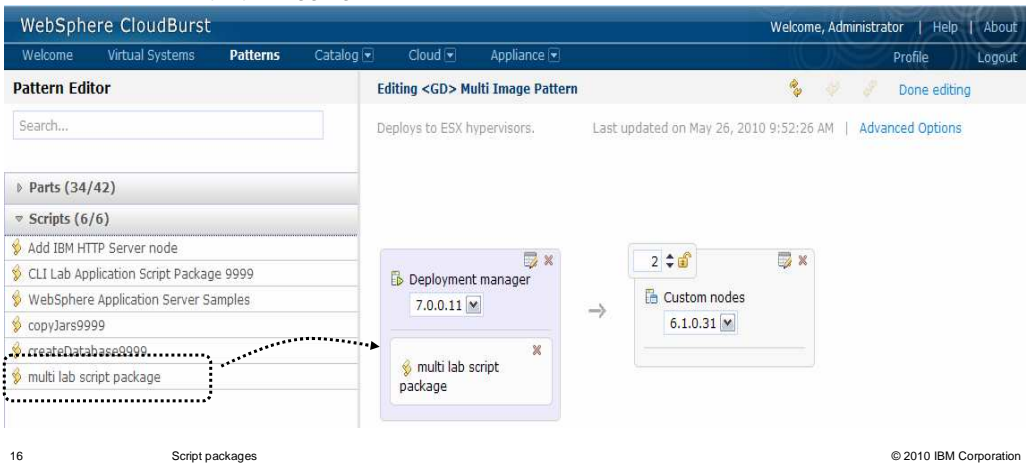
It is considered best practice to include a file called “cbscript.json” in the root of the script package archive. This file contains the script package’s definition in JSON format. By taking this approach you can move this script package from one appliance to another and not have to reenter the information using the user interface or command-line interface, thus reducing data entry errors.

Script package deployment

This section will cover how you go about deploying a script package.

Drag-and-drop pattern assembly

- Navigate to **Patterns > your_pattern >**  and expand “Scripts” from the pattern editor on the left panel
- Pattern assembly by dragging from the palette over to the canvas



WebSphere CloudBurst

Welcome, Administrator | Help | About

Welcome Virtual Systems **Patterns** Catalog Cloud Appliance Profile Logout

Pattern Editor Editing <GD> Multi Image Pattern Done editing

Search...

Parts (34/42)

Scripts (6/6)

- Add IBM HTTP Server node
- CLI Lab Application Script Package 9999
- WebSphere Application Server Samples
- copyJars9999
- createDatabase9999
- multi lab script package

Deployment manager 7.0.0.11

multi lab script package

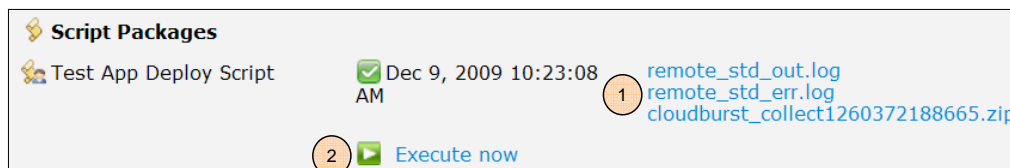
Custom nodes 6.1.0.31

16 Script packages © 2010 IBM Corporation

Adding a script package to a pattern is a drag and drop activity. You drag the script package from the palette in the left pane over the virtual image part you want script package started against. In the screen capture above a script package was added to the “Deployment manager” virtual image part. The script package “Script package” installs an application so it needs to be started against the deployment manager.

Script package execution and results

- Navigate to **Virtual Systems > your_vs > Virtual machines > your_target_vm**
 1. View execution results of script package
 2. Execute a user initiated script package (click “Execute now” highlighted in screen capture)



The “Script Packages” section located under Virtual Systems > your_virtual_system > Virtual machines > your_target_vm gives you access to the logs resulting from the execution of your script packages. In addition, if your script package was marked as a user initiated script package then there is a link labeled “Execute now” located next to the script. Click the link in order to start the script package on the virtual system.

Summary

- Covered the complete life cycle of a script package
- WebSphere CloudBurst makes available for your use a set of predefined variables
- Attaching a script package to a pattern is as simple as a drag and drop operation
- Script packages can be instructed to start at creation, deletion or when you initiate the execution

This presentation covered the complete life cycle of a script package. You were shown how to create one from scratch, attach to an existing pattern and deploy. In addition, you were shown that WebSphere CloudBurst makes available to you a set of predefined variables. Last, script packages can be instructed to start at virtual system creation, deletion or when you specifically initiate it.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback about CB20 ScriptPackages.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20CB20%20ScriptPackages.ppt)

This module is also available in PDF format at: [../CB20_ScriptPackages.pdf](http://.../CB20_ScriptPackages.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, CloudBurst, and WebSphere 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 other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.