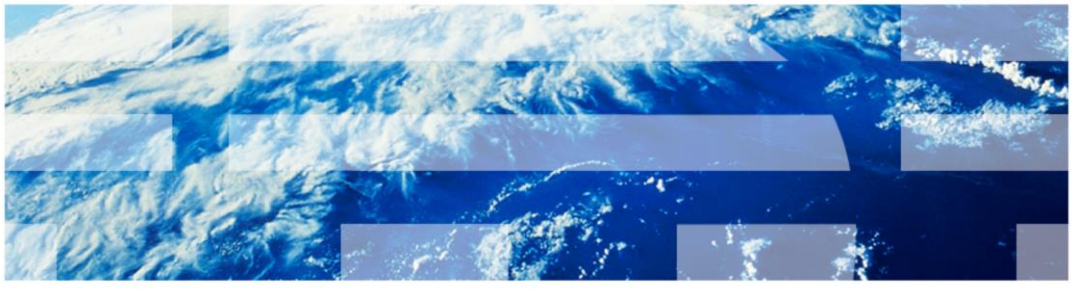
IBM

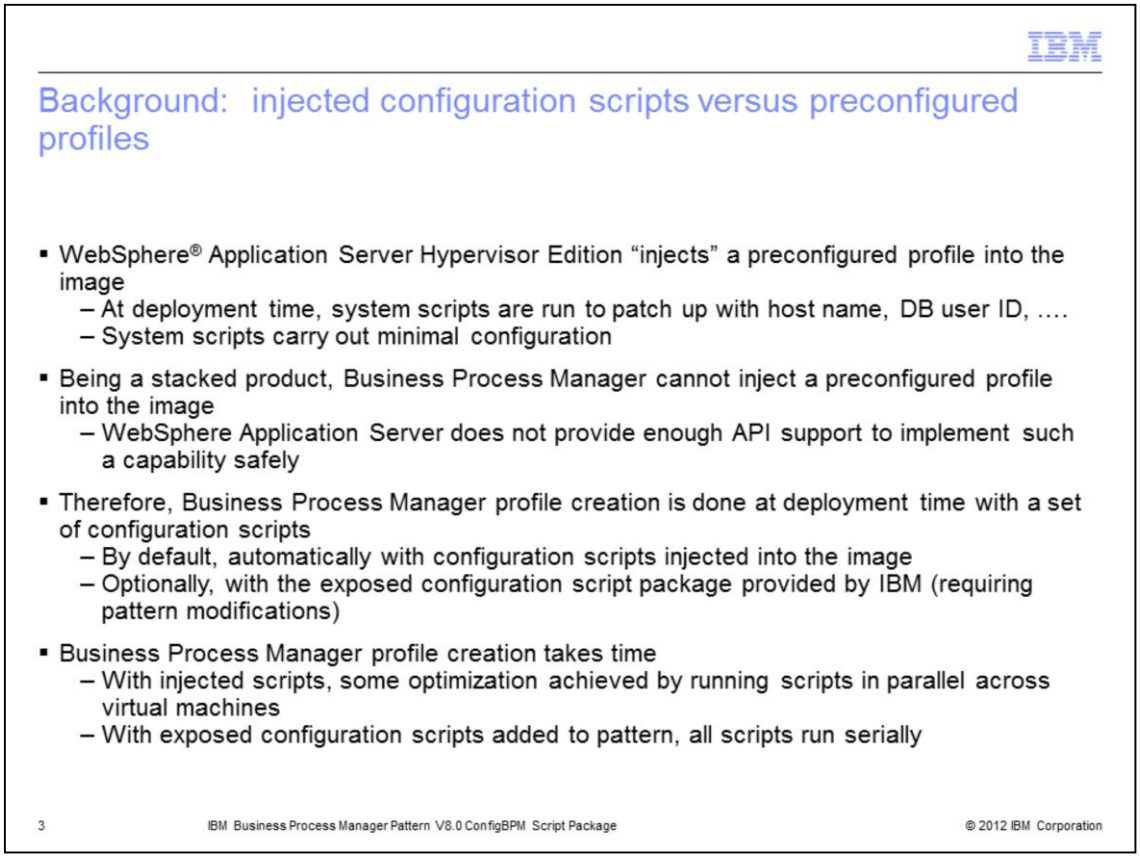
IBM PureApplication System and IBM Workload Deployer
IBM Business Process Manager Pattern V8.0

ConfigBPM Script Package



This presentation will cover the ConfigBPM script package in the Business Process Manager Pattern V8.0 in IBM PureApplication System and IBM Workload Deployer.

You will be introduced to the ConfigBPM script package.

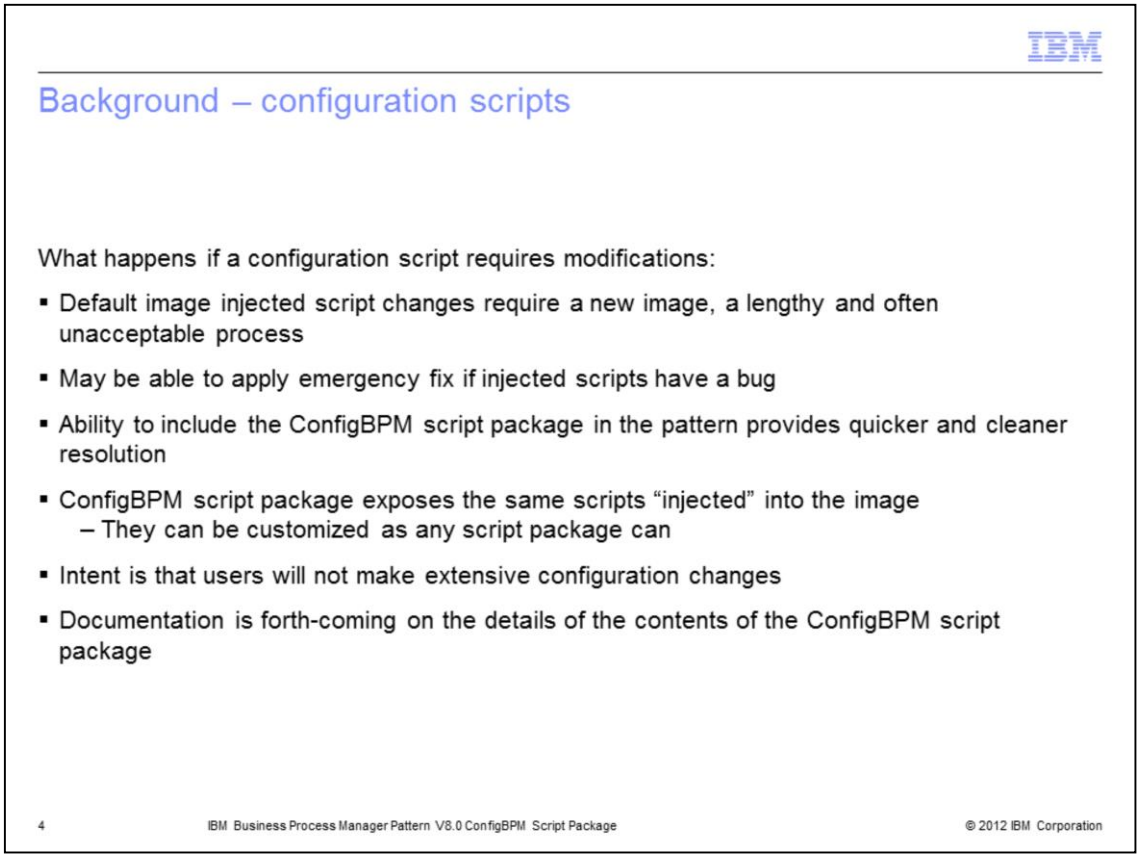


Here is background information on why IBM exposed these configuration scripts.

Business Process Manager is a "stacked" product, which means it was implemented on top of WebSphere Application Server. Virtual images for WebSphere Application Server Hypervisor Edition are "injected" with a preconfigured profile. Therefore very little configuration needs to be done at deployment time and the deployment will complete relatively quickly. However, at this time WebSphere Application Server does not have sufficient API support for stacked products to safely inject similar preconfigured profiles into their virtual images. So it is necessary to use a different technique to build the profiles for your Business Process Manager deployment.

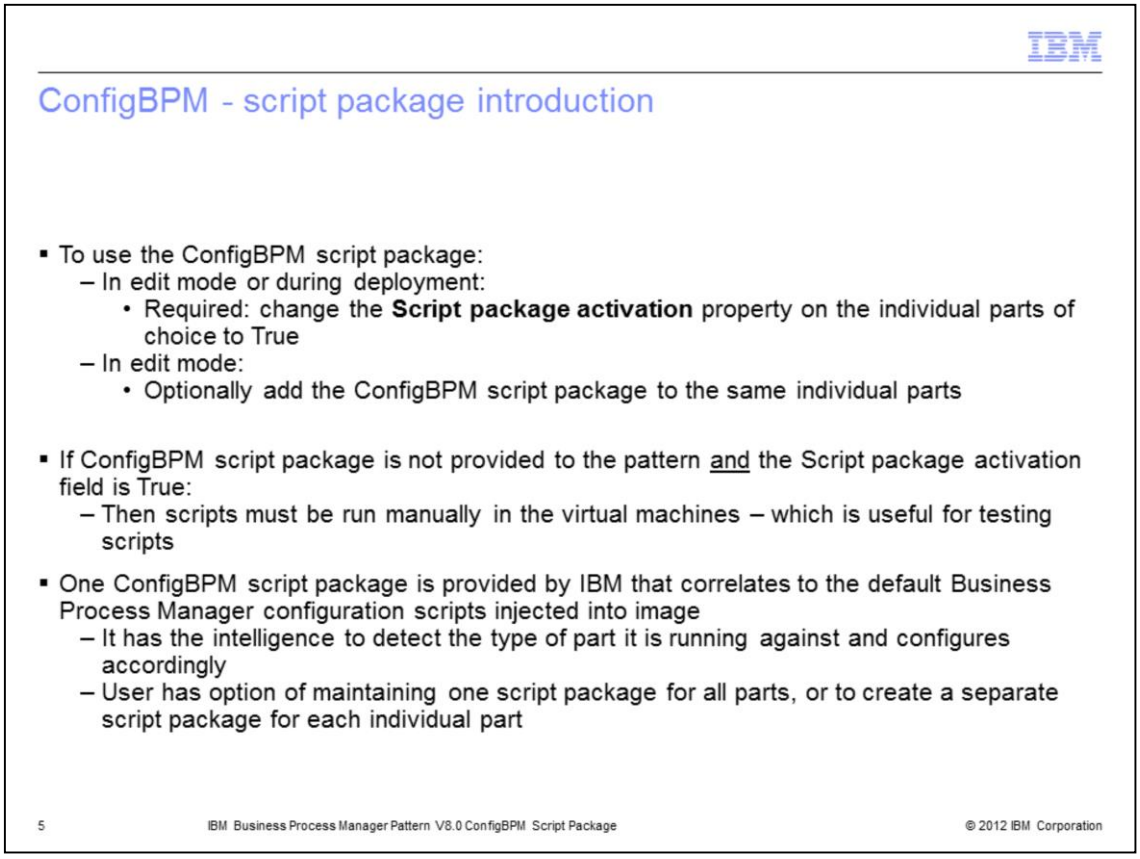
For Business Process Manager patterns, by default, profile creation is done at deployment time by a set of configuration scripts that have been injected into the image. These scripts are automatically ran on every virtual machine. But IBM has also exposed all of these configuration scripts in the ConfigBPM script package. Therefore you can modify the exposed scripts and run them during deployment. If you choose this option, then you can't use the default deployment pattern; running the ConfigBPM script package during deployment requires pattern changes.

Regardless of whether you use the injected scripts or the ConfigBPM script package, having to run these configuration scripts during Business Process Manager pattern deployment makes it take longer than a WebSphere Application Server pattern deployment. If you use the default pattern with the injected scripts, some optimization is achieved by running the scripts in parallel on multiple virtual machines. If you use the exposed scripts in the ConfigBPM script package, the deployment will take even longer because all the scripts have to run serially.



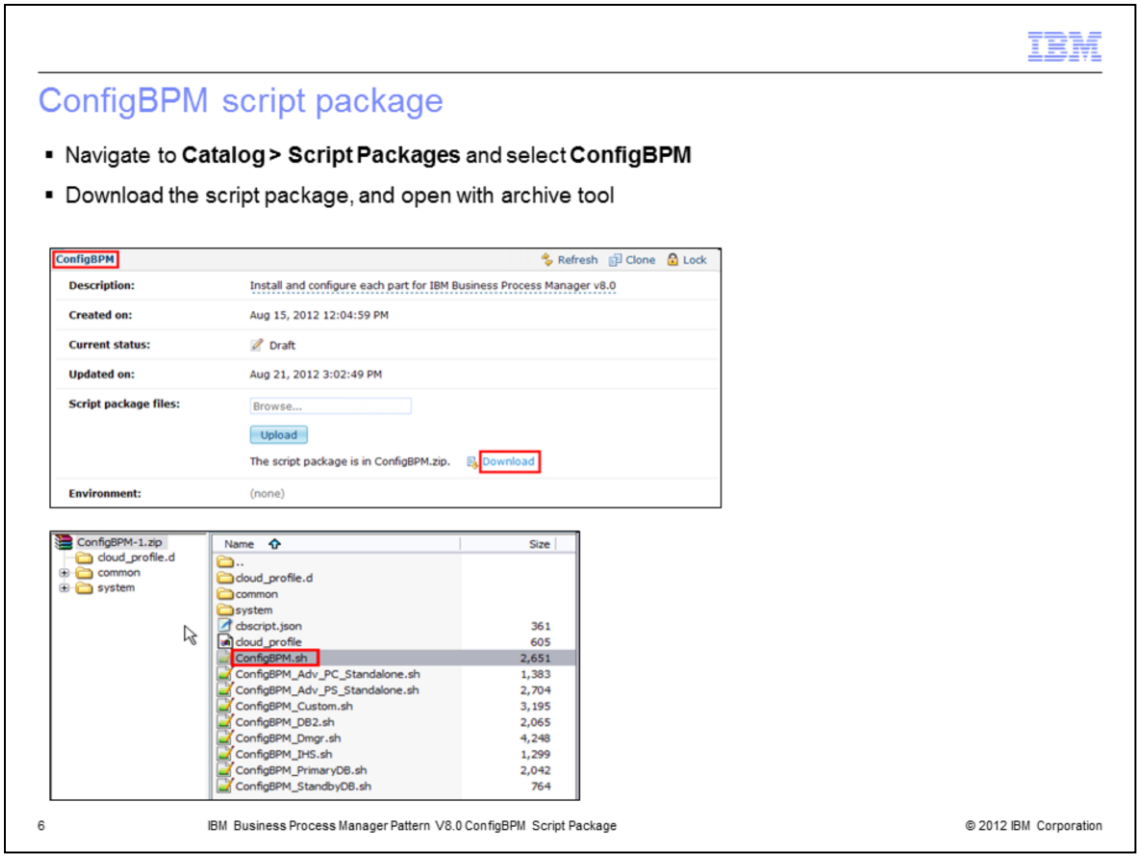
Here is another reason why IBM exposed the configuration scripts. Consider the scenarios where one of the configuration scripts has a defect or a customer needs a small configuration change. If the configuration scripts were available only as part of the virtual image, then IBM would have to create new images. Image building is a lengthy process. The time required would be unacceptable in these scenarios. Instead, IBM would have to create an emergency fix, a solution that should be avoided, if possible.

With the configuration scripts exposed, the process for modifying a script is very easy. The customer modifies the script package, modifies the pattern to pull in the script package and re-deploys the pattern. Note that you should not make significant changes to the exposed configuration scripts. At this time, there is very little documentation available about the details of these scripts, such as the scope of customization possible and each script's functionality.

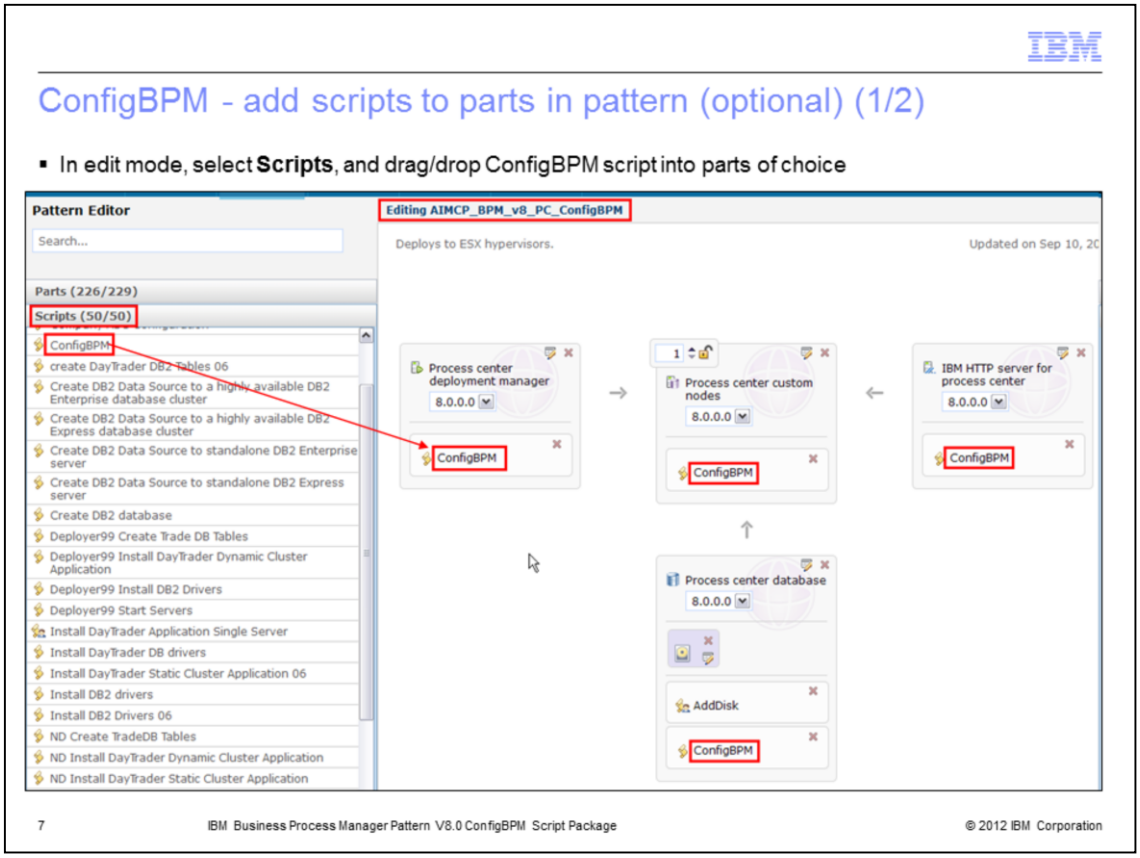


To deploy a Business Process Manager pattern with the script package, you must change the Script package activation property from False to True for each part in the pattern that runs the script package. Then typically you add the script package to the part. This step is optional in order to facilitate testing of the script. When testing the script, deploy the part without adding the script package and then manually run the script on the virtual machine until you are satisfied with the script. Once the script has been tested, then add it to the part and test the full, multi-hour deployment.

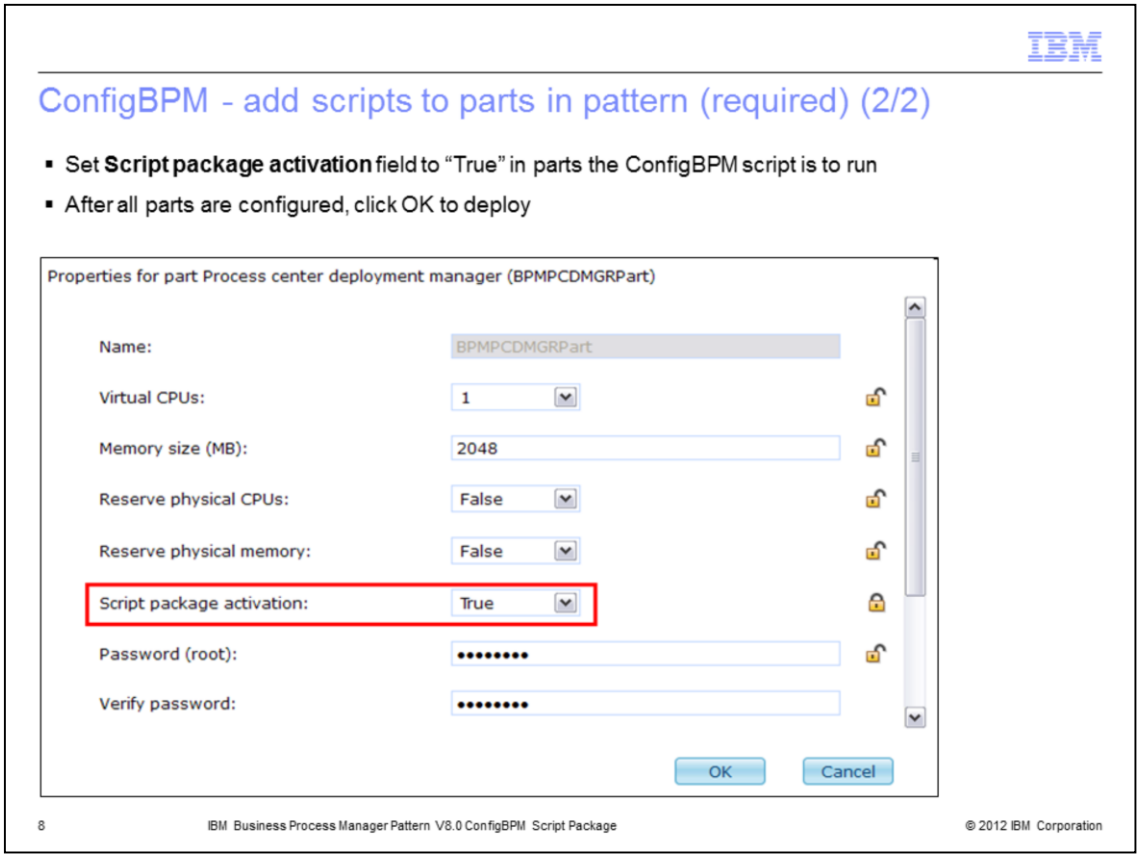
The ConfigBPM script package provided by IBM works for all parts of the Business Process Manager image. When ConfigBPM runs on any of the parts, it has the built-in intelligence to detect what kind of part it is running on and invokes the correct sub-scripts accordingly. If you modify the configuration scripts, you have the option of keeping all the scripts together in one script package or creating a separate script package per part.



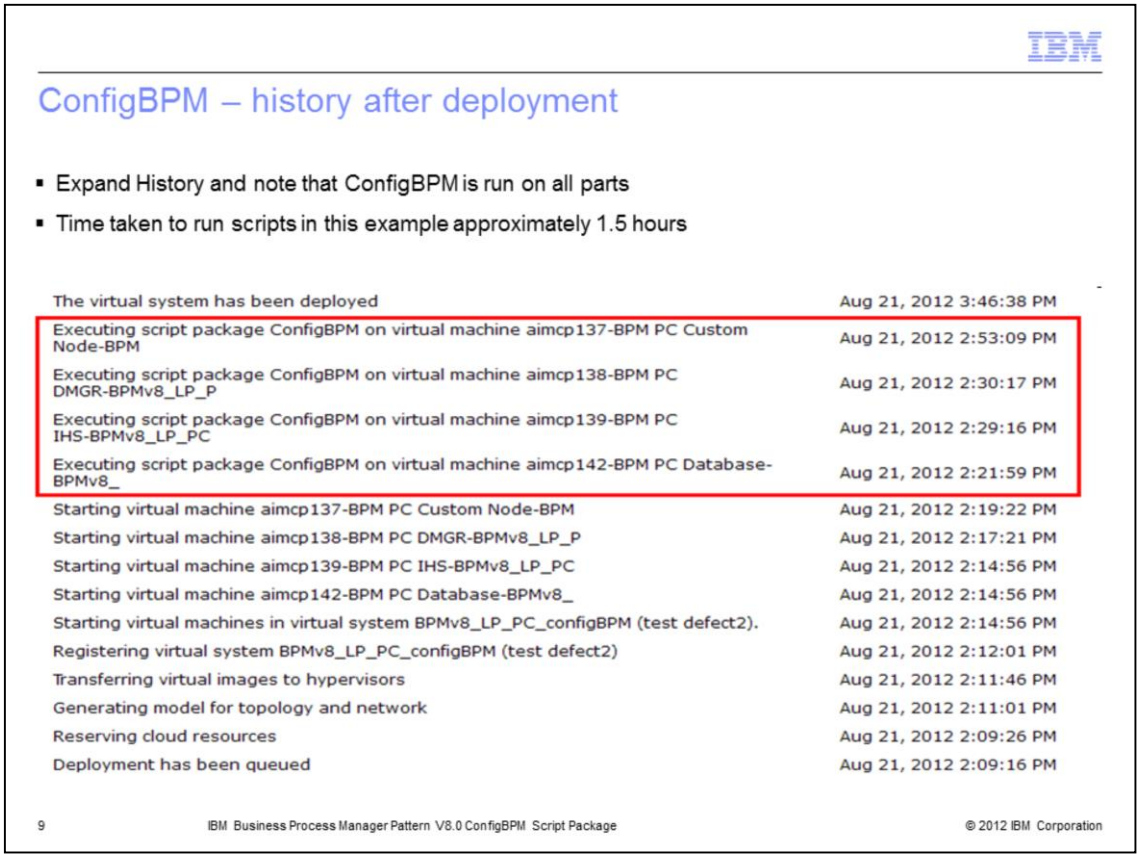
The next few slides walk through the screens related to the ConfigBPM script package. To view the contents of the script package, navigate to **Catalog**, and then **Script Packages** and select the **ConfigBPM** script. Once selected, you can download the script package archive file to your workstation and view it. When viewing it, remember that ConfigBPM.sh is the first script that runs, for any virtual image part.



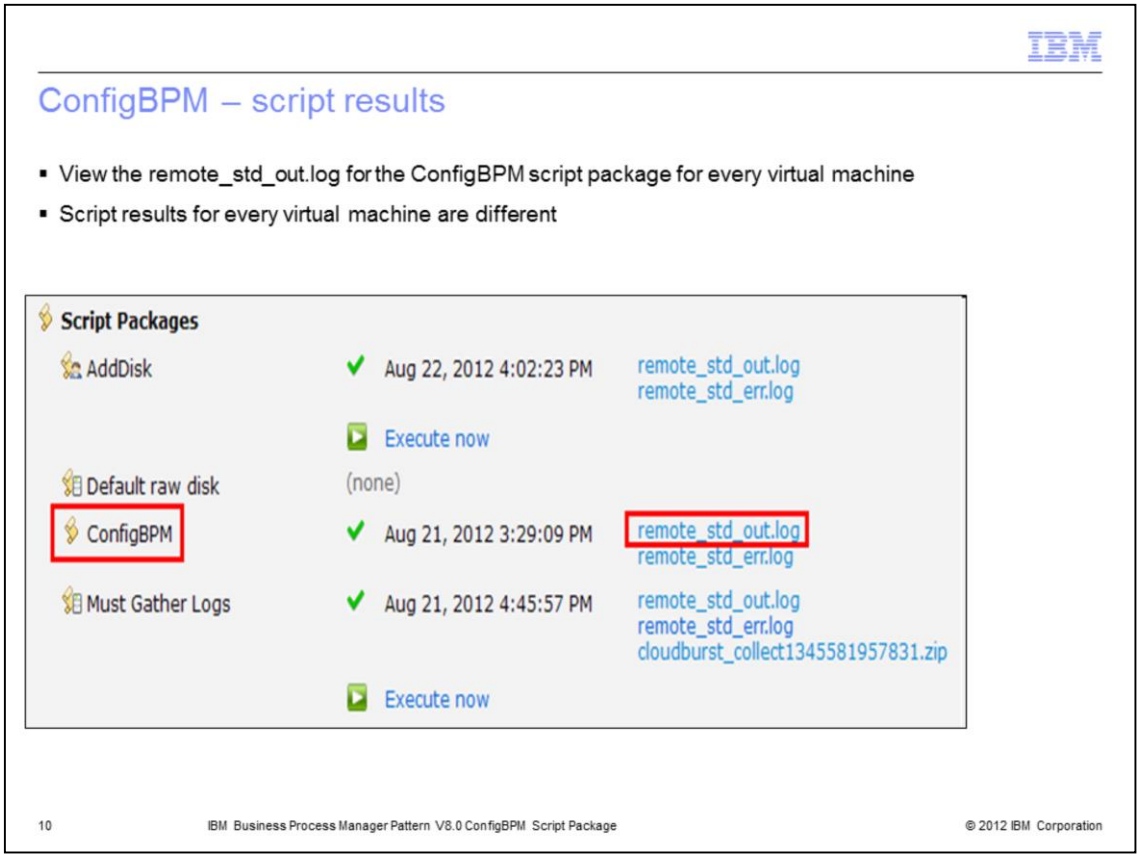
To add the ConfigBPM script package to any of the Business Process Manager patterns, while editing the pattern, drag the ConfigBPM script package into the part or parts where you want to run it. As mentioned before, before you add the script package to the deployment pattern you should test it by running it manually after deployment.



Additionally, for each part, you need to change the **Script package activation** property from **False** to **True**.



After the deployment completes, the History for the deployed instance reflects the fact that the configuration script package ran for the parts that you specified. On this slide, look at the length of time the scripts ran for the four parts in this example. The times varied from about one minute for the HTTP server to about 53 minutes for the custom node.



After the deployment completes, you can also view the results of the ConfigBPM script package. Select the “remote standard out log” to view that log.

ConfigBPM – script results example

```
Configuration type: Advanced Process Center Dmgr
In ConfigBPM_Dmgr.sh: configure dmgr.procctr.adv
[2012/08/21 19:30:59] Entry of NDutil configuration
[2012/08/21 19:49:15] Exit of NDutil configuration
[2012/08/21 19:49:15] Entry of setting SSH trust to IHS
Generating local ssh keys if needed
Generating public/private rsa key pair.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
67:95:e4:4d:2d:78:66:cb:9b:2e:c2:a8:5e:7b:17:84 root@aimcp138
The key's randomart image is:
+--[ RSA 2048]----+
|            .... |
|           o.+= .|
|           .+=.o |
|          E.. o  |
|        S o.   o |
|         o  . o  |
|        .o   o   |
|       ...+ o .  |
|     .o... o .   |
+-----------------+
Installing public key on aimcp139.austin.ibm.com
Buildfile: /opt/IBM/BPM/cloud/common/utils/setupSSH.ant

default:
  [sshexec] Connecting to aimcp139.austin.ibm.com:22
  [sshexec] cmd : mkdir -p /root/.ssh ; touch /root/.ssh/authorized_keys ; echo ssh-rsa
  AAAAB3NzaC1yc2EAAAABIwAAAQEArPss62Uff7KJL4EYCizkVsDtss0d/W12RQPUrS1KaaUHCwsQdeTeq6ptZrvNchwkeROwmOr+I59Xfqn5u/9e+UR4wTeYCQ0GC+5K4JebF51xn9XAlaf18WQP
  vZhupe1/rpxr0XkyaBRyOV3Of8q1hKDzzRb/9VfZZhcJVsH7xAQezMjnozTBzXDZKf1ReoEtB5pIh12GXwyRyvIdekQc58aIBm73XprO7WaySzDaWs0BKoAL5inClqVW6ibChy7+u4K9MY1pjSaF
  YwJv8mcwqupJl5YUw0OBa2cV1Bmyy9baLwxcX5FQae3pxFblC8f2pNeNcdp/RA5QKiLSc/rvvQ== root@aimcp138 >> /root/.ssh/authorized_keys; chmod -R go-rwx
  /root/.ssh
```


This slide shows an example log. You can review the entire log to see what configuration has taken place.

IBM

Trademarks, disclaimer, and copyright information