**Production Deployuri One Pager**

**1. Introduction**

**1.1. Project/Component Working Name:**

Support a new command to deploy the application by accepting the path as URI

**1.2. Name(s) and e-mail address of Document Author(s)/Supplier:**

Jeremy\_Lv: [lvsongping@cn.fujitsu.com](mailto:lvsongping@cn.fujitsu.com)

**1.3. Date of This Document:**

02/26/2013

**2. Project Summary**

**2.1. Project Description:**

Support a new command to deploy the application by accepting the path as URI

**2.2. Risks and Assumptions:**

This feature is done by external contributor Jeremy and there is certain risk associated with external contributors.

**3. Problem Summary**

**3.1. Problem Area:**

1). we prepare to introduce some new commands, Reasons are as follows:

1. As the original syntax about deploy command should be support, It is hard to reuse it to declare a command parameter which will both support the File and URI. It is because the file path can only accept one type of parameter.
2. The new command will keep things simple. It doesn’t need to change any codes related to the deploy command. We can define a new type of command parameter to support the URI syntax.

2). Not only allow users to deploy the application to the server by accepting the path as URI but also support the application to be deployed to the cluster.

3). In order to deploy the application remotely, the application should be download to the local temp directory first and then deploy the application to the server or cluster.

4). When it comes to the http protocol, there exist many situations as follows:

1. I think it is unreasonable to support two options like username and password, we should offer some friendly messages to the user under this situation. When we got the URI address base on the http protocol but we can’t download the application from the outside website because the website needs some user certification, it is hard and illegal to pass the username and password from the GF admin client or server to the other website’s database in a common situation.
2. When the suffix about the offered URI is not end with .jar, .war or .ear. I think we should offer some error messages to this situation because there’s no JDK to analysis these file types according to http protocol.(As the HttpURLConnection can judge other content type except .jar, .war and .ear)

5).When it comes to the ftp protocol, there exist many situation as follows:

1. It is no need to define two options as username and password, it is because the standard about ftp will be like <ftp://username:password@host:port/filename>, we can divided it into pieces to make sure we can connect to the ftp server and download the application to our local temp directory.

**3.2.** **Justification:**

The feature is more of ease of use, a convenience for avoiding a separate download step for an application that is available at a URI.

**4. Technical Description:**

**4.1. Details:**

**4.1.1. Expected behavior**

**4.1.1. 1. DAS mode**

With production deployuri feature, the user will be able to deploy the application by accepting the path as a URI type. This feature aims at two targets: to allow the user to deploy the application with the different types of URI and to allow users to deploy the application to the cluster by accepting the type of path as a URI.

1. Deploy the local application to the local server: This syntax is only aimed at the file:// syntax, the admin client will convert the file:// syntax to File syntax, Then the server will handle the process similar to the deploy command.
2. Deploy the remote application to the local server: This syntax is aimed at the http:// syntax and ftp:// syntax, the admin client will download the remotely application to the local temp directory(i.e: $Users\AppData\Local\Temp), then the server will accept the temp file address and handle the process similar to the deploy command.

1. Deploy the remote application to the remote server: When the application server is remotely and we need to specify the host address, the admin client will download the application to the temp directory first and then the remote server will handle the process similar to the existing remote deploy command.

**4.1.1.2. Standalone Instance mode**

The deployuri command support the application to be deployed to the instance, there exist many situations as follows:

1. Deploy the local application to the instance delivered by local server: This syntax is only aimed at the file:// syntax, the local admin client will convert the file:// syntax to File syntax, Then the application will be deployed to the sever, finally, the local server will deliver the application to the instance which is need to be deployed.
2. Deploy the local application to the instance delivered by the remote server: This syntax is only aimed at the file:// syntax, the local admin client will convert the file:// syntax to File syntax, Then the application will be deployed to the remote server, finally, the remote server will deliver the application to the instance which is need to be deployed.
3. Deploy the remote application to the instance delivered by local server: Firstly, the admin client will handle the URI and download the application to the temp directory. Secondly, the admin client will send the temp directory to the local server and the local server will handle the process to deploy the application. Finally, the local server will deliver the application to the related instance to make sure the application to be deployed to the instance.
4. Deploy the remote application to the instance delivered by remote server: Firstly, the admin client will handle the URI and download the application to the temp directory. Secondly, the admin client will send the temp directory to the local server and the remote server will handle the process to deploy the application. Finally, the remote server will deliver the application to the related instance to make sure the application to be deployed to the instance.

**4.1.1. 3. Cluster mode**

The deployuri supports running on a cluster. Deployuri commands which is executed in the server will be replicated to the each instance referenced by the cluster, then the cluster will handle the process about deploying the application. The information which is defined in the domaim.xml under instance will replicate from the server to each instance referenced by the cluster(The situation about the operation as asadmin, DAS, instance will be same to the standalone situation).

**4.1.2. Restricted features**

* --upload: The option of --upload is only support to the file syntax.
* We needn’t support the proxy option because it can be set into the domain.xml if any users need to set it.

**4.1.3. Use case examples**

**4.1.3.1. Context of examples**

For the command of deployuri to be applied, there must be an active version of the application.

|  |
| --- |
| $ asadmin deployuri file://test.war |

|  |
| --- |
| $ asadmin deployuri　–-host host\_address　file://test.war |

|  |
| --- |
| $ asadmin deployuri ftp://username:password@host:21/test.war |

|  |
| --- |
| $ asadmin deployuri  http://java.net/jira/secure/attachment/50467/test\_sample1.war |

|  |
| --- |
| $ asadmin deployuri  http://dldx.csdn.net/fd.php?i=271450501666412&s=35428f4bec874a3f4124a14beb2ec986 |

As the suffix String about the above http address is not end with .jar , .war or .ear, I think some of the error messages should be thrown out to the user.

|  |
| --- |
| $ asadmin deployuri –-host host\_address --target instancei1 ftp://test.war |

|  |
| --- |
| $ asadmin deployuri –-host host\_address --target instancei1　ftp://username:password@host:21/test.war |

|  |
| --- |
| $ asadmin deployuri　--host host\_address --target instancei1 http://java.net/jira/secure/attachment/50467/test\_sample1.war |

|  |
| --- |
| $ asadmin deployuri　–-host host\_address --target cluster1  http://java.net/jira/secure/attachment/50467/test\_sample1.war |

**4.2. Bug/RFE Number(s):**

<http://java.net/jira/browse/GLASSFISH-12699>

**4.3. In Scope:**

The command of “deployuri” will accept the URI as a path to deploy the application which is still not been downloaded. It will skip the step about downloading the application during the Internet.

**4.4. Out of** **Scope:**

**4.5. Interfaces:**

**4.5.1 Public Interfaces**

None.

**4.5.2 Private Interfaces**

None.

**4.5.3 Deprecated/Removed Interfaces:**

None.

**4.6. Doc Impact:**

Update man pages for the new/modified CLI commands

Application Deployment Guide/Developer Guide for new features

**4.7. Admin/Config Impact:**

CLI:

* Support a new command to accept the type of the path as a URI.

**4.8. HA Impact:**

No impact.

**4.9. I18N/L10N Impact:**

No impact.

**4.10. Packaging, Delivery & Upgrade:**

**4.10.1. Packaging**

The code could be packaged/delivered in the existing admin module and deployment module. (Presumably in the same module with the deploy command and do some changes about RemoteRestAdminCommand.)

**4.10.2.** **Delivery**

No impact.

**4.10.3. Upgrade and Migration:**

No impact.

**4.11. Security Impact:**

No impact.

**4.12. Compatibility Impact**

No impact.

**4.13. Dependencies:**

**4.13.1 Internal Dependencies**

None.

**4.13.2 External Dependencies**

If we support the application to be deployed with ftp:// syntax, I think we should inject external jar file called org.apache.commons.net.jar

**4.14. Testing Impact:**

Add new tests (automated) to test production deployuri feature.

**5. Reference Documents:**

**6. Schedule:**

**6.1. Projected Availability:**

* At production quality level - 3/13/2013 (EE7 MS6)