ADM v1.0 User's Guide

Draft

Dec 28, 2013

# Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.No | Version No. | Release Date | Author(s) | Section(s) Affected | Change History |  |
| 1 | 1.0.0.0 |  | Lorne | Initial Edition | Initial |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Contents

[Revision History 2](#_Toc376874904)

[1. Introduction 4](#_Toc376874905)

[2. Requirements 4](#_Toc376874906)

[3. Product Management 5](#_Toc376874907)

[4. Metadata Management 10](#_Toc376874908)

[4.1 Set Environment 10](#_Toc376874909)

[4.2 Manage Servers 10](#_Toc376874910)

[4.3 Manage Service Provider 11](#_Toc376874911)

[5. Product Deploy 12](#_Toc376874912)

[5.1 Deploy Product 12](#_Toc376874913)

[5.2 Update Product 15](#_Toc376874914)

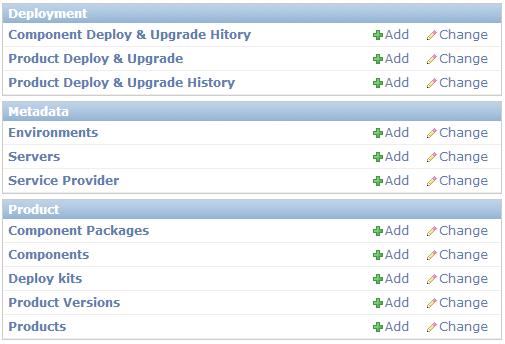
[5.3 Browse Product Deploy & Upgrade History and Component Deploy & Upgrade Hitory 17](#_Toc376874915)

Note:

1. All bolded word(s) wrapped with percentage mark, means the word(s) can be and should be replaced with meaningful value, such as **%RemDBUser%**.
2. Assume the previous version as **%PreVersion%** and current version as **%PackVersion%**.

# Introduction

ADM(Automatical Deployment Management) is the management web site of automatical deployment system.ADM include three database table group - deployment, metadata, product and four user group –Developer, Maintenance, ProjectManager, Tester. The following chapter will guide how to use this system.



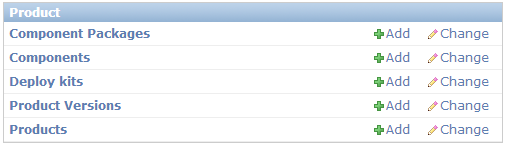
As screenshot above, every data entity have Add and Change buttons. Add button is used to create a entity while Change button for modify or browse.

1. Add or Change button are not shown for some entity means you have no perssion to do that
2. If you entered Change view found can not edit it’s field,that means you have no perssion or you have no necessary to edit them

# Requirements

ADM should be already installed successfully, open ADM’s website (http://cnemoptestpuppet:8000/admin or http://cnemopprodpuppet:8000/admin) and use you own account to login.

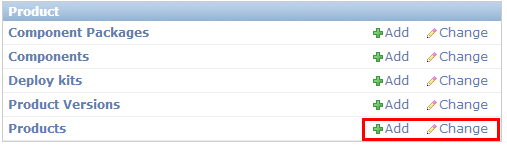
# Product Management



Because of dependency relation between each entity, you’d better create product data as following sequence.

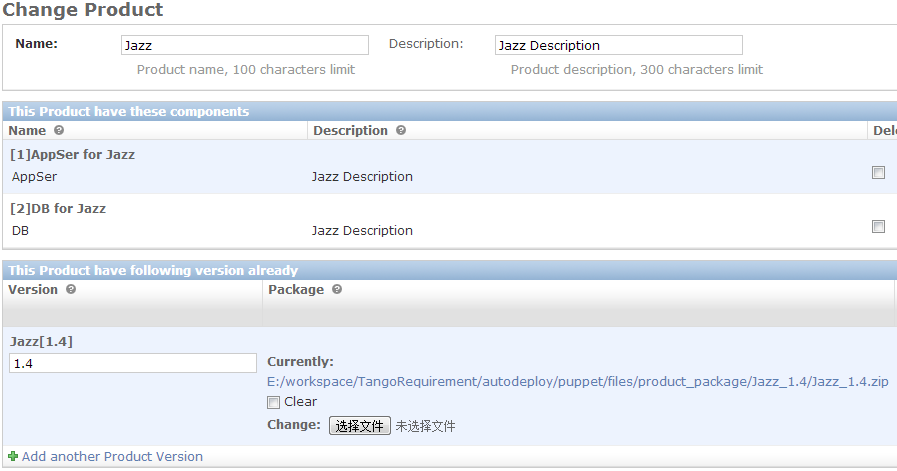
According to permission configuration:

1. only ProjectManager group can add Product and Component
2. Developer group can only browse Product and Component, but have full permission of Product Version and Component Package except set Product Version as release.
3. Maintenance group can only browse release Product Version
4. Test group can browse all entiry and set a Product Version as release version.
   1. **Manage Product**



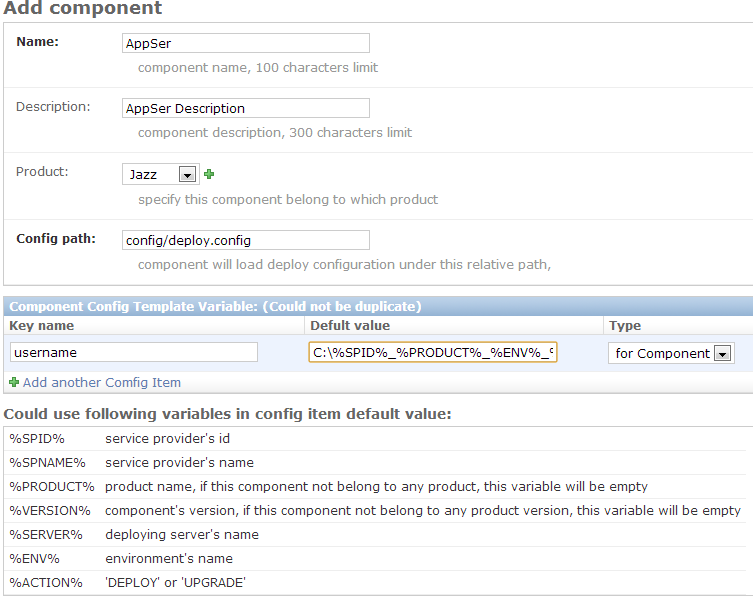
Click add button of Products, fill up all blank and click save button to create a Product.

Click Change button of Products to browse or modify, in Product’s change view, you can see all its Component and Product Version.



* 1. **Manage Component**

Click add button of Component to create Component, meanwhile coulde add config item for this Component in Component Config Template Item panel below. In the Default value field could use these variable : %SPID%, %SPNAME%, %PRODUCT%, %VERSION%, %SERVER%, %ENV%, %ACTION%. Take care Type field, if this config Item is for component you shoulde choose for Component, if this config item is for Deploy Kit please choose for Deploy Kit.



* 1. **Manage Deploy Kit**

Click add button of Deploy Kit, fill up all blank and click save button to create a DeployKit.

Take care that package field can only accept zip file which will be upload to /etc/puppet/files/deploy\_kit/ on the server.

The kit package must keep the **packageroot/{deploykit name field}.exe** is the entry excutable file.



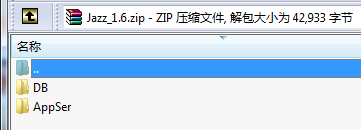
* 1. **Manage Product Version**

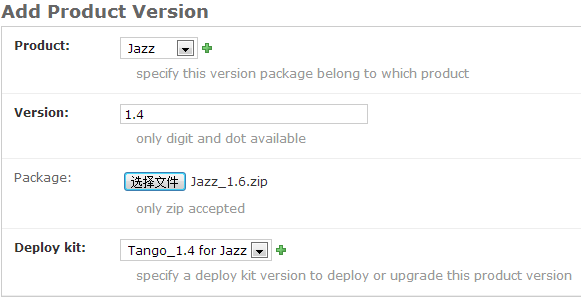
Click add button of Product Versions to create a Product Version.

Here is automatically process to extract each Component Package from the whole Product Package, please make sure :

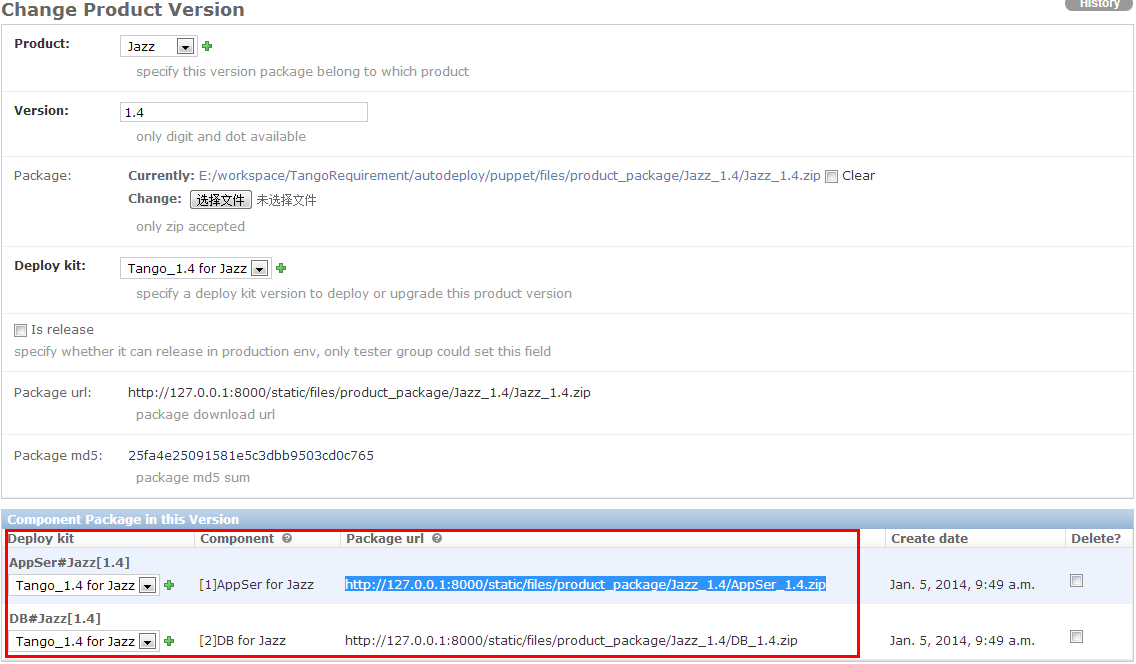
1. Select the right Product
2. Component folder should be in the root of this zip package
3. Every component folder’s name in this zip package should be equal with name field just created in step 3.2
4. If doesn’t meet the condition above,the automatically extract process may not work or make mistakes. In this situation you can refer to step3.5 to upload Component Package manually and individually.

The product version packge should ensure **every component folder in the root position**, for instance:



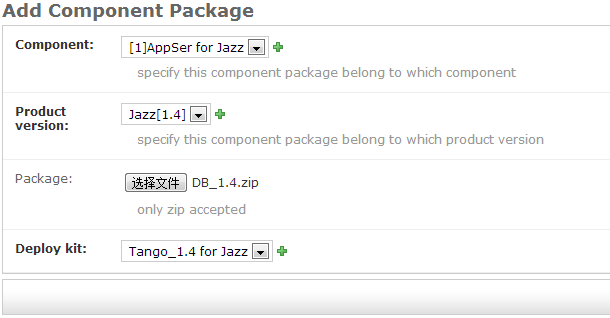


After created, you can enter Change view to check whether all Component Package extracted correctly.

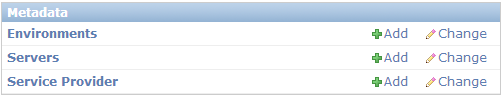


* 1. **Manage Component Package**

If your Product Version zip package is correct, it’s no necessary to go through this step. But you still could create Component Package manually or edit Component Package record at here.



# Metadata Management



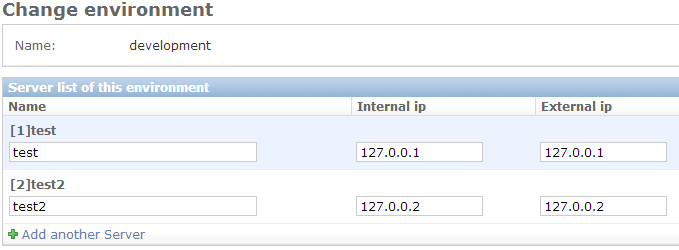
In this section you can manage Environment, Servers and Service Provider.

# Set Environment

Click Add button for Environment to create a Product Version, your can only add the environment you need.

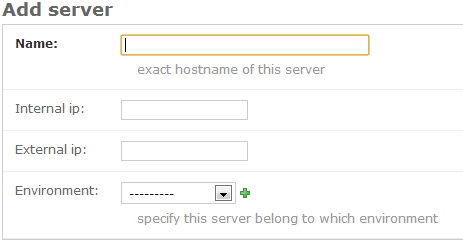


In the Change view of Environment, you can browse all server in this environment.



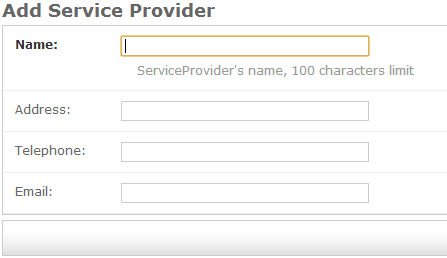
# Manage Servers

Click Add button of Server to create a server.

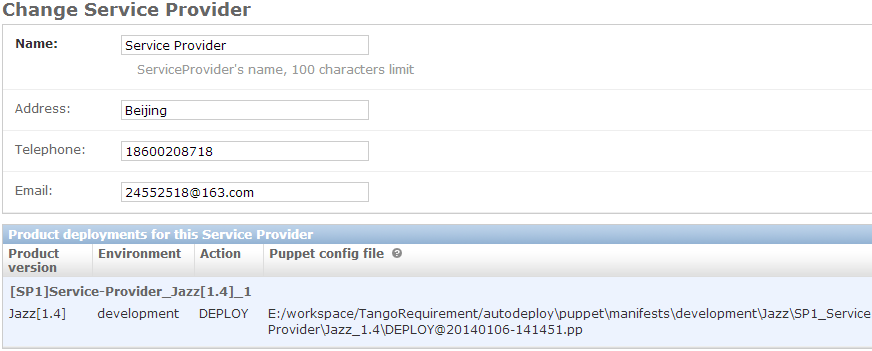


# Manage Service Provider

Click Add button of Server to create a server.



In Service Provider’s change view will list all ProductDeploymentHistory record.



# Product Deploy

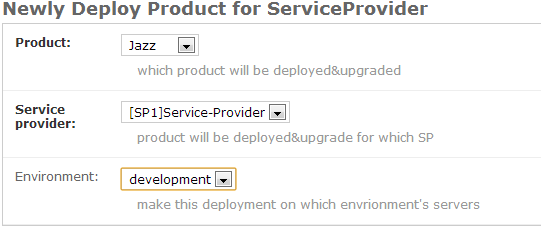


If no script is providered, you can initially configure ADM following below step.

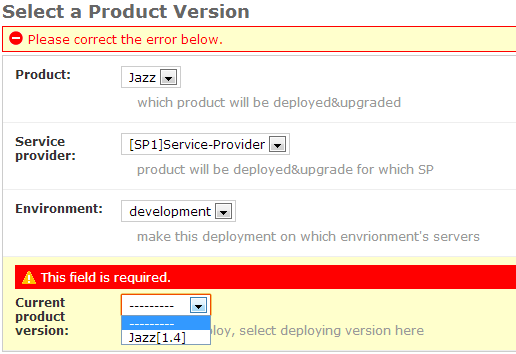
# Deploy Product

Click Add button for Product Deploy & Upgrade to create a product deployment.

1. Select Product, Service Provider and Environment in the first view, then click save button



1. Select Product Version in second view, then click save button



1. Select server for every component and fill in configuration in the next view , then click save button.

If you not wanna deploy some Component of this Product, you simplely leave the Server field of this Component as empty.



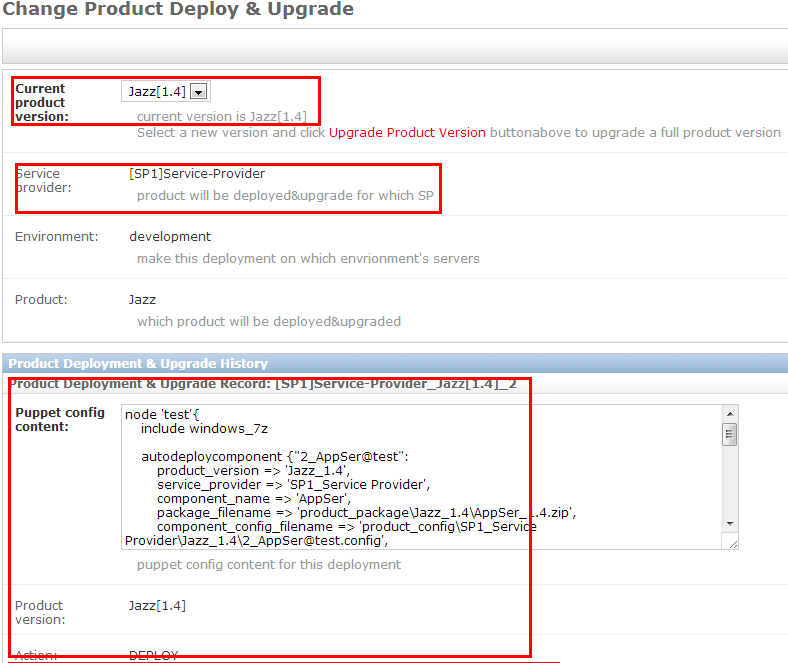
1. All deploy configuration will print on the deploy report, deploy finished.



1. When puppet agent next time polling puppet master, this deploy specification will be executed.

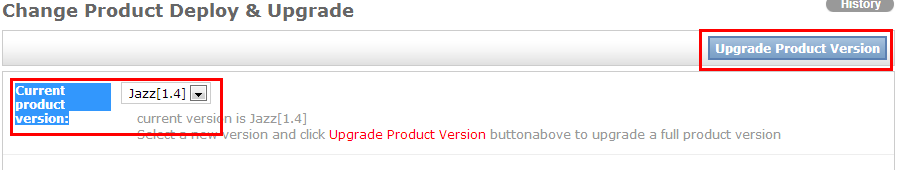
# Update Product

1. Click Change button for Product Deploy & Upgrade to browse a product deployment. In the panel at the bottom, you can see the ProductDeployHistory record belong to current Service Provider.

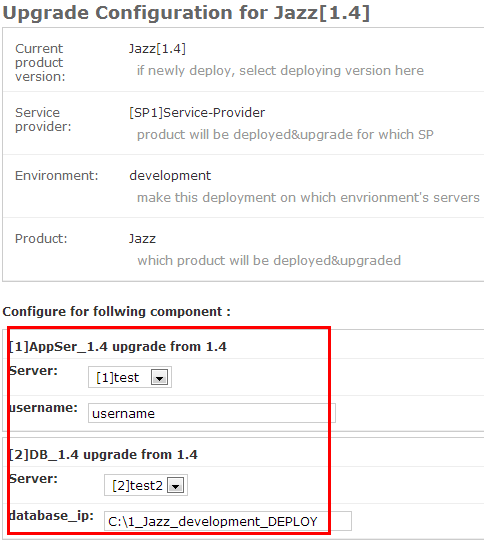


1. Select a new product version in the Current product version field then click Upgrade Product Version button.

If you just wanna re-deploy some component of current product version, you can directly click Upgrade Product Version button.



1. Fill configuration for this upgrade then click Upgrade button.



1. Upgrade report will be print out on the nex view
2. When puppet agent next time polling puppet master, this upgrade specification will be executed.

# Browse Deploy&Upgrade Hitory



After Entered the change view of Product Deploy & Upgrade History and Component Deploy & Upgrade Hitory, you can see all product and component deploy history and its deployed configuration.

In normal situation these two entity can only allow to browse, not allow to modify or delete. If have necessary to modify or delete these record , you must login as superuser.