CQ5 Setup with Eclipse

Posted on January 14, 2015 by Kedarnath Varadi • 0 comment

Below are the steps to set up workspace and start development of CQ5 components in Eclipse:

Prerequisites:

1. CQ5 Installation
2. File Vault (VLT)
3. Eclipse
4. Apache Maven

Search

Recent Posts

Adobe CQ5 Questions and Answers
CMS Migration to Adobe Experience Manager AEM/CQ5
Adobe Experience Manager – Digital Experience Delivery Made Easy for Marketers
CQ5 Setup with Eclipse
Which Organizations

Step 1

- Create a project in CQ5 using CRXDE LITE
- Under the /apps folder, create the nt:folder myApp.
- Under the myApp folder, create the nt:folder components.
- Under the myApp folder, create the nt:folder templates.
- Under the myApp folder, create the nt:folder install.
- Click save button after creating the above things.

Step 2 – Installing File Vault (FLT)

- In our file system, go to <cq-installation-dir>/crx-quickstart/opt/filevault. The build is available in both tgz and zip formats. Extract either of this two archives.
- Add <cq-installation-dir>/crx-quickstart/opt/filevault/vault-cli-<version>/bin to your environment PATH so that the command files vlt or vlt.bat are accessed as appropriate. For example, <cq-installation-dir>/crx-quickstart/opt/filevault/vault-cli-1.1.2/bin
- Open a command line shell and execute vlt –help. Make sure it displays the following help screen:

Which Organizations implementing Adobe CQ5/AEM Web CMS

Recent Comments

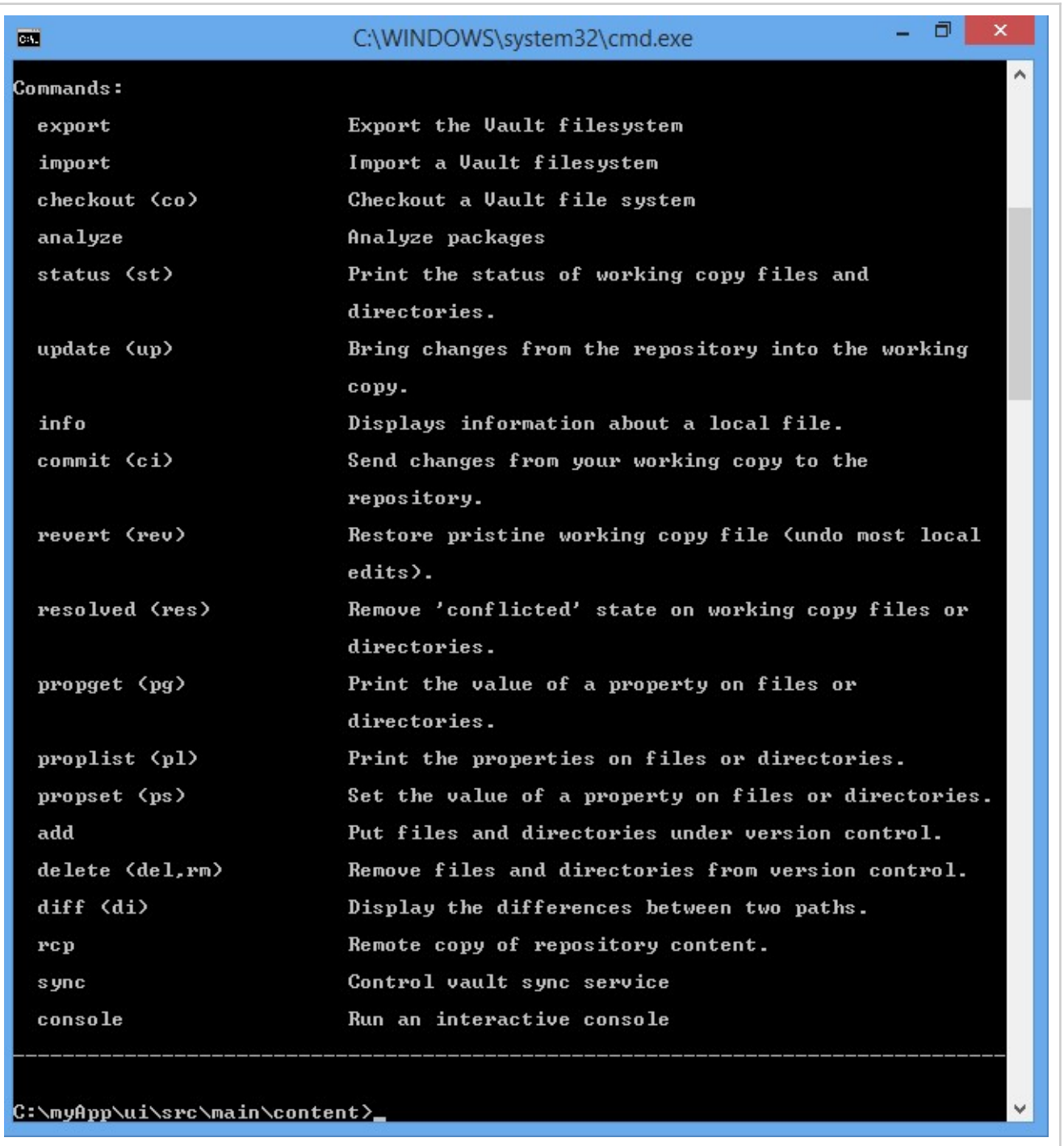
Archives

February 2015
January 2015

Categories

Adobe AEM
Adobe CQ5
Adobe Experience Manager
AEM Installation
CQ5 Installation
CQ5/AEM Development
CQ5/AEM Implementation

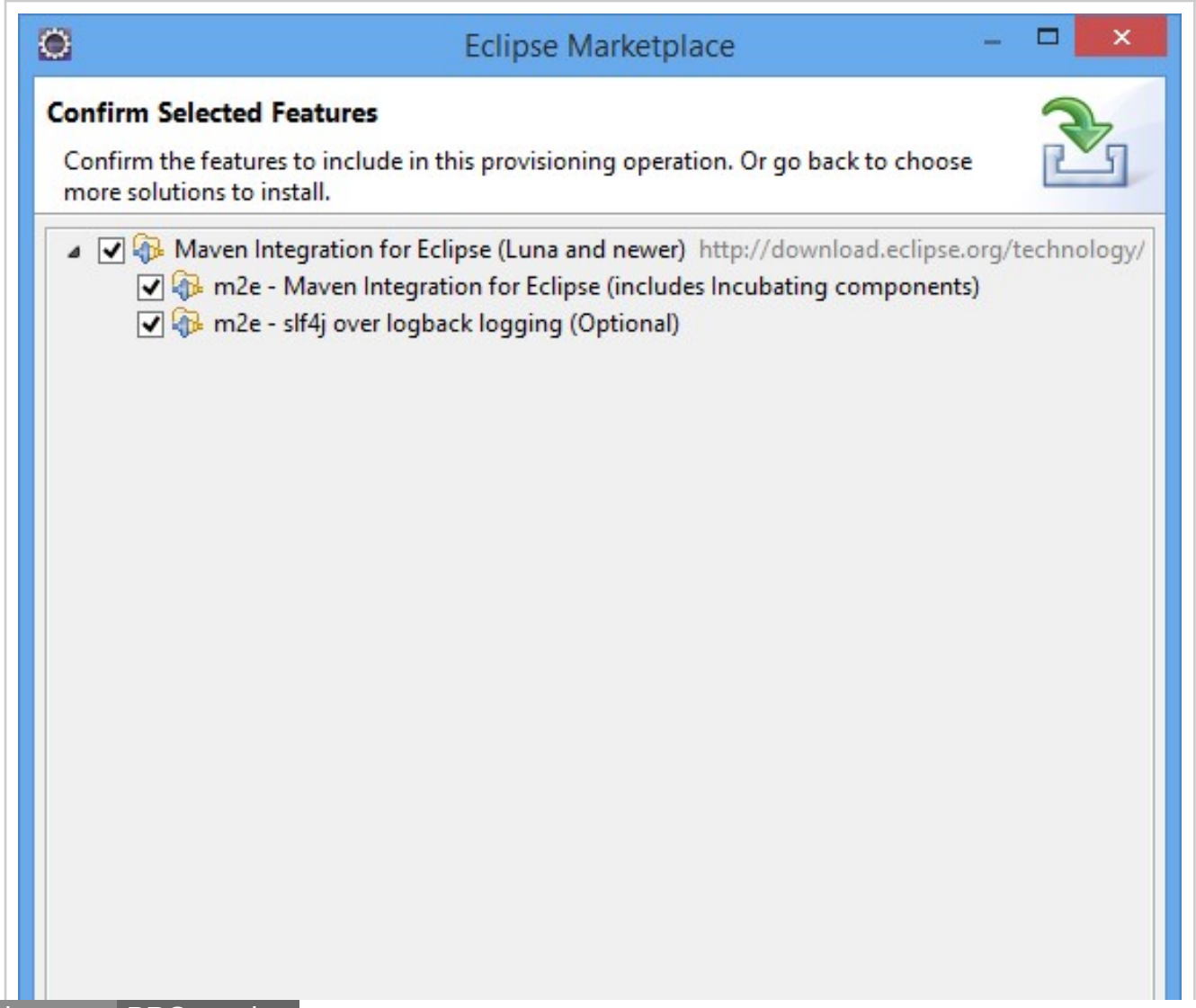
Meta

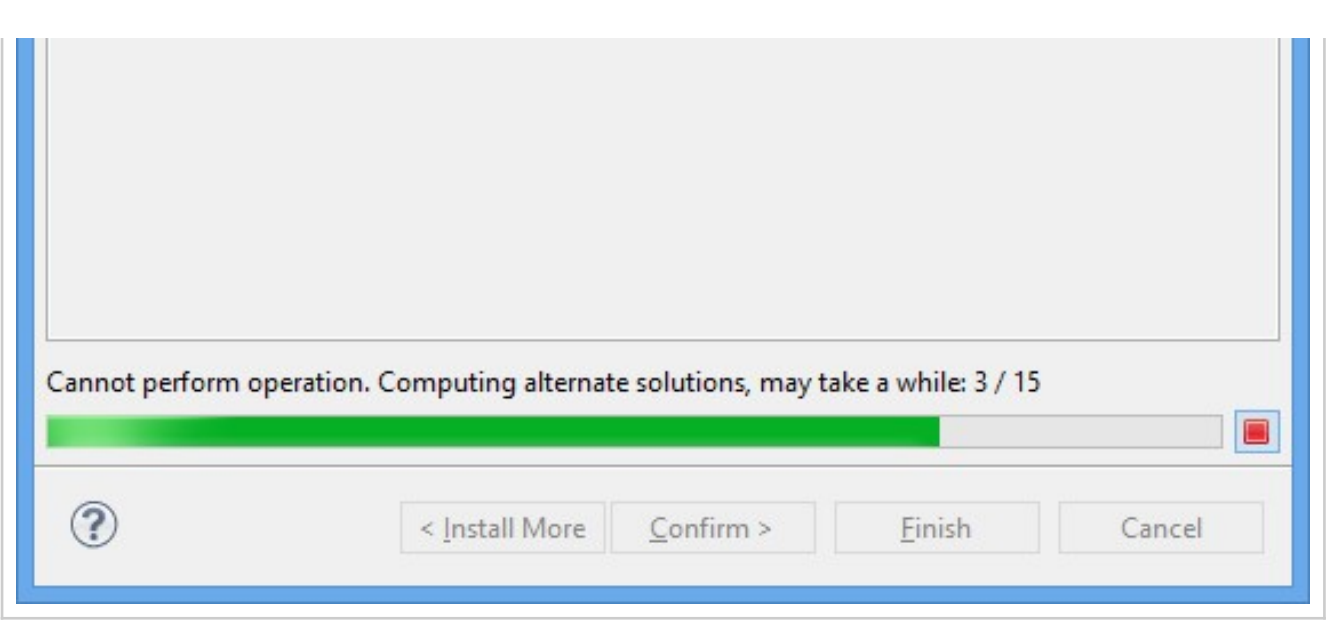


Log in
Entries RSS
Comments RSS
WordPress.org

Step 3

- Open eclipse by double clicking eclipse.exe
- Create a new workspace for your project and name it myApp.
- install maven plugin through eclipse market place





- After installation, restart Eclipse.

Step 4

In this section, we create two Maven projects: One called UI (after User Interface) which contains the Adobe Experience Manager AEM/CQ5 project structure with the JSP scripts and the other called Core which contains the Java code (source and compiled). The compiled code is stored in a jar file. The advantage of such a structure is that it adds modularity and autonomy to the logic of your application because each jar file (bundle) can be managed separately. Follow the below steps:

1. In Eclipse, click File > New > Other.

2. In the dialog, select Maven > Maven Project and click Next.
3. Select the Create a simple project option and the Use default Workspace location option, then click Next.
4. Specify the following property values for the Maven project, and accept the default values for all other properties:
    - Group Id: com.day.cq5.myapp
    - Artifact Id: ui
    - Name: CQ5 MyApp UI
    - Description: This is the UI module
5. Click Finish.

Now, set the Java Compiler to version 1.5:

1. In the Project Explorer view, right-click the ui project and click Properties.
2. Select Java Compiler and set following properties to 1.5:
    - Compiler compliance level
    - Generated .class files compatibility
    - Source compatibility
3. Click OK

Now, create the filter.xml file that defines the content that VLT exports:

- In the Project Explorer view, in the ui/src/main folder, create the content folder.
- Under content, create the META-INF folder.
- Under META-INF, create the vault folder.

- Under vault, create the filter.xml file.

  In filter.xml, copy the following code to filter.xml:

  ```
  <?xml version=”1.0″ encoding=”UTF-8″?>

  <!– Defines which repository items are generally included –>

  <workspaceFilter version=”1.0″>

  <filter root=”/apps/myApp” />

  <filter root=”/etc/designs/myApp” />

  </workspaceFilter>
  ```

- Now, save filter.xml.

Step 5

Use VLT to check out the CQ5 content into your ui project:

1. In the system command line, change the current directory to the following directory in your Eclipse workspace:
<eclipse>/<workspace>/myApp/ui/src/main/content.

2. Execute the command: vlt –credentials admin:admin co http://localhost:4502/crx
3. This command creates the folder jcr_root under <eclipse>/<workspace>/myApp/ui/src/main/content. This maps to the CRX root (/). Under jcr_root the following files and folders are created, as defined in filter.xml:

- apps/myApp
- etc/designs/myApp

4. It also creates two files, config.xml and settings.xml in <eclipse>/<workspace>/myApp/ui/src/main/content/META-INF/vault. VLT uses these files.
5. In the ui Eclipse project, create a link to the apps folder that you just checked out. This link enables your JSP scripts to reference the files.
6. Right-click ui, select New, and then Folder.
7. In the dialog box, click Advanced and select Link to alternate location.
8. Click Browse, specify <eclipse>/<workspace>/myApp/ui/src/main/content/jcr_root/apps, and then click OK.
9. Click Finish.

For any queries, please feel free to contact – Kedarnath Varadi, kvaradi@nextrow.com

Category: Adobe AEM, Adobe CQ5, Adobe Experience Manager, AEM Installation, CQ5 Installation | Tags: Adobe AEM Setup with Eclipse, Adobe CQ5 Setup with Eclipse, Adobe Experience Manager, AEM Setup with Eclipse, CQ5 Setup with Eclipse

Leave a Reply

Your email address will not be published.*Required fields are marked

Name
*

Email
*

Website

Comment

Post Comment

← Previous

Next →