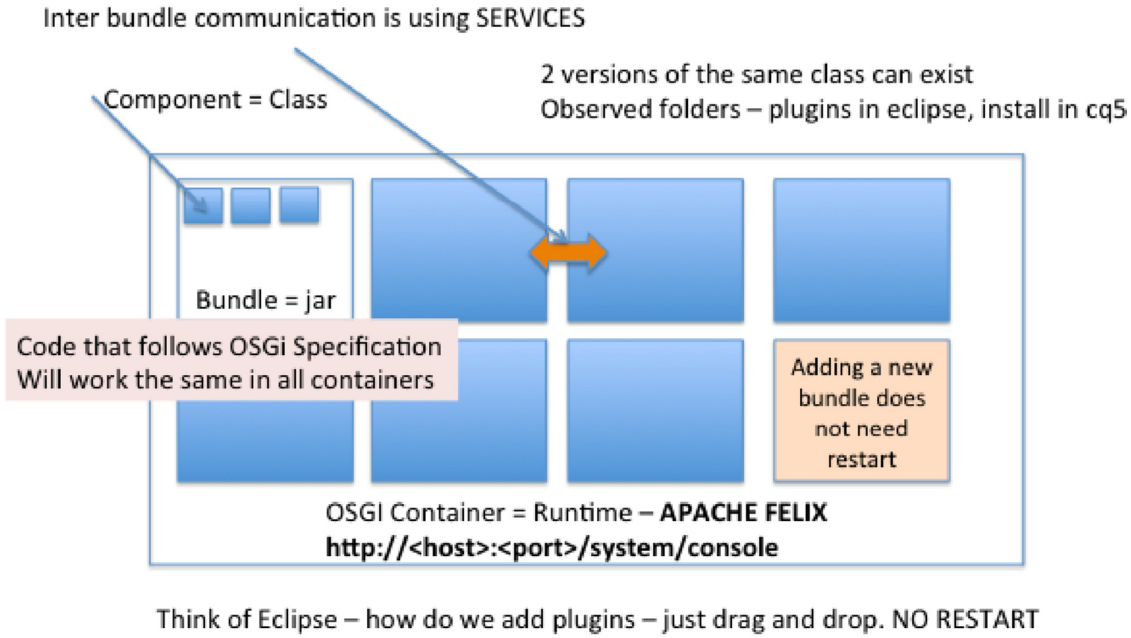
AEM CHEAT SHEET

OSGi



Think of Eclipse – how do we add plugins – just drag and drop. NO RESTART

1. Bundles = jar files + Manifest
2. Components = java classes
3. Services = special components used to interact between bundles
4. OSGi is a container in which any bundle can be added at runtime
5. Bundle jar files are dropped in folders called **install** to add them to CQ

Annotations: To add properties

```
@Properties({

    @Property(name = "scheduler.expression", value = "*/5 * *
* * ?") ,// Every 5 seconds     @Property(name =

"scheduler.expression.meta", options={


            @PropertyOption(name="prop1", value="Option 1"),

@PropertyOption(name="prop2", value="Option 2")


}),

@Property(name =
```

```
"scheduler.expre
ssion.count",
value =
"somevalue", cardinality=5)


    }
   )
```

To read properties: `componentContext.getProperties().get("prop-name").getString();`

Sling

1. URL Decomposition – get the contentPath, method and selectors from the URL
2. Content Resolution – identify the content and access the sling:resourceType Property
3. Resource Resolution - /apps or /libs
4. Script Resolution – selection+extn, selector, extn, method, default (same name as component name)

Sling Post Hanlder

1. Post using <form> will create new objects
2. To override point the form to currentNode *<form action="<%=currentNode.getPath()%> .html" method ="post"/>*
3. Create a **<comp-name>.POST.jsp** in the component folder
4. Handle all things in the JSP code

Interfaces:

1. CRX Explorer - http://localhost:4502/crx/explorer/browser/index.jsp
2. CRX DE Lite - http://localhost:4502/crx/de/index.jsp
3. Apache Felix - http://localhost:4502/system/console
4. Site admin - http://localhost:4502/siteadmin
5. etc/Tools - http://localhost:4502/miscadmin

WebDAV

To put multiple files in CRX repository use a tool like CyberDuck

AEM Cheat Sheet

Important Locations

Path in CRX	Content type
/content	all user generated content
/etc	all configuration
/apps	all code
/etc/design	all deisgn realted information

Key Properties

Entity	Type	Properties/Note
Template	cq:Template	allowedPaths, allowedParents, sling:ResourceType
Page Component	cq:Component	sling:resourceSuperType, dialog, design_dialog, cq:editConfig
Page	cq:Page	Is created in /content based on template by end user
Global Component	cq:Component	sling:resourceSuperType, componentGroup, allowedParents
Client Libs	cq:ClientLibararyFolder	dependencies, categories

Dialogs and Design Dialogs

1. Dialog must be named **dialog**
2. Design Dialog must be named **design_dialog**
3. All widgets should have be withing a node of type **cq:widgetCollection** and it should be named **items**
4. In the widget the key properties are
    i) xtype – the type of the component you want to use (textfield, pathfield, richtext)
    ii) fieldLabel – comes from the widgets API
    iii) name (with ./<prop-name>) will create a property <prop-name> in jcr under current node with value from the xtype as entered by the user

AEM Cheat Sheet

API usage

1. *properties.get("prop-name")* – gets the property value from current node
2. *currentStyle.get("prop-name")* – gets globally stored value by design_dialog
3. *currentPage* – reference of CQ Page API
4. *currentNode* – reference of CQ Node API
5. *resource* – reference of the object Sling is working on
6. *pageManager* – page helper class
7. *sling* – helper class to get services

Code Snippets

// JSP ready reckoner

```
 <%@ %> - Directive - used for imports and other attributes

 <%! %> - Declaration - used to instantiate objects

 <% %> - Scriptlet - java code goes here  <%= %> - Expression

- evaluate jsp variable and show on html
```

// to get list of selectors in an array

```
String[] selectors =
slingRequest.getRequestPathInfo().getSelectors(); //
```

to get a page from a path

```
Page page = pageManager.getPage(path); //
```

to get any containing page for a resource

```
Page page =
pageManager.getContainingPage(resourceResolver.getResource(pat
h));
```

// to get a session in JSP

```
final SlingRepository repos =
sling.getService(SlingRepository.class); session
= repos.loginAdministrative(null);
```

// to get query object and run a query //build query using search in crx explorer or crxdelite

```
String stmt = "select * from cq:Page where jcr:path like
'/content/training/%' and contains(*, '" +
slingRequest.getParameter("q") + "') order by jcr:score desc";
Query query =
currentNode.getSession().getWorkspace().getQueryManager().crea
teQuery(stmt, Query.SQL);

QueryResult results = query.execute();
```

// Get node from path

```
Node n = session.getItem(path); // where session = JCR session

session.save(); session.logout();
```

// add a property to node

```
Node n;

n.setProperty("prop-name","value");   //find WCM
mode (if the author is in edit mode)
if(WCMMode.fromRequest(request) == WCMMode.EDIT)

//for publish instance it will be WCMMode.DISABLED
```

Java Code

// to get repository

```
@Reference

SlingRepository repository;  // get session from
repository                session                =
repository.loginAdministrative(null);
```

// creating a logger instance for a class ReplicationLogger

```
private static final Logger LOGGER =
LoggerFactory.getLogger(ReplicationLogger.class);
```

//to get pagemanager

```
@Reference

SlingRepository repositroy;

@Reference

ResourceResolverFactory resolverFactory; Session session =

null;    session = repositroy.loginAdministrative(null);

Map<String,Object> authInfo = new HashMap<String, Object>();

authInfo.put(JcrResourceConstants.AUTHENTICATION_INFO_SESSION,
session);

ResourceResolver resourceResolver =
resolverFactory.getAdministrativeResourceResolver(authInfo);

PageManager pageManager =
resourceResolver.adaptTo(PageManager.class);
```

//to replicate a resource

```
@Reference
Replicator //com.day.cq.replication. Replicator
r.replicate(session, ReplicationActionType.ACTIVATE, <path-
toreplicate);
```

//to test if the crx instance is master of the cluster

```
@Reference
private Repository repository;

Boolean isMaster =
getBoolean(repository.getDescriptor("crx.cluster.master"));
```

//to send an event

```
@Reference
EventAdmin eventAdmin;
```

```
final Dictionary<String, Object> props = new Hashtable<String,
Object>();
props.put("path", “dummy”); org.osgi.service.event.Event
evt = new Event("com/cgi/training/store",props);
eventAdmin.sendEvent(evt);
```

// to schedule a job

```
@Reference
org.apache.sling.commons.scheduler.Scheduler scheduler;

  public void scheduleJob()

{ log.info("scheduling");

final Runnable job = new Runnable() {
public void run() { log.info("firing
job now");
scheduler.removeJob("training");
}};

// Date fireJobAt = ;
//scheduler.fireJobAt("training", job, null,);

try {
scheduler.addPeriodicJob("training", job, null, 10, true);
}

catch (Exception e) {
log.info("error in scheduling job"); }
```

Dev URLS

Dumplibs: To see the location of client libraries in cq

http://localhost:4502/libs/granite/ui/content/dumplibs.html

CQ5 System Administration

Determine the capacity and hardware requirements from here:

http://dev.day.com/docs/en/cq/current/managing/capacity-guide.html

CQ5 Deployment Steps – For Production

1. Install the servers

   a. Java –jar (cq-jar-name) –Xmx –Xms –XX:MaxPermSize to whatever
2. Set the repository.xml and workspace.xml in order
   a. Repository.xml
      i. Set the data cluster properties
   b. Workspace.xml
      i. Set tar pm properties
         1. autoOptimizeAt
         2. bundleCacheSize – 10% of –Xmx on JVM ii. Set search indexing
         1. Create indexing_config.xml
3. Set up publish instance
4. Set up Replication and Reverse Replication agents on Author
5. Set up Dispatcher
6. Set up dispatcher flush agent
7. Set up OSGi Configurations
   a. Disable or remove webdav
   b. Disable or remove Debug mode
   c. Any other project specific changes

CQ5 Server Maintenance

1. Optimize Tar – Everyday (off peak time)
2. Data Store Gabage Collection – Atleast once a week
3. Delete Tar Journal – Every 2 week
4. Back ups – as needed (Every Day)
5. CQ5 health check and security check

Backing up as a Package

1. /content/<site>
2. /apps/<site>
3. /etc/design/<ur-site>
4. /etc/workflow/models/<ur-workflow-models>
5. */etc/replication/agents-on-author*
6. /apps/foundation/<ur-code>
7. /campaigns/<ur-campaign>
8. /home/groups/<ur-user-groups> 9. /home/users/<ur-users>

segmentation, users, profiles

Security

CQ OSGi Production Checklist -

AEM Cheat Sheet

http://dev.day.com/docs/en/cq/current/deploying/osgi_configuration_settings.html feed.xml disable from dispatcher.any

Change passwords

Performance http://helpx.adobe.com/experience-manager/kb/performancetuningtips.html

BundleCacheSize – workspace.xml **FineGrainedISMLocking** – workspace.xml

ItemState Manager – Maintains state of Item (Node/Property)

ISM Locking – FineGrained and Default(Coarse)

resultFetchSize – workspace.xml/SearchIndex – set to 50

Caching TAR PM indexes

```
<PersistenceManager
class="com.day.crx.persistence.tar.TarPersistenceManager">
        <param name="indexInMemory" value="true" />  </PersistenceManager>
```

http://dev.day.com/docs/en/crx/current/administering/performance.html

Testing

Load Test: http://www.gastongonzalez.com/tech-blog/2013/11/17/aem-adobecq-load-testing

Getting system information

http://localhost:4502/system/console/config

Configuration tab – shows the configuration that CQ5 instance is working with. If there is a mismatch from what has been configured in the crxde-lite sling:OsgiConfig node then the configuration done there might be incorrect else the fault may be with the underlying code

Clustering 2 CQ instances

1. Delete the slaves (crx-quickstart/repository/) cluster_node.id file
2. Change the slave and masters cluster_id in cluster.properties files to the masters cluster_node.id
3. Start the master
4. Start the slave

Managing Access to CQ Interface

Console	Path
Site Admin	/libs/wcm/core/content/siteadmin
DAM Admin	/libs/wcm/core/content/damadmin
Tools	/libs/wcm/core/content/misc
Security	/libs/cq/security/content/admin
Workflow	/libs/cq/workflow/content/console
Tagging	/libs/cq/tagging/content/tagadmin

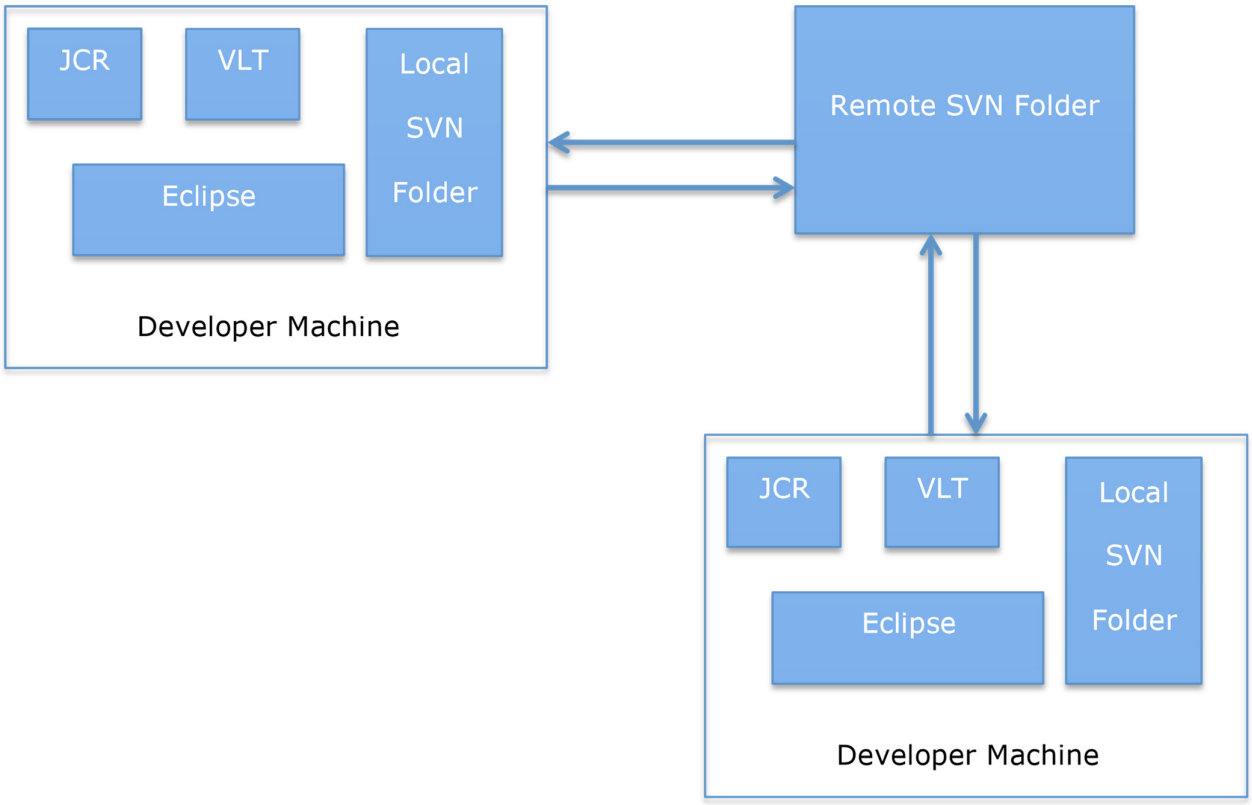
CQ Advanced Topics

Factory Service is a service with multiple implantations

Regular Services have Singleton implementations

AEM Cheat Sheet

Setup



User Management

1. All the authorization priviliges are stored in the rep:policy node
2. If a resource/node does not have a rep:ploicy node then it will refer to the parents rep:policy node

Using Events, ObserverationManager, Wrokflow

1. If events have to be trigerred from one class to other use Sling Event
2. If the event is to be handled on JCR Node or Property change user observation
3. For any other: use workflow (process, participant)

To configure pdf settings

Go to /libs/wcm/core/content/pdf/page2fo.xsl and modify the xsl

Tenats of Content Modelling

1. No Node types
2. Use nt:unstructured
3. Pref Mixin over CND
4. Content hierarchy determines a lot of things. Every bit of content should be well thought about

5. Don't create a workspace
6. NO same name siblings - use rules to name content nodes
7. Use weak ref use URI of the related content
8. Put files in folders
9. Don't use nt:binary
10. Don't assign an id to a node it’s not needed use the node URI instead

Comparing CQ versions

Using the COI tool to get bundle comparison from different versions of CQ:
http://dev.day.com/content/dam/day/onlinetool/COI.html

Slign Dynamic Includes

http://www.cognifide.com/blogs/cq/sling-dynamic-include/