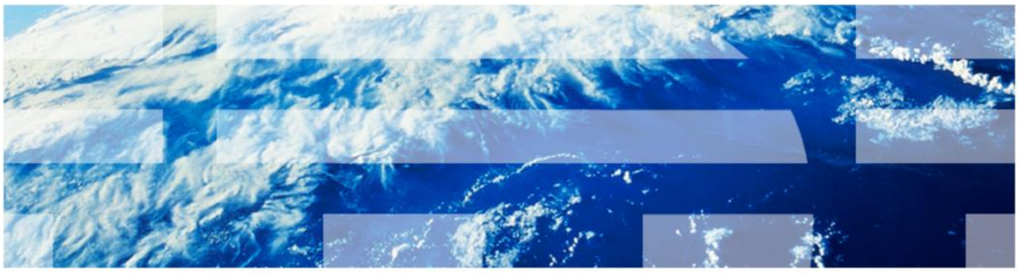
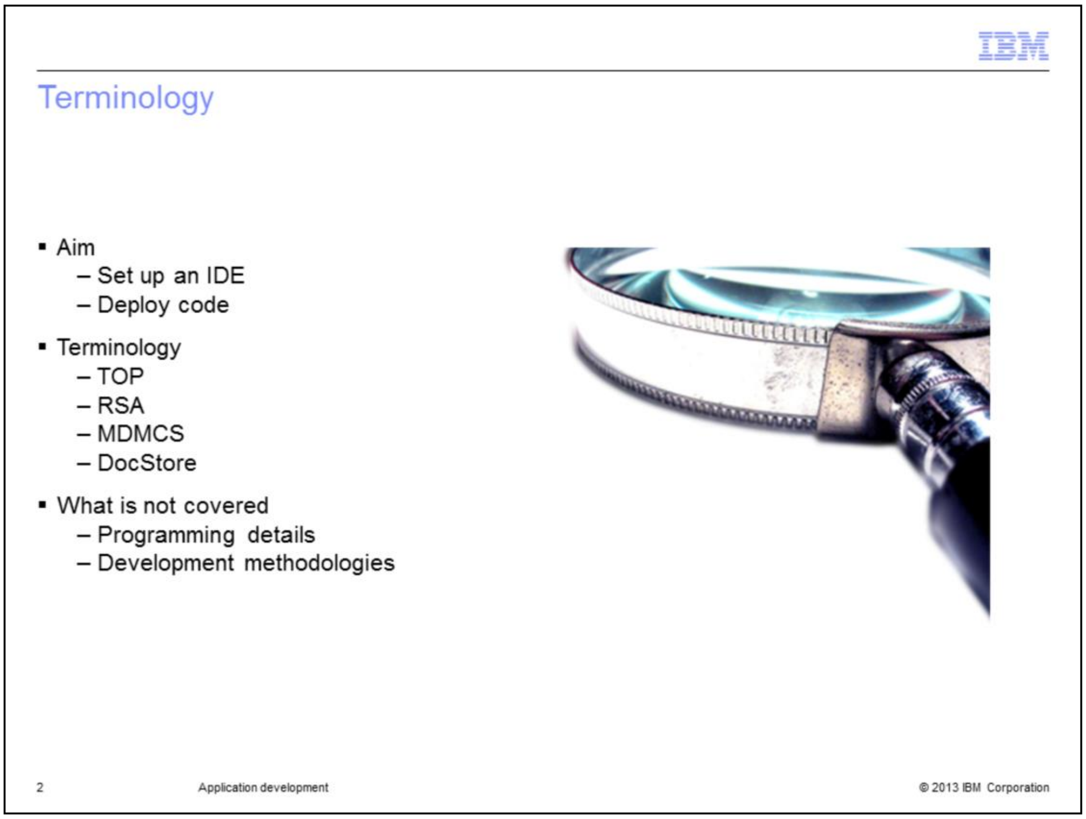
InfoSphere Master Data Management Collaboration Server V10

Application development



The topic of this presentation is how to develop custom applications and integrate custom code with IBM InfoSphere® Master Data Management Collaboration Server version 10. This is important because it will allow you to develop a custom solution that is ideal for your business case. IBM InfoSphere Master Data Management Collaboration Server is referred to as InfoSphere MDMCS throughout this presentation.



Terminology

- Aim
  - Set up an IDE
  - Deploy code
- Terminology
  - TOP
  - RSA
  - MDMCS
  - DocStore
- What is not covered
  - Programming details
  - Development methodologies

2 Application development © 2013 IBM Corporation

The aim of this presentation is to give you an overview of how to develop and deploy custom applications compatible with InfoSphere MDMCS. The steps necessary to configure your integrated development environment, or IDE, is reviewed.

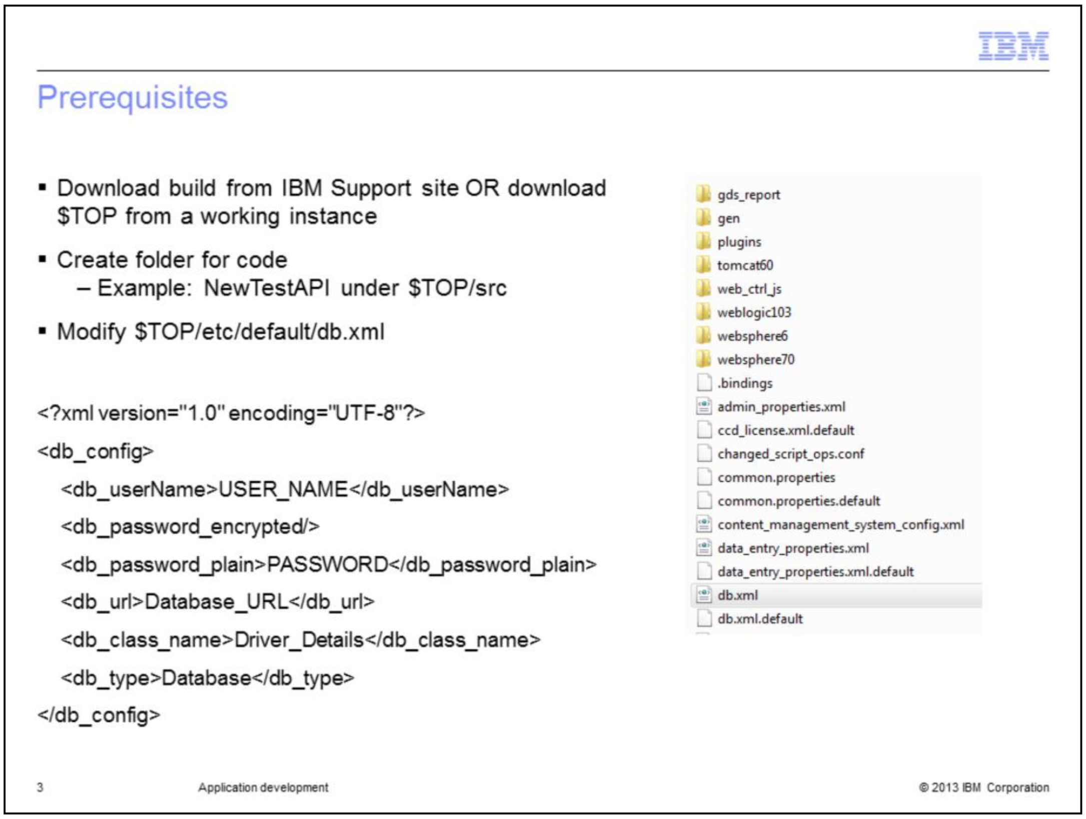
Before continuing, there are some terms you need to be familiar with.

TOP is an environment variable which points to the installation directory of the product.

Rational® Software Architect, or RSA, is IBM's modeling and development environment built on Eclipse open-source software framework. It can be used for designing architecture for Java 2 Enterprise Edition (J2EE) or C++ applications and web services. Note that RSA is the only Eclipse platform that the product is tested with and hence, it is the only one supported.

DocStore is an abbreviation for document store which is used by the application to manage all incoming and outgoing files, including import feeds, scripts, reports, and specs.

This presentation is not meant to help you with programming specifics or development methodologies. Those are beyond the scope of this presentation.



Prerequisites

- Download build from IBM Support site OR download $TOP from a working instance
- Create folder for code
  - Example: NewTestAPI under $TOP/src
- Modify $TOP/etc/default/db.xml

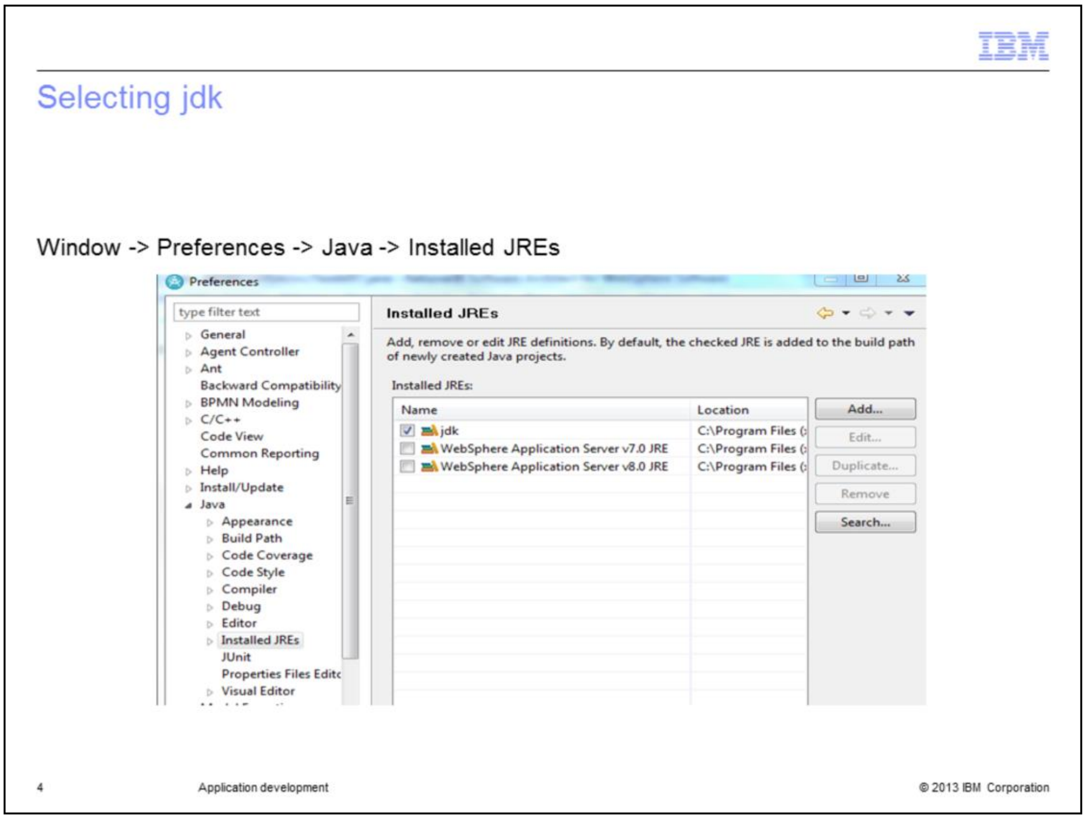
```
<?xml version="1.0" encoding="UTF-8"?>
<db_config>
   <db_userName>USER_NAME</db_userName>
   <db_password_encrypted/>
   <db_password_plain>PASSWORD</db_password_plain>
   <db_url>Database_URL</db_url>
   <db_class_name>Driver_Details</db_class_name>
   <db_type>Database</db_type>
</db_config>
```

```
gds_report
gen
plugins
tomcat60
web_ctrl_js
weblogic103
websphere6
websphere70
.bindings
admin_properties.xml
ccd_license.xml.default
changed_script_ops.conf
common.properties
common.properties.default
content_management_system_config.xml
data_entry_properties.xml
data_entry_properties.xml.default
db.xml
db.xml.default
```

3 Application development © 2013 IBM Corporation

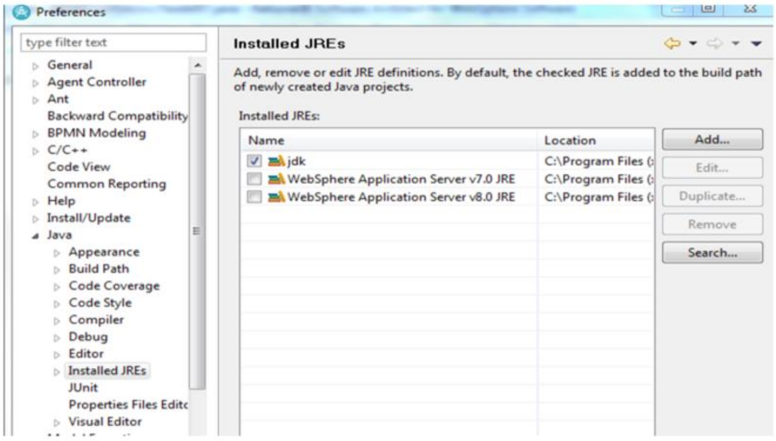
Before beginning the configuration, you must have the InfoSphere MDMCS build local to your machine. There are two ways to do this. You can download the same patch level from the IBM Support Site as the one you have deployed or you can download the entire $TOP structure directly from the instance which you want to connect to your Eclipse like environment. Either way, create a new folder under the $TOP/src directory for your local build to contain the .class files.

If you download the build from an IBM site, you will need to configure the $TOP/etc/default/db.xml file in your local build. This file should have the database connection details to be used by Eclipse to access the database. If you download $TOP, then this file should already be populated with the correct information. A sample db.xml is displayed on this slide.



Selecting jdk

Window -> Preferences -> Java -> Installed JREs



Installed JREs

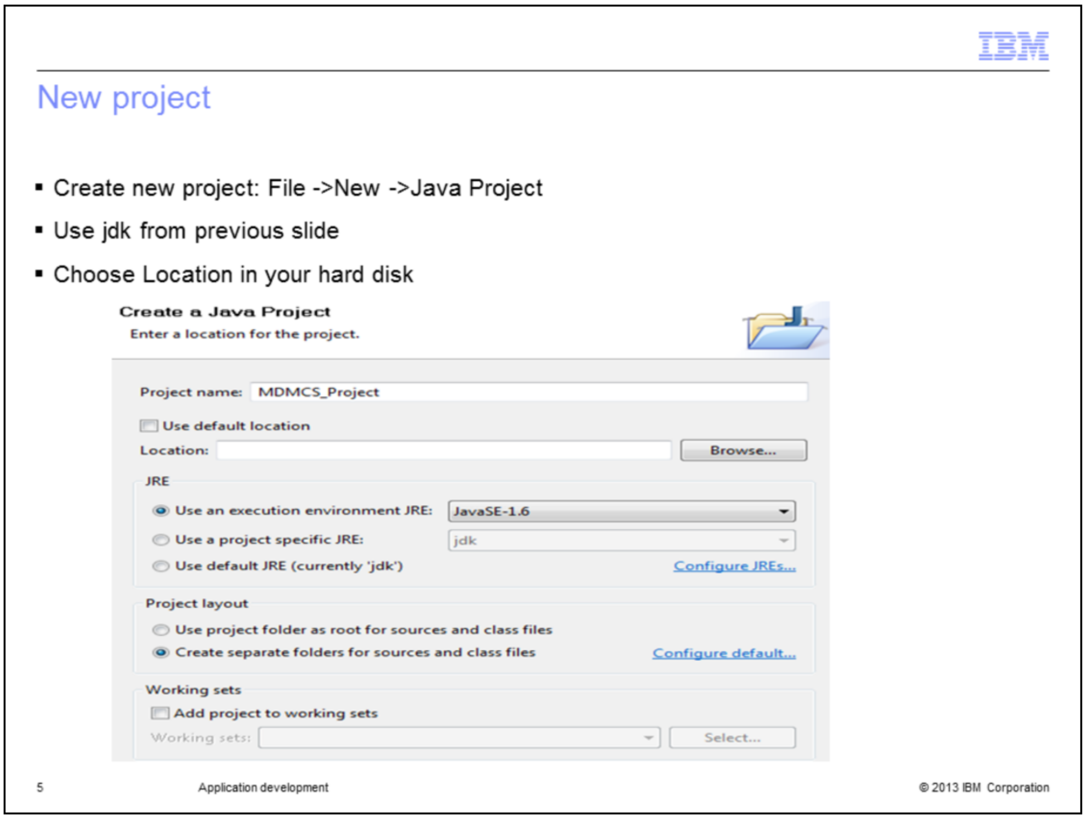
Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location
✓ jdk	C:\Program Files (:
WebSphere Application Server v7.0 JRE	C:\Program Files (:
WebSphere Application Server v8.0 JRE	C:\Program Files (:

4 Application development © 2013 IBM Corporation

To begin configuration, open RSA or Eclipse and create a new workspace to use. After creating the workspace, you need to configure the instance. Go to the “Installed JRE” prompt by following the path displayed on this slide and select the JRE which the workspace should use. It is recommended you use jdk 1.6 and using a jdk path which has no blank spaces in it such as avoiding directories like “Program Files”.



New project

- Create new project: File ->New ->Java Project
- Use jdk from previous slide
- Choose Location in your hard disk

Create a Java Project
Enter a location for the project.

Project name: MDMCS_Project

[ ] Use default location

Location: [Browse...]

JRE
[✓ Use an execution environment JRE: JavaSE-1.6]
Use a project specific JRE: jdk
Use default JRE (currently 'jdk') Configure JREs...

Project layout
Use project folder as root for sources and class files
[✓ Create separate folders for sources and class files] Configure default...

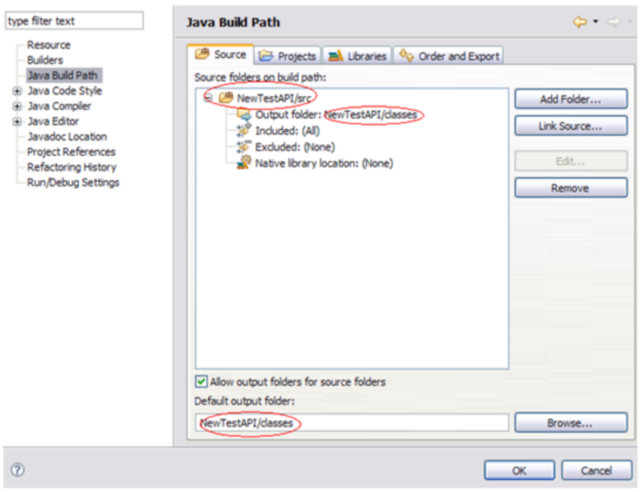
Working sets
[ ] Add project to working sets
Working sets: [Select...]

5 Application development © 2013 IBM Corporation

After selecting a jdk, create a new project by following the path displayed on this slide. Choose the previously added jdk and select the location in your hard disk where you want your project to be created. Click Finish to complete project setup.

Java build path (1 of 3)

- Right click created project and select Properties
- Select “Java Build Path” and configure Source Tab



Source folders on build path:

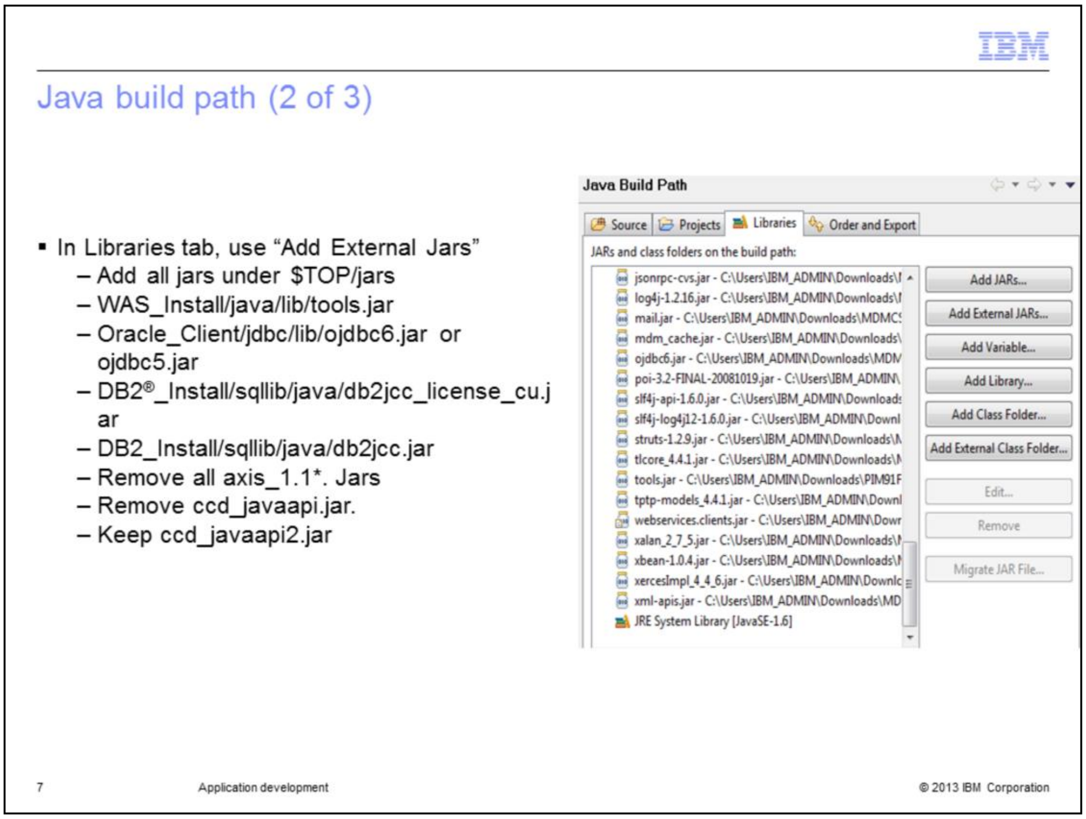
```
NewTestAPI/src
    Output folder: NewTestAPI/classes
    Included: (All)
    Excluded: (None)
    Native library location: (None)
```

☑ Allow output folders for source folders

Default output folder: NewTestAPI/classes

Next, you need to configure the run time properties of the workspace. Right click the name of the newly created project name in the left navigation pane and select Properties. A pop up box opens as displayed on this slide. Select “Java Build Path” and then configure the Source tab as displayed on this slide.

Tasks that need to be done include putting a check mark in the Allow output folders for source folders box. Under the source folders, select Output folder and Edit. Provide a new folder called classes through Specific output folder. In the Default output folder section, click Browse > Create New Folder. Specify the new folder name as classes and click OK. NewTestAPI refers to the new folder under $TOP/src directory that you created earlier. Finally, click Add Folder, select classes folder and click OK.



Java build path (2 of 3)

- In Libraries tab, use "Add External Jars"
  - Add all jars under $TOP/jars
  - WAS_Install/java/lib/tools.jar
  - Oracle_Client/jdbc/lib/ojdbc6.jar or ojdbc5.jar
  - DB2®_Install/sqllib/java/db2jcc_license_cu.jar
  - DB2_Install/sqllib/java/db2jcc.jar
  - Remove all axis_1.1*. Jars
  - Remove ccd_javaapi.jar.
  - Keep ccd_javaapi2.jar

Java Build Path

JARs and class folders on the build path:

```
jsonrpc-cvs.jar - C:\Users\IBM_ADMIN\Downloads\I
log4j-1.2.16.jar - C:\Users\IBM_ADMIN\Downloads\I
mail.jar - C:\Users\IBM_ADMIN\Downloads\MDMCS
mdm_cache.jar - C:\Users\IBM_ADMIN\Downloads\
ojdbc6.jar - C:\Users\IBM_ADMIN\Downloads\MDM
poi-3.2-FINAL-20081019.jar - C:\Users\IBM_ADMIN\
slf4j-api-1.6.0.jar - C:\Users\IBM_ADMIN\Downloads
slf4j-log4j12-1.6.0.jar - C:\Users\IBM_ADMIN\Downl
struts-1.2.9.jar - C:\Users\IBM_ADMIN\Downloads\N
tlcore_4.4.1.jar - C:\Users\IBM_ADMIN\Downloads\N
tools.jar - C:\Users\IBM_ADMIN\Downloads\PIM91F
tptp-models_4.4.1.jar - C:\Users\IBM_ADMIN\Downl
webservices.clients.jar - C:\Users\IBM_ADMIN\Dowr
xalan_2_7_5.jar - C:\Users\IBM_ADMIN\Downloads\N
xbean-1.0.4.jar - C:\Users\IBM_ADMIN\Downloads\N
xercesImpl_4_4_6.jar - C:\Users\IBM_ADMIN\Downlc
xml-apis.jar - C:\Users\IBM_ADMIN\Downloads\MD
JRE System Library [JavaSE-1.6]
```

7 Application development © 2013 IBM Corporation

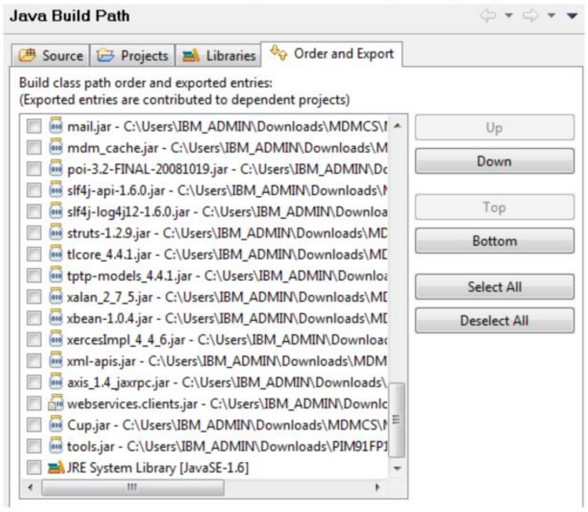
Go to the "Libraries" tab, click External Jars to add all the run time jars that the program will need to run and connect to the database. Add all the product jars located under the jars folder of the downloaded build.

The tools.jar and the database jars are not shipped with the product; you have to copy them from a working instance. Review the slide for the location of Oracle, DB2, and WebSphere® Application Server specific jars.

Ensure that you remove all instances of axis_1.1*.jar because InfoSphere MDMCS uses axis 1.4 and not 1.1. Also if both ccd_javaapi.jar and ccd_javaapi2.jar exist, remove ccd_javaapi.jar to avoid duplicate jars.

Java build path (3 of 3)

In “Order and Export” tab, make sure ‘JRE System Library’ entry is pushed to bottom

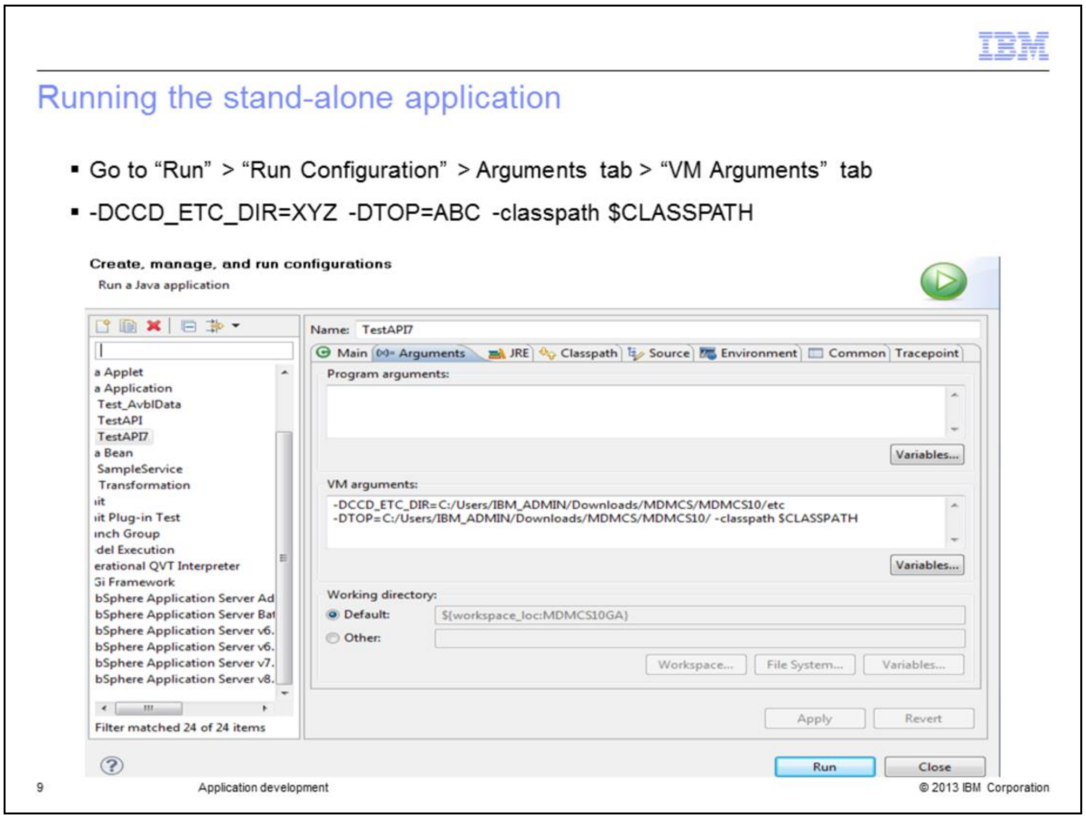


Build class path order and exported entries:
(Exported entries are contributed to dependent projects)

```
mail.jar - C:\Users\IBM_ADMIN\Downloads\MDMCS\I
mdm_cache.jar - C:\Users\IBM_ADMIN\Downloads\M
poi-3.2-FINAL-20081019.jar - C:\Users\IBM_ADMIN\Dc
slf4j-api-1.6.0.jar - C:\Users\IBM_ADMIN\Downloads\I
slf4j-log4j12-1.6.0.jar - C:\Users\IBM_ADMIN\Downloa
struts-1.2.9.jar - C:\Users\IBM_ADMIN\Downloads\MD
tlcore_4.4.1.jar - C:\Users\IBM_ADMIN\Downloads\MD
tptp-models_4.4.1.jar - C:\Users\IBM_ADMIN\Downloa
xalan_2_7_5.jar - C:\Users\IBM_ADMIN\Downloads\MD
xbean-1.0.4.jar - C:\Users\IBM_ADMIN\Downloads\MD
xercesImpl_4_4_6.jar - C:\Users\IBM_ADMIN\Downloac
xml-apis.jar - C:\Users\IBM_ADMIN\Downloads\MDM
axis_1.4_jaxrpc.jar - C:\Users\IBM_ADMIN\Downloads\
webservices.clients.jar - C:\Users\IBM_ADMIN\Downlc
Cup.jar - C:\Users\IBM_ADMIN\Downloads\MDMCS\I
tools.jar - C:\Users\IBM_ADMIN\Downloads\PIM91FP1
JRE System Library [JavaSE-1.6]
```

8 Application development © 2013 IBM Corporation

The last tab in this screen is the “Order and Export” tab. Ensure that the jars added in Libraries tab show up here and that JRE System library is pushed all the way to the bottom. This order determines how the jars are searched and classes executed.



Running the stand-alone application

- Go to “Run” > “Run Configuration” > Arguments tab > “VM Arguments” tab
- -DCCD_ETC_DIR=XYZ -DTOP=ABC -classpath $CLASSPATH

Create, manage, and run configurations

Run a Java application

Name: TestAPI7

Program arguments:

VM arguments:

```
-DCCD_ETC_DIR=C:/Users/IBM_ADMIN/Downloads/MDMCS/MDMCS10/etc
-DTOP=C:/Users/IBM_ADMIN/Downloads/MDMCS/MDMCS10/ -classpath $CLASSPATH
```

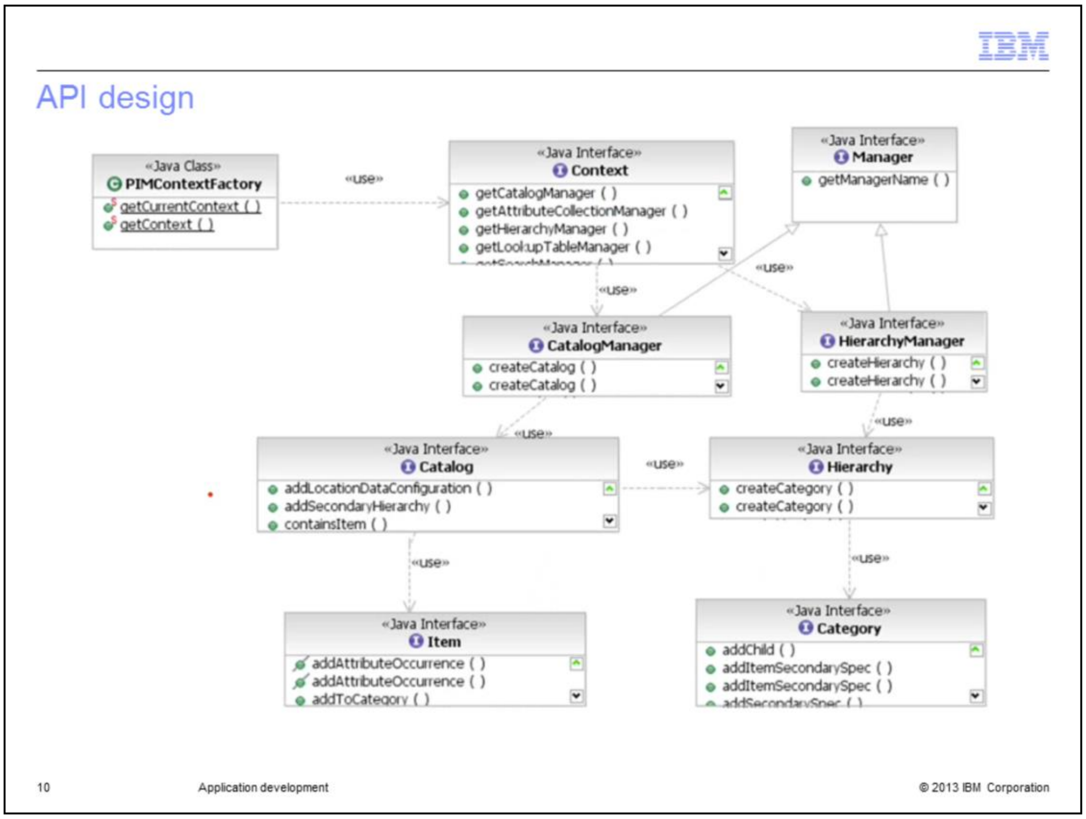
Working directory:

Default: ${workspace_loc:MDMCS10GA}

Filter matched 24 of 24 items

To configure run configurations, go to “Run” in the top panel and from the dropdown select “Run Configuration”. A Run dialog box will come up. Go to the Arguments tab and enter the value for VM Arguments as shown in the slide. These should point to the location of $TOP and $TOP/etc/default in your hard disk. Make sure you do not have any spaces before or after the ‘=' above.

Make sure that you configure run parameters properly before compiling a program.

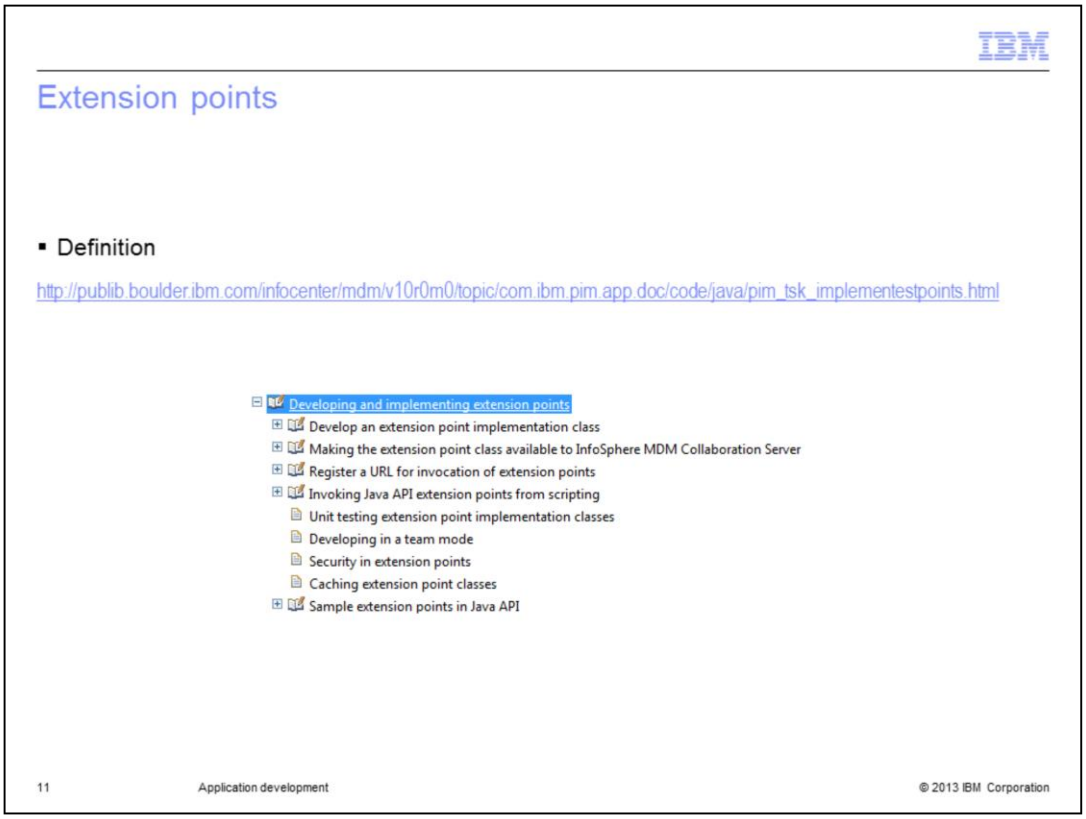


API design

10 Application development © 2013 IBM Corporation

This slide displays a high level overview of the API design. As shown, you can use methods in the PIMContextFactory to get the context and using that context, you can get a reference to catalog manager, lookup table manager, hierarchy manager, and so on. Using these references, you can get catalogs, items, attributes, and more. This is how you can add, edit, or delete data or data structures using code and without logging into the web user interface.

All available functions are documented in the Information Center.



Extension points

- Definition

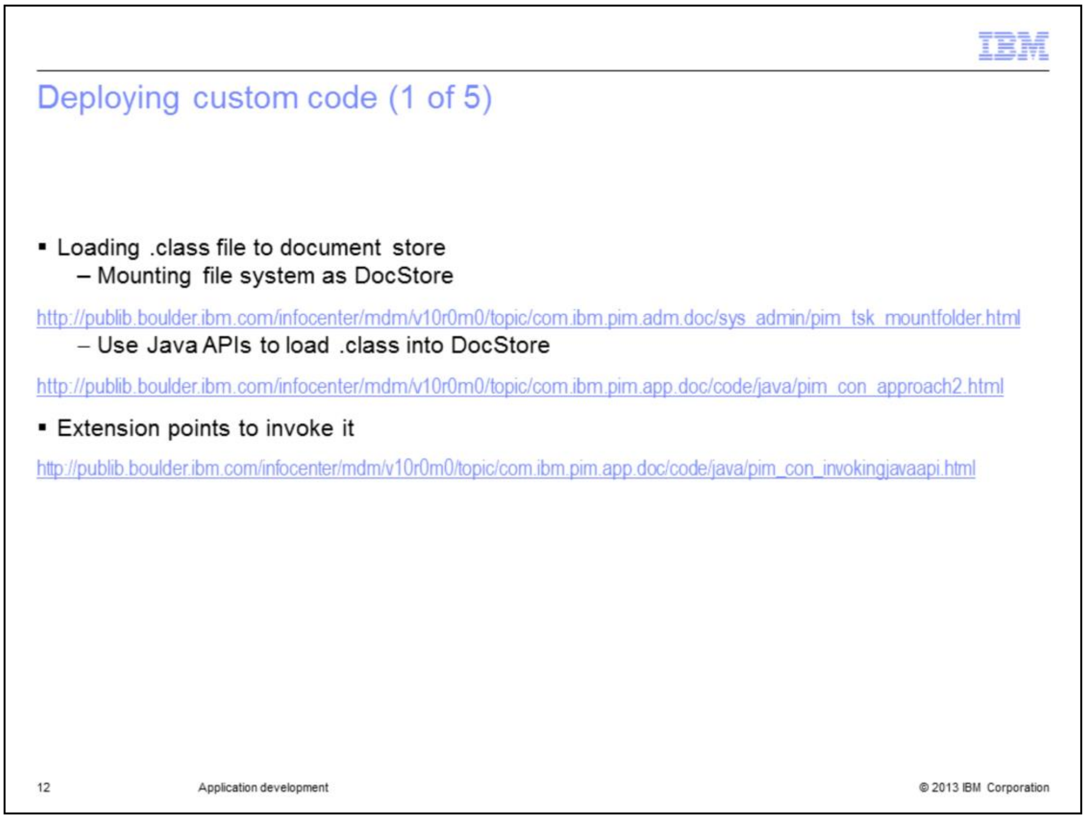
http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.app.doc/code/java/pim_tsk_implementestpoints.html

- Developing and implementing extension points
  - Develop an extension point implementation class
  - Making the extension point class available to InfoSphere MDM Collaboration Server
  - Register a URL for invocation of extension points
  - Invoking Java API extension points from scripting
  - Unit testing extension point implementation classes
  - Developing in a team mode
  - Security in extension points
  - Caching extension point classes
  - Sample extension points in Java API

11 Application development © 2013 IBM Corporation

An *extension point* is a point in the application where custom code can be invoked, such as entry preview script, post save script, and validation rule script. They are the various points within InfoSphere MDMCS where you can modify the behavior by running some user defined business logic. More than 20 extension point interfaces are shipped with InfoSphere MDMCS. You can write a Java class to implement custom code and then invoke that class from any of the available extension points as per your business use case.

Review the Information Center for more details including how to develop extension points, invoking them within InfoSphere MDMCS, and view some samples.



Deploying custom code (1 of 5)

- Loading .class file to document store
  - Mounting file system as DocStore

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.adm.doc/sys_admin/pim_tsk_mountfolder.html

  - Use Java APIs to load .class into DocStore

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.app.doc/code/java/pim_con_approach2.html

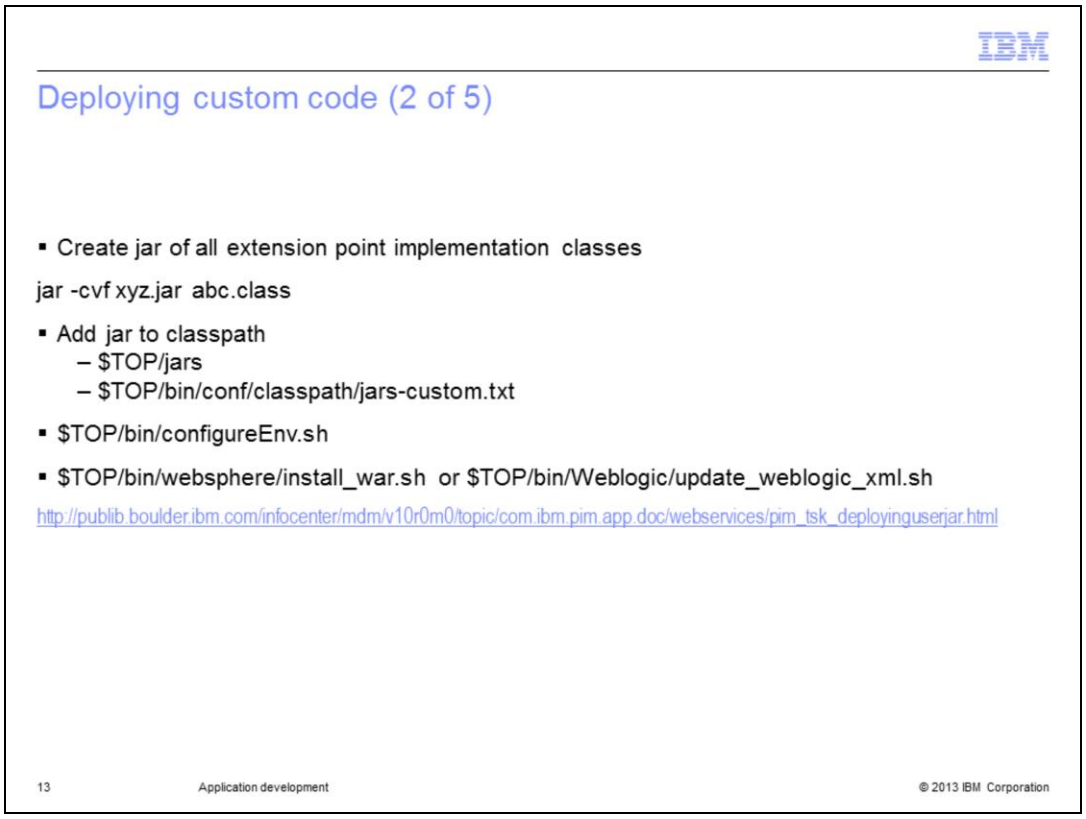
- Extension points to invoke it

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.app.doc/code/java/pim_con_invokingjavaapi.html

12 Application development © 2013 IBM Corporation

After compiling your code into a class or a jsp, you need to invoke it. There are four main ways of doing this. The first is to upload a Java class to the document store and then start it using an extension point.

You have two options for uploading files to the DocStore. The first is to mount the DocStore onto a directory in the server where InfoSphere MDMCS is installed. Once a DocStore is mounted, copying the file into the mounted directory of your server will upload it to the DocStore. The second option is to use Java API to upload the class to the DocStore. Once uploaded, the code can be invoked using an extension point. Multiple invocations and argument passing are supported in these extension points. Refer to the links displayed on the slide for more details on any of these options.



Deploying custom code (2 of 5)

- Create jar of all extension point implementation classes

```
jar -cvf xyz.jar abc.class
```

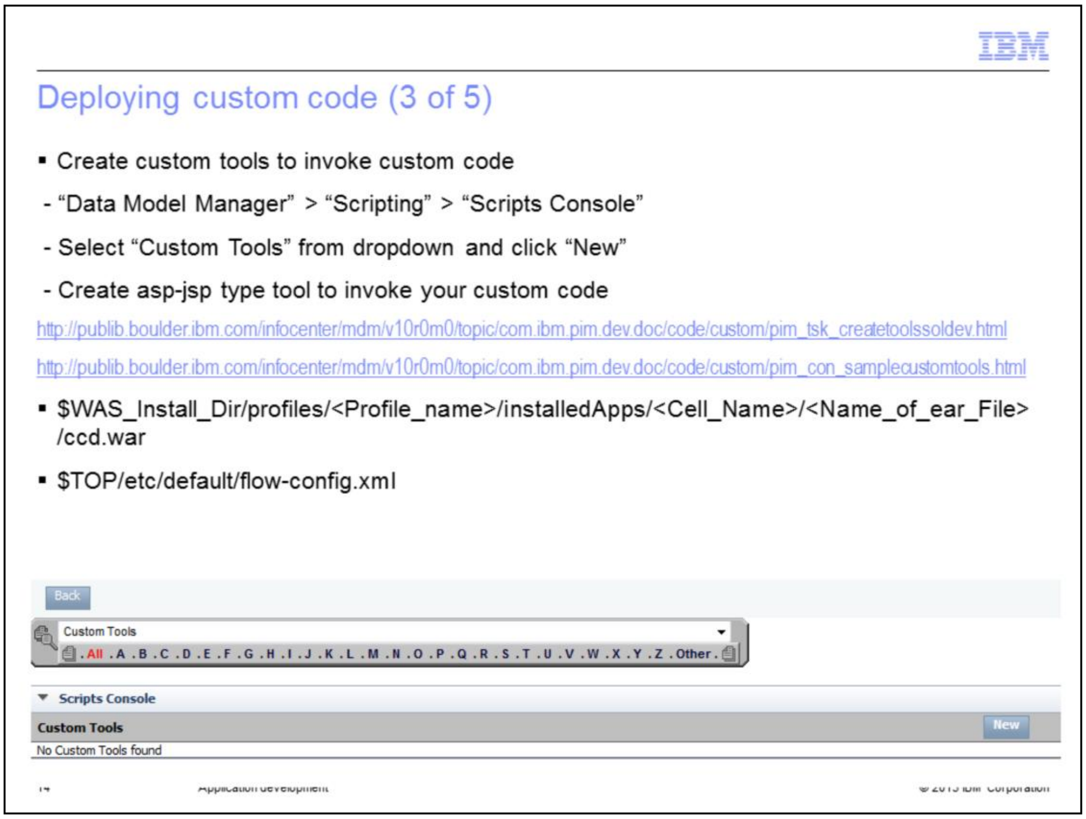
- Add jar to classpath
  - $TOP/jars
  - $TOP/bin/conf/classpath/jars-custom.txt
- $TOP/bin/configureEnv.sh
- $TOP/bin/websphere/install_war.sh or $TOP/bin/Weblogic/update_weblogic_xml.sh

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.app.doc/webservices/pim_tsk_deployinguserjar.html

13 Application development © 2013 IBM Corporation

The second way of deploying custom code is to create a jar of all the custom implementation classes. Then you can add the custom code to the InfoSphere MDMCS library to be used later. This slide contains the command to create a new jar or to add a class to an existing jar.

If you are deploying a new jar, it needs to be added to the classpath. Adding a jar to the classpath is a three step process. First, configure the application to recognize the location of this jar. You can do this by copying the jar to the $TOP/jars directory or by adding the location of the jar to the jars-custom.txt file. The second step is to update the classpath in InfoSphere MDMCS. You need to run configureEnv.sh in the $TOP/bin directory. Finally, you need to update the classpath in the application server. You can do this by running install_war.sh for WebSphere Application Server or update_weblogic_xml.sh for Weblogic. Now, your new jar and custom code has been deployed.



Deploying custom code (3 of 5)

- Create custom tools to invoke custom code
  - "Data Model Manager" > "Scripting" > "Scripts Console"
  - Select "Custom Tools" from dropdown and click "New"
  - Create asp-jsp type tool to invoke your custom code

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.dev.doc/code/custom/pim_tsk_createtoolssoldev.html

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.dev.doc/code/custom/pim_con_samplecustomtools.html

- $WAS_Install_Dir/profiles/<Profile_name>/installedApps/<Cell_Name>/<Name_of_ear_File>/ccd.war
- $TOP/etc/default/flow-config.xml

Back
Custom Tools
. All . A . B . C . D . E . F . G . H . I . J . K . L . M . N . O . P . Q . R . S . T . U . V . W . X . Y . Z . Other .
Scripts Console
Custom Tools
New

No Custom Tools found

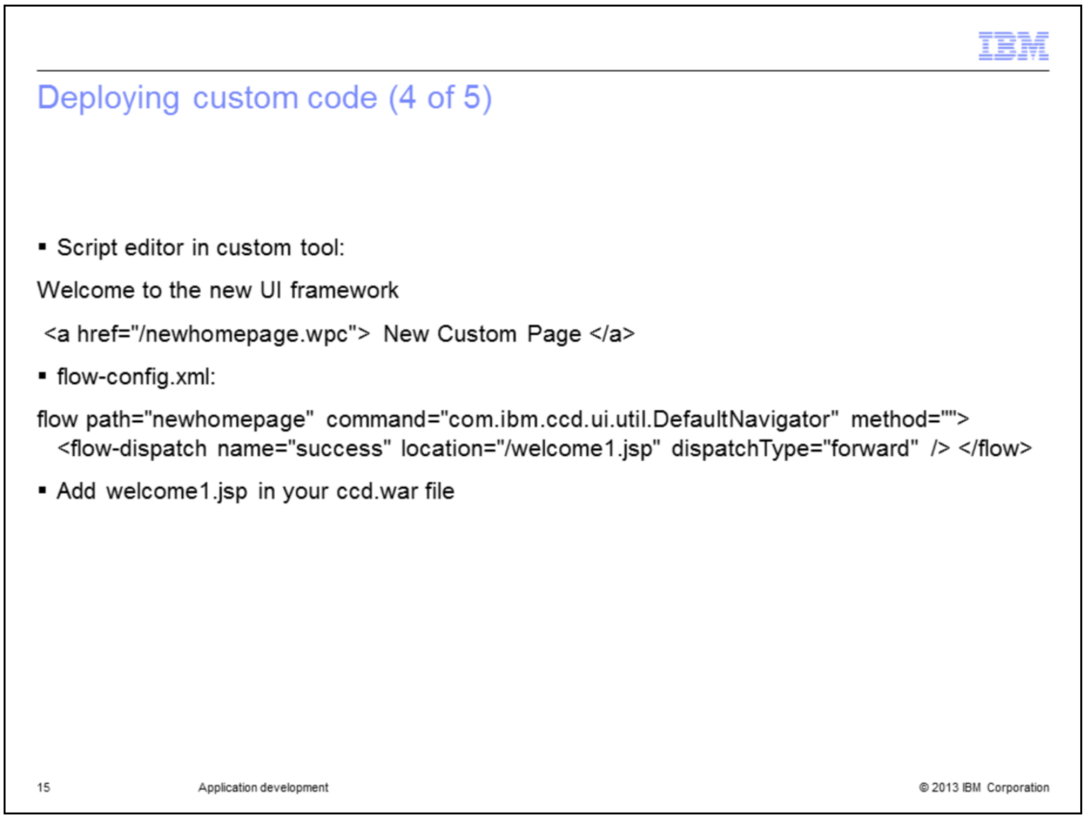
14 Application development © 2013 IBM Corporation

You can also build custom tools to deploy Java code. Custom tools provide a custom user interface and functionality that is not provided by the product by default. For example, you can use custom tools to present an item and category together on a screen when the user wants to view the item and category together on a screen or to display a custom audit trail of a user record.

To build your own custom tools, go to the Scripts Console and select "Custom Tools" from the drop down as displayed on this slide. Then, click "New" to define your tool. Specify any input parameters, provide a custom tool name, and select the ASP/JSP type.

In parallel, you need to define the code which is started by this tool. You do so by coding a stand-alone program in the form of an asp or jsp and then deploying it in the ccd.war file of the application server. The absolute path for a Websphere Application Server directory for ccd.war file is provided on this slide.

The flow-config.xml file is the core configuration file for the framework and it maps events in the web user interface to the function calls in the ccd.war file. For a custom tool to function properly, you need to map the code reference in the custom tool definition to the location of the jsp in the ccd.war file by way of a flow-config.xml file.



Deploying custom code (4 of 5)

- Script editor in custom tool:

Welcome to the new UI framework

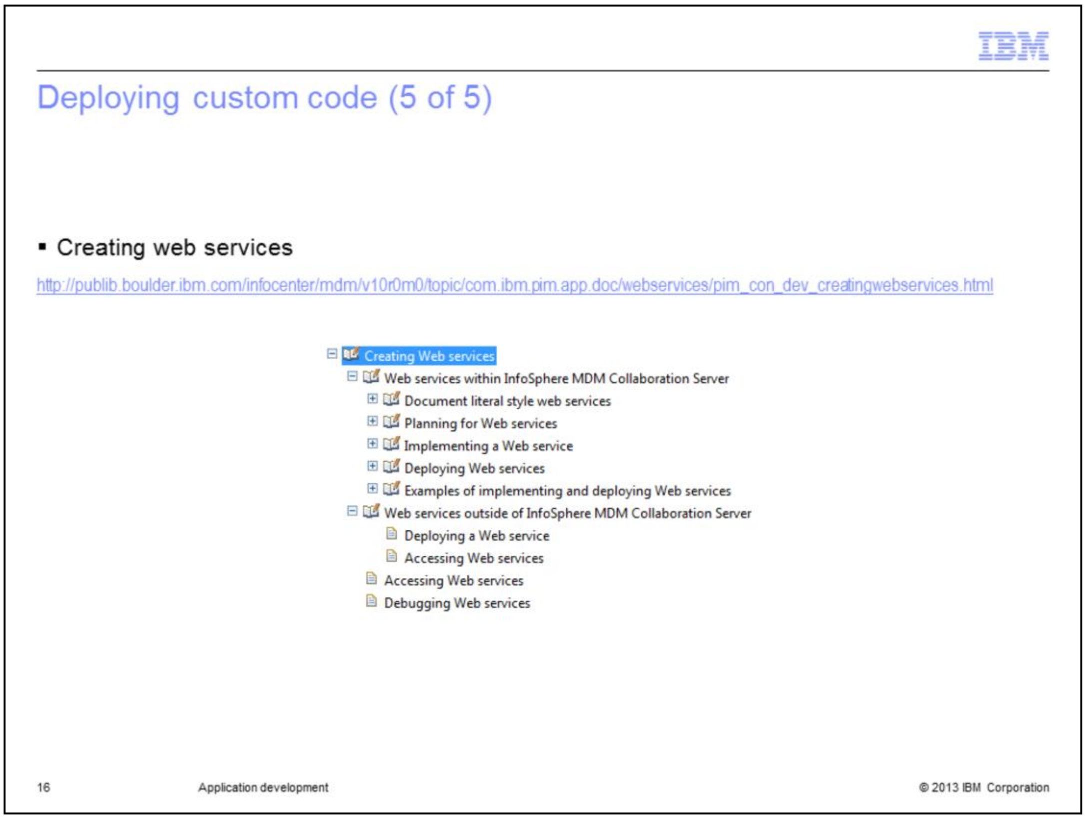
```
<a href="/newhomepage.wpc"> New Custom Page </a>
```

- flow-config.xml:

```
flow path="newhomepage" command="com.ibm.ccd.ui.util.DefaultNavigator" method="">
  <flow-dispatch name="success" location="/welcome1.jsp" dispatchType="forward" /> </flow>
```

- Add welcome1.jsp in your ccd.war file

This slide is a continuation of the last one to demonstrate the interdependencies. In this sample custom tool, you specify newhomepage.wpc as the target code in the definition. The system will then look in the flow-config.xml file to search for the location of the corresponding jsp. In this case, the jsp is named welcome1.jsp. The application will start the code within the jsp and the result is rendered on the web user interface. It should be noted that this is a purely custom application and IBM Support will not troubleshoot it.



Deploying custom code (5 of 5)

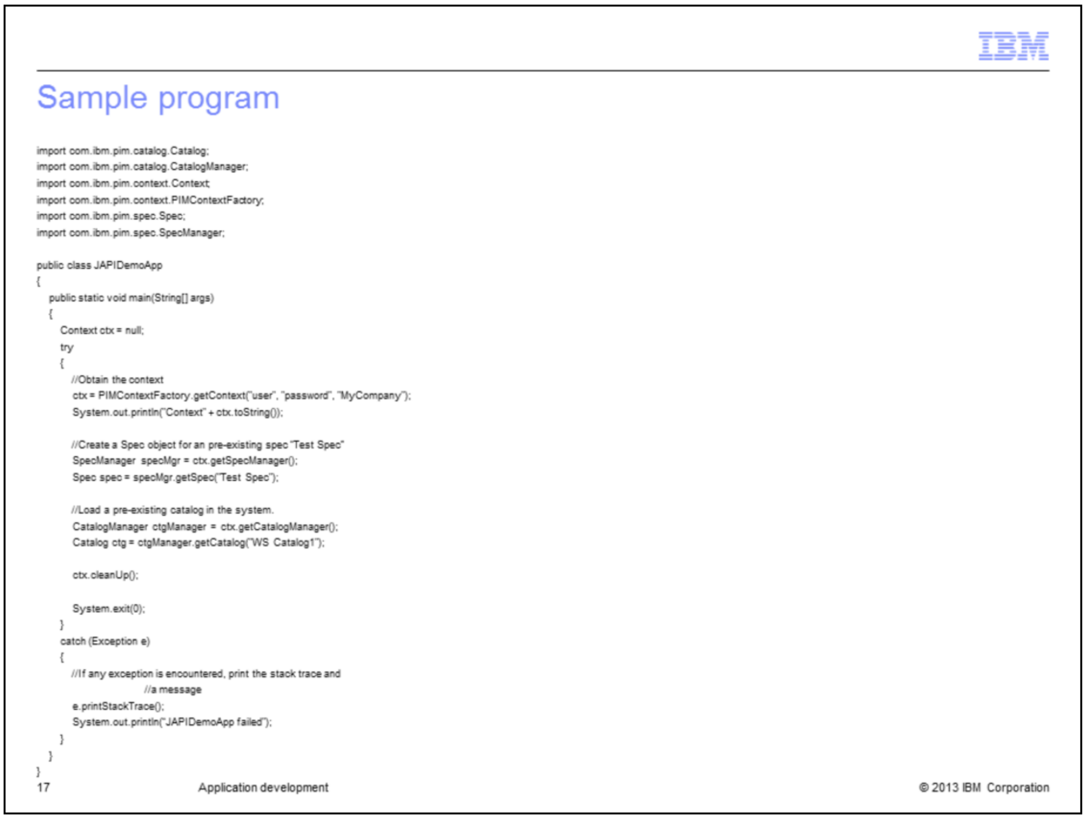
- Creating web services

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.app.doc/webservices/pim_con_dev_creatingwebservices.html

- Creating Web services
  - Web services within InfoSphere MDM Collaboration Server
    - Document literal style web services
    - Planning for Web services
    - Implementing a Web service
    - Deploying Web services
    - Examples of implementing and deploying Web services
  - Web services outside of InfoSphere MDM Collaboration Server
    - Deploying a Web service
    - Accessing Web services
  - Accessing Web services
  - Debugging Web services

16 Application development © 2013 IBM Corporation

The final method for deploying custom tools is web services. A web service is a method of communication between two electronic devices over the World Wide Web. It is a software system designed to support interoperable machine-to-machine interaction over a network. InfoSphere MDMCS supports web services with Axis standard 1.4 and some web services are also bundled with the product. Details of web services is beyond the scope of this presentation. Review other IBM Education Assistant modules or consult the Information Center documentation for more details. A direct link is provided on this slide along with the structure of the Information Center.

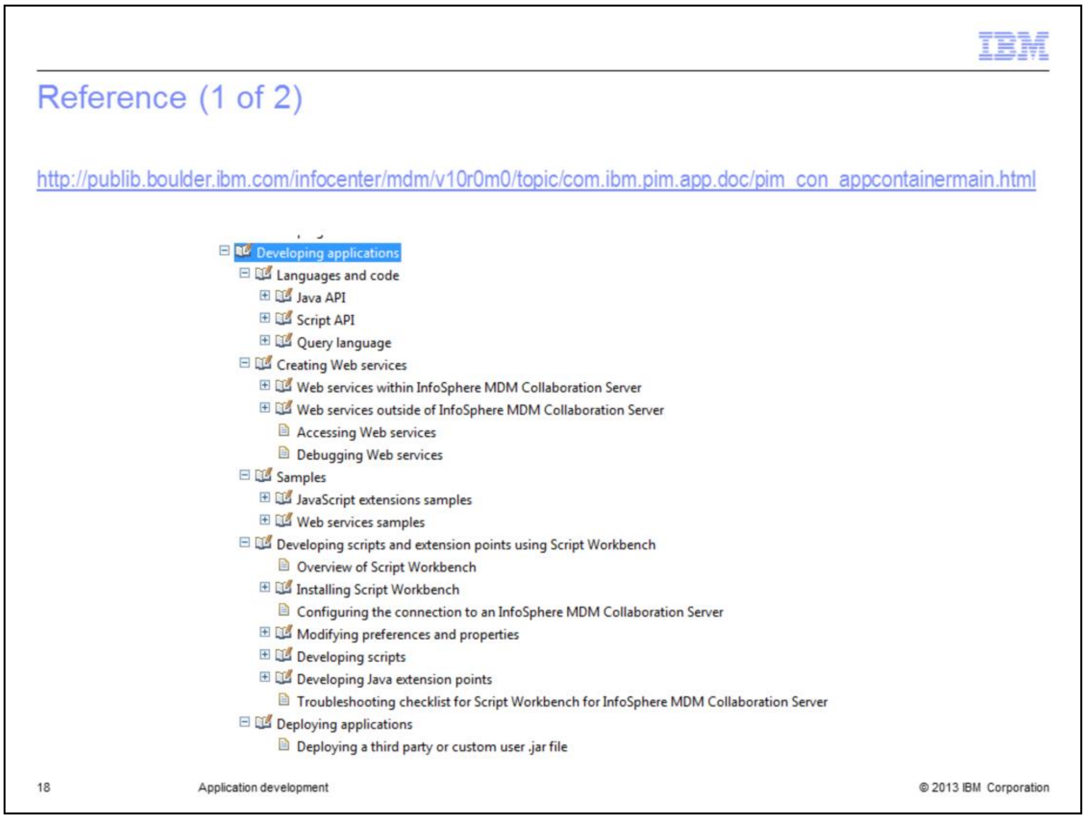


Sample program

```
import com.ibm.pim.catalog.Catalog;
import com.ibm.pim.catalog.CatalogManager;
import com.ibm.pim.context.Context;
import com.ibm.pim.context.PIMContextFactory;
import com.ibm.pim.spec.Spec;
import com.ibm.pim.spec.SpecManager;

public class JAPIDemoApp
{
   public static void main(String[] args)
   {
      Context ctx = null;
      try
      {
         //Obtain the context
         ctx = PIMContextFactory.getContext("user", "password", "MyCompany");
         System.out.println("Context" + ctx.toString());

         //Create a Spec object for an pre-existing spec "Test Spec"
         SpecManager  specMgr = ctx.getSpecManager();
         Spec spec = specMgr.getSpec("Test  Spec");

         //Load a pre-existing catalog in the system.
         CatalogManager ctgManager = ctx.getCatalogManager();
         Catalog ctg = ctgManager.getCatalog("WS  Catalog1");

         ctx.cleanUp();

         System.exit(0);
      }
      catch (Exception e)
      {
         //If any exception is encountered, print the stack trace and
                        //a message
         e.printStackTrace();
         System.out.println("JAPIDemoApp failed");
      }
   }
}
```

17 Application development © 2013 IBM Corporation

This slide displays a sample program which will loop through the database and can be used to get context, reference to a spec, and catalog.



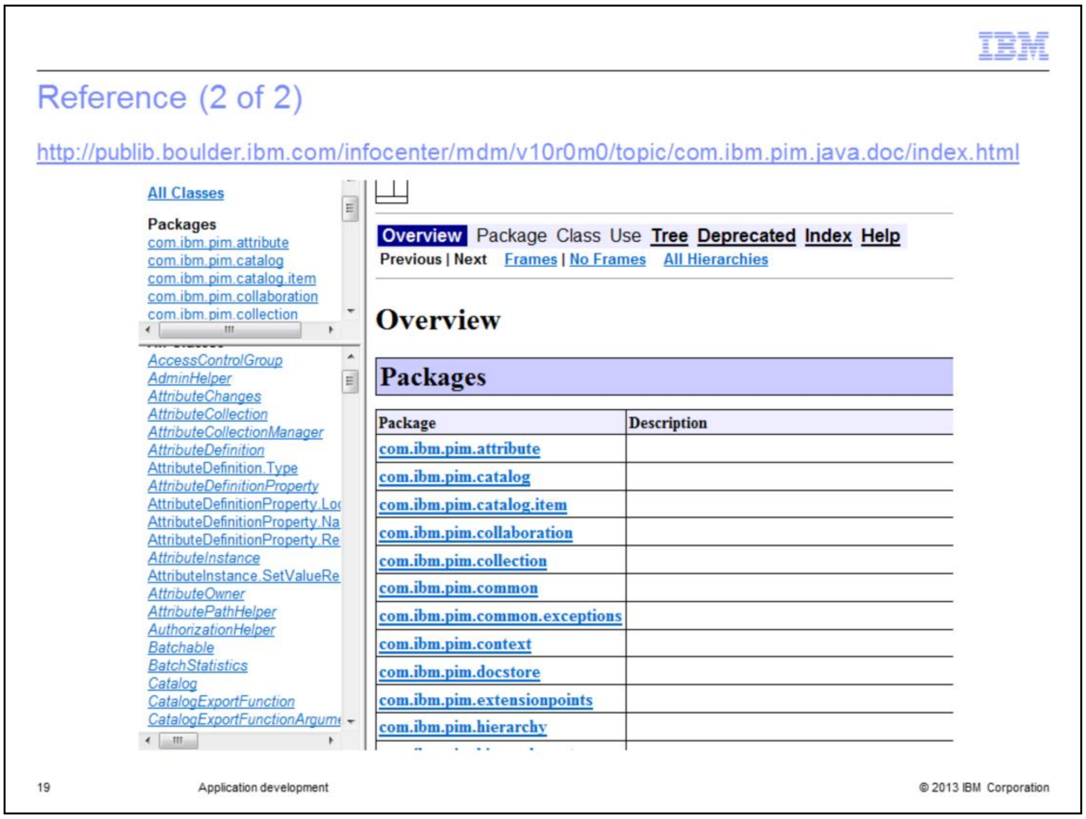
Reference (1 of 2)

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.app.doc/pim_con_appcontainermain.html

- Developing applications
  - Languages and code
    - Java API
    - Script API
    - Query language
  - Creating Web services
    - Web services within InfoSphere MDM Collaboration Server
    - Web services outside of InfoSphere MDM Collaboration Server
    - Accessing Web services
    - Debugging Web services
  - Samples
    - JavaScript extensions samples
    - Web services samples
  - Developing scripts and extension points using Script Workbench
    - Overview of Script Workbench
    - Installing Script Workbench
    - Configuring the connection to an InfoSphere MDM Collaboration Server
    - Modifying preferences and properties
    - Developing scripts
    - Developing Java extension points
    - Troubleshooting checklist for Script Workbench for InfoSphere MDM Collaboration Server
  - Deploying applications
    - Deploying a third party or custom user .jar file

18 Application development © 2013 IBM Corporation

The Information Center has extensive documentation to assist you in developing custom applications. Review the link on this slide and the structure of the documentation.



Reference (2 of 2)

http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.pim.java.doc/index.html

All Classes

Packages
com.ibm.pim.attribute
com.ibm.pim.catalog
com.ibm.pim.catalog.item
com.ibm.pim.collaboration
com.ibm.pim.collection

AccessControlGroup
AdminHelper
AttributeChanges
AttributeCollection
AttributeCollectionManager
AttributeDefinition
AttributeDefinition.Type
AttributeDefinitionProperty
AttributeDefinitionProperty.Lo
AttributeDefinitionProperty.Na
AttributeDefinitionProperty.Re
AttributeInstance
AttributeInstance.SetValueRe
AttributeOwner
AttributePathHelper
AuthorizationHelper
Batchable
BatchStatistics
Catalog
CatalogExportFunction
CatalogExportFunctionArgume

Overview Package Class Use Tree Deprecated Index Help
Previous | Next Frames | No Frames All Hierarchies

Overview

Packages

Package	Description
com.ibm.pim.attribute	
com.ibm.pim.catalog	
com.ibm.pim.catalog.item	
com.ibm.pim.collaboration	
com.ibm.pim.collection	
com.ibm.pim.common	
com.ibm.pim.common.exceptions	
com.ibm.pim.context	
com.ibm.pim.docstore	
com.ibm.pim.extensionpoints	
com.ibm.pim.hierarchy	

19 Application development © 2013 IBM Corporation

You can get an extensive list of available Java packages and classes functions from the Information Center. Follow the link displayed on this slide.

Trademarks, disclaimer, and copyright information