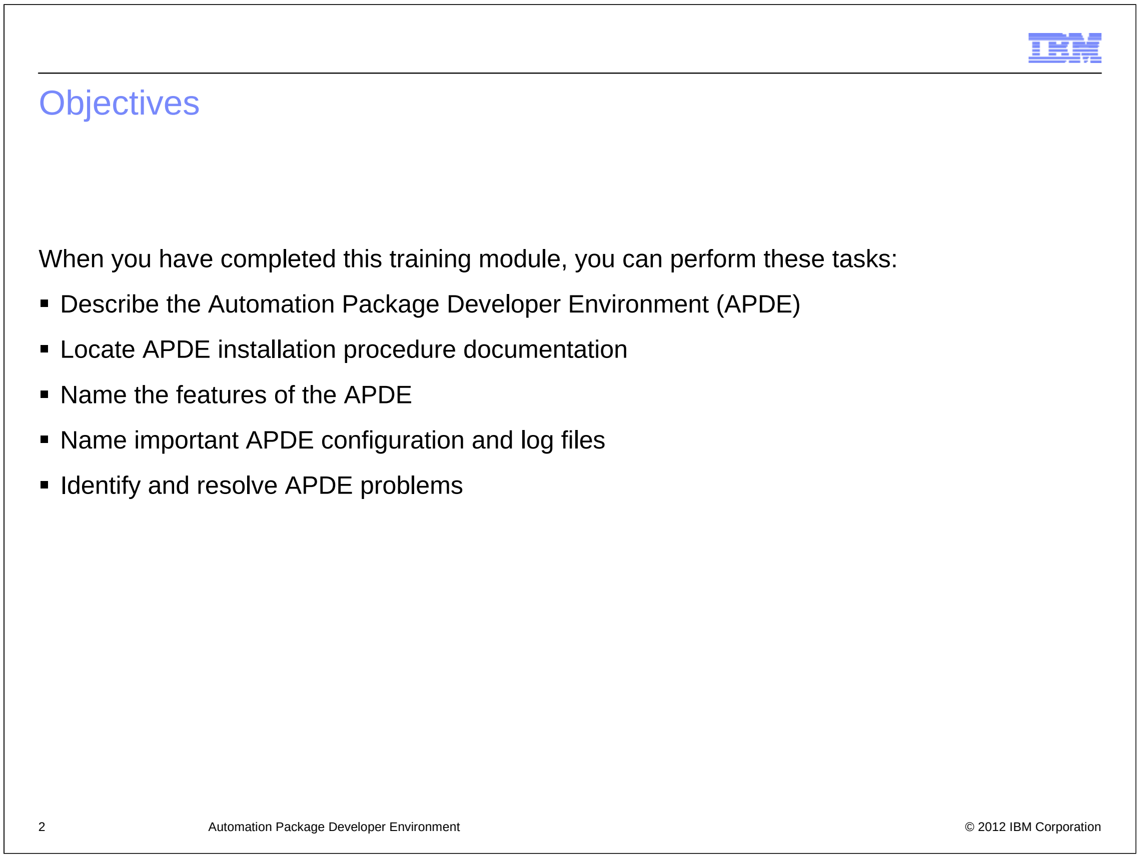
IBM Tivoli Provisioning Manager V7.2.1

Automation Package Developer Environment

In this training module, you learn about the Automation Package Developer Environment for IBM Tivoli Provisioning Manager V7.2.1.



Objectives

When you have completed this training module, you can perform these tasks:

- Describe the Automation Package Developer Environment (APDE)
- Locate APDE installation procedure documentation
- Name the features of the APDE
- Name important APDE configuration and log files
- Identify and resolve APDE problems

When you have completed this training module, you can perform these tasks:

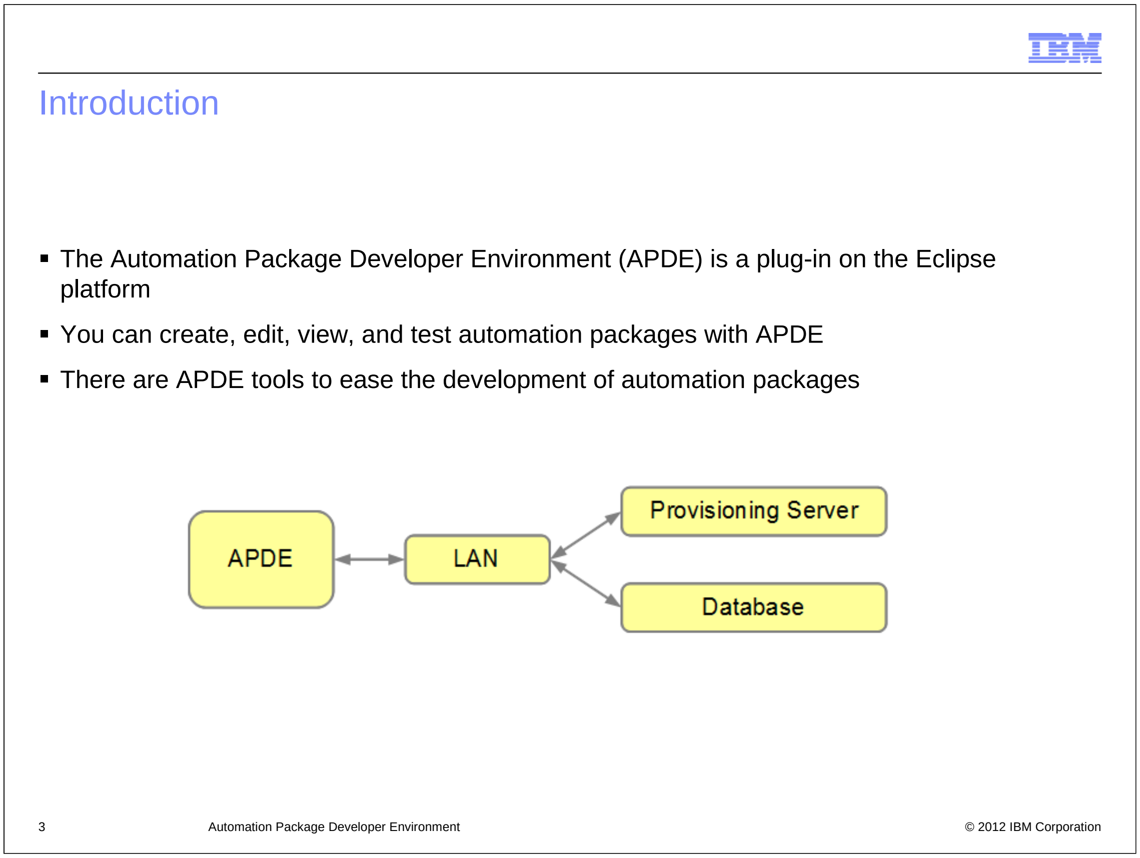
- Describe the Automation Package Developer Environment (APDE)

- Locate APDE installation procedure documentation

- Name the features of the APDE

- Name important configuration and log files

- Identify and resolve APDE problems



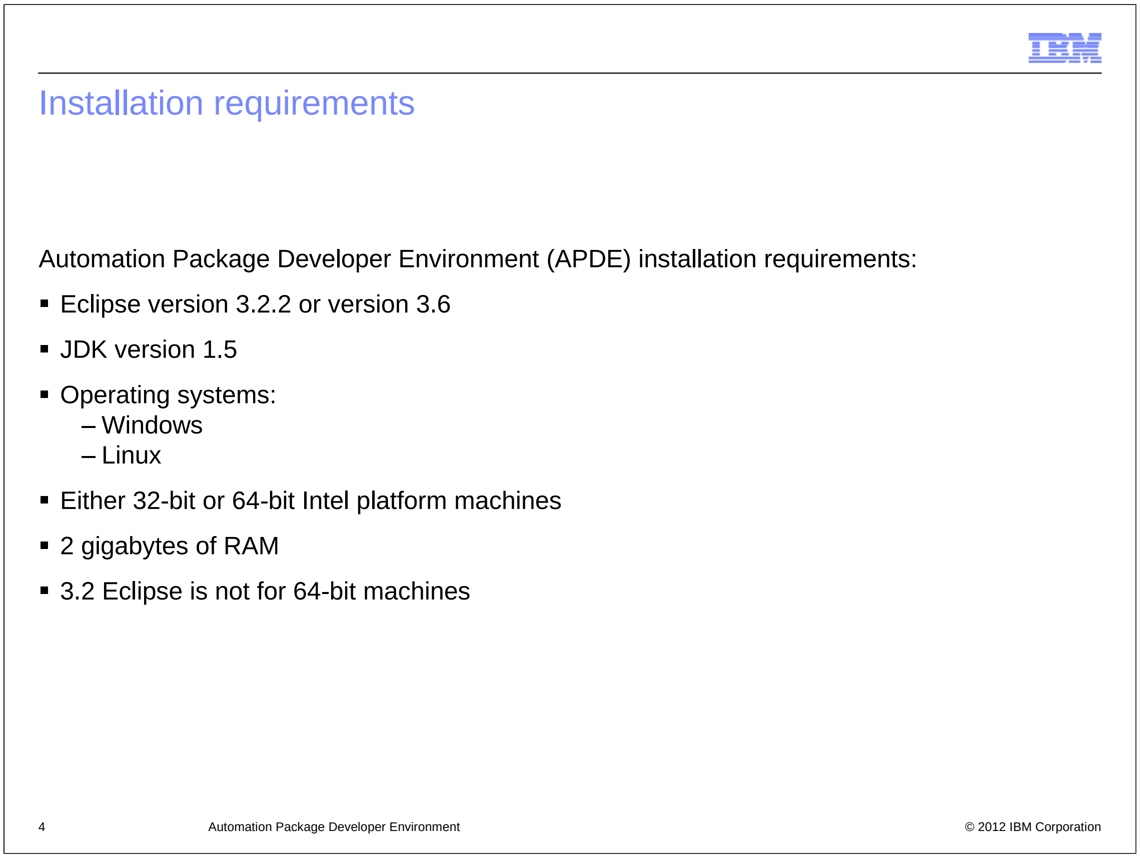
Introduction

- The Automation Package Developer Environment (APDE) is a plug-in on the Eclipse platform
- You can create, edit, view, and test automation packages with APDE
- There are APDE tools to ease the development of automation packages

APDE
LAN
Provisioning Server
Database

This slide shows a block diagram of the APDE, LAN, IBM Tivoli Provisioning Manager server, and a database server. The APDE remotely accesses both the provisioning server and the database server. There already exists a place in the IBM Tivoli Provisioning Manager user interface where you can write the tcdriver workflows without using an APDE.

You can complete the installation, compile and then run the workflows without an APDE. However, a formal IDE or application development environment is needed. The APDE is used when writing workflows and packages. The APDE allows developers to create, edit, and view workflows and packages in a more sophisticated way. The APDE is an Eclipse plug-in that is delivered with IBM Tivoli Provisioning Manager. The Eclipse editor is used to write code. Embedded Eclipse features are quite useful in development scenarios for those writing code.



Installation requirements

Automation Package Developer Environment (APDE) installation requirements:

- Eclipse version 3.2.2 or version 3.6
- JDK version 1.5
- Operating systems:
  - Windows
  - Linux
- Either 32-bit or 64-bit Intel platform machines
- 2 gigabytes of RAM
- 3.2 Eclipse is not for 64-bit machines

4 Automation Package Developer Environment © 2012 IBM Corporation

Lets review the APDE installation requirements:

- Eclipse version 3.2.2 or version 3.6

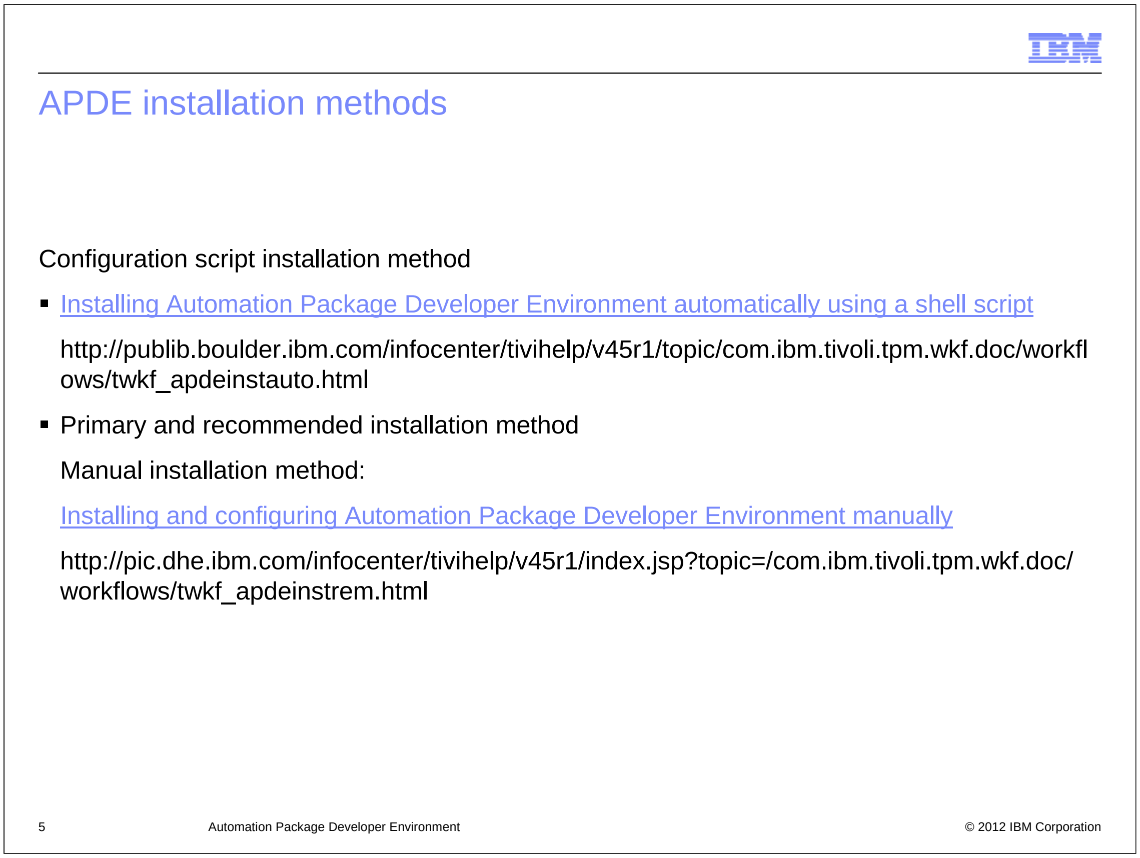
- JDK version 1.5

- 2 gigabytes of RAM

- Either Windows or Linux operating system

- Either a 32-bit or a 64-bit Intel platform machine

As a mental note remember that 3.2.2 Eclipse is not for 64-bit machines



APDE installation methods

Configuration script installation method

- Installing Automation Package Developer Environment automatically using a shell script

  http://publib.boulder.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_apdeinstauto.html

- Primary and recommended installation method

  Manual installation method:

  Installing and configuring Automation Package Developer Environment manually

  http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/index.jsp?topic=/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_apdeinstrem.html

The first step is to log in on the IBM Tivoli Provisioning Manager provisioning server to create the APDE package. After creating the APDE package, the next step is to deploy it on a remote workstation. Download the eclipse compressed file. It can be either version 3.2.2 or version 3.6. The IBM JDK version 1.5 is included with the IBM Tivoli Provisioning Manager package. The 32-bit JDK package is not supported for 64-bit operating systems. There are two methods of installation. They are the manual installation method and the configuration script installation method. The manual installation method is discussed in the IBM Information Center. The web address is: http://publib.boulder.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_apdeinstrem.html

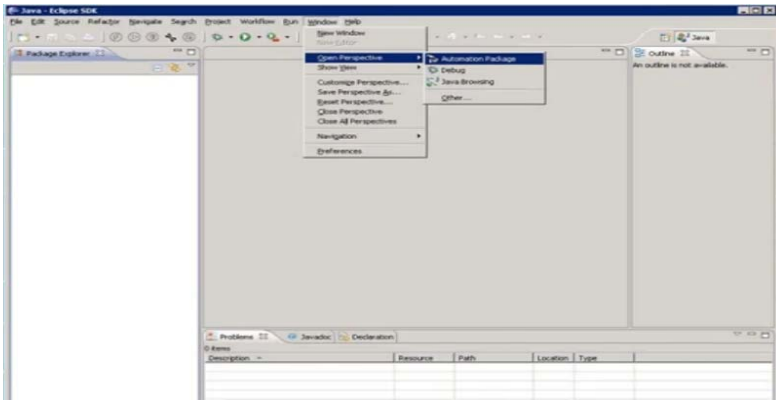
It is much easier to accomplish an APDE installation by using the configuration script provided with IBM Tivoli Provisioning Manager. The web address for the configuration script installation method is: http://publib.boulder.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_apdeinstauto.html

The configuration script is located under **TIO_HOME/apde**. All of the related installation packages are also under **TIO_HOME/apde**. The **Configure_APDE.sh** script can be used to generate the installation package for both Windows and Linux operating systems. The configuration script requires the APDE_install_location. The recommended directory path for Windows is **C:\APDE**. This directory must be created before deploying the package on the remote workstation. Additional parameters required for the **Configure_APDE.sh** script are the remote machine name, the username to connect to the remote machine, whether the eclipse version is 3.2.2 or 3.6, the eclipse_path, and the JDK_path. The eclipse_path and the JDK_path are the locations of the eclipse and Java compressed files on the IBM Tivoli Provisioning Manager server. The **Configure_APDE.sh** script extracts the eclipse and the Java files. It builds eclipse and Java files into a single APDE installation file for the target workstation. When the **Configure_APDE** script is run without any syntax it displays an informational message describing all the available parameters and usage.

At the completion of the **Configure_APDE.sh** script run, a compressed package is created. The compressed package is now ready to be deployed on a remote workstation. The compressed package is only created and not automatically copied and deployed on the remote workstation if the installer did not specify a machine name, username, and proper command syntax. In this scenario the installer must manually complete the installation on the remote workstation by copying the compressed file and extracting it on the remote workstation.

Starting APDE and opening the automation package

- Start Automation Package Developer Environment (APDE) using **eclipseLauncher.bat** (present in the **<APDE_HOME>/eclipse** folder)
- To load the automation package perspective, click **Window** > **Open Perspective** > **Automation Package**



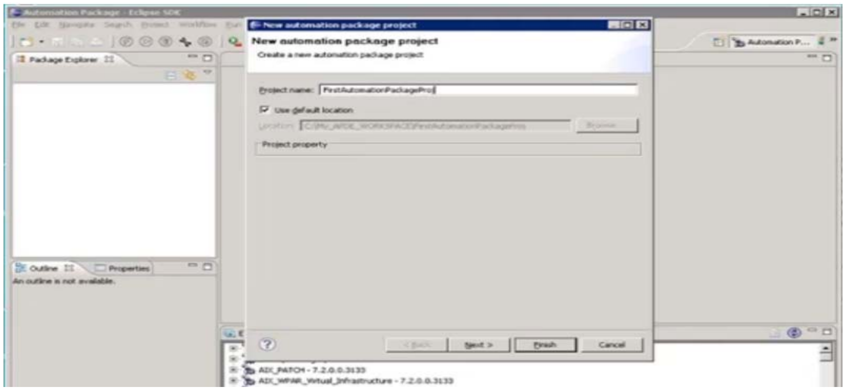
An outline is not available.

6 Automation Package Developer Environment

The APDE is now deployed on your workstation. Start the APDE by using the **eclipseLauncher.bat** file located in the **APDE_home/** folder. Next, you are prompted to define the APDE workspace. The APDE workspace is a user-specified location where the eclipse workspace files are stored. Open the automation package perspective after the APDE starts successfully. From the **Window** menu, select **Open Perspective** and select **Automation Package**. This activates the automation package plug-in within the eclipse environment. Installers see additional view tabs that display example workflows and packages after the perspective has fully loaded.

Creating an automation package project

- Create a sample automation package
- Click **File** > **New** > **Automation Package Project**



New automation package project

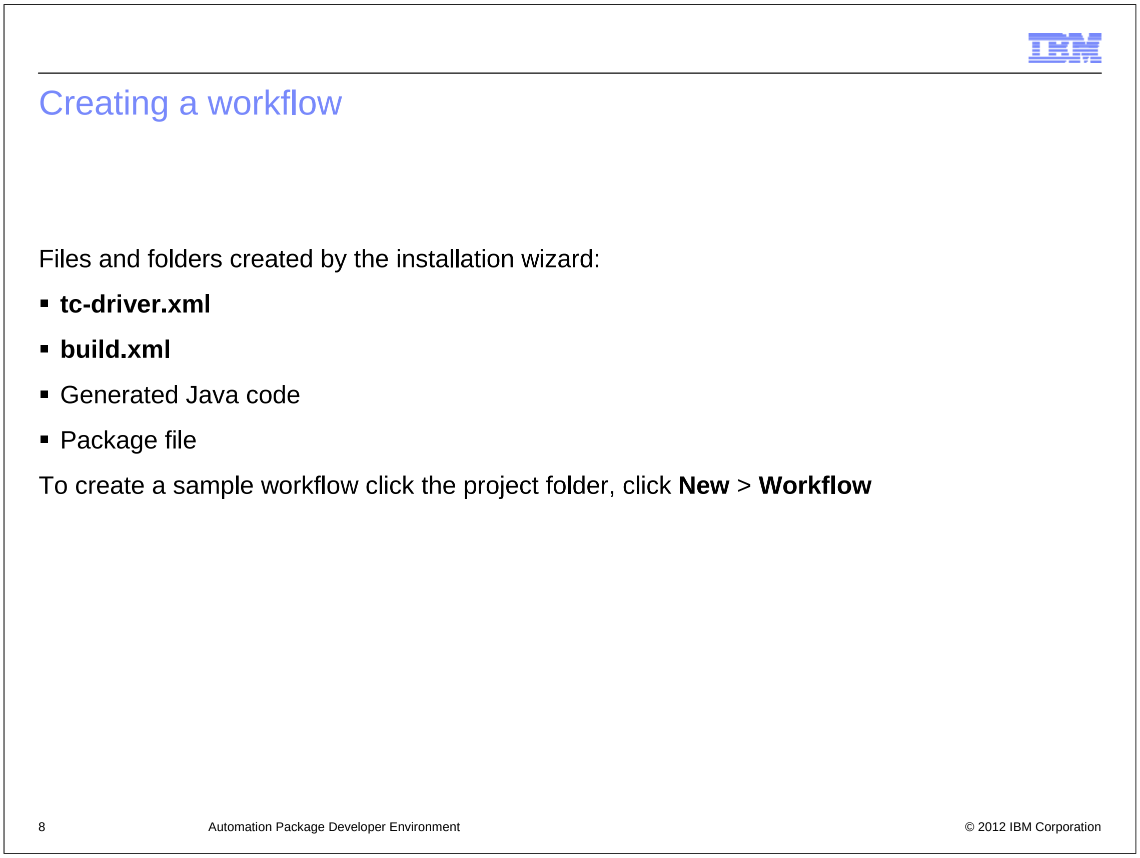
Create a new automation package project

Project name: FirstAutomationPackageProj

☑ Use default location

Location: C:\My_APDE_WORKSPACE\FirstAutomationPackageProj

Begin a new automation package project by selecting **File**>**New**>**Automation Package Project**. An empty project is created on the left panel of the screen.



Creating a workflow

Files and folders created by the installation wizard:

- **tc-driver.xml**
- **build.xml**
- Generated Java code
- Package file

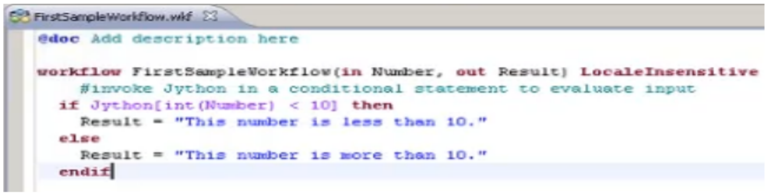
To create a sample workflow click the project folder, click **New** > **Workflow**

In this slide, you see names of the files and folders that are created by the installation wizard. These files and folders are needed when creating workflows. To create a new workflow, right-click the new project folder of the new automation package. Next, select **New** > **Workflow**. Specify a name for the new workflow and select any logical management operation that is implemented by the workflow. If unsure, then do not select any logical management operations. Click **Finish**. A new **.wkf** workflow file is created. This workflow file has the basic file structure in place.

Sample workflow code example

Sample workflow code example

```
          workflow test123(in Number, out Result) LocaleInsensitive

                #invoke jython in a conditional statement to evaluate input
                 if Jython[int(Number) < 10] then
                        Result = "This number is less than 10."
                 else
                        Result = "This number is greater than 10."
                 endif
```



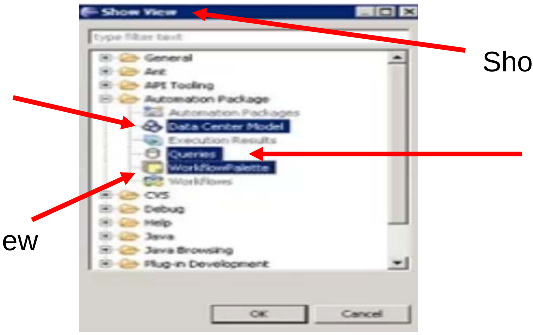
FirstSampleWorkflow.wkf

```
@doc Add description here

workflow FirstSampleWorkflow(in Number, out Result) LocaleInsensitive
    #invoke Jython in a conditional statement to evaluate input
  if Jython[int(Number) < 10] then
    Result = "This number is less than 10."
  else
    Result = "This number is more than 10."
  endif
```

Each APDE workflow has code that is available for viewing. On this slide, a sample code example is displayed. Use the sample code from this slide for your first workflow.

APDE features and views

- Data Center Model view
- Queries view
- Workflow Palette view



Show view

Data Center Model

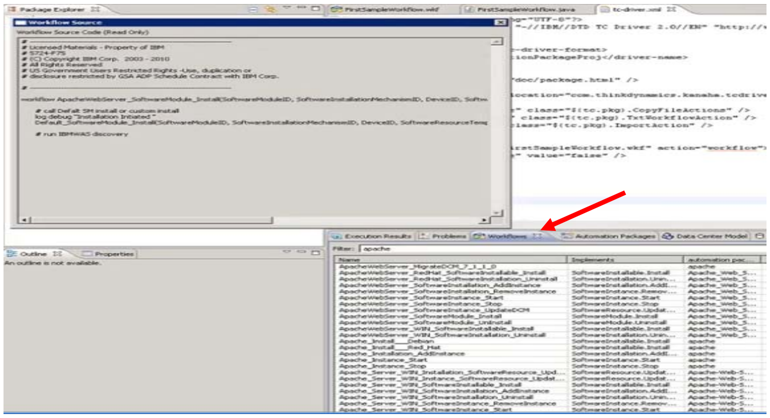
Queries

Workflow Palette view

There are additional views available within the automation package perspective. From the **Window** menu, click **Show View**. Next, select **Other**. There are six available views. By default the automation package perspective includes automation package, execution results, and workflows. Select additional views **Data Center Model**, **Queries**, and **Workflow Palette**. Now find the additional tabs in the lower panel of the screen.

How to view the source code of a workflow

Provisioning workflows view (view source code)



Workflow Source Code (Read Only)

```
# ------------------------------------------------
# Licensed Materials - Property of IBM
# 5724-F75
# (C) Copyright IBM Corp.  2003 - 2010
# All Rights Reserved
# US Government Users Restricted Rights -Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# ------------------------------------------------

workflow ApacheWebServer_SoftwareModule_Install(SoftwareModuleID, SoftwareInstallationMechanismID, DeviceID, Softw

    # call Defalt SM install or custom install
    log debug "Installation Initiated "
    Default_SoftwareModule_Install(SoftwareModuleID, SoftwareInstallationMechanismID, DeviceID, SoftwareResourceTemp

    # run IBMWAS discovery
```

```
ng="UTF-8"?>
 "-//IBM//DTD TC Driver 2.0//EN" "http://

-driver-format>
ionPackageProj</driver-name>

"doc/package.html" />

location="com.thinkdynamics.kanaha.tcdrive

r" class="${tc.pkg}.CopyFileActions" />
" class="${tc.pkg}.TxtWorkflowAction" />
class="${tc.pkg}.ImportAction" />

irstSampleWorkflow.wkf" action="workflow">
e" value="false" />
```

An outline is not available.

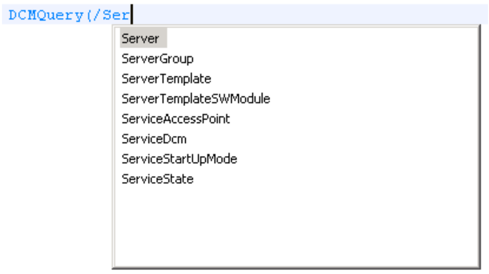
Filter: apache

Name	Implements	automation pac...
ApacheWebServer_MigrateDCM_7_1_1_0		apache
ApacheWebServer_RedHat_SoftwareInstallable_Install	SoftwareInstallable.Install	Apache_Web_S...
ApacheWebServer_RedHat_SoftwareInstallation_Uninstall	SoftwareInstallation.Unin...	Apache_Web_S...
ApacheWebServer_SoftwareInstallation_AddInstance	SoftwareInstallation.AddI...	Apache_Web_S...
ApacheWebServer_SoftwareInstallation_RemoveInstance	SoftwareInstance.Remov...	Apache_Web_S...
ApacheWebServer_SoftwareInstance_Start	SoftwareInstance.Start	Apache_Web_S...
ApacheWebServer_SoftwareInstance_Stop	SoftwareInstance.Stop	Apache_Web_S...
ApacheWebServer_SoftwareInstance_UpdateDCM	SoftwareResource.Updat...	Apache_Web_S...
ApacheWebServer_SoftwareModule_Install	SoftwareModule.Install	Apache_Web_S...
ApacheWebServer_SoftwareModule_Uninstall	SoftwareModule.Uninstall	Apache_Web_S...
ApacheWebServer_WIN_SoftwareInstallable_Install	SoftwareInstallable.Install	Apache_Web_S...
ApacheWebServer_WIN_SoftwareInstallation_Uninstall	SoftwareInstallation.Unin...	Apache_Web_S...
Apache_Install____Debian	SoftwareInstallable.Install	apache
Apache_Install____Red_Hat	SoftwareInstallable.Install	apache
Apache_Installation_AddInstance	SoftwareInstallation.AddI...	apache
Apache_Instance_Start	SoftwareInstance.Start	apache
Apache_Instance_Stop	SoftwareInstance.Stop	apache
Apache_Server_WIN_Installation_SoftwareResource_Upd...	SoftwareResource.Updat...	Apache-Web-S...
Apache_Server_WIN_Instance_SoftwareResource_Updat...	SoftwareResource.Updat...	Apache-Web-S...
Apache_Server_WIN_SoftwareInstallable_Install	SoftwareInstallable.Install	Apache-Web-S...
Apache_Server_WIN_SoftwareInstallation_AddInstance	SoftwareInstallation.AddI...	Apache-Web-S...
Apache_Server_WIN_SoftwareInstallation_Uninstall	SoftwareInstallation.Unin...	Apache-Web-S...
Apache_Server_WIN_SoftwareInstance_RemoveInstance	SoftwareInstance.Remov...	Apache-Web-S...
Apache_Server_WIN_SoftwareInstance_Start	SoftwareInstance.Start	Apache-Web-S...

Search for a specific workflow by using the workflows view tab at the bottom of the screen. Double-click a workflow to view the source code.

Obtaining context-sensitive help while writing workflows

Simultaneously press the Ctrl + Space keys for context-sensitive help

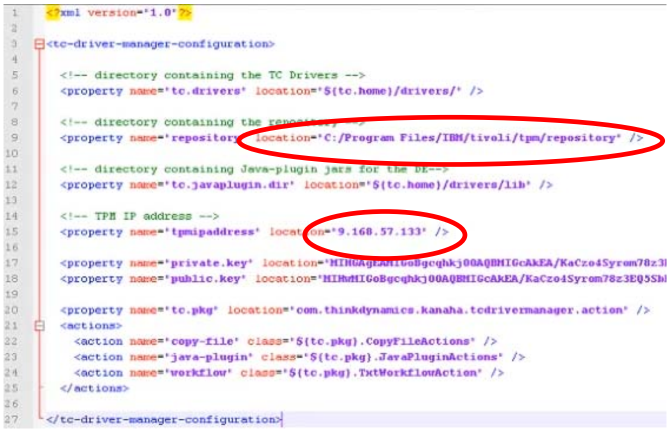


```
DCMQuery(/Ser
```

Sometimes a developer finds it handy to access help files when writing workflows. Simultaneously press the Ctrl + Space keys to obtain APDE context-sensitive help. The help file reveals the jython syntax options available for that particular code.

Important configuration and log files (1 of 3)

tcdrivermanager.xml

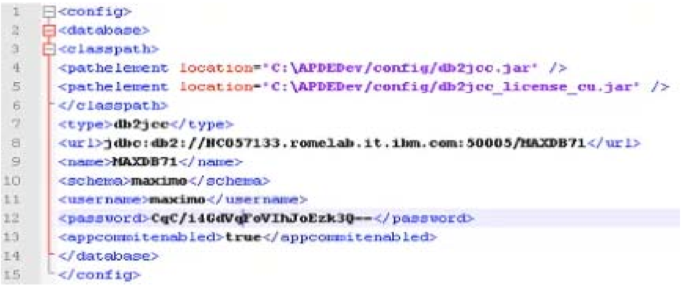


```
<?xml version='1.0'?>

<tc-driver-manager-configuration>

  <!-- directory containing the TC Drivers -->
  <property name='tc.drivers' location='${tc.home}/drivers/' />

  <!-- directory containing the repository -->
  <property name='repository' location='C:/Program Files/IBM/tivoli/tpm/repository' />

  <!-- directory containing Java-plugin jars for the DE-->
  <property name='tc.javaplugin.dir' location='${tc.home}/drivers/lib' />

  <!-- TPM IP address -->
  <property name='tpmipaddress' location='9.168.57.133' />

  <property name='private.key' location='MIHGAgEAMIGoBgcqhkj00AQBMIGcAkEA/KaCzo4Syrom78z3I
  <property name='public.key' location='MIHwMIGoBgcqhkj00AQBMIGcAkEA/KaCzo4Syrom78z3EQ5Sh

  <property name='tc.pkg' location='com.thinkdynamics.kanaha.tcdrivermanager.action' />
  <actions>
    <action name='copy-file' class='${tc.pkg}.CopyFileActions' />
    <action name='java-plugin' class='${tc.pkg}.JavaPluginActions' />
    <action name='workflow' class='${tc.pkg}.TxtWorkflowAction' />
  </actions>

</tc-driver-manager-configuration>
```

There are a few important files that are associated with the APDE runtime. The first is the **tcdrivermanager.xml** file. In this slide, the XML file displays the default repository location where tcdriver files are stored. This slide also displays the IBM Tivoli Provisioning Manager provisioning server’s IP address.

Important configuration and log files (2 of 3)

dcm.xml configuration XML file

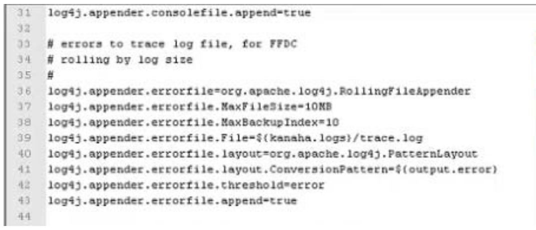


```
<config>
<database>
<classpath>
<pathelement location='C:\APDEDev/config/db2jcc.jar' />
<pathelement location='C:\APDEDev/config/db2jcc_license_cu.jar' />
</classpath>
<type>db2jcc</type>
<url>jdbc:db2://HC057133.romelab.it.ibm.com:50005/MAXDB71</url>
<name>MAXDB71</name>
<schema>maximo</schema>
<username>maximo</username>
<password>CqC/i4GdVqFoVIhJoEzk3Q==</password>
<appcommitenabled>true</appcommitenabled>
</database>
</config>
```

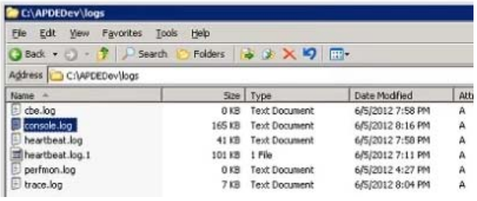
The **dcm.xml** file is a configuration file. It stores the location of the JDBC type4 jar files. This XML file also stores DB2 connection-specific information. The APDE log in password is stored as an encrypted value.

Important configuration and log files (3 of 3)

- **APDEDev\logs** folder
- **Console.log** file
- **Trace.log** file



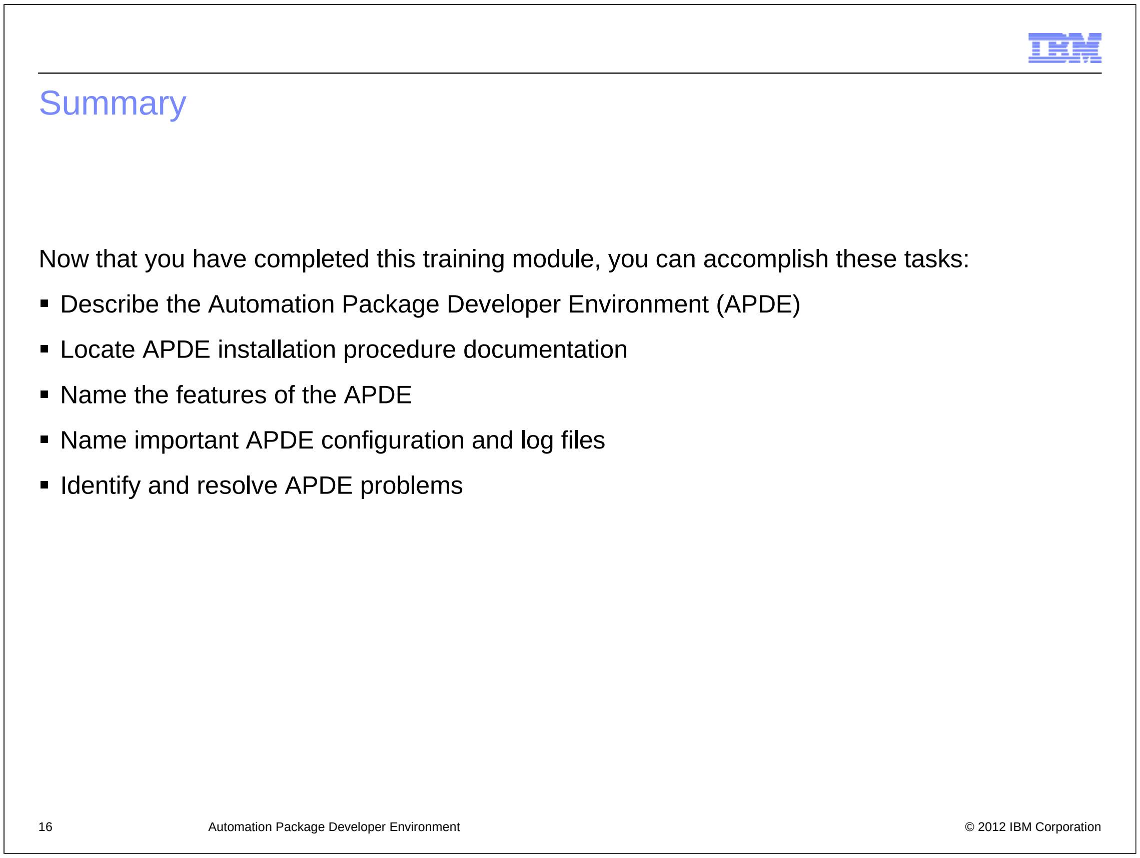
```
log4j.appender.consolefile.append=true

# errors to trace log file, for FFDC
# rolling by log size
#
log4j.appender.errorfile=org.apache.log4j.RollingFileAppender
log4j.appender.errorfile.MaxFileSize=10MB
log4j.appender.errorfile.MaxBackupIndex=10
log4j.appender.errorfile.File=${kanaha.logs}/trace.log
log4j.appender.errorfile.layout=org.apache.log4j.PatternLayout
log4j.appender.errorfile.layout.ConversionPattern=$(output.error)
log4j.appender.errorfile.threshold=error
log4j.appender.errorfile.append=true
```



C:\APDEDev\logs

Name	Size	Type	Date Modified	Attı
cbe.log	0 KB	Text Document	6/5/2012 7:58 PM	A
console.log	165 KB	Text Document	6/5/2012 8:16 PM	A
heartbeat.log	41 KB	Text Document	6/5/2012 7:58 PM	A
heartbeat.log.1	101 KB	1 File	6/5/2012 7:11 PM	A
perfmon.log	0 KB	Text Document	6/5/2012 4:27 PM	A
trace.log	7 KB	Text Document	6/5/2012 8:04 PM	A

The **APDEDev\logs** folder contains both the **console.log** file and the **trace.log** file. These log files are used in system problem resolution scenarios. Search for information in these log files when a workflow is generating error messages or when problems occur while compiling workflow code.



Summary

Now that you have completed this training module, you can accomplish these tasks:

- Describe the Automation Package Developer Environment (APDE)
- Locate APDE installation procedure documentation
- Name the features of the APDE
- Name important APDE configuration and log files
- Identify and resolve APDE problems

Now that you have completed this training module, you can accomplish these tasks:

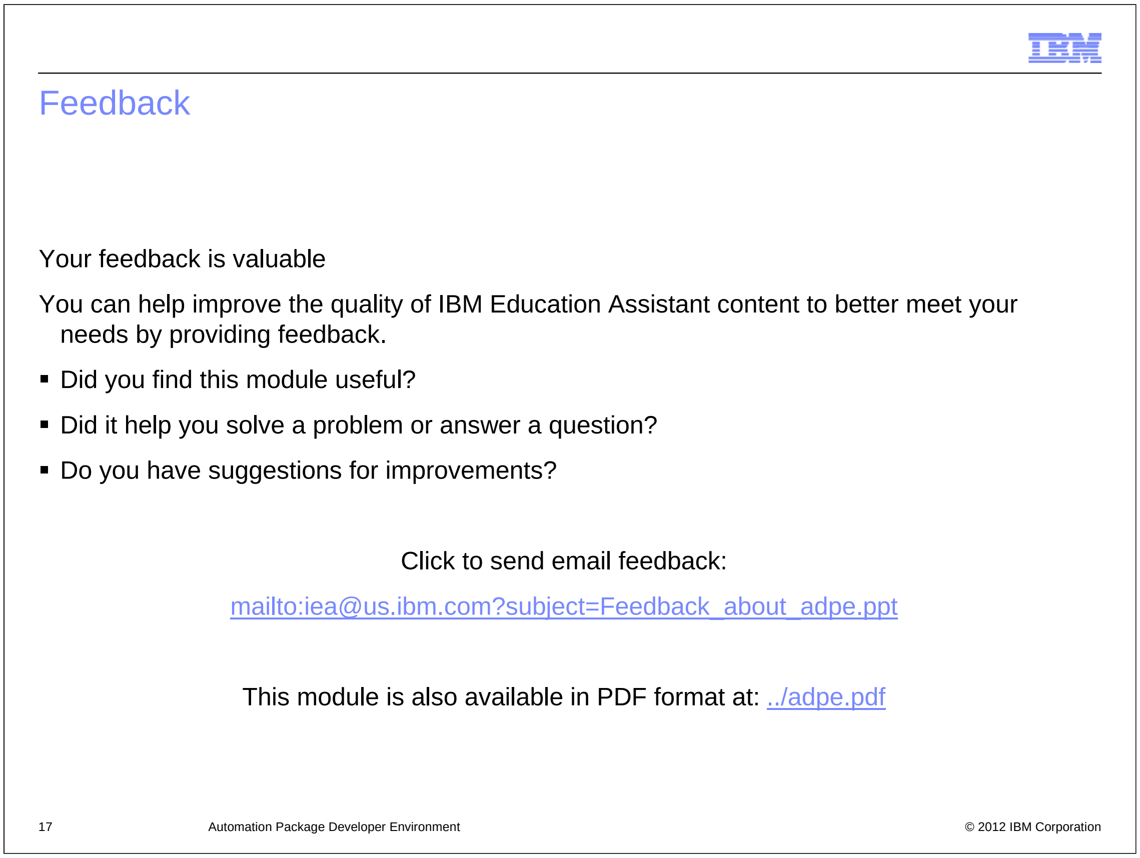
- Describe the Automation Package Developer Environment (APDE)

- Locate APDE installation procedure documentation

- Name the features of the APDE

- Name important APDE configuration and log files

- Identify and resolve APDE problems



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

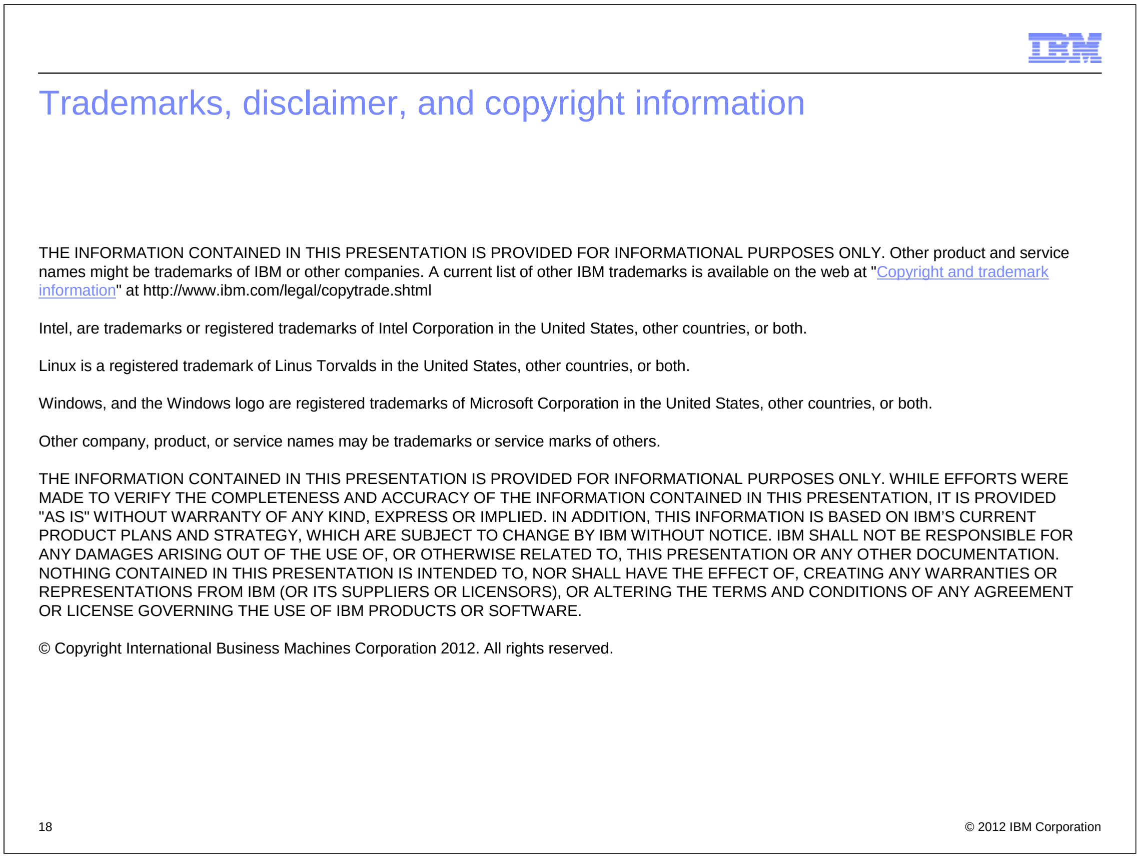
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_adpe.ppt

This module is also available in PDF format at: ../adpe.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

Intel, are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.