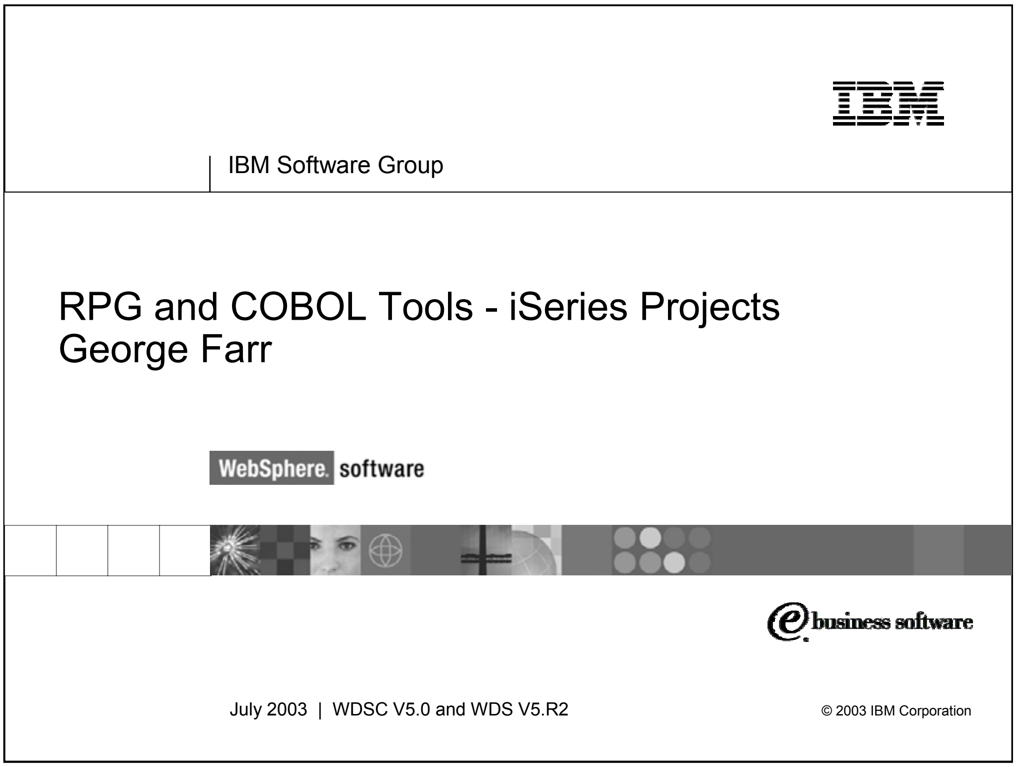


IBM Software Group

RPG and COBOL Tools - iSeries Projects
George Farr

WebSphere. software

e business software

July 2003 | WDSC V5.0 and WDS V5.R2

ADTS has been the traditional method for developing and maintaining server-side iSeries applications. WebSphere Development Studio Client for iSeries, V5.0, includes new highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offer programmers a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications, lowering the learning curve for all.

These new generation tools include the Remote System Explorer for a PDM-like experience, and iSeries projects for team-based development (together with a Eclipse-compliant software change management repository). They offer rich support for exploring the file system, compiling/building, editing, running, and debugging. The iSeries Projects support enables effective team support leveraging any iSeries or LAN resident source repository which supports Eclipse.

This presentation covers the iSeries Projects.

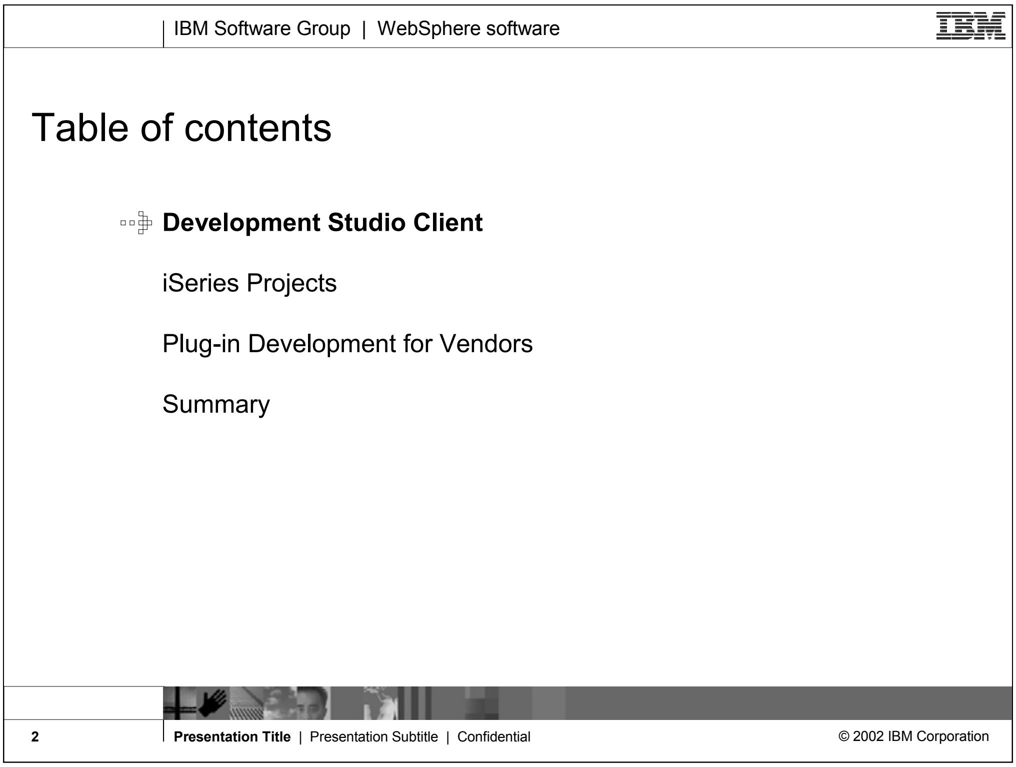


Table of contents

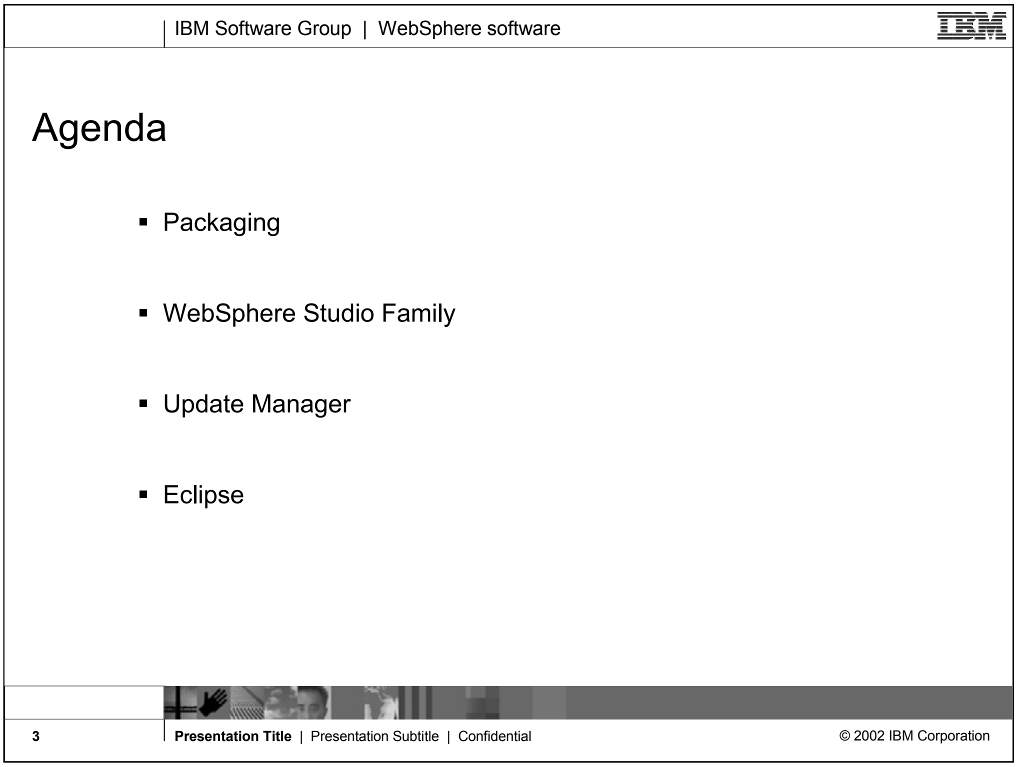
Development Studio Client

iSeries Projects

Plug-in Development for Vendors

Summary

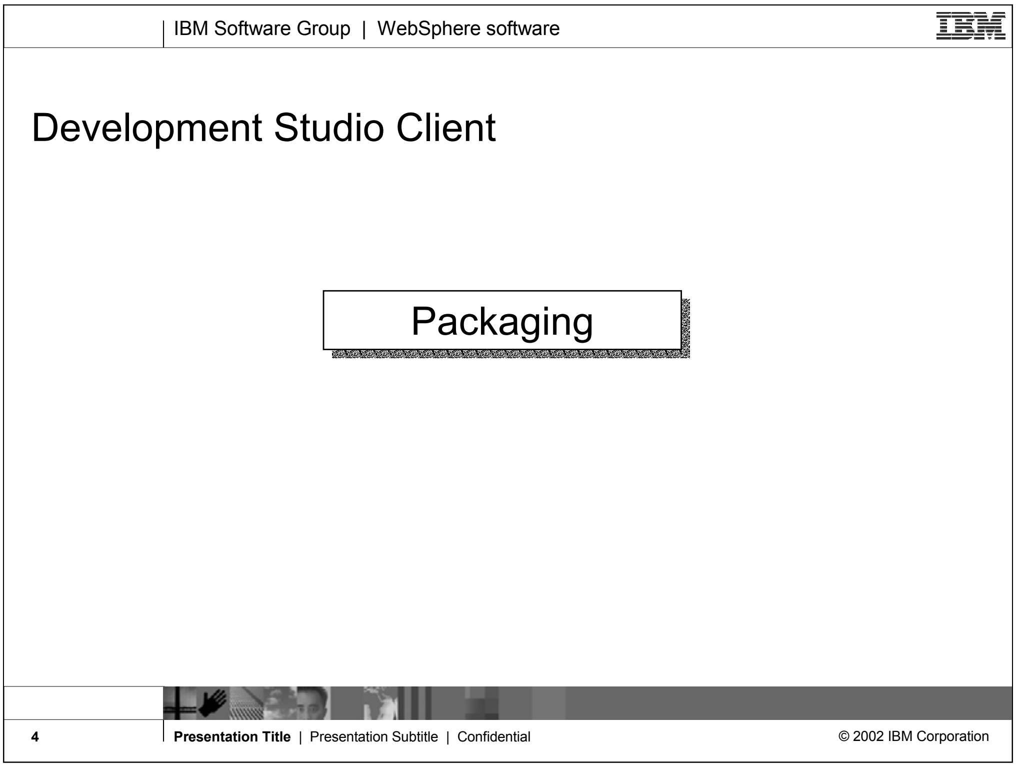
This presentation covers first the strategy behind Development Studio Client and is then followed by a review of the iSeries Projects in Development Studio Client Version 5.0. The final section reviews how business partners can easily leverage and extend iSeries Projects by easily integrating their tools with the IBM base development environment.



Agenda

- Packaging
- WebSphere Studio Family
- Update Manager
- Eclipse

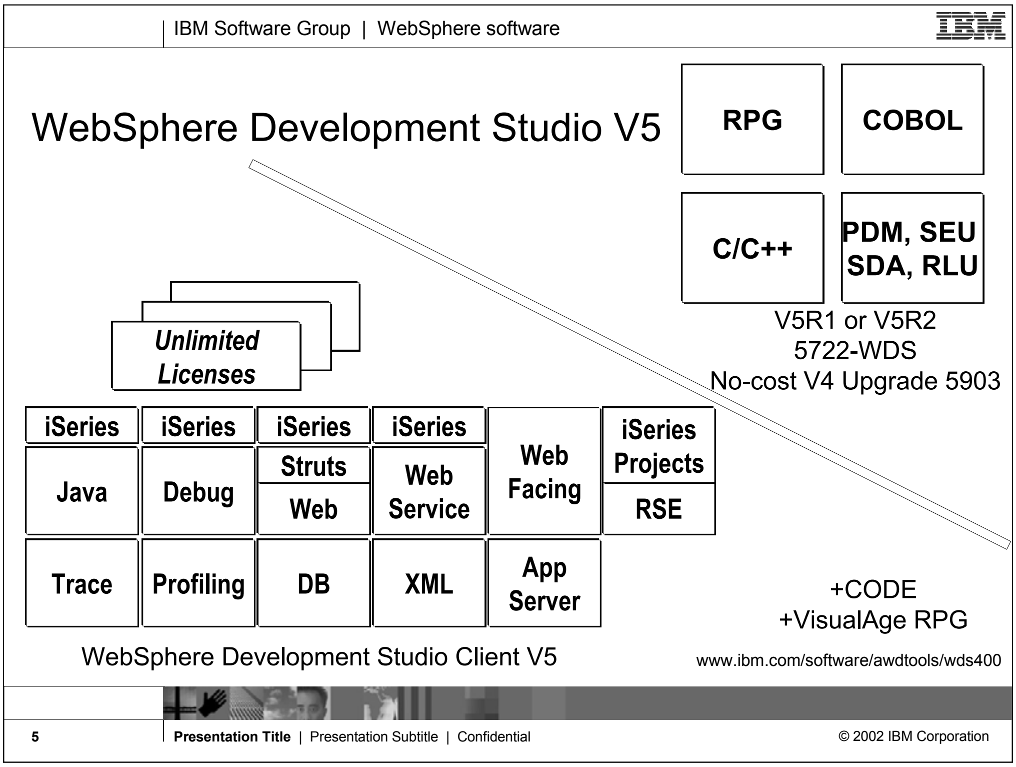
The first section of this presentation covers Development Studio Client packaging, how Development Studio Client fits into the WebSphere family of products, how Update Manager can easily be used to apply the latest product fixes and how the Eclipse technology forms the foundation of the Development Studio Client and all WebSphere products.



Development Studio Client

Packaging

The new packaging was actually done in May 2001, with the introduction of WebSphere Development Studio for iSeries, which as we will see offers a single product with all the host and client tools needed for all application development needs, from traditional to e-business.



WebSphere Development Studio V5

V5R1 or V5R2
5722-WDS
No-cost V4 Upgrade 5903

+CODE
+VisualAge RPG

WebSphere Development Studio Client V5

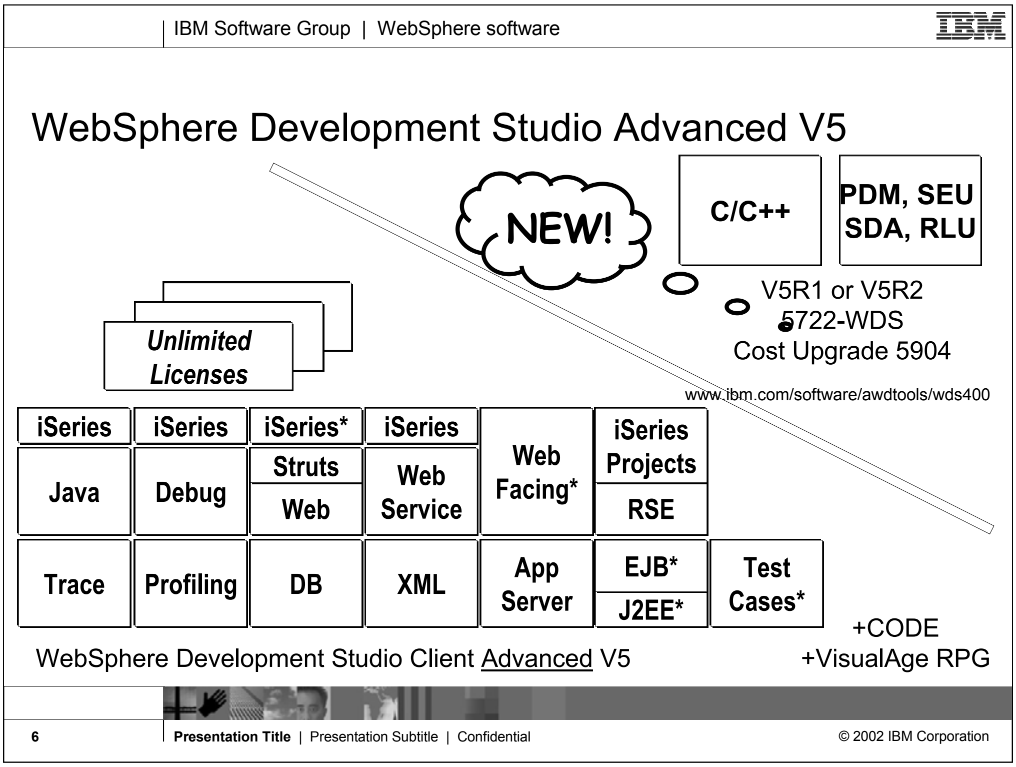
www.ibm.com/software/awdtools/wds400

There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.



WebSphere Development Studio Advanced V5

NEW!

C/C++

PDM, SEU SDA, RLU

V5R1 or V5R2
5722-WDS
Cost Upgrade 5904

www.ibm.com/software/awdtools/wds400

Unlimited Licenses

iSeries Java | iSeries Debug | iSeries* Struts Web | iSeries Web Service | Web Facing* | iSeries Projects RSE

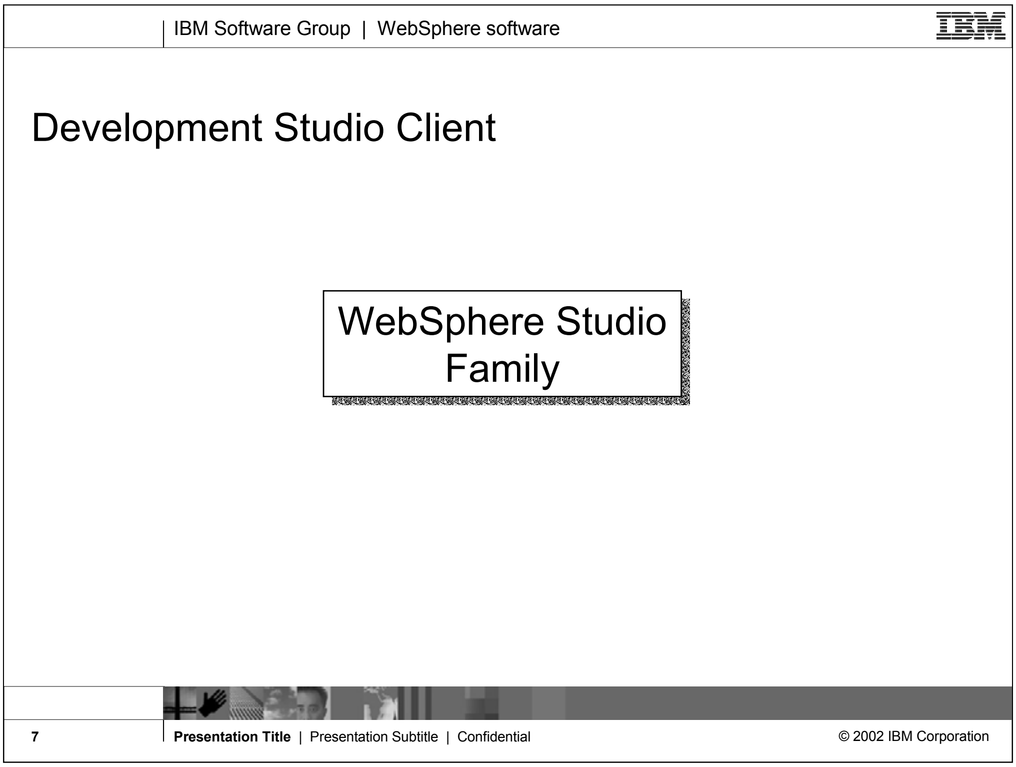
Trace | Profiling | DB | XML | App Server | EJB* J2EE* | Test Cases*

+CODE
+VisualAge RPG

WebSphere Development Studio Client Advanced V5

The Advanced edition of Development Studio Client, and Development Studio, is new as of April 25th 2003. Development Studio Advanced is currently the same as Development Studio, except the customer is entitled to unlimited licenses of Development Studio Client Advanced versus just Development Studio Client.

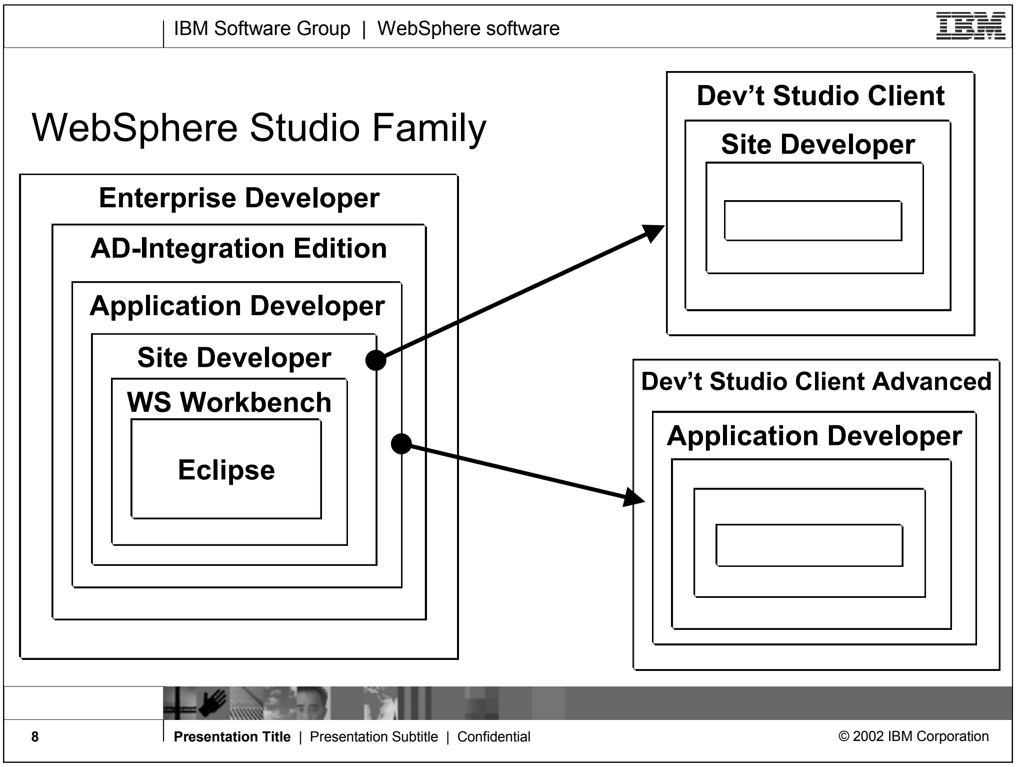
The difference of Development Studio Client Advanced over Development Studio Client is that it has additional tools (blue boxes with asterisks) and some enhancements to existing iSeries tools (green boxes with asterisks) which will be discussed later in this presentation. The majority of the new tools are related to Enterprise Java Bean (EJB) development. You’ll see later that Development Studio Client Advanced is based on WebSphere Application Developer (Application Developer) versus WebSphere Studio Site Developer (Site Developer).



Development Studio Client

WebSphere Studio Family

Now you will see how Development Studio Client fits in the WebSphere Studio family of products.



WebSphere Studio Family

Enterprise Developer

AD-Integration Edition

Application Developer

Site Developer

WS Workbench

Eclipse

Dev’t Studio Client

Site Developer

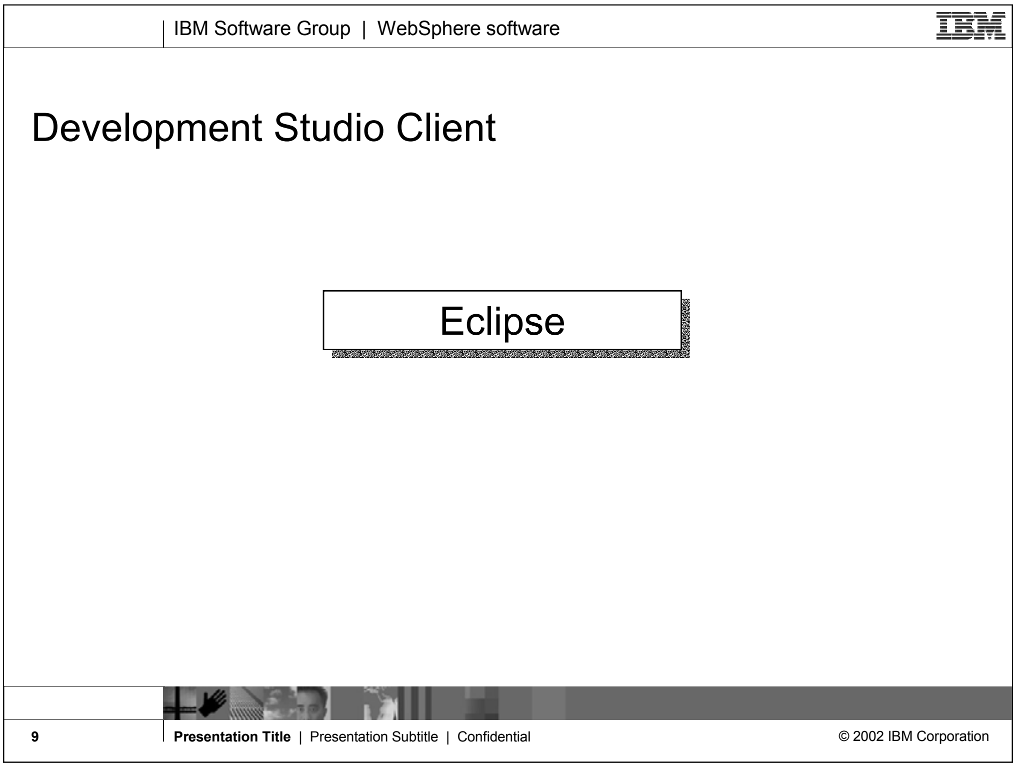
Dev’t Studio Client Advanced

Application Developer

Here you see that Development Studio Client is based on WebSphere Studio Site Developer, while Development Studio Client Advanced is based on WebSphere Studio Application Developer.

The Workbench is based on the open-source Eclipse technology about to be discussed. It is not for sale, but is the basis of all IBM WebSphere Studio products, and is available to business partners.

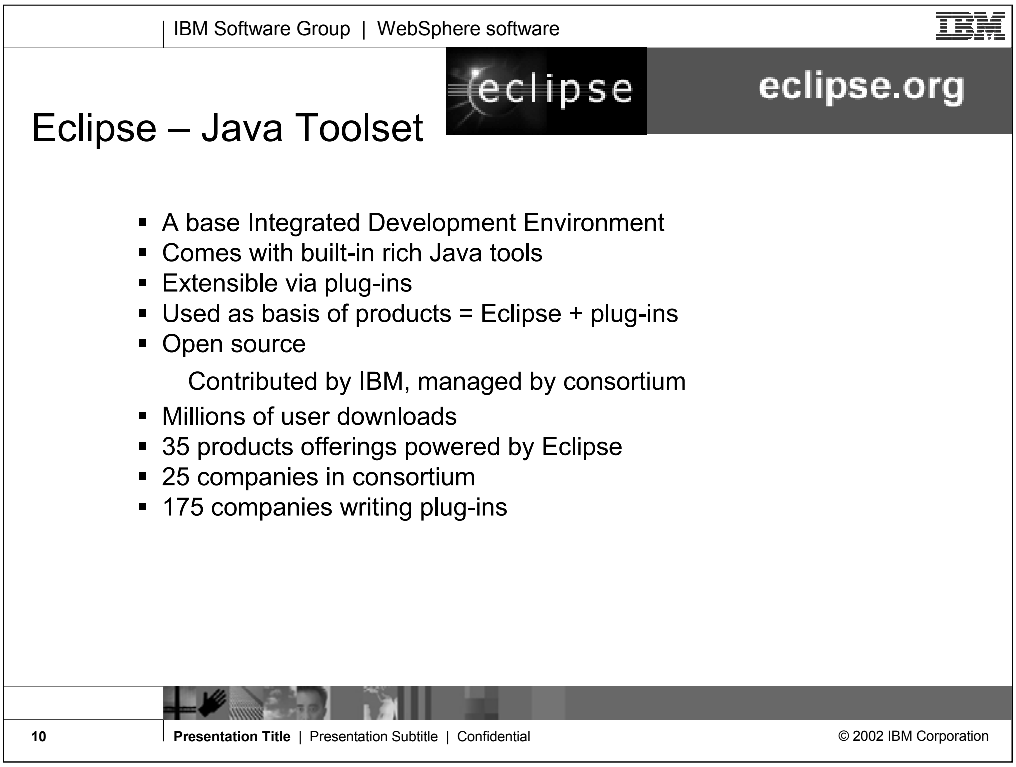
Site Developer is IBM’s entry level offering based on eclipse, and it is for building dynamic Web sites out of non-EJB Java. Application Developer extends Site Developer and adds support for EJBs. Application Developer-Integration Edition extends Application Developer and adds support for JCA Connectors and for Workflow. Enterprise Developer extends Application Developer-Integration Edition and adds support for S/390 and Enterprise Generation Language (EGL), the follow-on to VisualAge Generator.



Development Studio Client

Eclipse

With the 4.0 release of the client tools in June 2002, we introduced new Eclipse technology and totally new Eclipse-based tools. Further, for technology we also plan to keep up with technology both in the compilers and the tools, so iSeries programmers are very current. This was evident in the V5R2 release of the compilers in Sept 2002, and will be again in the V5R3 release of the compilers. With the April 2003 new 5.0 release of the client tools, we are in the second release of the eclipse-based technology and continue to improve the tools and introduce new technology.

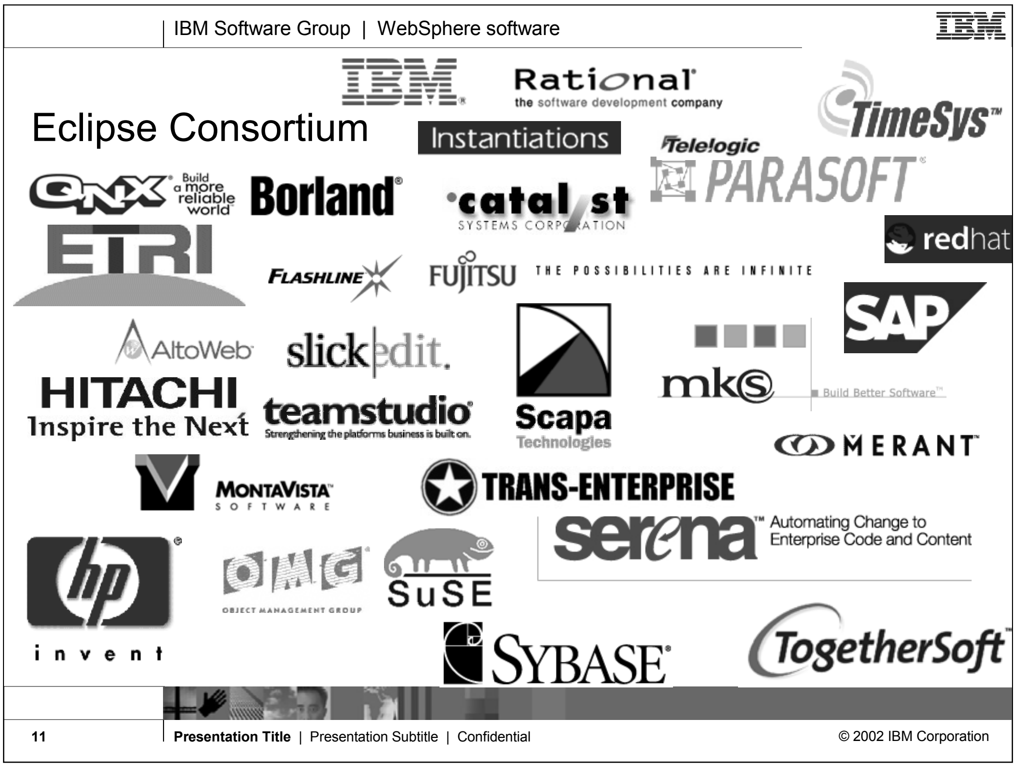


Eclipse – Java Toolset

- A base Integrated Development Environment
- Comes with built-in rich Java tools
- Extensible via plug-ins
- Used as basis of products = Eclipse + plug-ins
- Open source

  Contributed by IBM, managed by consortium
- Millions of user downloads
- 35 products offerings powered by Eclipse
- 25 companies in consortium
- 175 companies writing plug-ins

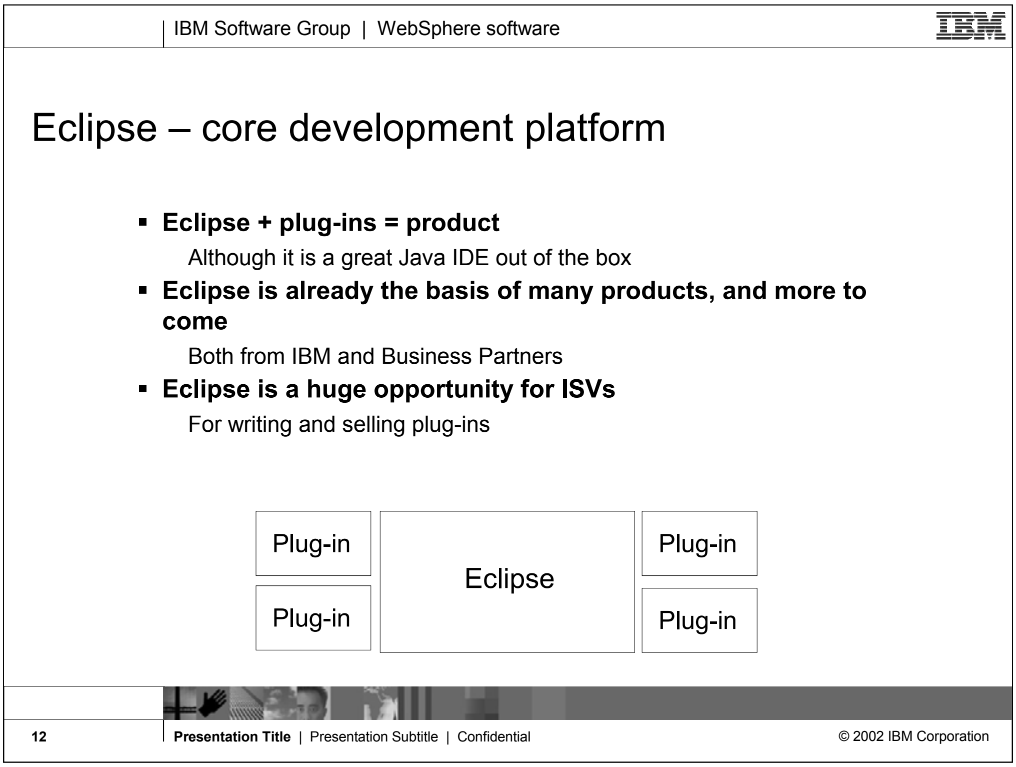
Eclipse was developed by IBM and donated to the open source community. That donation is estimated to be worth $40 million. Anyone can download Eclipse for free, including the source code, from www.eclipse.org. Eclipse has generated extraordinary excitement in the development community and the tools community. It is written in Java, and can be extended by tools that are also written in Java. These tools are known as plug-ins. Out of the box, Eclipse offers an integrated development environment (IDE) that has built-in support for teams and projects and a robust and revolutionary user interface framework. It also has tools built-in to create Eclipse plug-ins. Further, there are extensive and very powerful tools built-in for developing Java applications with Eclipse. So, if all you want is the world's best Java toolset, then all you need is Eclipse. You can't beat the price!



Eclipse Consortium

IBM
Rational the software development company
TimeSys™
Instantiations
Telelogic
PARASOFT
QNX Build a more reliable world
Borland
catalyst SYSTEMS CORPORATION
redhat
ETRI
FLASHLINE
FUJITSU THE POSSIBILITIES ARE INFINITE
SAP
AltoWeb
slickedit.
mks Build Better Software™
HITACHI Inspire the Next
teamstudio Strengthening the platforms business is built on.
Scapa Technologies
MERANT™
MONTAVISTA™ SOFTWARE
TRANS-ENTERPRISE
serena™ Automating Change to Enterprise Code and Content
hp invent
OMG OBJECT MANAGEMENT GROUP
SuSE
SYBASE
TogetherSoft™

IBM is not alone with Eclipse. The open-source consortium that oversees contributions made to it include a number of large companies, and that list is growing. Note that MKS is an iSeries tool vendor. See www.eclipse.org for the latest list.



Eclipse – core development platform

- **Eclipse + plug-ins = product**
  Although it is a great Java IDE out of the box
- **Eclipse is already the basis of many products, and more to come**
  Both from IBM and Business Partners
- **Eclipse is a huge opportunity for ISVs**
  For writing and selling plug-ins

Plug-in
Plug-in
Eclipse
Plug-in
Plug-in

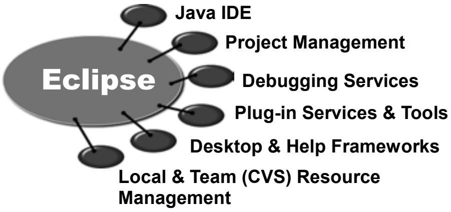
So what is Eclipse? With the exception of the built-in Java tools, Eclipse itself is not that interesting. It only gets interesting when you add to it some plug-ins (for example, tools) that do something interesting.

An Eclipse-based product is Eclipse plus a number of interesting plug-ins. IBM is building numerous such Eclipse-based products, including as we have seen Site Developer and Development Studio Client. Further, because Eclipse is free and business partners are free to include it in their products, there will be many other Eclipse-based products from other companies too, including Rational and TogetherSoft.

For business partners or software developers who write and sell application development tools, Eclipse is a fantastic opportunity. By writing plug-ins for Eclipse, those plug-ins can be sold to any developer using any product based on Eclipse or even just the raw Eclipse as downloaded from www.eclipse.org. This opportunity is not lost on iSeries tool vendors, who are all looking at offering Eclipse plug-ins for their tools. This will result in a rich offering of third party plug-ins for developers to choose from, all of which extend their core Development Studio Client development environment. One community, one core development platform, many IBM and 3rd party tools. This is community and excitement!

Eclipse Plug-in Development

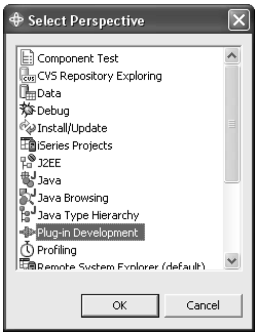
Products built with Eclipse inherit these capabilities plus 'plug-ins' built by others



Some of the common services Eclipse supplies include: the Java integrated development environment (IDE), project management, debug, plug-in frameworks, desktop and help frameworks, and resource managers.

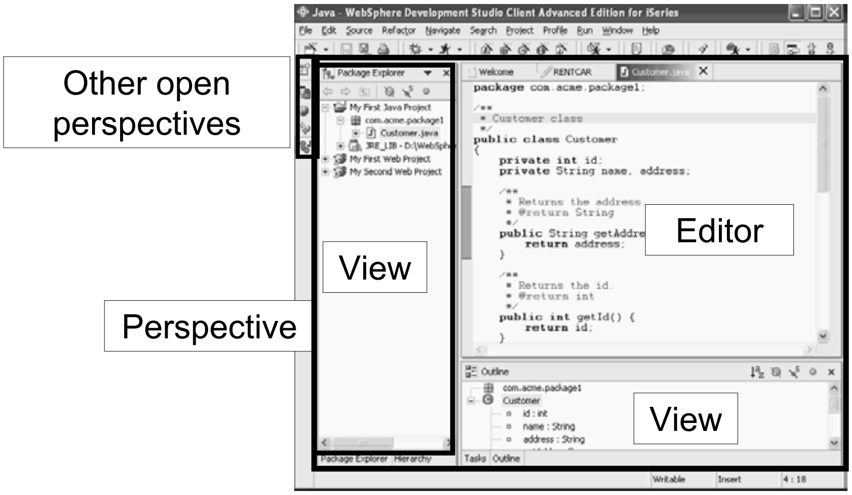
Eclipse User Interface – Select Perspective

- Users work with **perspectives**
  - –Collection of **views** and **editors**
    - –Tools for a particular task
    - –Allows for role-based development
  - –Many perspectives are pre-supplied for specific tasks like Java, Web, XML, RPG/CBL
  - –Users can create their own perspectives
- The user interface is very Windows-like
  - –Build from a Java wrapper of OS widgets
  - –Behaves and feels like any other native application
  - –Eg: views can be re-sized and re-positioned through drag ‘n drop



The core features of the Eclipse user interface include perspectives which is a collection of views and tools. Perspectives allow role based development. For example, if you are a Java developer you would use the Java perspective which includes tools and views for Java development. You can also create your own perspectives. Naturally, the Eclipse user interface applies to all Eclipse-based products, like Development Studio Client.

Eclipse User Interface – Open Perspective



```
package com.acme.package1;

/**
 * Customer class
 */
public class Customer
{
    private int id;
    private String name, address;

    /**
     * Returns the address
     * @return String
     */
    public String getAddre
        return address;
    }

    /**
     * Returns the id.
     * @return int
     */
    public int getId() {
        return id;
    }
```

Other open perspectives

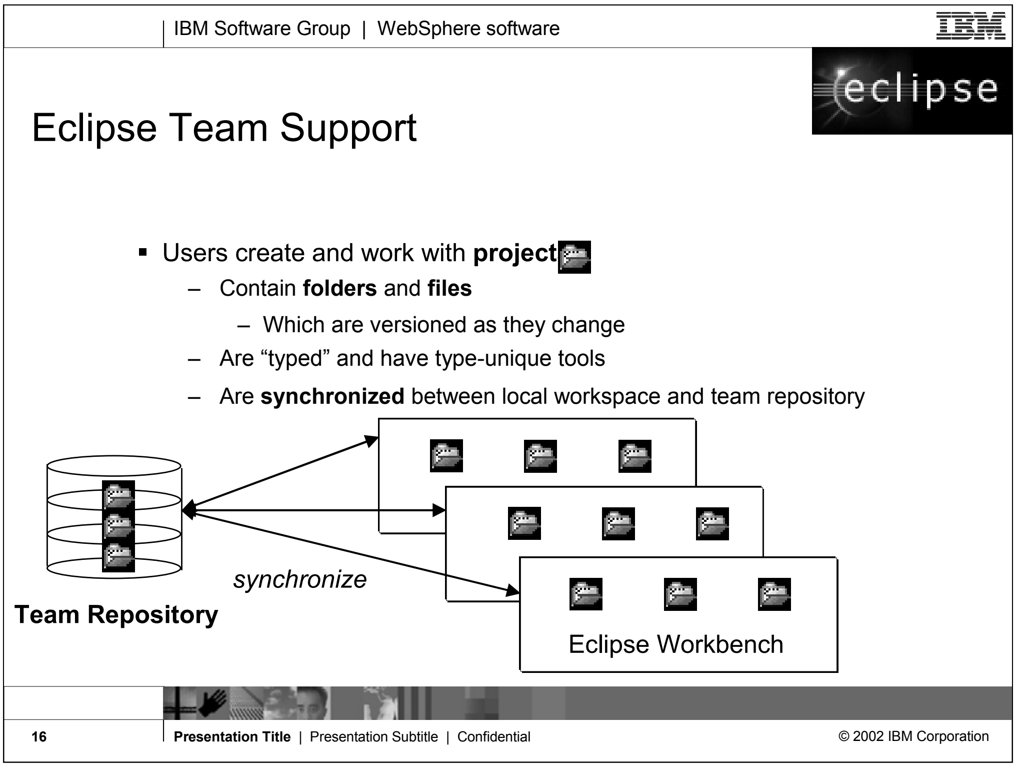
Perspective

View

Editor

View

The Eclipse workbench has four open perspectives. You can see them lined up on the left frame of the workbench. The current active perspective is the one whose icon is indented, which in this case is the Java perspective. To open a new perspective, you use Window->Open Perspective giving you the Select Perspective dialog that you saw previously. This particular perspective has a Package Explorer view which drives the Editor and the Outline view.



Eclipse Team Support

- Users create and work with **projects**
  - Contain **folders** and **files**
    - Which are versioned as they change
  - Are "typed" and have type-unique tools
  - Are **synchronized** between local workspace and team repository

Central to the Eclipse integrated development environment is support for projects. Projects are simply a grouping of folders and files.

Plug-ins have types, such as Java or Web or XML. Tools that plug into Eclipse can define their own new project types. Tools that plug into Eclipse work against resources (folders or files) within projects. They can be scoped to projects of a particular type, if appropriate.

All projects, regardless of type, have common behavior and support. This includes the ability for a team to share a project by using an Eclipse-supporting software change management (SCM) product such as Concurrent Version System (CVS) or Rational Clearcase. This SCM acts as a central repository for one or more projects. Each team member can easily keep their local copy of any project "in synch" with the central repository copy. CVS is a free open-source SCM. It runs on Linux, UNIX and Windows. It comes on the Linux distribution CDs for iSeries Linux LPAR.

All IBM SCM vendors for iSeries are enabling, or have enabled, their products to Eclipse.

Table of contents

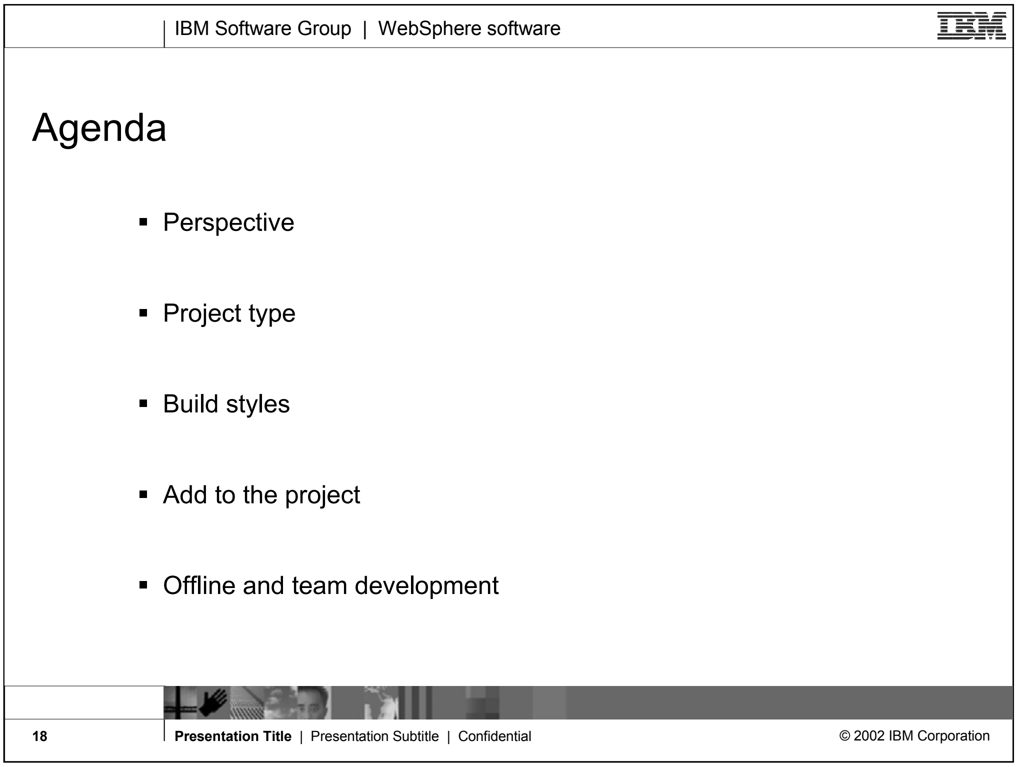
Development Studio Client

iSeries Projects

Plug-in Development for Vendors

Summary

Now you know the story behind Development Studio Client. Lets look at iSeries Projects, the tool for iSeries project based team sharable development.



Agenda

- Perspective
- Project type
- Build styles
- Add to the project
- Offline and team development

The second section of this presentation covers the iSeries Projects which has its own perspective and views. Unlike the Remote Systems Explorer, an iSeries project fully leverages the Eclipse support for resources. This means an iSeries project contains folders and files that exist in the local file system, yet can be shared and synchronized among a team, if an eclipse-compliant repository is used. The options for this today include Rational's ClearCase or the free and open-source product CVS or Concurrent Versioning Systems, or the MKS Integrity. CVS is available on the distribution CDs for the Linux LPAR on iSeries. In the near future, the remaining iSeries change management vendors will also be available as repository options. As you will see, an iSeries project allows RPG and COBOL developers to fully exploit the power Eclipse, while developing applications targeted to run on iSeries.

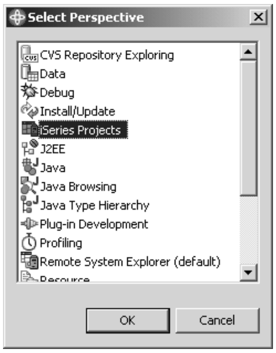
iSeries Projects

- A special project type

  For holding source destined to be compiled on iSeries

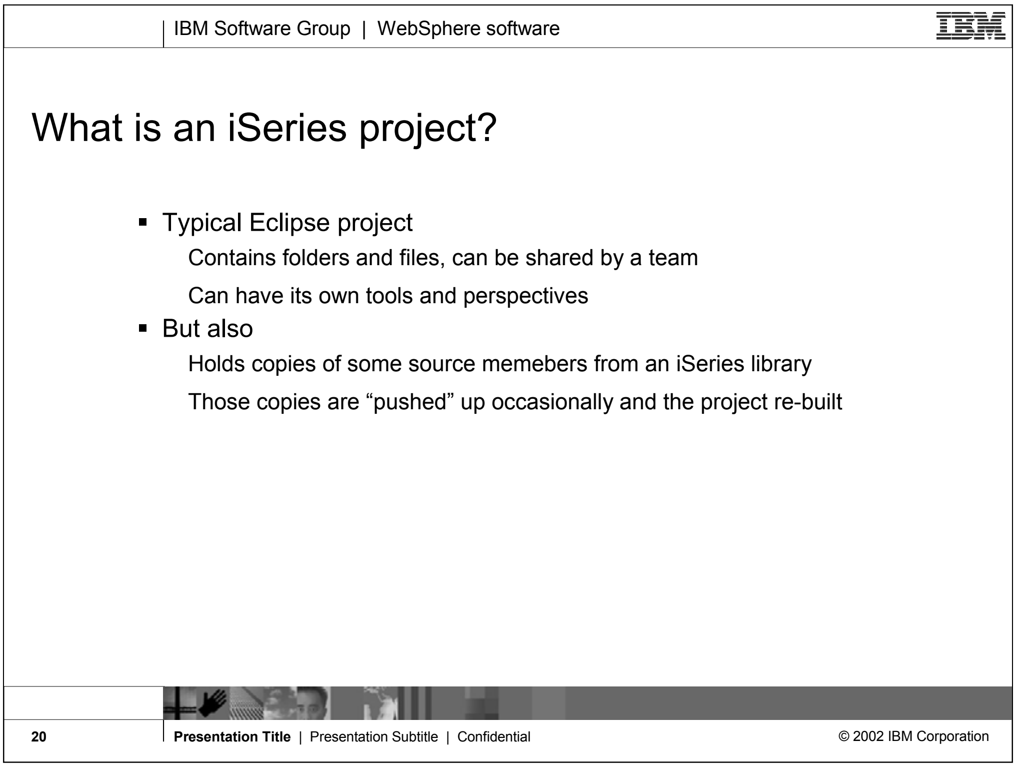
- A dedicated perspective

  For working with all and only iSeries projects



There is a special project type designed to hold source that is destined to be compiled and run on an iSeries. There is a wizard for creating a new iSeries Project.

There is a special perspective for working with iSeries projects.

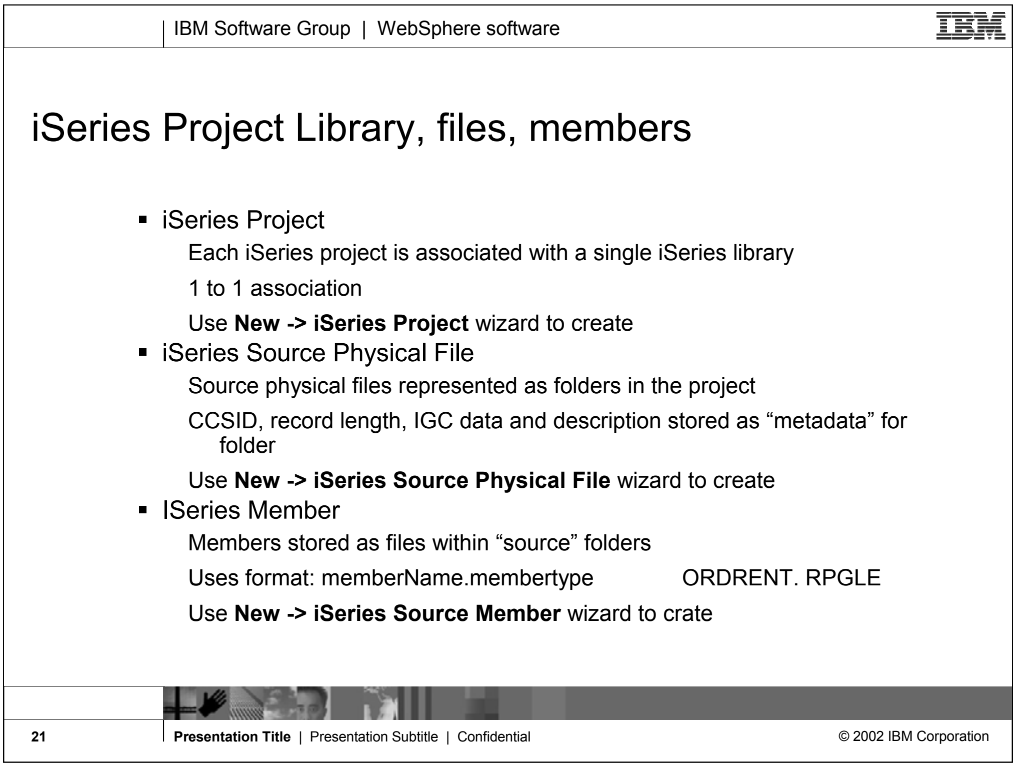


What is an iSeries project?

- Typical Eclipse project
  - Contains folders and files, can be shared by a team
  - Can have its own tools and perspectives
- But also
  - Holds copies of some source memebers from an iSeries library
  - Those copies are “pushed” up occasionally and the project re-built

An iSeries Project is a typical Eclipse project. This means it contains folders and files that are in the local file system of the IDE, but these can be synchronized with a central repository for team support.

Since the files are local, there is actions to support "pushing" those files up to an iSeries library, where they become source members. This is done prior to compiling or "building" the source for the purpose of testing.



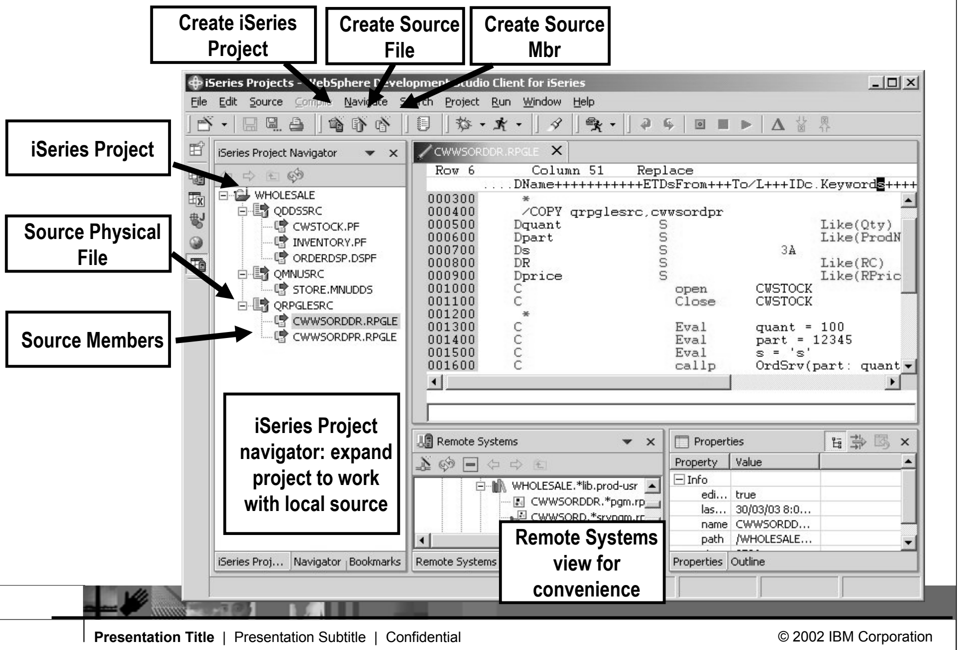
iSeries Project Library, files, members

- iSeries Project
  - Each iSeries project is associated with a single iSeries library
  - 1 to 1 association
  - Use **New -> iSeries Project** wizard to create
- iSeries Source Physical File
  - Source physical files represented as folders in the project
  - CCSID, record length, IGC data and description stored as “metadata” for folder
  - Use **New -> iSeries Source Physical File** wizard to create
- ISeries Member
  - Members stored as files within “source” folders
  - Uses format: memberName.membertype ORDRENT. RPGLE
  - Use **New -> iSeries Source Member** wizard to crate

iSeries Projects are projects that are dedicated for holding iSeries source members destined to be pushed and compiled on an iSeries machine. They follow the Workbench paradigm in that they have a special perspective, the iSeries Projects Perspective, dedicated to showing and working with these projects. When a new iSeries project is created three properties need to be associated with it. These are an iSeries Connection, an Associated Library, and a Build Style.

Only iSeries projects can hold a source physical files and members. Since you can create various types of projects with the Development Studio Client for iSeries, each project type is identified by a distinct icon *decorator*. Specifically, iSeries projects are distinguished by a blueon the top right corner of the project icon.

iSeries Projects Perspective

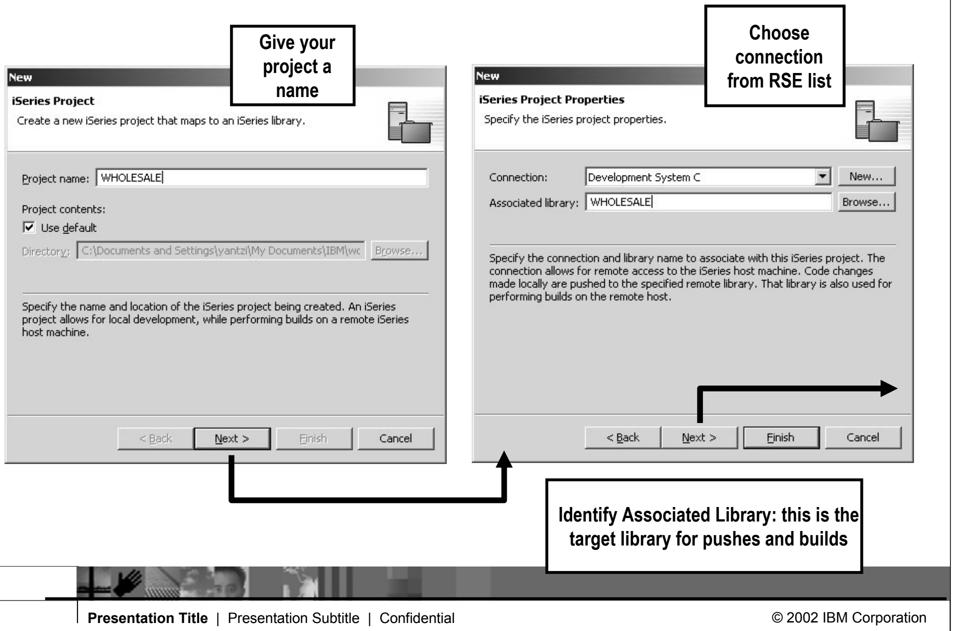


```
 Row 6         Column 51     Replace
          ....DName+++++++++++ETDsFrom+++To/L+++IDc.Keywords++++
000300      *
000400      /COPY qrpglesrc,cwwsordpr
000500     Dquant                S                        Like(Qty)
000600     Dpart                 S                        Like(ProdN
000700     Ds                    S                   3A
000800     DR                    S                        Like(RC)
000900     Dprice                S                        Like(RPric
001000     C                      open         CWSTOCK
001100     C                      Close        CWSTOCK
001200      *
001300     C                      Eval         quant = 100
001400     C                      Eval         part = 12345
001500     C                      Eval         s = 's'
001600     C                      callp        OrdSrv(part: quant
```

Property	Value
Info	
edi...	true
las...	30/03/03 8:0...
name	CWWSORDD...
path	/WHOLESALE...

Here we see the iSeries Projects perspective, which you can explicitly open. The primary view is the iSeries Project Navigator that allows exploration of all existing iSeries Projects. There are wizards lauchable from the toolbar for creating a new iSeries Project, or creating source physical files and source members within an existing project. The iSeries Projects perspective includes a full copy of the Remote Systems Explorer, as a view, for your convenience. This allows you to easily work with the contents of the associated library while simultaneously working with the local copies of the source within the project.

iSeries Project wizard



Give your project a name

New

iSeries Project

Create a new iSeries project that maps to an iSeries library.

Project name: WHOLESALE

Project contents:

☑ Use default

Directory: C:\Documents and Settings\yantzi\My Documents\IBM\wc

Specify the name and location of the iSeries project being created. An iSeries project allows for local development, while performing builds on a remote iSeries host machine.

Choose connection from RSE list

New

iSeries Project Properties

Specify the iSeries project properties.

Connection: Development System C

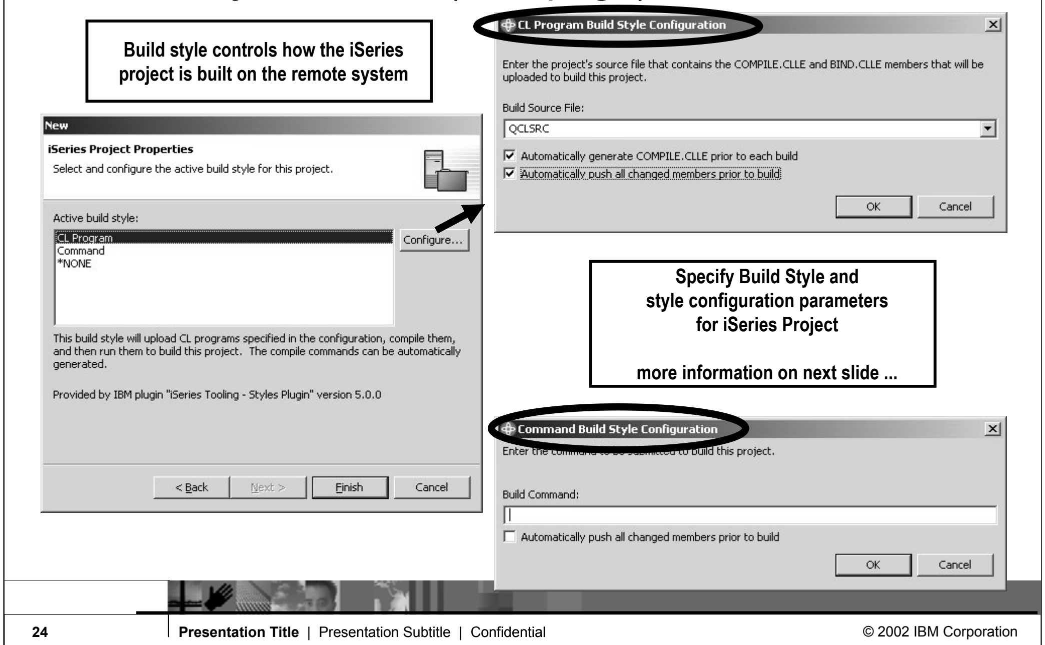
Associated library: WHOLESALE

Specify the connection and library name to associate with this iSeries project. The connection allows for remote access to the iSeries host machine. Code changes made locally are pushed to the specified remote library. That library is also used for performing builds on the remote host.

Identify Associated Library: this is the target library for pushes and builds

Here we see the iSeries Project wizard. First, give the project a name. Any name you want! Then select or create a connection (from the RSE!) that identifies the iSeries with which this project is associated. Also select the library on that iSeries where the contents of this project will be pushed to.

iSeries Project Wizard (next page)



Build style controls how the iSeries project is built on the remote system

New

iSeries Project Properties

Select and configure the active build style for this project.

Active build style:

- CL Program
- Command
- *NONE

This build style will upload CL programs specified in the configuration, compile them, and then run them to build this project. The compile commands can be automatically generated.

Provided by IBM plugin "iSeries Tooling - Styles Plugin" version 5.0.0

CL Program Build Style Configuration

Enter the project's source file that contains the COMPILE.CLLE and BIND.CLLE members that will be uploaded to build this project.

Build Source File: QCLSRC

☑ Automatically generate COMPILE.CLLE prior to each build

☑ Automatically push all changed members prior to build

Specify Build Style and style configuration parameters for iSeries Project

more information on next slide ...

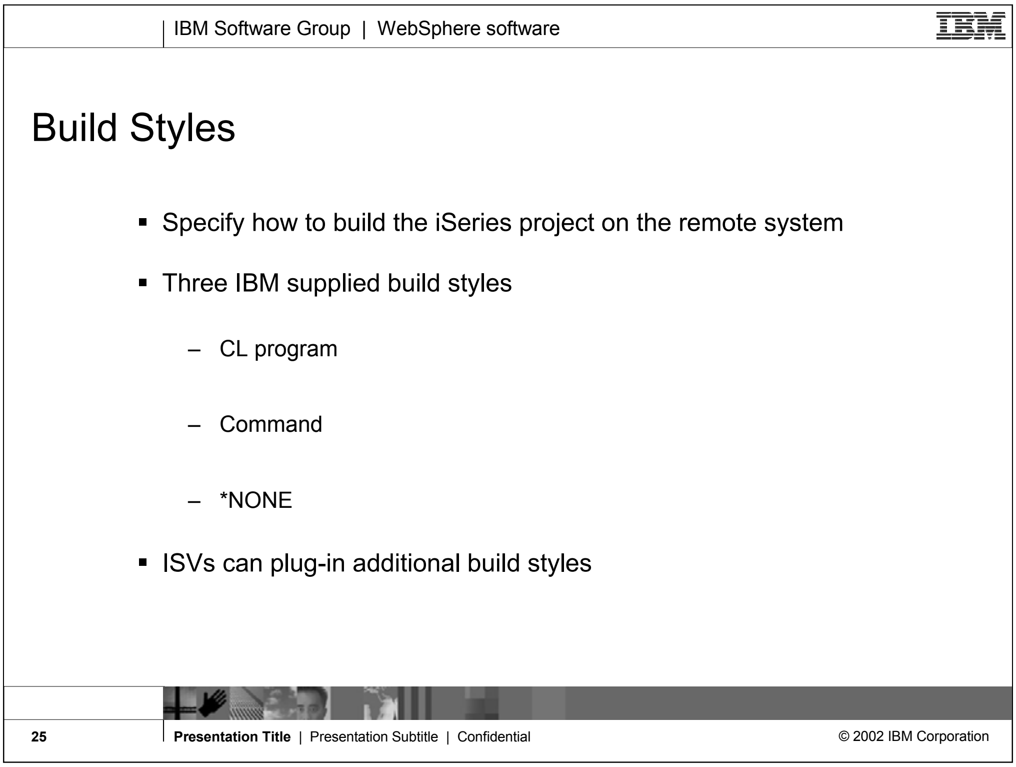
Command Build Style Configuration

Enter the command to be submitted to build this project.

Build Command:

☐ Automatically push all changed members prior to build

Here we see the rest of the iSeries Project wizard. Here you specify a CL command to call for the "Build" action. If you don't have such a CL command, enter a dummy value or leave it blank.



Build Styles

- Specify how to build the iSeries project on the remote system
- Three IBM supplied build styles
  - CL program
  - Command
  - *NONE
- ISVs can plug-in additional build styles

There are three IBM-supplied build styles:

***NONE:** This build style contributes no actions and is useful if you want to disable the menu items associated with a build style. This style will always be available.

Command: This build style runs a single command in batch to compile all the members of the project. The command can be configured through the build style's configuration page. This build style contributes the following three actions to the *Remote Actions* menu found in the iSeries Projects Navigator view:

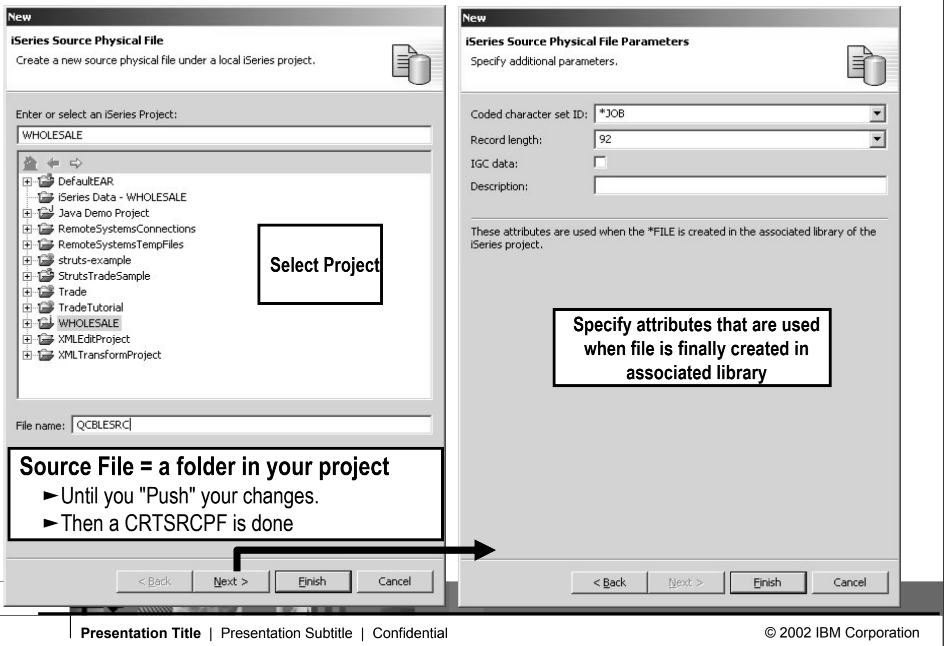
Push Changes: Pushes any new or changed resources found in the selected project. An item is changed if it has been updated locally since the last time it was pushed, either by Push Selected or Push Changes. The first time it is done, all resources are uploaded to the associated library. Each subsequent upload is incremental.

Push Selected: This action uploads all selected resources and their children to their corresponding iSeries source files and members in the associated library. It creates any resources that do not exist. It will overwrite any changes that have been made on the host system only if confirmed by the user. Use this action to force the contents of the associated library to match the workspace.

Submit Build Command: Submits the command found in this build style configuration to the host.

CL Program: This build style uploads the COMPILE.CLLE and BIND.CLLE members from the source file that you designated in the build style configuration for this project. It then compiles and runs them to build this project on the host system. This build style contributes the following actions to the *Remote Actions*

iSeries Source Physical File Wizard



New

iSeries Source Physical File

Create a new source physical file under a local iSeries project.

Enter or select an iSeries Project: WHOLESALE

- DefaultEAR
- iSeries Data - WHOLESALE
- Java Demo Project
- RemoteSystemsConnections
- RemoteSystemsTempFiles
- struts-example
- StrutsTradeSample
- Trade
- TradeTutorial
- WHOLESALE
- XMLEditProject
- XMLTransformProject

File name: QCBLESRC

Select Project

Source File = a folder in your project

- ►Until you "Push" your changes.
- ►Then a CRTSRCPF is done

New

iSeries Source Physical File Parameters

Specify additional parameters.

Coded character set ID: *JOB

Record length: 92

IGC data: □

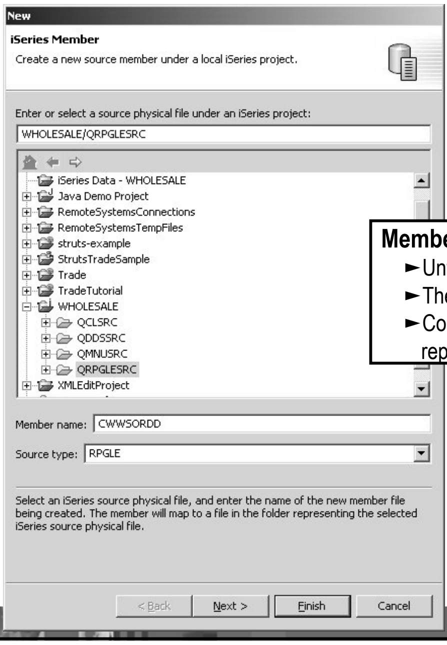
Description:

These attributes are used when the *FILE is created in the associated library of the iSeries project.

Specify attributes that are used when file is finally created in associated library

Here we see the wizard for creating a new iSeries source physical file. Within the project, this is actually a folder. When the project is pushed to its associated library, this result in a CRTSRCPF command being run to create a file with the attributes specified in this wizard.

iSeries Member wizard



New

iSeries Member

Create a new source member under a local iSeries project.

Enter or select a source physical file under an iSeries project:

WHOLESALE/QRPGLESRC

- iSeries Data - WHOLESALE
- Java Demo Project
- RemoteSystemsConnections
- RemoteSystemsTempFiles
- struts-example
- StrutsTradeSample
- Trade
- TradeTutorial
- WHOLESALE
  - QCLSRC
  - QDDSSRC
  - QMNUSRC
  - QRPGLESRC
- XMLEditProject

Member name: CWWSORDD

Source type: RPGLE

Select an iSeries source physical file, and enter the name of the new member file being created. The member will map to a file in the folder representing the selected iSeries source physical file.

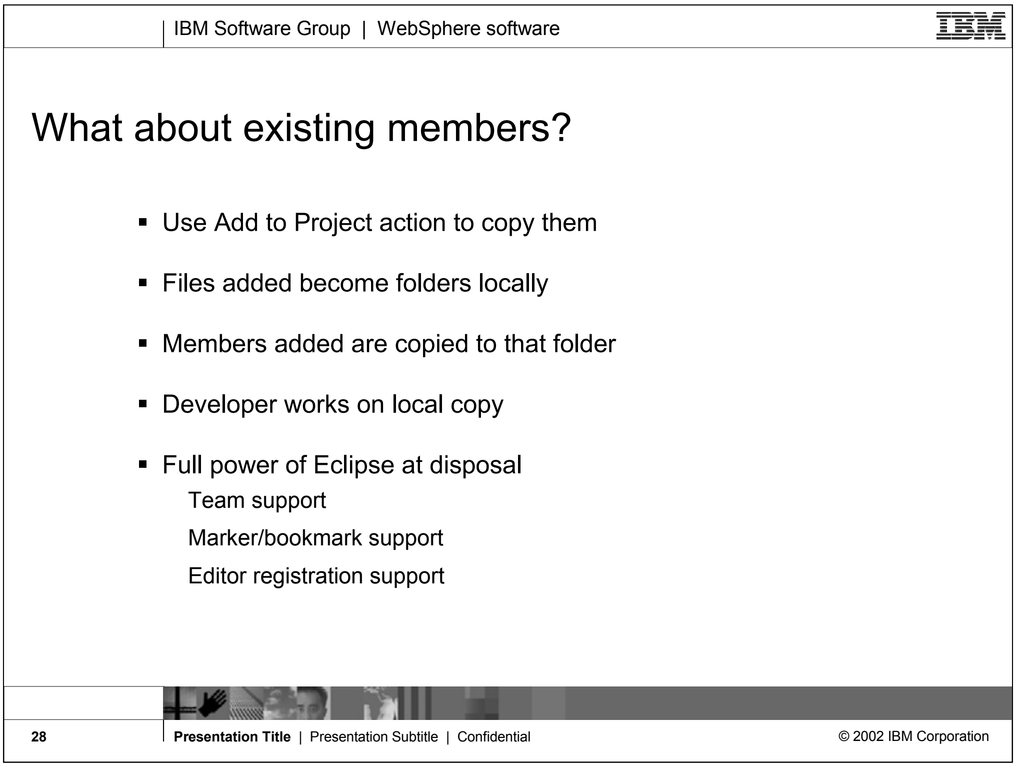
Select the parent folder (src file)

Enter the member name and select source type

Member = a file in your folder

- ►Until you "Push" your changes.
- ►Then an ADDPFM is done, or
- ►Contents of remote member are replaced

Here we see the wizard for creating a new iSeries source member within an iSeries project. Actually, it is within a source phyical file (aka folder) within an iSeries project. Locally, a member is really a file on disk. The file's extension is the member's type, as in ABC.RPGLE. When this project is pushed to its associated library, this will result in an ADDPFM command being run to create the file with the name and type, in its parent file.



What about existing members?

- Use Add to Project action to copy them
- Files added become folders locally
- Members added are copied to that folder
- Developer works on local copy
- Full power of Eclipse at disposal
  - Team support
  - Marker/bookmark support
  - Editor registration support

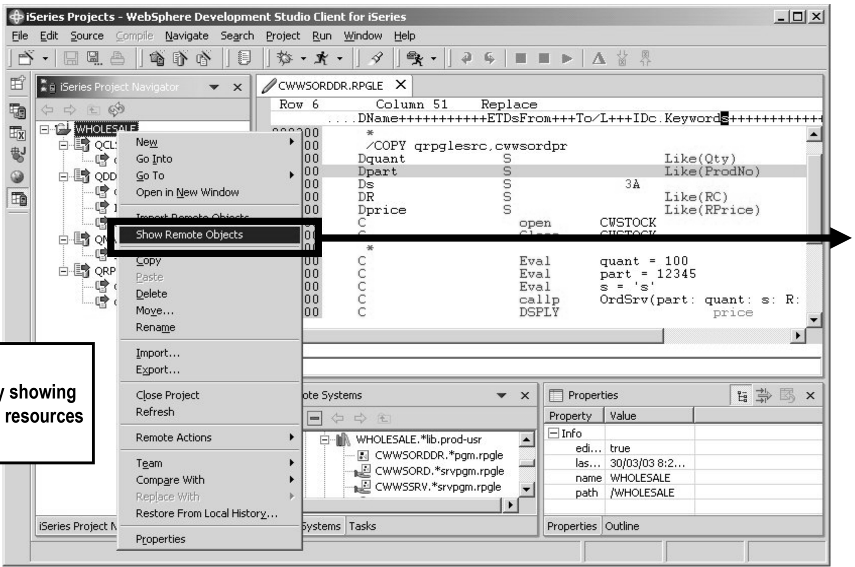
Add project action

- •Right click on project, select “Show Remote Objects”
- •Drill down to source file, right-click, select “Add to Project”
- •Drill down to members, right-click, select “Add to Project”

Developer works on local copy

- •Edits file
- •Pushes changes using “Push” action
- •Does a build using “Build” action
- •Shares with team using “Synchronize” action

Project Bleedthrough



```
Row 6       Column 51    Replace
    ....DName+++++++++++ETDsFrom+++To/L+++IDc.Keywords+++++++++++
00    *
00     /COPY qrpglesrc,cwwsordpr
00    Dquant              S                          Like(Qty)
00    Dpart               S                          Like(ProdNo)
00    Ds                  S                    3A
00    DR                  S                          Like(RC)
00    Dprice              S                          Like(RPrice)
00    C                     open      CWSTOCK
00    C                     Close     CWSTOCK
00    *
00    C                     Eval      quant = 100
00    C                     Eval      part = 12345
00    C                     Eval      s = 's'
00    C                     callp     OrdSrv(part: quant: s: R:
00    C                     DSPLY                            price
```

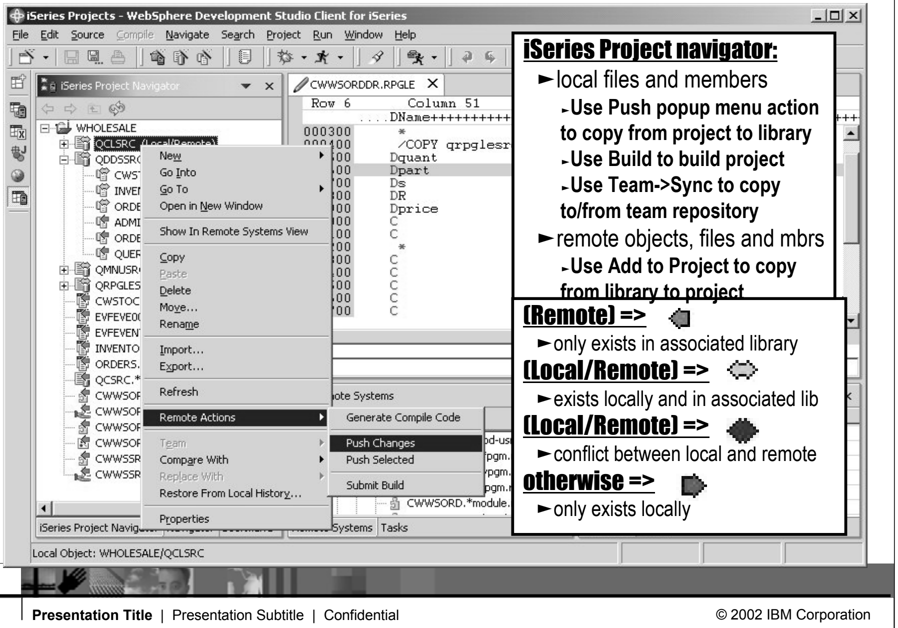
WHOLESALE.*lib.prod-usr
- CWWSORDDR.*pgm.rpgle
- CWWSORD.*srvpgm.rpgle
- CWWSSRV.*srvpgm.rpgle

Property	Value
Info	
edi...	true
las...	30/03/03 8:2...
name	WHOLESALE
path	/WHOLESALE

Currently showing only local resources

Here we see an iSeries project in the iSeries Navigator, within the iSeries Projects perspective, that has a number of local files and members. Right-clicking on the project or anything in the project, gives a popup menu with the all-important "Show Remote Objects" menu item....

iSeries Project Navigator



```
Row 6        Column 51
     ....DName+++++++++++
000300        *
000400        /COPY qrpglesr
   00    Dquant
   00    Dpart
   00    Ds
   00    DR
   00    Dprice
   00    C
   00    C
   00     *
   00    C
   00    C
   00    C
   00    C
   00    C
```

Local Object: WHOLESALE/QCLSRC

iSeries Project navigator:

- ►local files and members
  - ‣Use Push popup menu action to copy from project to library
  - ‣Use Build to build project
  - ‣Use Team->Sync to copy to/from team repository
- ►remote objects, files and mbrs
  - ‣Use Add to Project to copy from library to project

(Remote) =>
►only exists in associated library

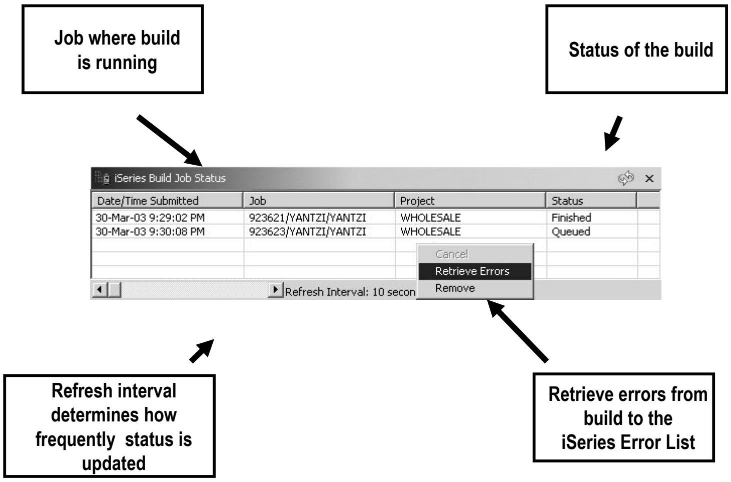
(Local/Remote) =>
►exists locally and in associated lib

(Local/Remote) =>
►conflict between local and remote

otherwise =>
►only exists locally

Once "Show Remote Objects" is enabled (it is a toggle) all the objects in the associated library for this project are shown in the iSeries Project Navigator. This navigator is called a "bleed through" view because it lists both local files/members and remote objects, files and members. However, if a file or member exists both locally and remotely, it is not shown twice. Rather, it is only shown once, and its icon and bracketed text identify it as existing in both places. The icons and text enable you to see at a glance which files/members exist only locally, only remotely, or both locally and remotely. In the latter case, the color of the icon indicates if the two are in synch or not. If not, it is an indication you need to "push" your local source to the associated library to get it in synch.

Submit Build



Job where build is running

Status of the build

iSeries Build Job Status

Date/Time Submitted	Job	Project	Status
30-Mar-03 9:29:02 PM	923621/YANTZI/YANTZI	WHOLESALE	Finished
30-Mar-03 9:30:08 PM	923623/YANTZI/YANTZI	WHOLESALE	Queued

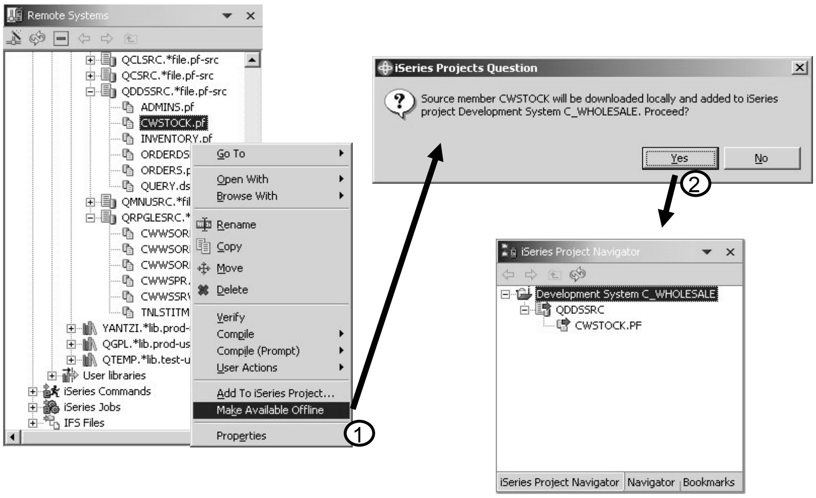
Refresh Interval: 10 secon

Refresh interval determines how frequently status is updated

Retrieve errors from build to the iSeries Error List

Once code has been edited locally and verified against any syntax errors, we are ready to perform a build (or compile) on the source members under a given iSeries project. Associated with each iSeries project is a **Build Style** which basically determines how source members under that project will be pushed and compiled on the host. A build style is a set of actions that provide push and build operations for a given iSeries Projects.

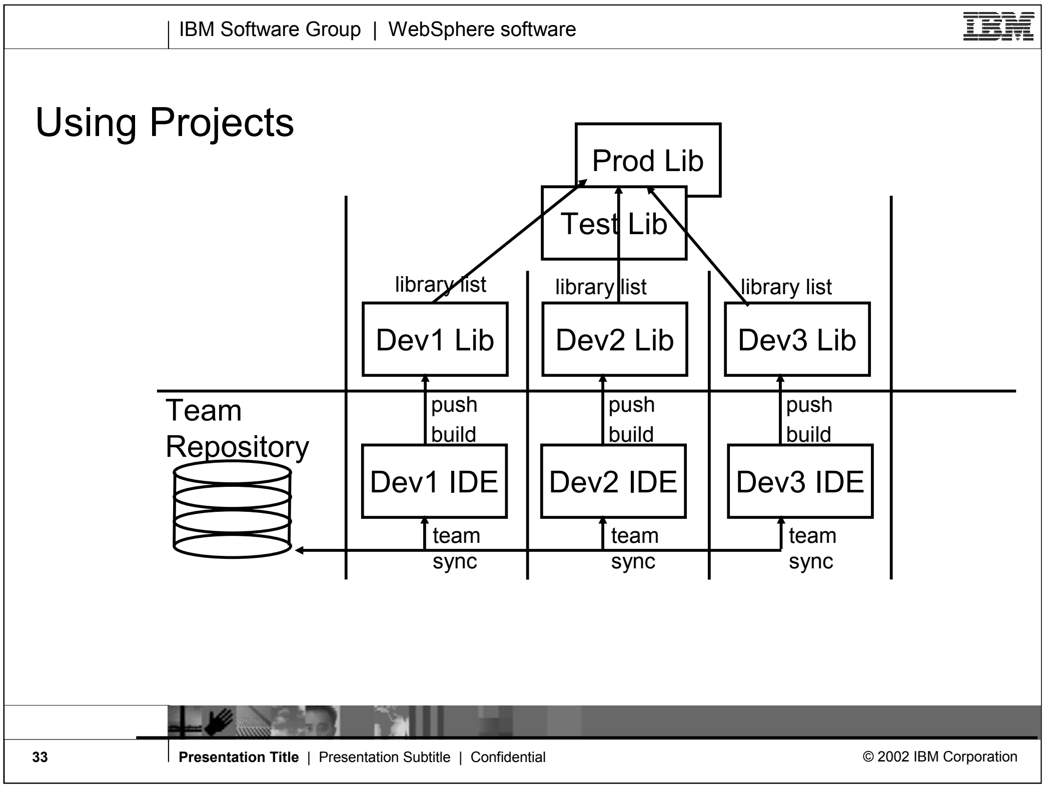
Disconnected development



Source member CWSTOCK will be downloaded locally and added to iSeries project Development System C_WHOLESALE. Proceed?

Offline development allows for a disconnected mode of development where a live connection to a host is only required when code updates or builds are needed.

The **Make Available Offline** action downloads the selected files or members locally to their corresponding iSeries project. The corresponding iSeries project has the same host and the same library properties as the host name and the library of the selected source physical files or members. If such an iSeries project is not found, one will automatically be created. In our case, this action created a new iSeries Project with the name being connectionName_libraryName. If there was one or more existing iSeries projects which had the same connection and mapped to the same library as the Associated Library, then this action would have prompted you to select one of these existing projects.



Using Projects

Prod Lib

Test Lib

library list

library list

library list

Dev1 Lib

Dev2 Lib

Dev3 Lib

Team Repository

push build

push build

push build

Dev1 IDE

Dev2 IDE

Dev3 IDE

team sync

team sync

team sync

The typical usage scenario for iSeries projects is to use them for tasks, such as adding a feature to an existing application. Rarely will you use an iSeries project to hold all the source for an application... use an iSeries change management vendor for that. Consider a change that involves 3 developers changing a number of files...

The lead developer will create the iSeries project and synchronize it with the repository. The other developers will use the Eclipse team support to add that project to their workspace. Each will change the properties of the project to associate it with their own private library. The master library containing all the source will be on their library list, so compiles will work, but their private library will only hold the members they work on. If they use an SCM product, then they will use it to check source members out to their private library. These members and files will then be imported into their project, where they will edit them.

After editing, they will "push" their changes to their library and do a "Build" or a compile, and run and test their unique changes.

Every so often, they will do a team synchronization to give their changes to their fellow team members, and pick up the changes of those team members. After a synch, they will push their colleaque's files to their own associated library and test the changes all work together.

At the end of the cycle, someone will pick one of the libraries, and use the SCM product to promote their changes back into the test and production stages.

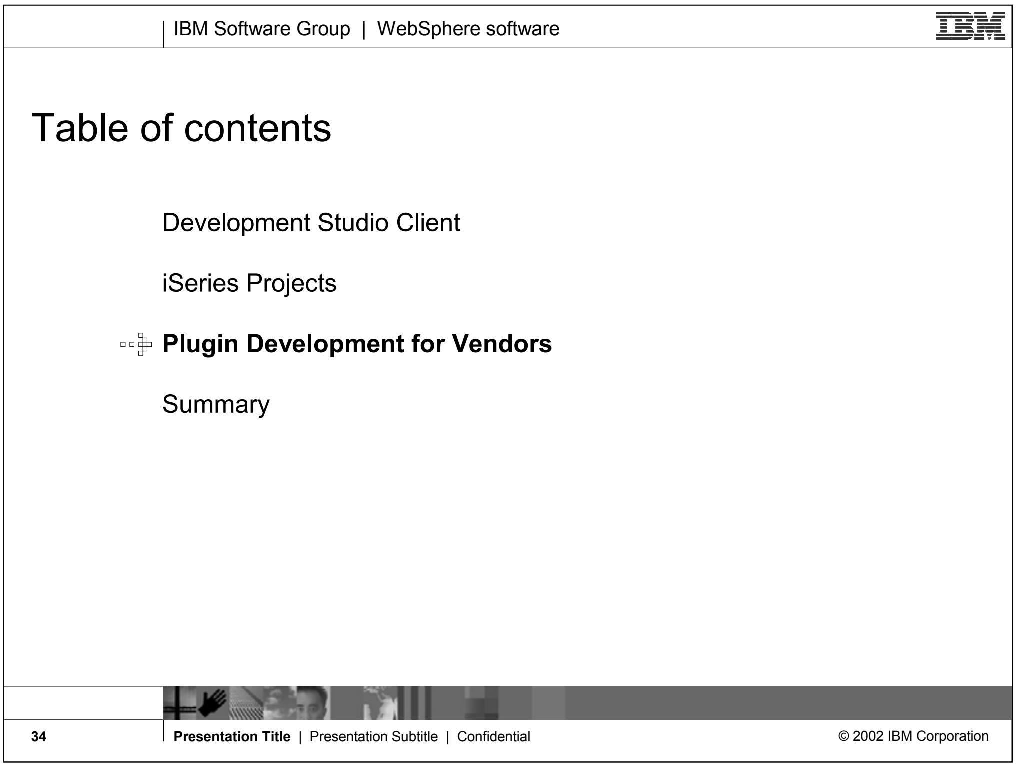


Table of contents

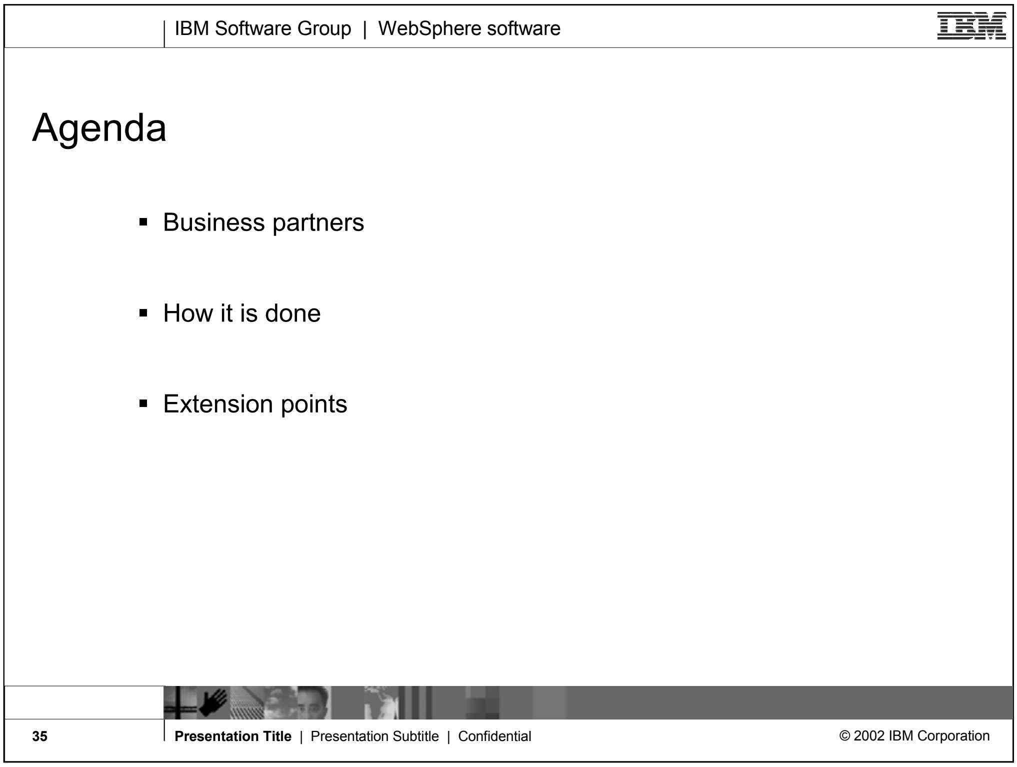
Development Studio Client

iSeries Projects

Plugin Development for Vendors

Summary

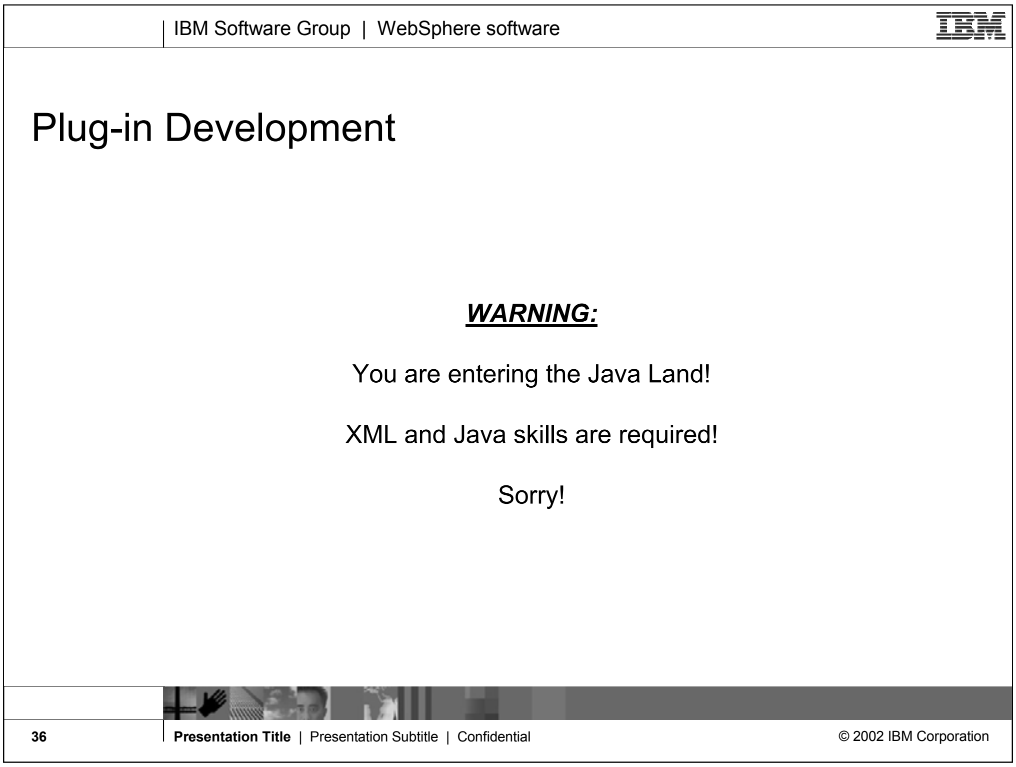
Now we have reviewed all the features of the iSeries Projects. Business partners can easily leverage and extend iSeries Projects.



Agenda

- Business partners
- How it is done
- Extension points

This section describes business partners role in extending RSE, how to extend and what the extension points are.



Plug-in Development

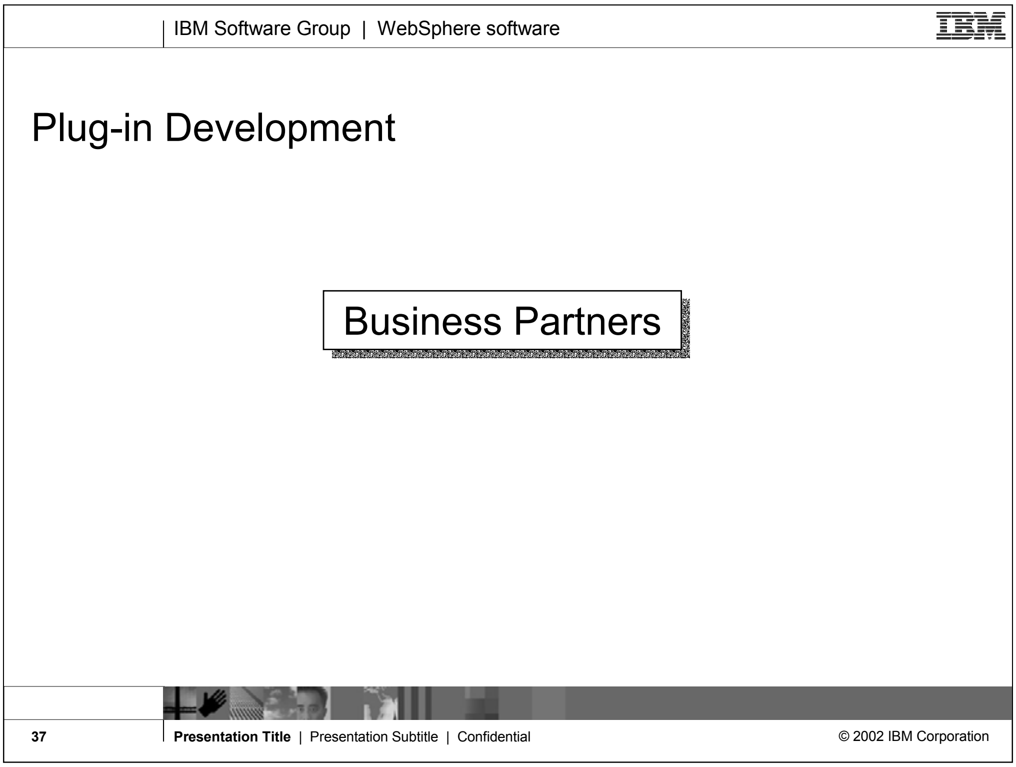
WARNING:

You are entering the Java Land!

XML and Java skills are required!

Sorry!

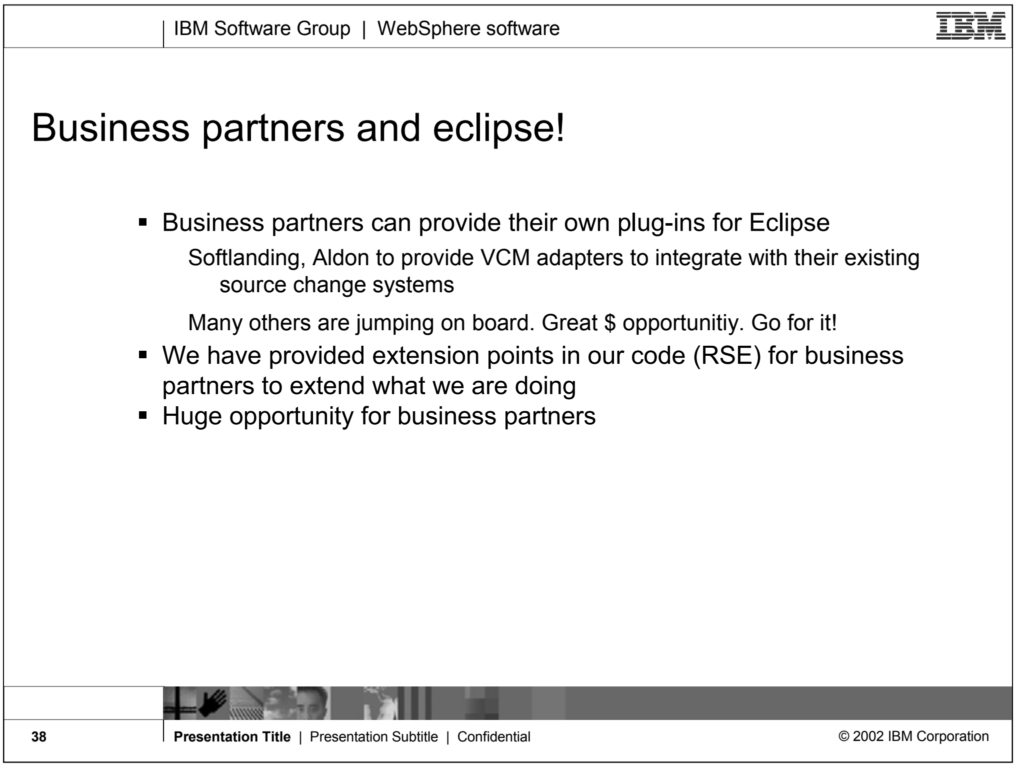
To do plug-in development you must have Java and XML skills.



Plug-in Development

Business Partners

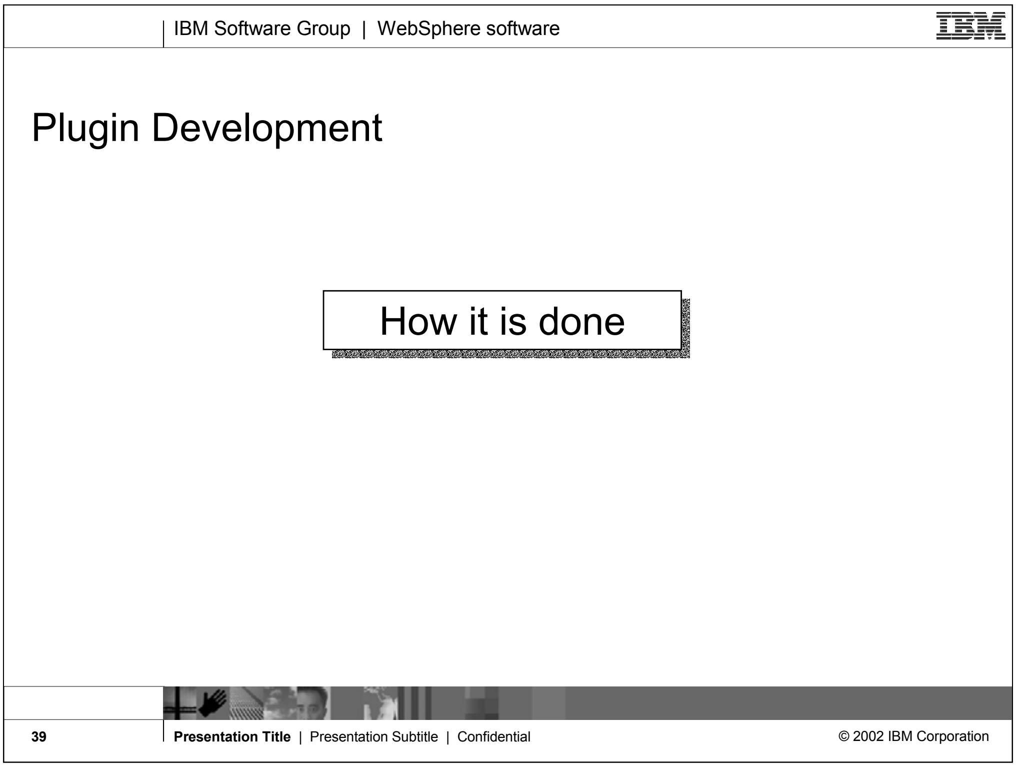
Many leading iSeries tool vendors are actively integrating their tools into this new release. These vendors include, but are not limited to change management and impact analysis tool suppliers.



Business partners and eclipse!

- Business partners can provide their own plug-ins for Eclipse
  - Softlanding, Aldon to provide VCM adapters to integrate with their existing source change systems
  - Many others are jumping on board. Great $ opportunitiy. Go for it!
- We have provided extension points in our code (RSE) for business partners to extend what we are doing
- Huge opportunity for business partners

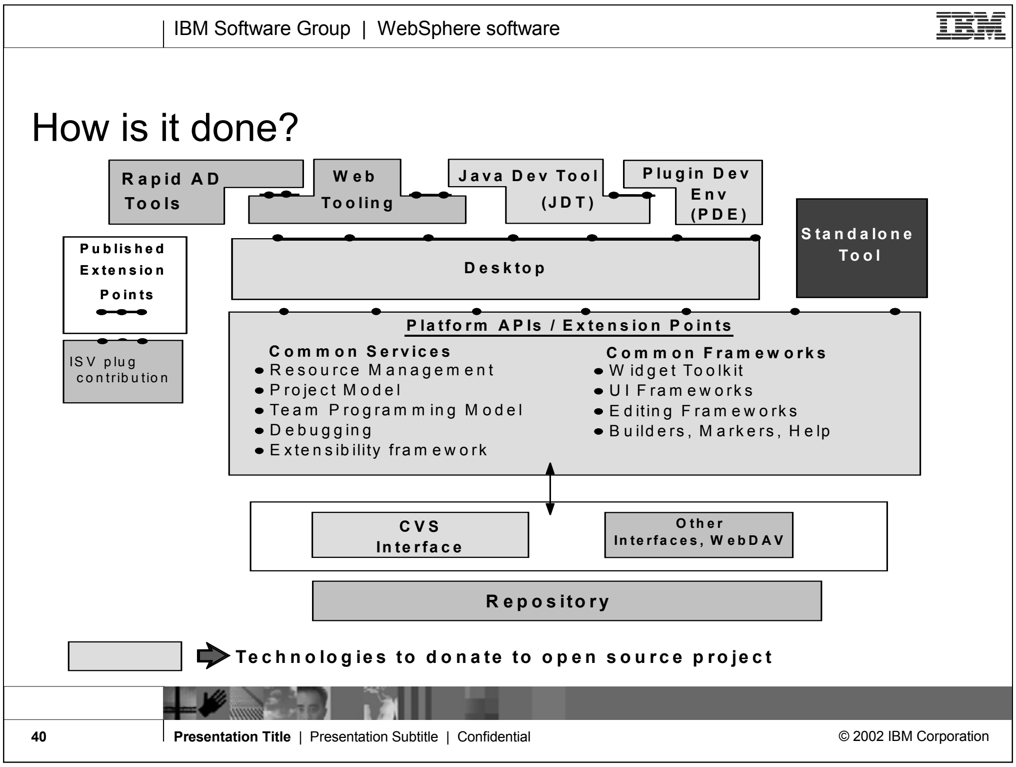
Business partners can provide their own tools and plug them into the Remote System Explorer. This is a tremendous opportunity for business partners.



Plugin Development

How it is done

Here is how plug-in your tools.



How is it done?

Rapid AD Tools

Web Tooling

Java Dev Tool (JDT)

Plugin Dev Env (PDE)

Standalone Tool

Published Extension Points

ISV plug contribution

Desktop

Platform APIs / Extension Points

Common Services
- Resource Management
- Project Model
- Team Programming Model
- Debugging
- Extensibility framework

Common Frameworks
- Widget Toolkit
- UI Frameworks
- Editing Frameworks
- Builders, Markers, Help

CVS Interface

Other Interfaces, WebDAV

Repository

▭ ➡ Technologies to donate to open source project

You can extend the Remote System Explorer by authoring Java code that adds additional items to the perspective, such as pop-up menu actions, property pages, and even subsystems. This feature is valuable if you are a business partner or independent software vendor, and you want to tightly integrate your tools with the Remote System Explorer.

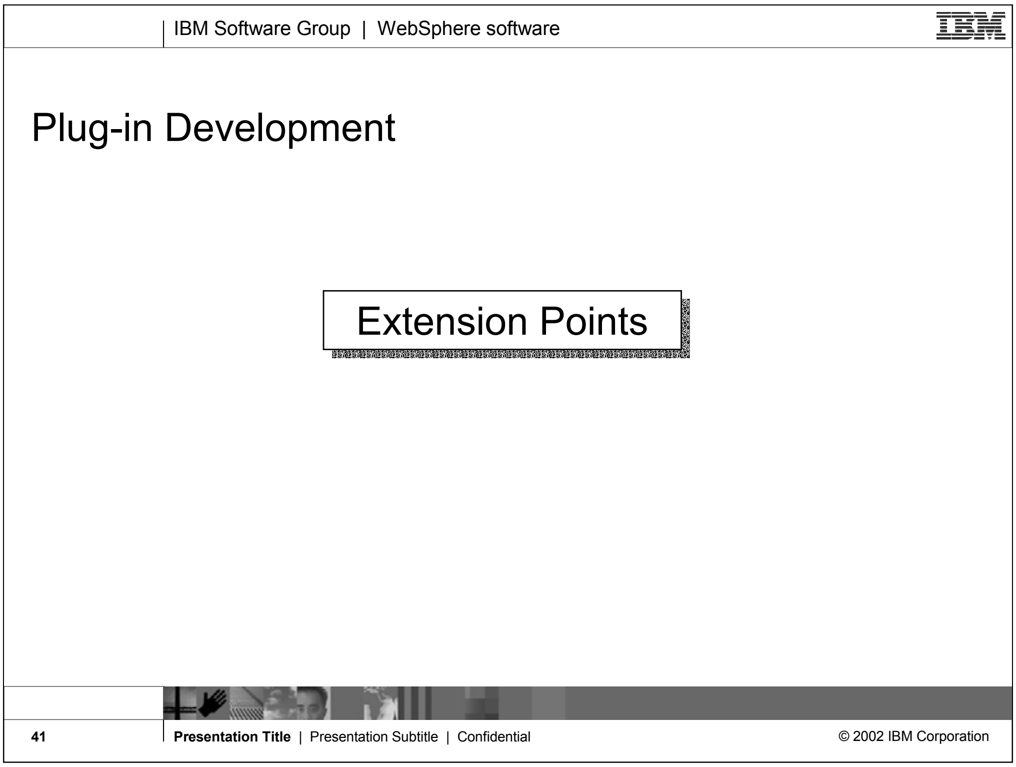
Programmer documentation for extending the Remote System Explorer is currently available in English only. To retrieve the documentation:

Navigate to the plug-in folder com.ibm.etools.systems.doc.isv, in the iseries\plugins directory where you installed this product.

Rename the file toc.xml.off to toc.xml.

Close and re-start the workbench.

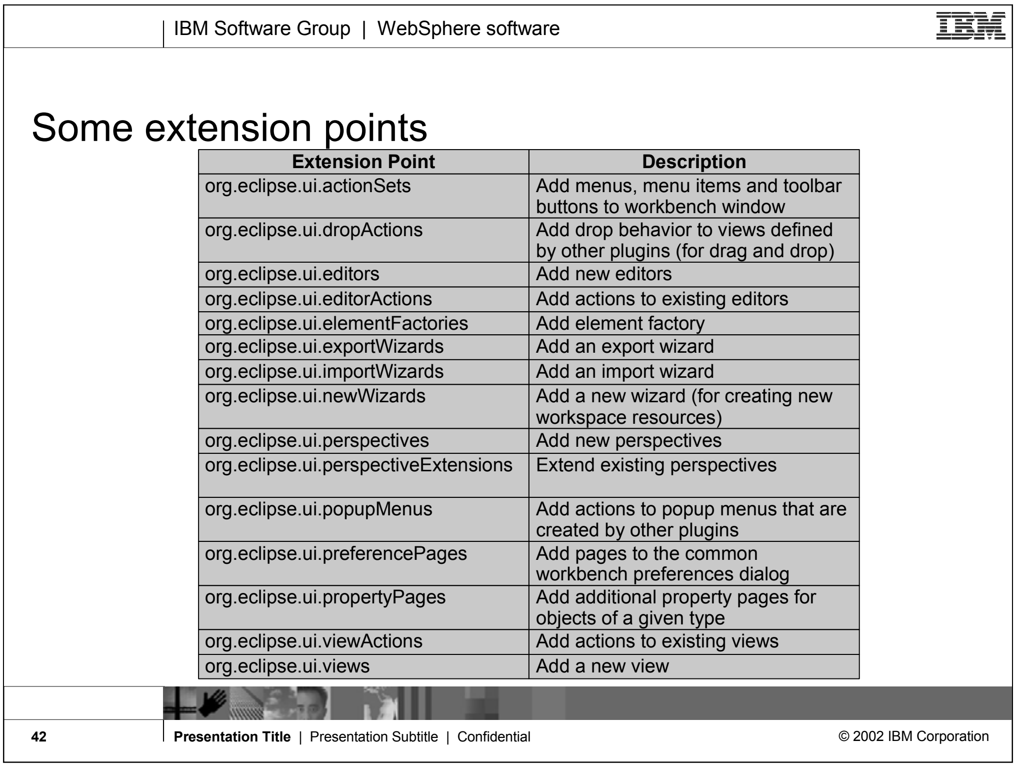
Select Help Contents from the Help menu, and click Extending the Remote System Explorer (RSE) in the Contents frame to read the new programmer documentation.



Plug-in Development

Extension Points

Extension points allow you to plug-in your tools into RSE.



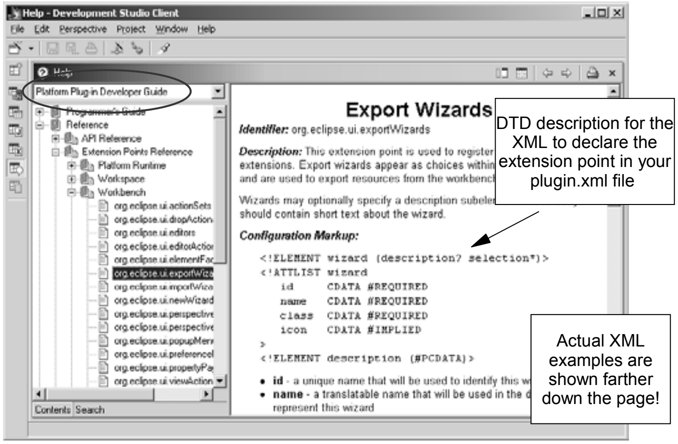
Some extension points

Extension Point	Description
org.eclipse.ui.actionSets	Add menus, menu items and toolbar buttons to workbench window
org.eclipse.ui.dropActions	Add drop behavior to views defined by other plugins (for drag and drop)
org.eclipse.ui.editors	Add new editors
org.eclipse.ui.editorActions	Add actions to existing editors
org.eclipse.ui.elementFactories	Add element factory
org.eclipse.ui.exportWizards	Add an export wizard
org.eclipse.ui.importWizards	Add an import wizard
org.eclipse.ui.newWizards	Add a new wizard (for creating new workspace resources)
org.eclipse.ui.perspectives	Add new perspectives
org.eclipse.ui.perspectiveExtensions	Extend existing perspectives
org.eclipse.ui.popupMenus	Add actions to popup menus that are created by other plugins
org.eclipse.ui.preferencePages	Add pages to the common workbench preferences dialog
org.eclipse.ui.propertyPages	Add additional property pages for objects of a given type
org.eclipse.ui.viewActions	Add actions to existing views
org.eclipse.ui.views	Add a new view

Here are a list of extension points and what they do.

So where do you find all this information?

- The online help!


Help - Development Studio Client
File Edit Perspective Project Window Help
Help
Platform Plug-in Developer Guide
Programmer's Guide
Reference
API Reference
Extension Points Reference
Platform Runtime
Workspace
Workbench
org.eclipse.ui.actionSets
org.eclipse.ui.dropAction
org.eclipse.ui.editors
org.eclipse.ui.editorActior
org.eclipse.ui.elementFac
org.eclipse.ui.exportWiza
org.eclipse.ui.importWiza
org.eclipse.ui.newWizard
org.eclipse.ui.perspective
org.eclipse.ui.perspective
org.eclipse.ui.popupMen
org.eclipse.ui.preferencel
org.eclipse.ui.propertyPa
org.eclipse.ui.viewAction
Contents Search

Export Wizards

Identifier: org.eclipse.ui.exportWizards

Description: This extension point is used to register extensions. Export wizards appear as choices within and are used to export resources from the workbench

Wizards may optionally specify a description subele should contain short text about the wizard.

Configuration Markup:

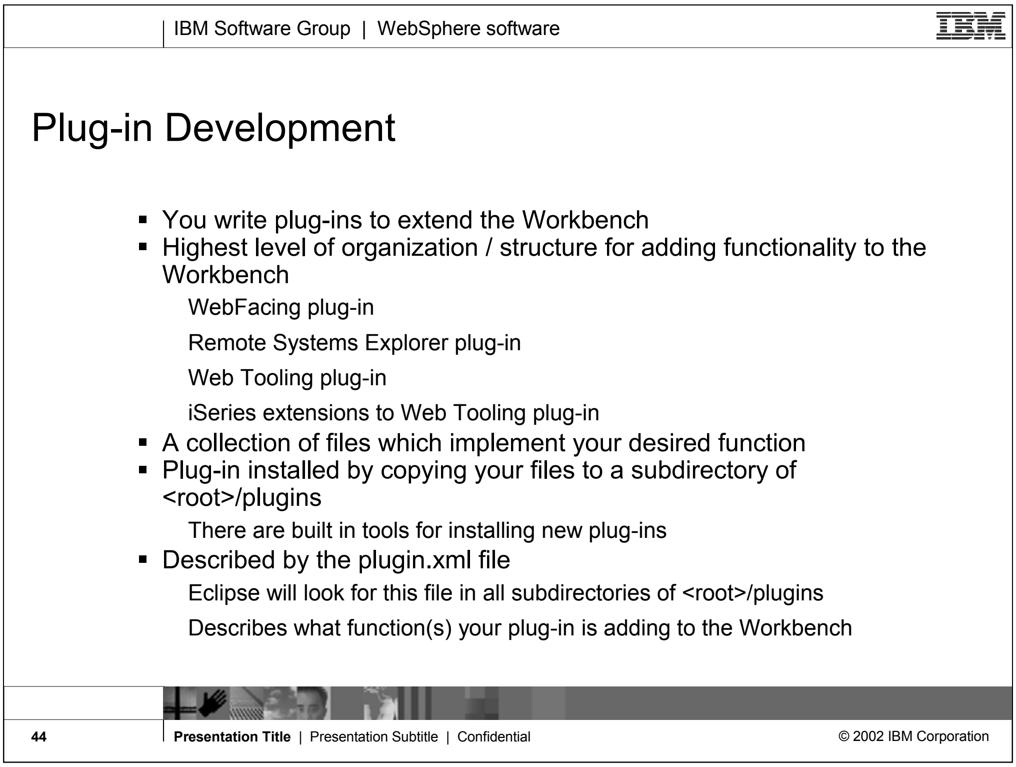
```
<!ELEMENT wizard (description? selection*)>
<!ATTLIST wizard
   id      CDATA #REQUIRED
   name    CDATA #REQUIRED
   class   CDATA #REQUIRED
   icon    CDATA #IMPLIED
>
<!ELEMENT description (#PCDATA)>
```

- **id** - a unique name that will be used to identify this w
- **name** - a translatable name that will be used in the d represent this wizard

DTD description for the XML to declare the extension point in your plugin.xml file

Actual XML examples are shown farther down the page!

Use the online help to view more information on how to declare the extension point in your plugin.xml file.

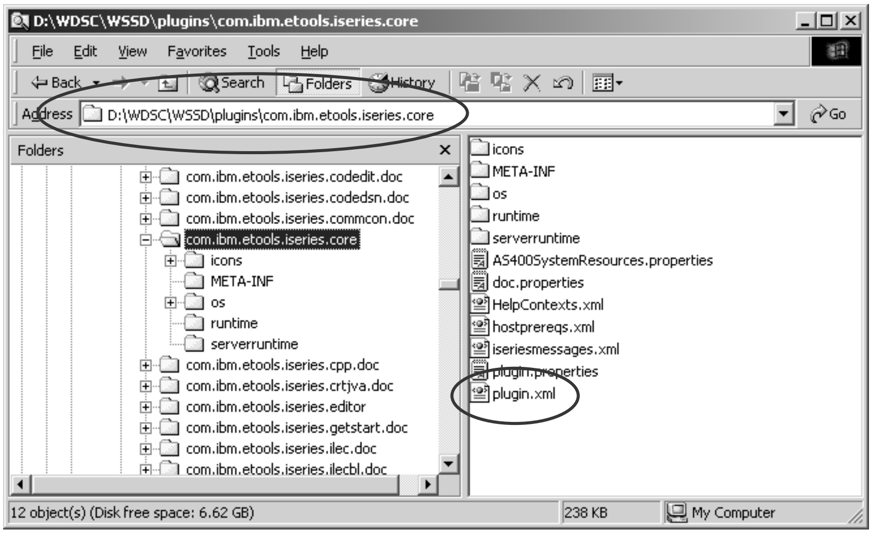


Plug-in Development

- You write plug-ins to extend the Workbench
- Highest level of organization / structure for adding functionality to the Workbench
  - WebFacing plug-in
  - Remote Systems Explorer plug-in
  - Web Tooling plug-in
  - iSeries extensions to Web Tooling plug-in
- A collection of files which implement your desired function
- Plug-in installed by copying your files to a subdirectory of <root>/plugins
  - There are built in tools for installing new plug-ins
- Described by the plugin.xml file
  - Eclipse will look for this file in all subdirectories of <root>/plugins
  - Describes what function(s) your plug-in is adding to the Workbench

Here are the steps to develop plug-in’s.

Plug-in Information



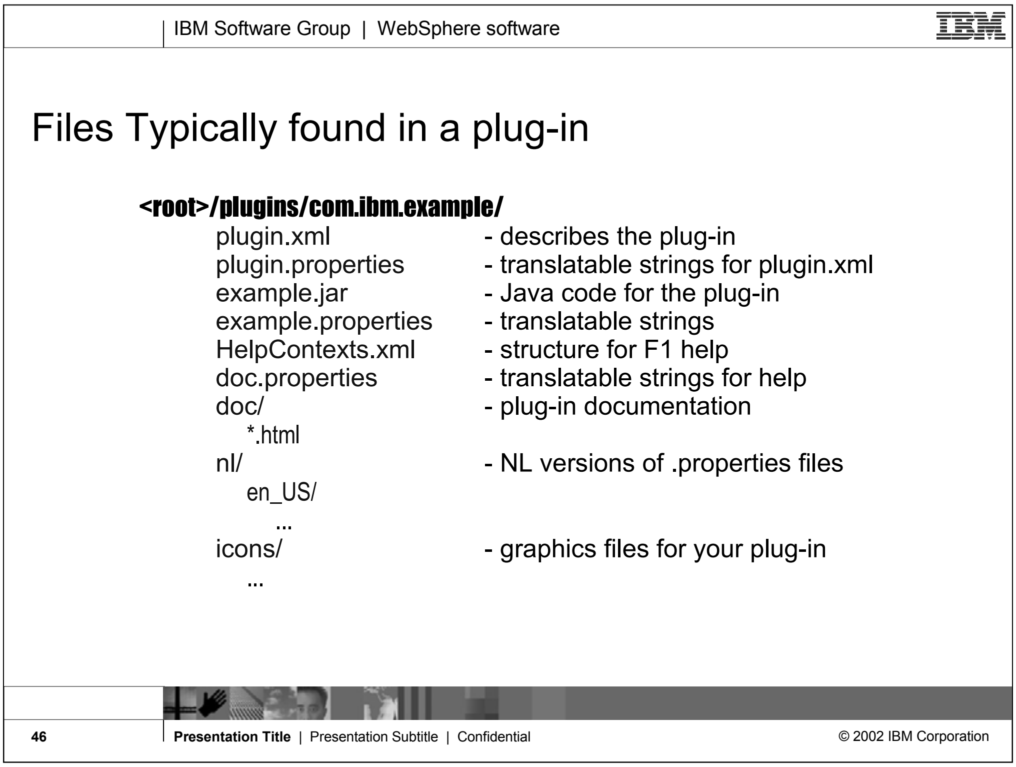
D:\WDSC\WSSD\plugins\com.ibm.etools.iseries.core
File Edit View Favorites Tools Help
Back Search Folders History
Address D:\WDSC\WSSD\plugins\com.ibm.etools.iseries.core
Go
Folders

- com.ibm.etools.iseries.codedit.doc
- com.ibm.etools.iseries.codedsn.doc
- com.ibm.etools.iseries.commcon.doc
- com.ibm.etools.iseries.core
  - icons
  - META-INF
  - os
  - runtime
  - serverruntime
- com.ibm.etools.iseries.cpp.doc
- com.ibm.etools.iseries.crtjva.doc
- com.ibm.etools.iseries.editor
- com.ibm.etools.iseries.getstart.doc
- com.ibm.etools.iseries.ilec.doc
- com.ibm.etools.iseries.ilecbl.doc

icons
META-INF
os
runtime
serverruntime
AS400SystemResources.properties
doc.properties
HelpContexts.xml
hostprereqs.xml
iseriesmessages.xml
plugin.properties
plugin.xml

12 object(s) (Disk free space: 6.62 GB) | 238 KB | My Computer

Here is the location of the plug-in information.

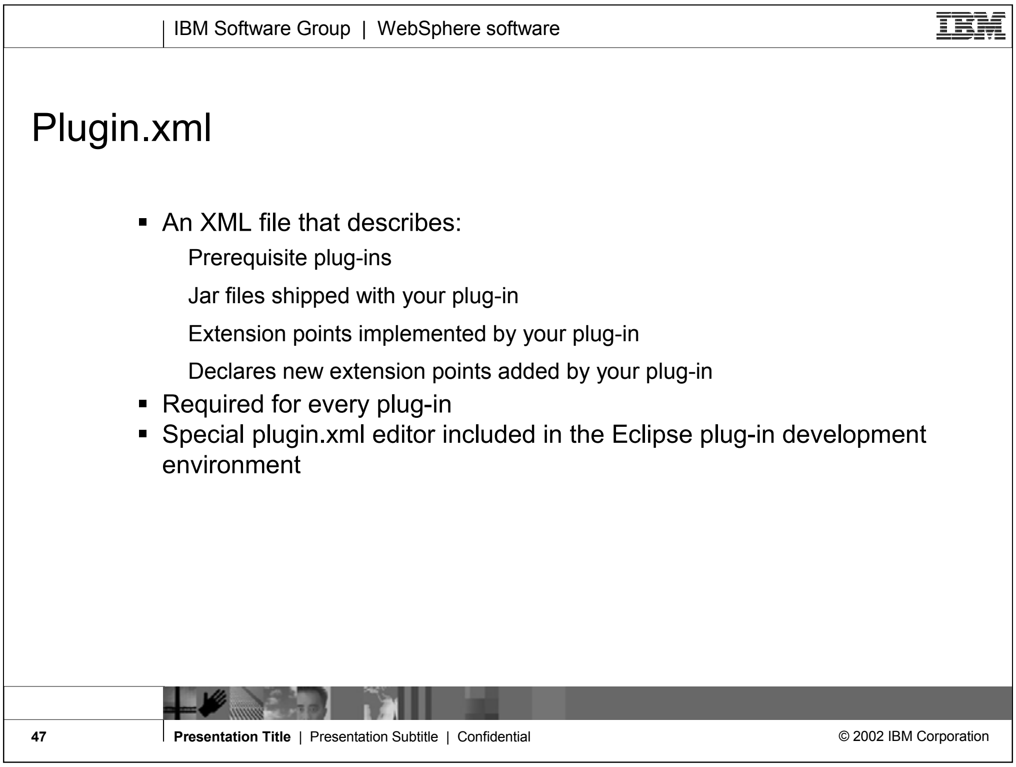


Files Typically found in a plug-in

<root>/plugins/com.ibm.example/

plugin.xml	- describes the plug-in
plugin.properties	- translatable strings for plugin.xml
example.jar	- Java code for the plug-in
example.properties	- translatable strings
HelpContexts.xml	- structure for F1 help
doc.properties	- translatable strings for help
doc/	- plug-in documentation
*.html	
nl/	- NL versions of .properties files
en_US/	
...	
icons/	- graphics files for your plug-in
...	

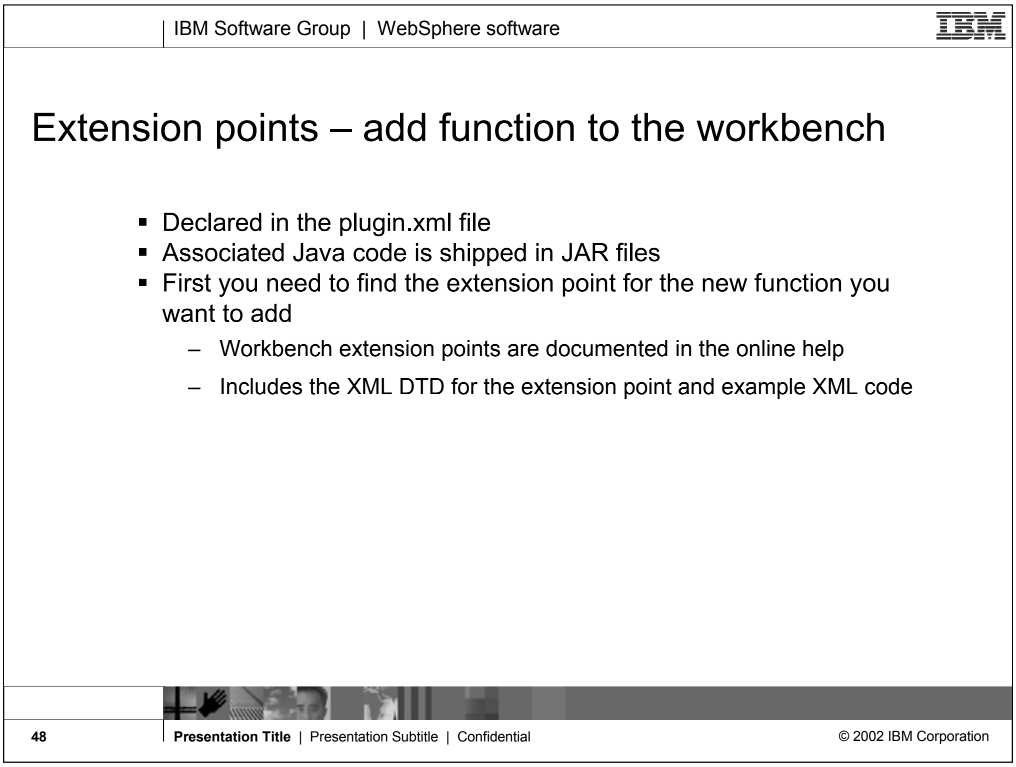
These are the files you typically find in the plug-in.



Plugin.xml

- An XML file that describes:
  - Prerequisite plug-ins
  - Jar files shipped with your plug-in
  - Extension points implemented by your plug-in
  - Declares new extension points added by your plug-in
- Required for every plug-in
- Special plugin.xml editor included in the Eclipse plug-in development environment

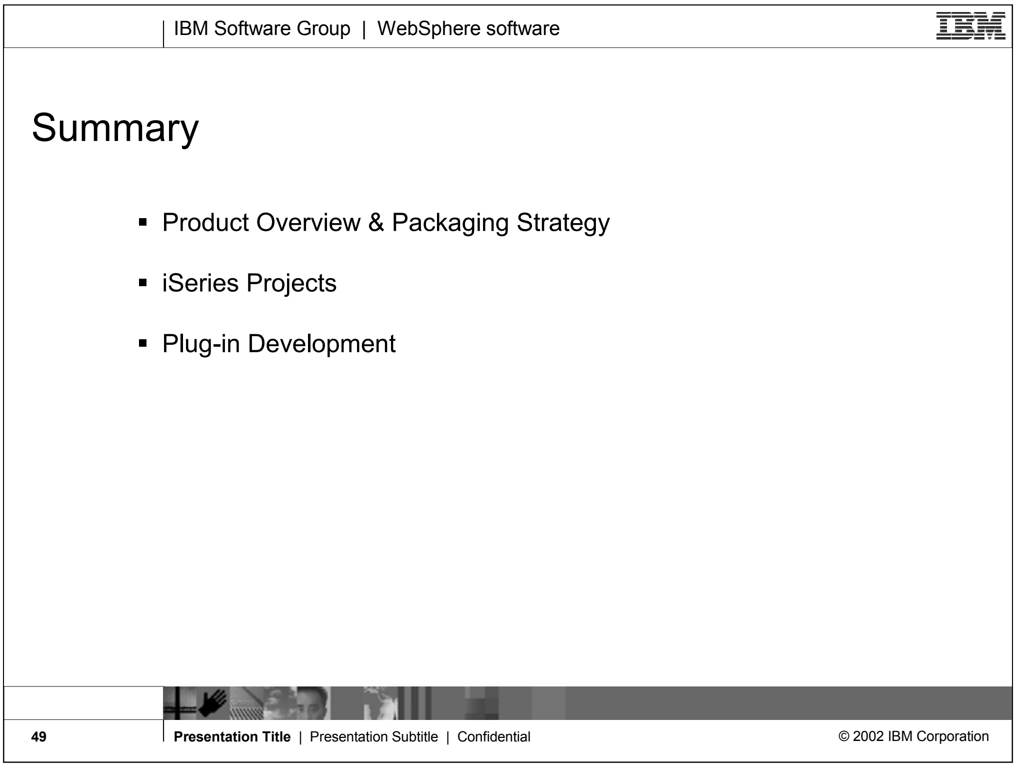
The plugin.xml file describes the extension points added by your plug-in. You need this file for your plug-in.



Extension points – add function to the workbench

- Declared in the plugin.xml file
- Associated Java code is shipped in JAR files
- First you need to find the extension point for the new function you want to add
  - Workbench extension points are documented in the online help
  - Includes the XML DTD for the extension point and example XML code

Extension points add function to the workbench and allow you to integrate your tools into the RSE. Check out the online help for information on the extension you want to use.



Summary

- Product Overview & Packaging Strategy
- iSeries Projects
- Plug-in Development

We hope this presentation helped you understand more about Development Studio Client iSeries application development tools. We started with an overview of our strategy for iSeries application development tools, went onto review the iSeries Projects, the perspective for iSeries programmers for project and team development of iSeries applications.

Legal information

Acknowledgement:

- This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:

  Phil Coulthard, George Farr, Claus Weiss, Don Yantzi, and David Slater

Disclaimer:

- The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Reproduction:

- The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

Thank you for coming.
Enjoy WebSphere Development Studio Client !