



IBM WebSphere Development Studio Client for iSeries product overview

Version 6.0 for Windows



IBM WebSphere Development Studio Client for iSeries product overview

Version 6.0 for Windows

Note

Before using this document, read the general information under Notices.

Second Edition (July 2005)

This edition applies to version 6.0.0 of WebSphere Development Studio Client for iSeries (product number D5165LL) and to all subsequent releases and modifications until otherwise indicated in new editions.

(C) Copyright IBM Corp. 2003, 2005. All rights reserved.

Contents

Chapter 1. PDF version 1

Chapter 2. Overview of IBM WebSphere Development Studio Client for iSeries . . 3

Understanding Rational and WebSphere products from IBM	4
iSeries developer roadmap.	5
Web technology choices.	6
Tutorials, scenarios, demos, and presentations	6
What's new in this release?	7
Tasks overview	10
Goal 1: Develop and maintain applications	10
Goal 2: Create Web front ends	11
Goal 3: Create GUI front ends	14
Tools overview	14
The workbench-based integrated development environment	14
iSeries development tools.	15
iSeries Web development tools	18
iSeries Java development tools	19
Struts environment support	19
Web Services development tools	20

Server development tools.	20
The IBM WebFacing Tool.	21
Integrated iSeries debugger	21
Separately Installable Features	23
VisualAge RPG	23
CoOperative Development Environment (CODE)	23

Chapter 3. Prerequisites 25

Working with your HTTP server and WebSphere Application Server	25
Server port numbers	26
Installing the sample libraries	27
Adding the sample library to your initial library list	28
Starting your WebSphere Application Server instance	29
Mapping a network drive to the iSeries system	29

Notices 31

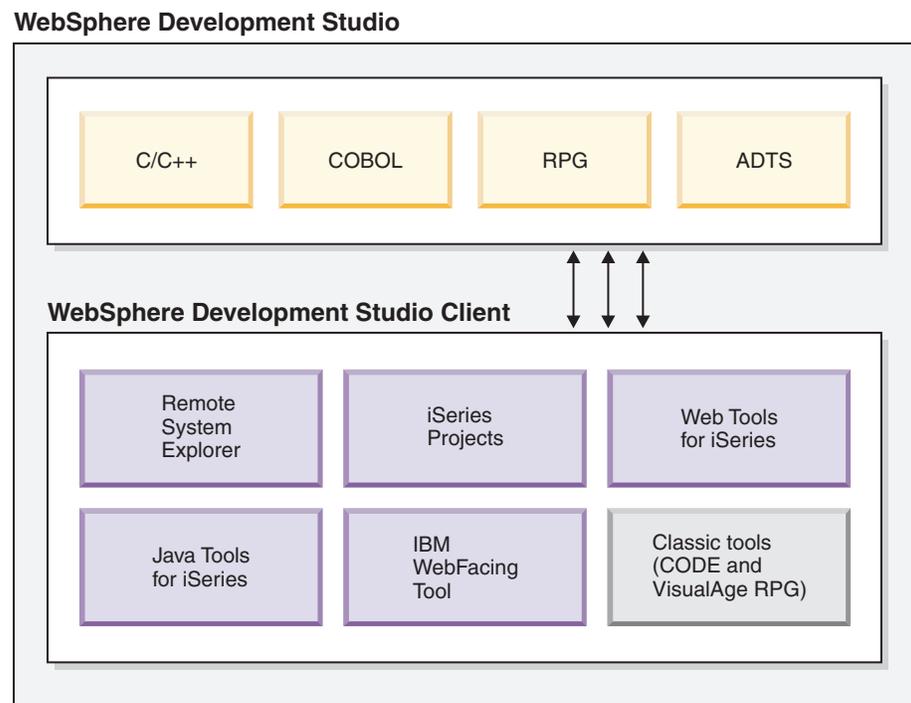
COPYRIGHT LICENSE:	32
Programming interface information	32
Trademarks and service marks	33

Chapter 1. PDF version

To open a PDF version of the product overview that you can print, click the following link: [Product overview PDF](#).

Chapter 2. Overview of IBM WebSphere Development Studio Client for iSeries

Development Studio Client is the ideal set of workstation development tools for creating, testing, deploying, and maintaining traditional and e-business applications for your iSeries™ server. Development Studio Client is included with the compiler-based server product, WebSphere® Development Studio. The following diagram illustrates the interaction between server and client tools:



The client component of the overall package is designed to help you accomplish three primary programming goals:

1. Develop and maintain iSeries applications
2. Create Web-enabled front ends to iSeries business logic
3. Create GUI front ends to iSeries business logic

You can use this product to perform all of the following tasks from a Windows® workstation.

Goal 1: Develop and maintain iSeries applications

- Connect to remote systems and perform programming tasks
- Edit, compile, run, and debug applications, whether green-screen, Web-based, or combined
- Work on iSeries applications when disconnected from the iSeries server
- Develop iSeries business logic in the Java™ programming language
- Develop in a team-based environment with version control

Goal 2: Create Web-enabled front ends to iSeries business logic

- Develop Java servlets and Java Server Pages (JSP files) that access iSeries programs and data, whether your core skills are in Java or ILE and Original Program Model (non-ILE) programming
- Customize the appearance of static HTML pages and JSP files for a Web site
- Test Web applications in a local WebSphere test environment
- Quickly add a Web interface to an existing application and deploy to an iSeries application server

Goal 3: Create GUI front ends to iSeries business logic

- Develop graphical user interfaces to RPG programs using visual design tools
- Produce Java applets, Java applications, and native Windows applications from the same source

Additional goals

- Develop XML resources
- Develop SQL statements
- Create Web services
- Generate Web pages from local and remote relational databases

Understanding Rational and WebSphere products from IBM

Both Development Studio Client and Development Studio Client Advanced Edition are built on the Rational® Software Development Platform. This platform offers a fresh look and feel for the Eclipse workbench, and helps make it easier for you to build, integrate, and extend your applications. The Rational Software Development Platform offers a tutorials gallery and a samples gallery, to help you get up and running with the product as quickly as possible. The platform also offers user roles, which you can select from the bottom-right corner of the Welcome perspective, (and from **Window > Preferences > Workbench > Capabilities**) that customize and simplify the user interface according to your programming role. These are just some of the new features provided in the Rational Software Development Platform.

The product comes in two editions for iSeries programmers. Both editions of the product are packaged with an additional base Rational product.

- Development Studio Client is packaged with Rational Web Developer. Rational Web Developer includes support for Web services, XML development tools, and core support for Java and Web development tools.
- **Advanced** Development Studio Client Advanced Edition is packaged with Rational Application Developer. This base product provides end-to-end support for the creation and maintenance of J2EE applications and Web services. It also provides extensive support for Enterprise JavaBeans™, EGL, and for Java messaging services.

The Tasks overview and Tools overview sections describe in more detail the main tasks that you can complete with the product, and the tools to use for these tasks:

- Read the Tasks overview if you are new to the product and want to understand how to choose the right tools to do your job.
- Read the Tools overview if you have used a previous version of this product or a similar product and you want to understand what new tasks you can perform with each of the tools.

Prerequisite information

1. If you have not yet installed the product, see the installation notes on the first product CD or the product DVD under `x:\install\help\help nl s\install.htm`, where nl s denotes the appropriate language.
2. You can use the integrated server and test environment for the development and testing stages of your product. To deploy and run your e-business applications in a production environment, you need to install and configure IBM WebSphere Application Server for iSeries and an HTTP server such as IBM HTTP Server for iSeries (powered by Apache) on the hosting iSeries server. Some configuration instructions and links to other sources of documentation are included in this guide.
3. You can also deploy your Web applications to WebSphere Application Server – Express. This is an entry-priced, entry-level server in the WebSphere Application Server family, providing support for static content, servlets, and JSP files.
4. If you purchased Development Studio Client as a workstation-only product, you need to ensure that is WebSphere Development Studio installed on any iSeries server where you perform the server portion of your development work. However, WebSphere Development Studio is not required on the iSeries server you use for Web application deployment.

iSeries developer roadmap

The iSeries developer roadmap describes discrete, achievable steps that move developers and applications on the path to an excellently implemented Web future. Search for the iSeries developer roadmap on <http://www.ibm.com/servers/index.html> and click the graphic to see the roadmap image.

Starting from the left, this is where most IT shops that use the iSeries platform and Business Partners are today. Typically, day-to-day development tasks still involve building and maintaining green-screen applications via traditional 5250 tools such as Programming Development Manager (PDM), Source Entry Utility (SEU), and Screen Design Aid (SDA).

The first step to Web application enhancement involves embracing modern tools to do the same development work previously accomplished via PDM, SEU, and SDA (see the "Better Tools" column).

The next step, which is considered to be urgent by end users, is a better user interface (UI) than the generations-old green screen. For most applications, this is best addressed by moving to a browser-based user interface.

"Better Architecture" is a significant step where—from scratch or from cut-and-paste—you create a Web application, an application enhancement, or even a new Web service. Separating the business logic from the user interface is a very important experience, and is fundamental to a superior architecture that allows for logic reuse.

"Better Portability" involves a move from creating business logic in traditional languages to writing it in Java. You use simple, standard Java—referred to as Java 2 Standard Edition (J2SE)—that accesses data in the familiar SQL ways.

Finally, on the far right, there is "Better Scalability." For companies requiring highly scalable Web applications or the full object-oriented and functional power of J2EE, the J2SE Java code is replaced with Enterprise JavaBeans (EJBs) and Message Driven Beans.

Also see education for the iSeries developer roadmap at http://www.ibm.com/servers/enable/site/education/abstracts/roadmapj2ee_abs.html for more information.

Web technology choices

In addition to the Roadmap and What's new sections in this overview, the following table provides a simple guide to help you choose which Web technology to use depending on your programming goal. Additional notes are provided to explain the features available for each Web technology.

Your goal	Technology	Notes
Quickly Web-enable existing interactive iSeries applications	The IBM WebFacing Tool	Tactical choice
Create Web user interfaces for iSeries business logic	iSeries Web development tools	Strategic choice
Take your existing iSeries data to the Web	JavaServer Faces (JSF)	Rich tagging options for user interface; strategic choice for database-centric applications that have little business logic
Re-architect your applications to get better Model-View-Controller separation	Struts tools	Rich tagging options for model-controller; leverages JavaServer Pages (JSP) for user interface
Encapsulate your applications from future technology changes and enable them for multiple platforms	Enterprise Generation Language (EGL) tools	Fourth generation language (4GL); abstracts your applications at a higher level and generates the appropriate Java or COBOL code on the iSeries server; leverages JSF; learning time relatively short for COBOL or RPG programmers

Tutorials, scenarios, demos, and presentations

Tutorials and samples help you get started with the product. See the Tutorials gallery and Samples gallery in the product workbench. Also be sure to see the product Web page, in particular, our library page at <http://www.ibm.com/software/awdtools/wdt400/library/index.html> for updates to the tutorials and samples.

What's new in this release?

The following new features have been added to this release, sorted by your role. (Please see "Tools overview" on page 14 for descriptions of all product components):

Your role	New features
RPG and COBOL iSeries programmer	<ul style="list-style-type: none"> • Enhancements to the Error List view: <ul style="list-style-type: none"> – Pop-up actions for Open, Copy, Remove, and Show Help – You can now display local events files from the Error List view – Fixes that you make in the source are reflected in the Error List view (with check marks beside errors that you have fixed) • While editing the CL source, you can now perform selective prompting, and add labels and comments. • You can now export a jar file to a remote system, such as the i5/OS™ Integrated File System, in one single step. • The settings on Remote file system export wizard are now saved to a workspace file, so that you can re-run the export action and re-execute previous export actions easily. • The syntax checker is now enabled to check SQL statements in COBOL source. • While debugging ILE RPG and COBOL source, the Variables view now shows the local variables for the currently selected stack frame at a V5R3 or higher level. In essence, you no longer need to enter local variables into the monitor manually.
Java, C, and C++ developer	<ul style="list-style-type: none"> • Advanced For Development Studio Client Advanced Edition: <ul style="list-style-type: none"> • You can now develop and debug C, C++ and Java applications for Linux™ and AIX® on the i5/OS eServer™ iSeries. • You can also debug your C/C++ applications on a remote AIX and Linux server.
Web developer	<ul style="list-style-type: none"> • Enhancements to the subfile component so that you can now perform the following tasks with the host program: <ul style="list-style-type: none"> – Set the page to the display mode (similar to the SFLRCDNBR DDS keyword) – Set the record to be available on the next read (similar to the SFLNXTCHG keyword) – Apply styles to individual subfile record fields • You can now apply inline styles to individual subfile columns. • You can create a more accurate diagram for a Web interaction with the new Interaction Action Mapping node in the Web diagram palette.
Web services developer	<ul style="list-style-type: none"> • From the palette of the Visual Editor for Java you can create Java classes with the following new visual components: <ul style="list-style-type: none"> – Toolbox – ObjectList beans – JFORMATTED – DFU • Enhancements to the Program Call wizard: <ul style="list-style-type: none"> – You can now reuse the configuration file for multiple program calls. – The user interface is streamlined for importing PCML files from the source. – You can now browse for procedures in a service program.

Your role	New features
iSeries programmer and IBM WebFacing Tool user	<ul style="list-style-type: none"> • You can now view and work with DDS files in the Outline view • A new Web Settings view in CODE Designer provides an integrated Eclipse version of the former Web Settings tool. In the former Web Settings view, DDS annotations were embedded for the purposes of setting WebFacing conversion preferences. The new Web Settings view is now ported to the Eclipse workbench as part of the IBM WebFacing Tool plug-in. • A new outline view for DSPF, PRPF, PF and LF DDS source in the Remote Systems LPEX editor to provide a structured view of the source. The new outline view is populated when the source is loaded into the Remote Systems LPEX editor, thus facilitating both understanding and navigating of the source.

Your role	New features
All developers	<ul style="list-style-type: none"> • Improved integration of the debugger with the Remote System Explorer. You can set source breakpoints in the iSeries editor prior to the startup of a debug session. • The WebSphere Host Access Transformation Services (HATS) Toolkit transforms the screens of a server application into a Web application that you can deploy to a Web server. (This tool, however, is for development use only. When you deploy the application, you need to acquire a user-based license.) In this release, the HATS Toolkit includes the following key enhancements: <ul style="list-style-type: none"> – Easier-to-use drag and drop screen customization – New wizards to customize HATS components and widgets – Enhanced macro support with a macro-stepping feature – Side-by-side preview of host screens and HATS customizations – JavaServer Faces support of Integration Objects, for easier Web page design – Improved 5250 subfile recognition, rendering, and customization – A new EGL form editor to provide a full-function graphical editor for creating 5250 screen layouts for use in EGL applications – A more powerful programming environment with new date and time functions, new data types, and a data initializer – When performing EBCDIC debugging on an iSeries server, you can now debug the EGL application with data that is kept internally in EBCDIC coding. You can also access remote databases without converting the data. • You can use Enterprise Generation Language (EGL) programming to create Java code • CODE VisualAge[®] RPG is now installed separately to reduce the amount of hardware resources needed. You only need to install these products if you are not using the Eclipse-based workbench. <ul style="list-style-type: none"> – VARPG is upgraded to the RPGIV V5R3 level, which includes: <ul style="list-style-type: none"> - A new %SUBARR built-in function to assign, sort, or return a subarray - Direct conversion of date/time/timestamp elements to a numeric format using %DEC - A second parameter for %TRIM, %TRIMR and %TRIML, to indicate which characters to trim - A new prototype option OPTIONS(*TRIM) to pass a trimmed parameter - Relaxation of the rules for using a result data structure for I/O to externally-described files and record formats – Enhancements for the VARPG GUI designer and runtime: <ul style="list-style-type: none"> - Improved usability for backup and installation of the packaging utility - The ability to specify different folders for authentication information at runtime. You can now use stored authentication information per user in the Windows server environment. - The open file dialog box is improved such that now you can select multiple files or an entire folder. - The line graphs in the Graph part are improved such that you can now click an end point on the line to set internal information about that point, and if some points have missing values, the tool uses a linear algorithm to calculate the values.

Your role	New features
All developers	<p>Advanced For Development Studio Client Advanced Edition</p> <ul style="list-style-type: none"> You can use Enterprise Generation Language (EGL) programming to create ILE COBOL code WebSphere Portal includes the following enhancements: <ul style="list-style-type: none"> You can create portals visually with a new Portal Designer utility that allows you to layout a Portal interface visually without the need to write any code. You can use a graphical Page Designer utility to layout the interface for portlets and to create portlets using the Struts framework and the Web Diagram Editor to visualize the structure. You can also now generate portlets with wizards, which can also automatically create portal deployment descriptors and EAR files The WebFacing Tool provides Cascading Style Sheets-Positioning (CSS-P) support, and allows pixel-based repositioning of 5250 fields on a web page that is delivered through the WebFacing Tool. You can associate a new pair of row, column coordinates to a 5250 field, and the WebFacing Tool will map the row,column coordinates to pixel-based coordinates for CSS-Positioning. This support allows the Webfacing Tool to reposition 5250 fields on a web page unconstrained by the boundaries of the original 5250 area. (This feature will be released in a future fixpack.)

Tasks overview

This section explains how to use the different tools in the product to accomplish your major development goals:

- “Goal 1: Develop and maintain applications”
- “Goal 2: Create Web front ends” on page 11
- “Goal 3: Create GUI front ends” on page 14

Goal 1: Develop and maintain applications

Development Studio Client is designed to ease the entire life cycle of developing and maintaining iSeries applications. Whether you work on traditional green-screen applications, or are responsible for making existing iSeries programs accessible as Web applications, your productivity is greatly increased as you work on the following tasks.

Edit, compile, and debug ILE and non-ILE applications

You can create and manage development projects on your iSeries server from your Windows-based workstation with the Remote System Explorer and iSeries projects. With these tools, you can view iSeries libraries, files, and members. You can also launch the server compilers, the workstation editor, and various debuggers. Your program editing tasks are simplified with the Remote Systems LPEX editor. The editor can access source files on your workstation or your iSeries server.

With the integrated iSeries debugger you can debug your program running on the iSeries server from a graphical user interface on your workstation. You can also set breakpoints directly in your source before running the debugger. The integrated iSeries debugger client user interface also enables you to control program execution. For example, you can run your program, set line, watch, and service entry point breakpoints, step through program instructions, examine variables, and examine the call stack. You can also debug multiple applications, even if they are written in different languages, from a single debugger window.

Develop iSeries business logic in Java

You can use iSeries Java development tools to create Java applications that can access iSeries data. More specifically, you can also call native iSeries applications from your Java code.

Create and edit DDzS definitions easily

You can create and update DDS definitions for display files, printer files, and physical files using CODE Designer when you are on or offline. This tool provides a graphical user interface you can use to define your DDS files, and then save them locally or upload them directly to your iSeries server. By creating your DDS screens visually you can improve their usability and your programming productivity (because more of your time is spent on visual design, and less on getting the syntax right).

Goal 2: Create Web front ends

With Development Studio Client you can make your iSeries applications and data accessible beyond the green-screen interface. You can generate a Web-based user interface for your program's DDS screens, or create a new Web interface that connects directly to your program's input and output parameters. Which path you choose depends on the way your iSeries application is designed, and on trade-offs you need to make between rapid deployment and customized look and feel. You can perform the following Web-enabling tasks for your iSeries applications.

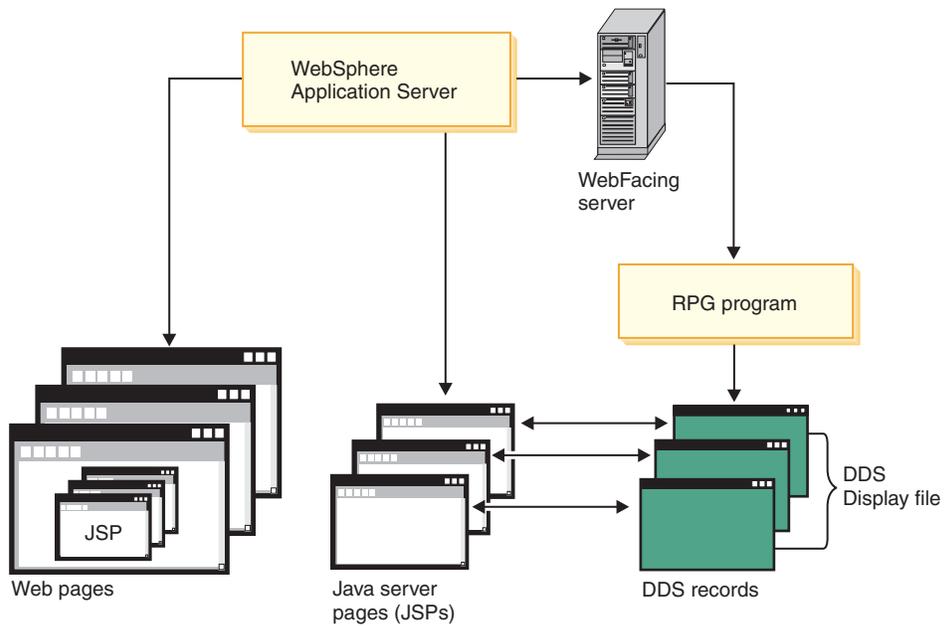
Generate Web front ends to existing applications

You can use the IBM WebFacing Tool to create Web front ends to iSeries applications that use DDS for their green-screen transactions. You choose a Web style for your green-screen application, generate a set of JSP and XML files that interact with the logic of your program, and then easily test your application in the WebSphere test environment of the workbench. When you are ready to deploy your application, you can generate standard J2EE WAR and EAR files that can be installed on a WebSphere Application Server.

The WebFacing Tool is ideal for applications that you want to deploy broadly over a corporate intranet or the Internet, where rapid deployment takes precedence over customizing the look and feel of each page. With the WebFacing Tool you can continue to deliver your application as a 5250 application, and use the same ILE and non-ILE programs to deliver the application through the Web. Note that there are some restrictions, described in the WebFacing documentation, on which DDS keywords can be converted to generate sets of JSP and XML files.

CODE Designer has been enhanced to work with the WebFacing Tool. A new **Web Settings** section helps you specify how your DDS fields and constants will be handled by a browser after they are converted using the WebFacing Tool, without affecting their appearance on the green-screen.

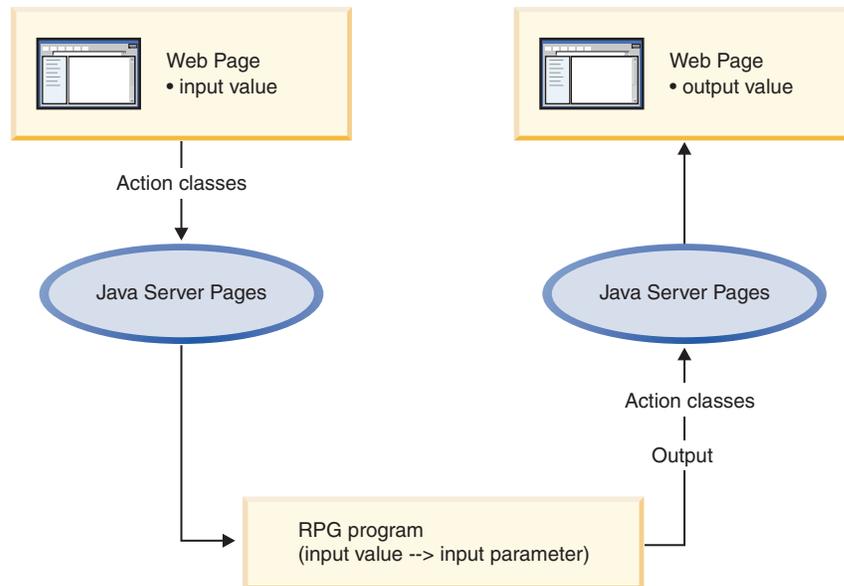
The following diagram illustrates the run-time process for the WebFacing Tool:



Use the WebFacing Tool for programs where the DDS user interface is interwoven with the business logic. If you architect programs with a set of distinct entry points for each element of business logic, and a user-interface component that simply generates appropriate calls to business logic entry points based on user inputs, you should consider using Web development tools to develop your own servlets and JSP files to access your code, as described in the next section.

Develop Web pages, Java Action classes, and JSP files to access iSeries programs and data

You can create interactive Web pages using Web development tools. Use the Web Interaction wizard to define how your pages interact with one or more ILE or non-ILE applications. This wizard generates Java Action classes and JSP files for use with data from HTML forms. When the end user enters data in a form, the input becomes data to your ILE and non-ILE programs, and the output from the programs is formatted for the Web. Your ILE and non-ILE logic can be separated into different programs for each stage of input and output (known as a Web interaction), or can be a single service program with entry points to handle each Web interaction. The following diagram illustrates the Web development tools process:



You can also use Web development tools to develop customized Java classes that run on the iSeries server. These servlets make use of iSeries data directly, or can access the data through ILE and non-ILE applications. IBM Toolbox for Java classes included with Java development tools let you access iSeries data files and issue ILE and non-ILE program calls from the Java servlets. You develop the servlets using Web development tools, and use the Record I/O and Program Call classes from the toolbox to retrieve and update iSeries data. You can launch your servlets or customize their output from the JSP files that you create in the Page Designer editor in Web development tools.

Customize the appearance of your Web site pages

You can easily customize your HTML and JSP files using the editing tools in Web development tools. Use these tools to create and update input forms, change the appearance or placement of blocks of text, and add backgrounds and images to your pages. The iSeries-specific Web components help you create Web versions of your input and output pages with the same kinds of input validation, output formatting, and subfile controls that native DDS screens provide. You do not need a detailed knowledge of HTML or Java to accomplish these tasks.

Test and deploy your Web-enabled applications

You need an easy test mechanism for your Web-enabled applications. These include applications you have hand-coded on your workstation that use:

- HTML
- JSP files
- Java servlets
- Web front-ends created using one of the code-generating wizards in Web development tools.

You can easily run your program in the WebSphere test environment of the workbench, quickly make changes, and retest, rather than redeploy your application every time you want to verify functions. When you are finished, you can package and deploy your Web applications as J2EE-based Web archive (WAR) and Enterprise Archive (EAR) files, and then install them in a WebSphere Application Server.

Goal 3: Create GUI front ends

You do not have to deploy your iSeries applications on the Web in order to give them a more modern look and feel. You can produce graphical user interface (GUI) front ends to your iSeries applications with Java development tools.

Produce applets and native graphical user interfaces from the same source

After you create a visual interface and associated logic on your workstation, you can deploy your application as a native Windows application, or as a Java applet that can run in any Web browser with an appropriate Java Virtual Machine (JVM™). This gives you extra flexibility in making your application available to users. If you want to control access tightly, you can deploy the application as a Windows executable. If you want widespread access you can place your applet (and associated files generated by Java development tools) on a Web site, so that users with browser access to the Web site can run the applet in their browser and communicate with data on the iSeries server.

Create Java-based graphical user interfaces to ILE and non-ILE applications

Use your own custom calls to Java GUI classes (such as Swing classes), and the Java visual editor to develop your graphical user interface, and then use classes provided by IBM Toolbox for Java or Java beans provided by Java development tools to access your iSeries server. Although this option gives you great flexibility over your program's look and feel, it requires a bit more work than using Web development tools to develop HTML-based interfaces to iSeries programs.

Tools overview

This section describes the main tools of Development Studio Client, and the kinds of tasks you can perform with each tool. The tools are:

- “The workbench-based integrated development environment”
- “iSeries development tools” on page 15
- “iSeries Web development tools” on page 18
- “iSeries Java development tools” on page 19
- “Struts environment support” on page 19
- “Web Services development tools” on page 20
- “Server development tools” on page 20
- “The IBM WebFacing Tool” on page 21
- “Integrated iSeries debugger” on page 21

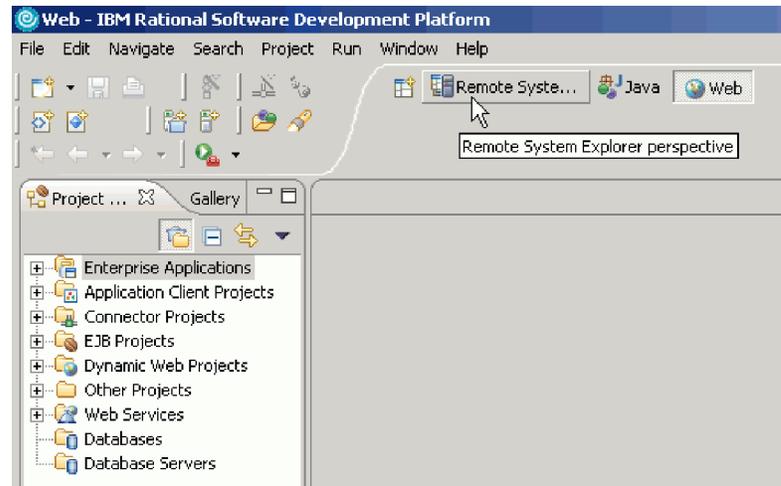
The workbench-based integrated development environment

Development Studio Client uses the Eclipse workbench. This workbench offers you a comprehensive development environment that integrates all your tools, teams, assets, and other workbench-based projects seamlessly. The platform provides the core frameworks and services upon which all plug-in extensions are created, the run time in which plug-ins are loaded, integrated, and executed, plus a common user interface model for working with e-business tools. Plug-in tools can be integrated into the workbench portable Application Program Interface (API), yet run unchanged on any supported operating systems. Thus, the architecture discovers these plug-ins and integrates them all into the new base environment, providing a standard user navigation model for building integrated Web and application development tool projects. Because development resources for a project

are stored in a single repository, the workbench is ideal for resource sharing and for providing consistent team support for shared projects.

Perspectives

Perspectives define the initial set and layout of views in the workbench. They provide a set of capabilities aimed at accomplishing a specific type of task or working with specific types of resources. To open a perspective, click **Window > Open Perspective**. Open perspectives are represented by icons on the shortcut bar (the toolbar at the top-left of the workbench window). When you have more than one perspective open, you can switch between them by clicking the icons on the shortcut bar. Your current perspective displays in the workbench title bar.



You can also save customized perspectives for reuse. To work with perspectives, use the workbench menu items for **Window > Customize Perspective** and **Window > Save Perspective As**.

iSeries development tools

Two major iSeries development tools, the Remote System Explorer and iSeries projects, provide the framework, user interface, editing capability, and performable actions on iSeries objects, commands, and jobs.

Tools to access and edit Native iSeries applications

iSeries development tools support views, editors, and tool extensions, to help you develop and maintain iSeries applications while working with remote operating systems. The Remote System Explorer and iSeries Projects perspectives are the iSeries programming environments you use to complete these tasks. The perspectives guide you through various connections to your remote systems, give you access to files and resources, and help you monitor jobs and run commands. They also provide support for the sharing of work through a team perspective. Most users can easily do their work using the Remote System Explorer perspective. The iSeries Projects perspective is designed more specifically to support structured programming, offline development, and team collaboration.

The Remote System Explorer perspective also supports system types other than iSeries systems, such as UNIX[®], Linux, and Windows systems.

The Remote System Explorer and iSeries projects provide the following additional support:

- Seamless access to local and server files
- The ability to use copy, paste, and drag-and-drop support, even between server systems
- The manipulation of library lists
- PDM-style filtering of iSeries objects from your library list
- The ability to extend the Remote System Explorer and iSeries projects by plugging in third-party tools
- A table view that lets you explore and sort remote files, customize column layouts, use function keys to support PDM-like actions, and specify command line parameters on remote objects.

A good place to begin in the help contents is: Introduction to the Remote System Explorer.

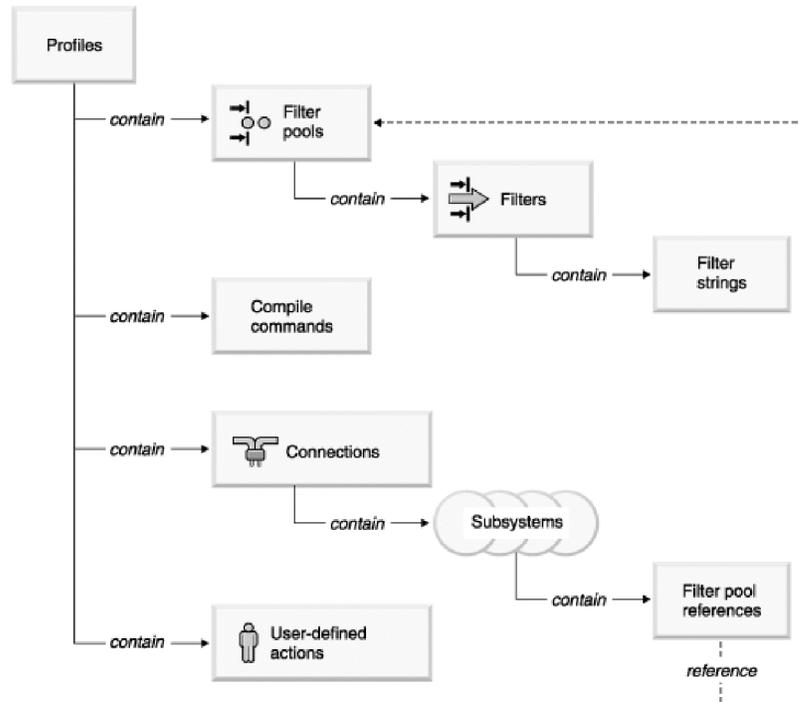
Universal file and command subsystems

You can export and import files, explore remote files, and execute remote commands for servers other than iSeries, such as UNIX, Linux, and Windows servers. Using the Remote System Explorer, you can use the workbench editor of your choice for any remote file on any type of remote server. Remote searching capabilities that help you search the contents of remote files and filters on a iSeries native file system, and perform asynchronous searching of the iSeries Integrated File System (IFS), Linux, Windows, and UNIX servers .

iSeries exploration and command execution

Use the Remote System Explorer to list remote libraries, objects, and source members, and organize these elements into filters. You can create and share filters for rapid access to specific artifacts and commands. Filters can also be partitioned into pools to allow for easier organization of filters. To save time, you can also open source members directly in the Remote Systems view without having to create a filter. You can work with iSeries native file system (QSYS) files, CL commands, and generic file and command subsystems that involve the Integrated File System files and QSHELL commands. You can also create your own actions and commands.

The following graphic serves as both a summary and an introduction to the structure of the Remote System Explorer and iSeries development tools:



iSeries actions for edit, compile, verify, run, and debug

The Remote System Explorer contains the Remote Systems LPEX editor, so that you can edit source code directly in the workbench. The interface provides right-click actions so that you can compile, run, and debug your programs.

The Remote Systems LPEX editor supports many rich editing features for RPG, COBOL, CL and DDS such as source prompting, file compare, syntax checking, automatic uppercasing, automatic indentation, and so on.

The following features also help you with your programming tasks:

- Local verification and error feedback
- The Outline view, as a navigational tool for your source, where you can click various nodes in the outline to move your cursor to the corresponding place in your file
- Command shells for running local or server commands with logged output
- The ability to create a new RPG procedure, D-specification, and Java method call
- Content assist, (by pressing CTRL+Space), to provide code completion choices as you write code, and automatic code completion when there is only one match for the code
- iSeries Job Status view for working with OS/400® jobs
- Run and Debug launch configurations that you can use to edit and save information required to run or debug your applications
- Reference documentation available from the Source workbench menu

iSeries Projects perspective

The iSeries Projects perspective is the ideal tool for team-based iSeries application development. The perspective encourages structured programming and allows for disconnected development. In the perspective, you can monitor error feedback, and inspect timestamps to resolve conflicts between remote and local files.

Enhancements to iSeries projects include the ability to compile individual members (in addition to the function that allows you to build whole projects).

Extension of the Remote System Explorer and iSeries Projects for business partners

Using the workbench, you can integrate your tool sets into the Remote System Explorer. Many leading iSeries tool vendors are actively integrating their tools into the product.

iSeries Web development tools

iSeries Web development tools give you the ability to create new e-business applications that use a Web-based front end to communicate with the business logic in an ILE or non-ILE program residing on an iSeries server. In contrast to the WebFacing Tool, with Web development tools you can build a Web application from scratch and control everything about the application's behavior and appearance, with more tools for customization. You can create the high-level design of your Web site and apply page templates using Web site designer. Then, you can create the individual pages with Page Designer, or generate input and output JSP files with the Web Interaction wizard. You can also add iSeries Web components to your pages, for example, Web equivalents of iSeries command keys, input fields that accept only particular types of data, or a Web component that provides subfile-like support. You can even use iSeries Web development tools to customize the JSP files generated by the WebFacing Tool.

The tools give you open standards, greater flexibility, and the ability to tie in existing applications with Web service wizards to help you write applications that are portable across platforms.

The specific iSeries extensions of Web development tools include:

- Support for the Struts run-time environment and the Struts Web Diagram editor
- iSeries Web Tools run-time configuration wizard that captures server name, user ID, password, and library list information
- The Web Interaction wizard helps you with the following tasks:
 - You can link the Web components for a Web page to parameters of your Program Call and define the Program Call parameters without the need to deal directly with JSP files and iSeries classes.
 - You can access iSeries programs from a Web application, or include them as the base for a Web service.
 - In addition to iSeries Web Components, you can use JSP files with plain HTML controls to build Web pages that interact with an iSeries program or procedure call.
 - You can choose a Page Template for the generated input or output pages in the Web Interaction wizard.
 - The message handling feature enables you to associate an iSeries Message file or a Java properties file with your Web application in the iSeries Web Tools Runtime Configuration wizard. This support enables your application to retrieve run-time messages from the properties or message file and display them on your Web pages.
 - The flow controller feature helps you connect multiple output pages from the Web Diagram editor and specify the flow condition in the Web Interaction wizard.
 - You can define a program or procedure to call when your Web application session times out.

- The flexibility of Java classes created by the Web Interaction wizard has been enhanced to include more methods and be organized into packages. These Java classes are generated from templates, and you can customize these templates as well.
- Support for cascading style sheets – you can define your own external or internal style sheet, or inline styles, to override the default styles of iSeries Web Components. This gives you extra flexibility in customizing the look and feel of your Web pages.
- It is easy to design your own JSP page with the enhanced Palette view, and entry field validation is provided in Attribute view to decrease the chance of entering erroneous data.

iSeries Java development tools

iSeries Java development tools give you the ability to develop Java applications and write, compile, test, debug, and edit Java programs. Java development tools are ideal for:

- Developing and compiling business logic in Java that runs on a remote server, such as an iSeries or Linux server
- Creating applications and Java GUIs that access existing server data, business logic, and resources
- Deploying or exporting your Java class or source files to a remote server
- Support for EJB query language, container-managed persistence (CMP), and message-driven Java beans
- **Advanced** Improved speed and memory performance for Enterprise JavaBean (EJB) deployment

The specific iSeries extensions of Java development tools include:

- iSeries-specific launch configuration support, so that you can remotely execute and debug Java programs on an iSeries server
- A Program Call wizard that can create Java beans and Program Call Markup Language (PCML) files to call your iSeries programs or service programs. The Java beans can also be used to create Web Services using Web Services development tools, and you can import any PCML file generated by COBOL or RPG iSeries compilers.
- A Remote Error List view to display feedback from shell commands on UNIX, Linux, Windows, and local servers (applicable to Java/C/C++ programmers)
- Support for Enterprise Generation Language (EGL) programming, to create Java or COBOL source, and prepare the output to produce executable objects. This support is available through the EGL and the EGL Web perspectives.
- The ability to trace and determine performance issues for iSeries Java programs with the Profiling perspective
- The IBM Toolbox for Java
- iSeries-specific Java beans
- iSeries-specific Java Visual Editor (JVE) palette extensions

Struts environment support

Development Studio Client offers support for Struts and the Web Diagram editor. Struts is a framework for building Web applications based on the Model-View-Controller paradigm, organizing the application into three separate components:

- Model: the application model with corresponding data representation and business logic
- View: data presentation, providing views for user input
- Controller: to dispatch requests and control data flow

The Web Diagram editor is the visual development environment for developing Struts-based applications. You can drag and drop JSP files and action servlets onto a free-form surface, which is like a blank screen, make connections between items, and then define those items by double-clicking and editing them through a wizard.

Web Services development tools

When components in a system are tightly coupled, and based on database records and flat files, they are rigid and sensitive to change. Web Services tackle the problem of inefficient distributed computing. This is accomplished by ensuring that system components are loosely coupled and dynamically bound to provide greater flexibility, scalability, lower overhead costs, and hence, better business growth. Web services are self-contained, modular applications that can be described, published, located, and invoked over the Internet, employing 'just-in-time' integration of services. The architecture involves a relationship between Service providers, brokers, and requesters. More specifically, Service requesters submit a request to a Service broker, which finds the right service from a Service provider. Web Services development tools allow you to develop this service.

The Web Service wizard creates a Web Service, using a Java bean generated by the iSeries Program Call wizard, to call one or more server programs or service program procedures on the iSeries server. More specifically, the Web Service wizard works in the context of a Web Project and allows for creation, deployment, testing, generation of a proxy, and publication to a Universal Description, Discovery, and Integration (UDDI) registry of Web services based on Java beans, input and output URLs, DB2 XML extender calls, DB2 Stored Procedures, and SQL queries.

Server development tools

Server development tools let you test your applications in locally or remotely installed run-time environments. You do so by creating a server project that represents your environment and can associate your Web projects with a server configuration. This tells your server tools how to configure the instance for that particular Web project.

Server development tools include the following components and features:

- The WebSphere test environment
- A local copy of the WebSphere Application Server (WAS) run-time environment so that you can test Web and Enterprise Application Archive projects
- A local copy of the TCP/IP Monitoring Server run-time environment, which forwards requests and responses, and monitors test activity
- Support for the local Apache TomCat run-time environment so that you can test Web projects that contain servlets and JSP files
- Support for Remote Agent Controller (installed on your remote machine) if you want WAS to create a server project instance and configuration for you when you run your Web application (rather than having you create the server project yourself)

Note: If you use an external server instance, such as WAS on your Windows machine or remote iSeries server, you need to deploy your project before running. To deploy, you can use Server development tools to create a remote file transfer instance to handle the details of how and where to copy files.

The IBM WebFacing Tool

With the IBM WebFacing Tool, you can quickly convert your DDS display file source members so that the user interface of your iSeries programs can run in a browser. When you convert your DDS display files, JSP and XML files are generated for you that substitute for the DDS code and make Web access possible.

In the WebFacing Project wizard, you can select one or more DDS source members to convert, and select a Web look and feel from one of several predefined styles, or you can design your own Web style to use with your applications. The tool creates three JSP and XML files for your record formats. The XML files hold the data for the record format, or control its appearance or other characteristics, and the JSP handles displaying the Web version of the screen, prompting for data, and handling input errors. The wizard generates an application home page to launch the Web-enabled version of your program.

When a user invokes a converted application from the browser, the WebFacing server on the iSeries server starts the application. The server intercepts all calls to Read, Write, and Exfmt operations to display files, so that in many cases your program can run without modifications, and without even detecting that it is being accessed using the WebFacing Tool. You might need to make coding changes if your application uses DDS keywords that are not supported by WebFacing, or if you want to modify the DDS screens so that the conversion to Web format produces a more attractive or consistent result.

The WebFacing Tool also provides:

- Automatic configuration for UTF-8 support when you deploy to WebSphere Application Server
- Support for custom tags and run-time extensions
- Support for function keys within window records
- Support for applications that use program-to-system fields for setting the display attributes of protected fields
- **Advanced** Support for portlet applications that can access DDS-based RPG, COBOL, and CL applications. You can run and test the portlet application inside the product, and then deploy the application to the WebSphere Portal Server for production.
- **Advanced** Support for applications that display system screens during the application flow
- **Advanced** Struts-compliant code generated by the WebFacing Tool conversion process

Integrated iSeries debugger

The integrated iSeries debugger helps you debug code that is running on the iSeries server or on your Windows system, using a graphical user interface on your workstation. It supports the following tasks:

- Service entry point management in the Remote System Explorer. The service entry point feature is designed to allow easy debugging of Web applications that invoke business logic written in RPG, COBOL, CL, or even C and C++. The

service entry point is a special kind of entry breakpoint that you can set directly from the Remote System Explorer, triggered when the first line of a specified procedure is executed in a job that is not being debugged. Thus, service entry points allow you to gain control of your job at that point. A new debug session is started and execution is stopped at that location. While the original service entry point design was meant to assist with debugging Web applications, the feature is also beneficial when debugging applications in an environment where the job name is not known ahead of time. Some sample scenarios include:

- Debugging WebFacing applications—Typically, the WebSphere Application Server is executing a JSP file that is driving your RPG or COBOL program. The RPG or COBOL code, however, is what you want to debug. Since ahead of time you would not know the name of the job in which your RPG or COBOL program will run, the use of service entry points is an ideal solution.
 - Debugging any application that uses the Toolbox program or service program calls—When a program or service program is called and the code where the service entry point is set is about to execute, the debugger can take control of the application. With this technique, you can put the program or service program invoked by the Toolbox under debug, even though you do not know which job it will be running in.
 - If your application runs in a multi-job environment, service entry points will significantly simplify your tasks while debugging. You can set a service entry point in the program that is about to be spawned in a new job. When the program is spawned and the line where the service entry point is set is about to execute, the debugger will be able to gain control of the program and stop at that line.
- Setting breakpoints on a source code line or on a function or procedure, and setting watchpoints to see when a variable or storage location is changed.
 - Three types of stepping operations: step-into, step-over, and step-return. You can even step between functions or procedures written in different languages.
 - Viewing the contents of global or local variables, call stacks, and data storage in specialized panes.
 - Quick restart of programs you debug repeatedly: Breakpoints and debugger window layouts for a given program are automatically saved between debugging sessions so that you don't have to set them again the next time.
 - Attaching to an already running job or Java Virtual Machine on the iSeries server.
 - Issuing a step return in your ILE programs while debugging. When you step return, all threads are allowed to run. The current execution point, up to the line immediately following the line that called the function or program, is executed. Execution stops after exiting the current function or program.
 - Debugging SQL stored procedures, complete with editor source view display.

The integrated iSeries debugger supports any program written in the following languages:

- ILE RPG, ILE COBOL, ILE CL, C, C++, as well as Original Program Model (non-ILE) RPG, COBOL, and CL.
- Java, when running the program under Windows, or when running on any iSeries server with OS/400 V4R2, V5R1, V5R2, or V5R4. You can even debug servlets, JavaServer Pages™, and Enterprise JavaBeans on the iSeries server.

Separately Installable Features

CoOperative Development Environment (CODE) and VisualAge RPG can be separately installed.

VisualAge RPG

Using VisualAge RPG you can develop and maintain client/server applications in a visual development environment on the workstation. You leverage your existing RPG skills to create graphical user interface (GUI) applications that you can deploy as native Windows applications, or as Java applets that can run on any Java-capable Web browser. These applications can access iSeries data and other iSeries objects. VisualAge RPG can also create stand-alone Java applications.

With VisualAge RPG you can create, edit, compile, and debug applications on your workstation. You can build an application from the top down. You start by focusing on the look and feel of the interface, and then you tie all the parts together with workstation RPG logic that you write in the VARPG language. You can reuse RPG logic and import display files (DSPF) from an existing application.

You can learn VisualAge RPG quickly thanks to its tightly integrated tools and its visual development environment. For example, you can quickly create text, buttons, and fields in your design window using a point-and-click action.

VisualAge RPG includes the following features and tools:

- The VARPG compiler incorporates the latest changes from the ILE RPG compiler.
- You can create interfaces in the GUI Designer by selecting visual parts and dropping them onto your design window, rather than by coding UI capabilities in source code. Select a part such as a push button or an entry field, drag it with the mouse, and drop it on the window you are designing. You can then select an event for the part from the part's pop-up menu, and use the editor to create the workstation RPG program logic behind the event. From the GUI Designer, you can also launch the editor, the compiler, and the debugger.
- The syntax checker helps you detect syntax errors quickly in your VARPG programs.
- The compiler performs compilations on the workstation, where your applications eventually run.
- The help and message compiler allows you to incorporate online help and messages into your workstation application.
- The editor allows you to add new editor functions or change existing ones. Language-sensitive editing and language-sensitive help are provided with the editor.

CoOperative Development Environment (CODE)

CODE gives you a suite of utilities for creating source and DDS files, and managing your CODE projects. CODE consists of CODE Designer, and CODE Editor.

CODE Designer

CODE Designer removes the drudgery from managing your DDS files. Using a customized visual editor, you perform the following DDS editing tasks with CODE Designer:

- Create new DDS screens, printer files, and physical files

- Add text to DDS screens, simply by clicking and typing
- Insert new screen fields for input, output or both, from a pop-up menu
- Manipulate fields and text using drag-and-drop actions
- Change the attributes and properties of data fields and plain text
- Link DDS fields to iSeries database fields, by connecting to the database and selecting appropriate fields from a pull-down list
- View the hierarchical relationships among files, records, fields, help specifications, keys, and keywords in each selected DDS object
- Organize records into groups for a particular screen, report, or printer file

You can view the DDS source for each element when you add or update it. You can even edit the DDS code that CODE Designer generates for you. CODE Designer also saves you time by flagging errors before the DDS source is compiled on the iSeries server.

CODE Editor

CODE Editor is the predecessor to the set of powerful editing features found in the workbench, primarily in the Remote Systems LPEX editor of the Remote System Explorer. Please see “iSeries development tools” on page 15 for more information.

Chapter 3. Prerequisites

Before you deploy Web applications using Development Studio Client, and any other relevant tools, you need to ensure that IBM WebSphere Application Server for iSeries and other services are properly configured on your iSeries system. This chapter describes the steps you need to take and provides links to appropriate documentation to prepare your iSeries system for operation. See the product readme and installation notes for more information on recommended memory and disk space allocations.

If you plan to undertake development work using the Remote System Explorer, Java development tools, or other ILE-focused tools, you only need to know how to access your iSeries system to begin development, and you can skip the majority of this chapter.

Therefore, if you do not need to deploy any Web applications, begin here: Getting connected to an iSeries server.

Working with your HTTP server and WebSphere Application Server

Your Web-enabled iSeries applications use WebSphere Application Server to run the Java servlets and JavaServer Pages™ (JSPs) that allow communication between the Web user's browser and the iSeries programs or data. **Advanced** In Development Studio Client Advanced Edition, WebSphere Application Server can also run your Enterprise Java Beans (EJB) based on J2EE technology.

To serve your HTML pages and JSPs from the same iSeries system, you also need an HTTP server on that host. We recommend that you use the IBM® HTTP Server powered by Apache. You can find documentation about how to use this server in the following place: IBM HTTP Server for iSeries Documentation Center.

WebSphere Application Server handles executing the JavaServer Pages, JavaBeans, Java servlets, and EJB for Development Studio Client processes. The primary documentation resources for IBM WebSphere Application Server for iSeries are available at the following Web sites:

- IBM WebSphere Application Server Version 4.0 Advanced Edition for iSeries at <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Version 4.0 Advanced Single Server Edition for iSeries at <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>

For versions 4.0 of WAS, minimally you need to carry out the steps under the *Installation* link. Becoming familiar with the IBM WebSphere Application Server documentation, in particular the sections on *Installation* and *Setting up multiple instances of the WebSphere administrative server*, is highly recommended.

For WebSphere Application Server version 5.0 and WebSphere Application Server Express:

- RedBooks page for WebSphere: Search for RedBooks for WebSphere Application Server Version 5.0 and WebSphere Application Server Version 5.0 Express Edition at <http://publib-b.boulder.ibm.com/redbooks.nsf/portals/WebSphere>

In order to prepare your system for Web application development and deployment, you need to complete the following tasks. iSeries-specific information is included below, but follow the links above for information about the IBM HTTP Server and WebSphere Application Server servers.

- Become familiar with server port numbers (information provided below)
- Find the port for your HTTP instance
- Find the port for your WebSphere Application Server instance
- Start the HTTP administration server job
- Create your HTTP configuration
- Create your HTTP instance (optional; you can use the default instance)
- Start your HTTP instance
- Install the sample libraries (information provided below)
- Configure your WebSphere Application Server
- Start your WebSphere Application Server instance (information provided below)
- Map a network drive to the iSeries system (information provided below)
- Start the WebSphere Administrative Console

Server port numbers

If you plan to use the default port numbers for WebSphere Application Server and the HTTP server, typically, these default ports are:

- The HTTP server instance uses port 80.
- The WebSphere Application Server version 4.0 instance uses port 900, but this could vary depending on the version of WebSphere Application Server you are using. For WAS version 5.0, there are many default ports. For example, the administrative port is 9090, and the default internal HTTP port is 9080.

If these are not the correct default port numbers, contact your system administrator to determine what these ports are. You can use these port numbers for your HTTP and WAS instances, or you can create your own. See the appropriate documentation for creating your own instances.

The following table lists the ports that Development Studio Client uses to communicate with the iSeries server:

Development Studio Client function	Required iSeries port on OS/400
Remote System Explorer functions: including filtering, editing, compiling, and running	<ul style="list-style-type: none"> • 8470: OS/400 central server • 8475: OS/400 remote command server • 8476: OS/400 sign-on server • 446: OS/400 DRDA (record access) server • 449: OS/400 server mapper server
Integrated File System (IFS) access	8473: OS/400 file server
Interactive job support (via the STRRSESVR command)	8472: OS/400 data queue server
WebFacing (only for run time)	4004: WebFacing server

You need the Remote System Explorer ports (8470, 8475, 8476, 446, and 447) to remain open for proper functioning of the product. However, you only need to keep the IFS, Interactive job support, and WebFacing ports open if you require those functions.

To change the local ports that your workstation uses to connect to the iSeries server:

- Remote System Explorer port:
 1. From the workbench menu bar, click **Window > Preferences**.
 2. Expand **Remote Systems** and click **Communications** to select it.
 3. You can change the port number in the **RSE Communications daemon port number** field.
- Debugger port:
 1. From the workbench menu bar, click **Window > Preferences**.
 2. Expand **Debug** and click **Debug Daemon** to select it.
 3. You can change the port number in the **Daemon port** field.

You also need to reserve two ports on your local machine for TCP/IP 'call-back' communication to the iSeries server.

Development Studio Client function	Required local port
Interactive job support (via the STRRSESVR command)	3001: Remote System Explorer communications daemon
Integrated Debugger	8001: Debugger daemon. This function also uses port 3001.

Installing the sample libraries

To work with the samples in this guide, you need to restore the WHOLESale and WDSCLAB libraries to your iSeries server. You should do this even if you have already restored the libraries for a previous release of the product, because their contents are different. You can use the Eclipse workbench to restore the save files for the library, rather than the 5250 emulator. The instructions describe how to restore the WHOLESale library. (To restore the WDSCLAB library, the instructions are exactly the same except in the workbench you select wdsclab.savf instead of wholesale.savf.) The following process uploads the save file to a temporary library on your iSeries server, QRSETEMP, and then restores that save file to a new library called *SAVLIB, or a library of your specification.

Note: The save files used to install the sample library are for use with a V5R1 or later iSeries system.

To restore the libraries:

1. In the workbench, switch to the Remote System Explorer perspective if you are not there already. From the workbench menu, click **Window > Open Perspective > Remote System Explorer**.
2. In the Remote Systems view, expand **Local > Local Files > Drives**.
3. Continue to expand the **Drives** directory to navigate to the directory where you installed the product. By default, this is **c:\WDSC**.
4. In the **WDSC** directory, expand **wdscsampl**.
5. Right-click **wholesale.savf** and select **Restore on iSeries** to invoke the Restore iSeries Save File dialog box.
6. The first field, **iSeries connection**, specifies the iSeries server to which you want to restore the savf file. In the workbench, you use a connection to communicate between your local workstation and the iSeries server. If you do

not have a connection, click **New** to define one to an iSeries server. If you already have a connection, this field is automatically pre-filled with the connection name.

7. Leave the default values for the following fields: **Save file library** and **Save file name**.
8. In the **Saved from library** field, enter WHOLESale.
9. In the **Restore to library** field, you can accept the default value, enter the name of an existing library you want to use, or create a new library by typing in a new name, for example, MYLIB.
10. Click **OK**. Enter your iSeries user ID and password if prompted.
11. When the wizard completes, go back to the Remote Systems view, and locate wdsclab.sav in the same directory (**Local > Local Files > Drives > C > WDSC > wdsclab**.)
12. Right-click **wdsclab.sav** and select **Restore on iSeries** to invoke the Restore iSeries Save File dialog box.
13. Select the same iSeries connection you used to restore the wholesale save file in the **iSeries connection** field.
14. Leave the default values for the following fields: **Save file library** and **Save file name**.
15. In the **Saved from library** field, enter WDSCLAB.
16. In the **Restore to library** field, you can accept the default value, enter the name of an existing library you want to use, or create a new library by typing in a new name, for example, MYLIB2.
17. Click **OK**. Enter your iSeries user ID and password if prompted.

Adding the sample library to your initial library list

To make your library addition show up permanently when you expand your library list in the Remote system explorer, you can change your job description (associated with your user profile) to add the library to your initial library list. To add the library:

1. Open a 5250 emulator and sign on to your iSeries server.
2. Enter `dspusrprf <userid>` to display your user profile.
3. Press the page down button to scroll through your profile until you find the lines for Job description and Library. (Library appears right after Job description.)
4. Make sure that the job description is yours or one used by your team. If not, create your own job description and associate it with your user profile. If you need assistance, see your system administrator.
5. Return to the main menu and enter `chgjobd` and then press F4.
6. In the **Job description** field, enter the name and library for your job description. If you do not know the name of your job description, perform a `dspusrprf` to display your user profile. Your job description is on one of the pages of your user profile.
7. Press F10 to access additional parameters.
8. Press the page down button to scroll through your job description until you find the line **Initial library list**.
9. In the field beside the line that says + for more values, type a + followed by a space and then press Enter.
10. On the **Specify more values** display, type WHOLESale in the first available field and press Enter to change your job description. Repeat this step for WDSCLAB.

Starting your WebSphere Application Server instance

You need to restart your WebSphere Application Server instance each time it is stopped, and each time your iSeries system is restarted. Follow these steps:

1. Log onto the iSeries system and open a QShell prompt by using the `qsh` command.
2. Change to the `bin` directory of the installation directory of WebSphere Application Server using the `cd` command.
3. Enter the following command:
`strwasinst -instance wasinst`

where:

`wasinst`

is the name of the WebSphere Application Server instance you are using, and

Note that for WebSphere Application Server Express edition you use the `startServer` command instead of `strwasinst`.

4. When the command completes and you see a prompt character (a \$ dollar sign), along with a confirmation message on the console, exit QShell by pressing F3.

Mapping a network drive to the iSeries system

Web development tools deploys your HTML, JSP, servlet, and other publishable files (EJB for Development Studio Client Advanced Edition) to the iSeries system on a shared network drive. If you can map a network drive to the `/QIBM` directory of your iSeries system using the `NET USE` command or the `Map Network Drive` command, you already have `NET USE` access set up correctly. Follow these steps to set up `NET USE` access to the iSeries system, depending on your operating system.

For Windows® 2000 and XP:

1. Open a command prompt and type the command `NET USE`.
2. If you see a drive letter assigned to the `/QIBM` directory on your iSeries system, you are already set up for `NET USE` access.
3. Enter the following command: `NET USE * \\MYHOST\QIBM /USER:USERNAME *` where `USERNAME` is your iSeries user ID.
4. Enter your iSeries password when prompted.

If you can successfully connect to the network drive, `NET USE` access is set up properly. If instead an error message is displayed indicating problems connecting to the iSeries system, check with the system administrator that `TCP/IP` is properly started on the iSeries system. You can start it yourself if you have administrative privileges, by logging onto the iSeries system, typing `STRTCPSVR *NETSVR` and answering any prompts as needed. When this command completes, try the steps described above again. If you are unable to map a network drive to your iSeries system, contact the system administrator.

Notices

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this Documentation in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this Documentation. The furnishing of this Documentation does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director IBM Canada Ltd. Laboratory 8200 Warden Avenue Markham, Ontario, Canada L6G 1C7

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this Documentation and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 2000, 2005. All rights reserved.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- AIX
- AIX windows
- Application System/400
- AS/400
- AS/400e
- C Set ++
- COBOL/2
- COBOL/400
- DB2
- DB2 Extenders
- DB2 Universal Database
- @server
- IBM
- IBMLink
- Integrated Language Environment
- iSeries
- Language Environment
- MQSeries
- Operating System/400
- OS/2
- Open Class
- OS/390
- OS/400
- RPG/400
- SQL/400
- VisualAge
- WebSphere

InstallShield is a trademark of InstallShield Corporation.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation, in the United States, other countries or both.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group.

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

(C) Copyright IBM Corp. 2003, 2005. All rights reserved.



Printed in USA

GC09-7881-01

