

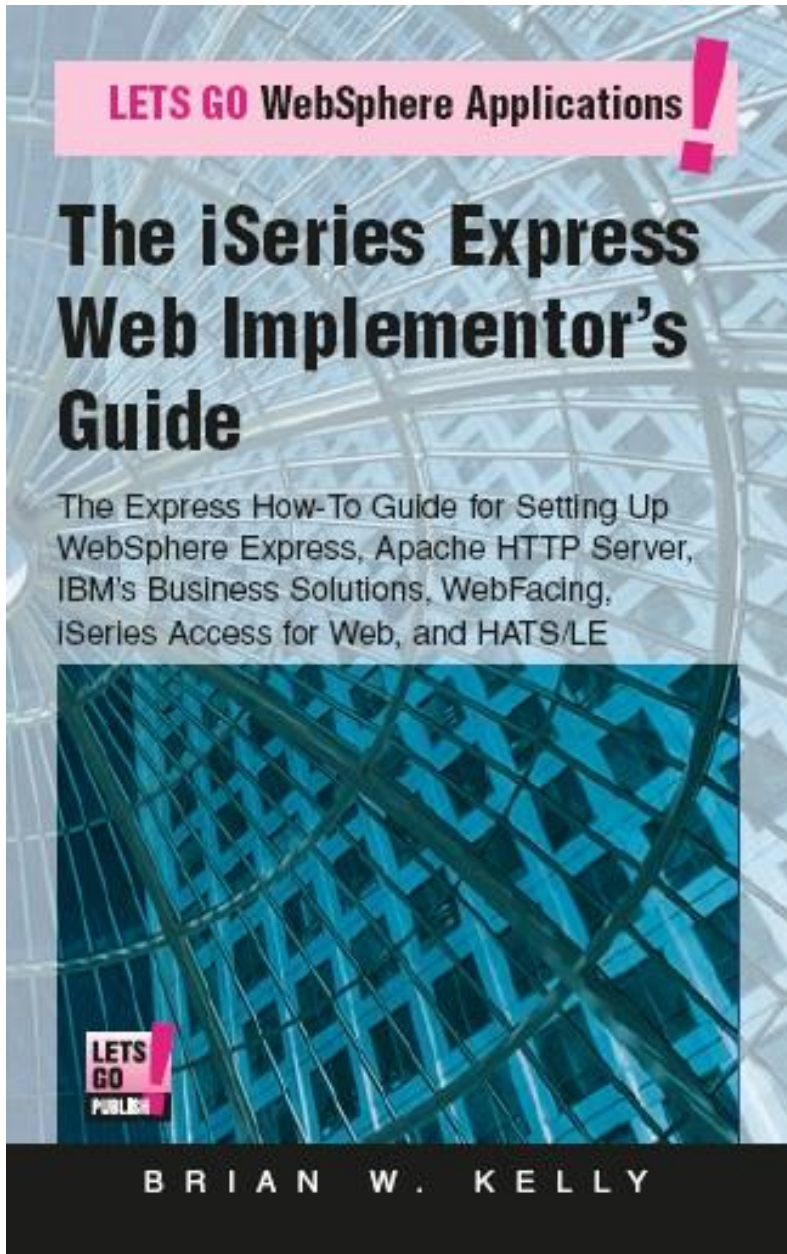
Dear Reader: Thank you for downloading this free book from Brian W. Kelly's finished book catalog. I finished the book titled **iSeries Web Implementer's Guide** at <https://letsGOPublish.com/technology/webguide> in 2004. This book shows how to put an AS/400 type machine on the Web.

Most of my books had previously been published on Amazon.

Click below if you would like to donate to help the free book cause:

<https://www.letsGOPublish.com/books/donate.pdf>

Enjoy!



Section I

IBM's Web Software for iSeries

**Apache HTTP,
WAS Express,
IBM Business Solutions,
WebFacing,
iSeries Access for Web, HATS/LE.**

Chapter 1

Software Ingredients for AS/400 and the Web

The Internet Is Covered

For many years IBM's subtle message to its AS/400 constituency was to do client server and the Web with Windows or Unix. For most of the 90's, while the Internet was in full swing, there was little sweat expended in IBM to make the AS/400 as fine and well known an Internet player as the native Unix box or the up and coming Microsoft Windows Web facilities.

During the early 1990's while Windows NT was just catching on and Unix owned the Internet world, IBM felt that it had the Internet covered in its product line since the company sold Unix boxes and PCs and both of these attacked the Web vigorously. Therefore, during this period, IBM let its AS/400 advanced development for the Web slide and ironically, the company did the same thing with the mainframe.

The Gerstner Influence

It took a "Cookie executive from Nabisco," an outsider to IBM to declare that all IBM servers were to fully Web capable. One might argue, however, that by the time Lou Gerstner issued his directive, the proverbial horse was already out of the barn. No matter how much

material and energy IBM could devote to the Web game for the AS/400 product and the mainframe product, the inertia of the Internet became a Unix thing and later a Bill Gates thing. Nonetheless, because of Lou Gerstner, Rochester and mainframe IBM were both empowered to bring their platforms to the Internet.

Free TCP/ IP

So, in 1993 and 1994, IBM's Endicott plant became the location in which a quality TCP/IP stack was to be built for both the AS/400 community and the mainframe community. It turns out that Endicott is just over an hour from my hometown; so I was able to meet some of these IBM developers one on one. At one of these sessions, IBM's TCP/IP architect at the time, a wonderful gentleman by the name of Armando Fratezi shared his big performance secret with me. Not only was the new TCP/IP eight times faster than the laggard product that had been the only thing available on the AS/400 prior to this effort, it was actually faster than the SNA/SDLC protocol that IBM had been championing since 1974. Mainframe performance was equally good compared to SNA/SDLC.

With a solid performing and highly functional base TCP/IP stack running on the system, the next good news was that it was free with the operating system, just as SNA/SDLC. Free and high performing meant that the new TCP/IP was nothing like the low grade licensed product and the \$26,000 toll that IBM had been exacting from unsuspecting AS/400 TCP/IP stalwarts before Gerstner's mandate. Unfortunately, however, IBM did cheat a bit with some of the applications such as FTP. The company reused the old Pascal programs that were not industry compliant. Therefore, it took awhile for what are termed the "well known TCP/IP applications" to mature to the point that they were not embarrassing for me to explain.

IBM chipped away at the big nut that had to be accomplished to make the AS/400 a real TCP/IP Internet machine. In the late 1990's, I wrote a book about the new TCP/IP, called the AS/400, The Internet, and email. It was the first book about the AS/400 and the Internet and for many months I am pleased to say, it was on Midrange Computing's bestseller list. Though IBM had done Gerstner's bidding with TCP/IP, there were still lots of missing or poorly capable pieces.

Missing Pieces

A few of the tell tale signs that my friends at College Misericordia, where I toiled in the mid 1990's used to help me know that the AS/400 was not quite the Internet machine that I would like it to be can be defined in the following missing functions list:

- ✓ Standard FTP facilities for transferring non AS/400 objects
- ✓ Domain Name Server (DNS)
- ✓ Apache HTTP Server (AS/400 used the old, less functional CERN HTTP Server).(Apache)
- ✓ Dynamic Host Configuration Protocol (DHCP).
- ✓ Compliant Telnet Client
- ✓ Secure Sockets Layer (SSL)
- ✓ Virtual Private Networking (VPN)
- ✓ Lightweight Directory Access Protocol (LDAP)
- ✓ Integrated Web Development Environment
- ✓ Integrated Dynamic Web (servlet) server

I can recall meetings with the College's ISP in my days in charge of networking at the institution. I had to bite my tongue as he positioned the AS/400 as anything but an Internet machine because it was missing such essential features as those above.

Preoccupation with Java

By 1999, the Web had finally reached AS/400 shops for sure but loyal AS/400 aficionados had already given the control of such facilities to the Windows and Unix crowd. Though Lou Gerstner had promised, the AS/400 still was not a fully functional Internet machine.

Rather than give the AS/400 native Web facilities in the same fashion as the System/34 pioneered native workstation facilities, IBM was preoccupied with Java. The IBM strategy was for AS/400 shops to learn Java as a prerequisite to getting on the Web. Just like IBM could not nudge its System/36 customers to like the AS/400, it could not

move AS/400 shops to Java. This strategy backfired and there was very little movement in AS/400 shops to the Web.

Soon, the trade press labeled the AS/400 a legacy system, and the loyal AS/400 constituency, once proud to deliver applications ten times faster than any other platform was forced to maintain silence. After all, IBM had not given them a functional tool bag with which they could build Internet applications.

AS/400 Web Development Begins

In 2000 at the Fall COMMON Conference, IBM previewed its 5722-WDS product, which, for the first time included a Web development component. The Toronto Lab Chieftains made a big deal about the fact that all of the new facility would be available on WebSphere Standard Edition V3.5, which was a free product.

By mid 2001, IBM released the WebSphere Development Tools to very little applause. Few people expected them and IBM did a poor job of telling people what they were all about. It looked to me at the time like IBM had finally become serious about Web development for the AS/400 using both RPG and COBOL, though it may have been too late. Those who saw the WebFacing component, in particular were very encouraged at this new IBM initiative. I for one was very impressed with what I saw.

Major Setback

Then, just about a month later, IBM canceled the free WebSphere Server version 3.5 and initially left its customer set the choice of purchasing the \$12,000 Advanced WebSphere version or being excluded from using the new Web facilities. Most chose the former.

For three more years IBM's AS/400 leaders fumbled in their attempts to figure out a way to circumvent the IBM Software Division's attempts to make a piece parts system of the integrated AS/400. At one point, in apparent frustration, AS/400 leaders announced that the then too expensive WebSphere would no longer be the AS/400's entry Web application server. For a brief period among the confusion,

Rochester selected the free Jakarta Tomcat server to take WebSphere's place in the AS/400 Web starter kit. Finally, IBM came to its senses and Rochester had to renege on its promise of support. But, there was still no free of charge Web servlet server for the AS/400 during this period, and there was little if any strategy direction coming from IBM.

A Change for the Good

In spring 2004, IBM introduced i5/OS V5R3. With it, the company reinstated a free WebSphere Server as a staple of AS/400 computing. By this time, the version had changed from 3.5 to 5.1 and the product was known as WebSphere Express.

Just as the ten years before hand, during the years from 2001 to 2004, after the super AD tooling had been promised, AS/400 shops had no real solution at the low end. Though WAS Express had come out in 2003, at the time, it too carried a price tag, albeit far lower than the WebSpheres of the past.

For three years on the workstation front, from 2002 through 2004, the company switched its tooling from the WDT model to the Eclipse model called WebSphere Development Studio Client (WDS_C). The current version is now based on the Eclipse framework and it is far more functional and far more stable than the early product that I used in 2001. So, perhaps it was providence that these tools are now ready to complement each other.

For this book, the only real WDS_C tool that we care about is the WebFacing capability that has been provided with the Version 5.1.2 or better WDS_C Eclipse product since it premiered in 2002 as Version 4. WDS_C provides many more facilities including a scratch Web development studio, a visual RPG, and a PDM like GUI development environment for green screen programming known as the Remote Systems Explorer. Overall, WDS_C is a very powerful set of tooling.

With the WebFacing facilities in WDS_C and the WebSphere Express provided with i5/OS (V5R3 of OS/400), AS/400 shops can convert RPG, ILERPG or COBOL or ILE COBOL or CL or ILECL

applications and menus to run on the Web under the control of WebSphere Express and the Apache Web Server.

Note:

In 2000, IBM premiered a functional Apache Web Server on the AS/400 platform and began to develop all applications, including WebSphere to work in this environment. With V5R3, the original HTTP server based on the CERN directives is no longer supported.

With V5R1 in 2001, IBM also introduced the iSeries Access for the Web product. This product was to provide ad hoc 5250 to Web translation for all AS/400 panels. However, the product relied on another product called IBM Host Publisher to provide full capabilities. It also required a WebSphere Server. It was a lot of baggage for an AS/400 shop to handle and therefore, most chose to give this product a pass. Those who actually tried to implement often stopped as soon as they learned about the cost of the WebSphere Server or the work involved in implementation.

Again, by 2003, this too was solved. IBM added a nice 5250 emulator that runs in a browser to the iSeries Access for the Web packaging. The product runs like a telnet on the Web and yet does not require a heavy 5250 Telnet client. It is very slick. Moreover, the new version does not need Host Publisher and it is much easier to install and make work in the free WAS Express environment.

IBM provided the final piece of the puzzle with the introduction of the Host Application Transfer Server Limited Edition (HATS/LE) product brought out in 2003 and almost perfected with i5/OS. I say almost because the installation is still somewhat obscure. From my perspective, it should be a natural installation with little work involved. Though it is not that, it is far simpler than HOST Publisher.

The HATS/LE environment provides a Web conversion to a GUI form in much the same fashion as WebFacing. Both run under WebSphere. However, HATS/LE provides its facility on the fly without any pre-conversion activity. Moreover, HATS/LE works with all OS/400 panels.

In the following chapters, we will describe each of these facilities in far greater detail. Please note however that the objective of this book is to get you to the point where you can install all of the tools, configure all of these tools, and be prepared to use all of these tools. This book does not address the operational aspects of these products other than to demonstrate their capability as a side bar to the set up and configuration effort.

My experience has been that once you have these tools up and running, using them is a fun next step and not as intimidating as the installation and setup stages.

Best wishes as you embark upon your AS/400 Web Journey.

Hopefully, one day, once again, you'll see your friendly Unix and Windows peers in your rear view mirror.

The Rest of the Book!

This book is divided into four sections and twenty-nine chapters. The sections are as follows:

Section I: Introducing IBM's Web Software for iSeries: Apache HTTP, WAS Express, IBM Business Solutions, WebFacing, iSeries Access for Web, and HATS/LE.

Section II: Prerequisites, Ordering, Installing, Fixing (PTFs) for IBM's Web Software for iSeries--- Apache HTTP, WAS Express, IBM Business Solutions, WebFacing, iSeries Access for Web HATS / LE

Section III: Setup and Configuration for IBM's Web Software for iSeries--- Apache HTTP, WAS Express, IBM Business Solutions, WebFacing, iSeries Access for Web HATS / LE

Section IV: Managing and Using IBM's Web Software for iSeries--- Apache HTTP, WAS Express, IBM Business Solutions, WebFacing, iSeries Access for Web, and HATS/LE.

Starting with Chapter 2, we take you through what is necessary to work with these AS/400 Web products from describing them to ordering, installing, fixing, configuring, and using them. We divide the tasks by section, so that you can get all your ordering done at once, including prerequisites, PTFS and whatever it takes to get these products up and running under WebSphere. And, of course a big part of that is getting WebSphere Express up and running itself. This is the only book that you need to install all of these products and get them to work in your shop.

I hope you enjoy the ride.

Chapter 2

What Is the Apache HTTP Server?

What is an HTTP server?

An HTTP server is another name for a Web Server. It uses a specific protocol (set of rules) known as the Hypertext Transfer Protocol (HTTP). With the HTTP protocol, Web servers can provide hypertext document support with the lightness and speed necessary for distributed, collaborative, hypermedia information systems.

The initial language for HTTP servers was the hypertext markup language, which was designed to provide hyper spots within documents to permit immediate links to other documents anywhere within the network of HTTP servers. This network of HTTP servers across the world is still referred to as the World Wide Web. The idea is that each server on the network is accessible from a browser running from a user's PC. HTTP has been in use as the server protocol by the World-Wide Web global information initiative since 1990.

If the whole world wanted to share documents and nothing more, the original HTTP protocol would have been all that was needed. But it was not enough. Organizations realized that computers had the capabilities to provide live, dynamic information to Web clients and

there were powerful business reasons as well as commerce reasons to make this happen.

Dynamic Data?

The first attempts at semi-dynamic data were to fill up documents on the fly. For example, for you to see your online transactions from your bank, the bank could have refreshed all the files in batch operations at night and embedded the necessary html for the files to appear nicely on your browser. When the Web document was fetched from the server, the data would be a day old but that was lots better than getting mailed statements once a month.

In this scenario, the server would also provide some security so that you could only see your transactions. Though the batch programs were much easier to write than interactive or browser based programs, this all or nothing approach would mean that the bank had to update all of these files on a very regular basis. If perhaps one of every 100 customers requested an inquiry each day, there was a lot of wasted computer work to establish all the files for all the customers once a day. Data was produced that few looked at. It would be far more efficient and flexible if the transaction data were produced only when it were requested.

CGI To the Rescue

Without application servers, the early HTTP servers solved this problem with scripting languages that operated through something called the Common Gateway Interface, or CGI. Though CGI has never had its own consortia or Internet task force as HTTP and applications servers do; the fact that it worked was enough for developers to begin to use it. However, CGI methods require lots of coding.

In the book titled Getting Started with the WebSphere Development Studio Client (WDSc) distributed by IT Jungle, iSeries Network, and MC Press, I defined a simple application, which consisted of one RPG program and a display file. The program took no more than 20

statements to do its whole job. My objective was to show how the program would look in many different scenarios, such as VARPG, WebSphere Studio, and WebFacing. To answer the question once and for all about CGI or Web Servers, I rewrote the application as CGI. The RPG program ballooned to more than ten times its original size.

From my perspective, CGI is not an inherently bad method, but it forces developers to perform all the user interface (UI) work in the application code. The WebSphere approach to AD takes away all UI coding from the program and places that work in the hands of Web Designers and developers working with Java Server Pages (JSPs). So, what would have been the display file in an interactive program is now a JSP in a Web program. Where the EXFMT would have gone is now a parameter list in the program that you use to pass the data back and force to the user interface. The RPG WebSphere program to perform the same function as the CGI program is about the same size as the original RPG code.

Note :

Java Server Pages, or JSPs, are a special form of html that makes it easier to build Web Pages.

Web Application Servers Replace CGI

In the mid 1990's the Web and the Java community got together to come up with a better way than CGI. They created application servers to support servlets on the Web. Servlets are server based Java-based applications. As e-business began to proliferate, more and more companies adopted the Web application server notion to help convert their Web sites from brochure-ware into enterprise-class application environments. It was just a few years ago in 1998, after its successful Nagano Japan Olympics Web hosting that the IBM Corporation entered this new market with its WebSphere Application Server.

WebSphere provides the base technology for supporting Web applications. A number of add-ons and upgrades to the technology, such as J2EE and XML, along with performance

upgrades, have had a major impact in helping make Web sites more reliable and scalable. Over the years, IBM and others have introduced a number of developers' toolsets to make it easier to deploy applications in this environment. The latest of these is the WDS*c*, which is built on Eclipse technology.

Tip:

For a description of the WDS*c* and WebFacing, you might want to read the **iSeries Pocket WebFacing Primer** from Lets Go Publish! It is distributed by IT Jungle, MCPress, and iSeries Network.

Server Side Applications Are In

Growth in the application server market has been driven by the growing need to support enterprise-class applications over the Web as well as internally. Companies are no longer interested in posting the same pretty brochures that their desktop publishing folks had been creating for years in paper form. They are interested in application server environments that they can use to support both Web and in-house distributed component applications using the browser as the universal client.

Basically, WebSphere, as all Web application servers, supports server-side Java applications that execute inside a Web server (such as Apache), and build Web pages dynamically, on the fly. Though Java started out as a client tool in applets, increasingly, it has been deployed as a server-side environment. There has been a trend, especially in larger companies toward using Java as a server technology (servlet), and away from using it as client technology (applet) in the browser.

And, as you would expect, WebSphere works hand in glove with your friendly Apache HTTP server.

Chapter 3

What is WebSphere Application Server? (WAS) Express?

What Does IBM Say?

If you check IBM's verbiage, you will find that WebSphere Express is a cost-effective, out-of-the-box solution with tools to create and run a simple dynamic Web site. IBM boasts that it gives you an onramp to e-business by providing the following major functions:

1. An application server and development tools in one tightly integrated package
2. A simplified programming model with wizards and samples that allows you to leverage existing skills
3. Minimal administrative effort
4. Smooth migration to other WebSphere Application Server (WAS) and WebSphere Studio configurations

Well, there you have it. Unfortunately, that's just about all that IBM has to say about WebSphere Express. Is that enough for you to know? It really doesn't tell you very much.

Basically, IBM is talking about how Express differs from its mainstream WebSphere versions. If you read the four points above, you learned that Express has its own Web application server and it also contains development tools (WDS). It also has a

simplified programming model and minimal administrative effort and it provides smooth migration from other WebSphere servers and other WebSphere development platforms.

What Is an Application Server?

SearchDatabase.com defines an application server as a server program in a computer in a distributed network that provides the business logic for an application program. This notion relies on the view that the application server is part of a three-tier application, consisting of a graphical user interface server, an application (business logic) server, and a database and transaction server.

A computer application of course is software that provides a solution to a business problem. So, an application server would be the component of a networked distributed system that provides the code to accomplish the purpose of the application. If the application is Web storefront catalog sales, for example, then the application server provides the ability to link to the elements in the computer system to provide the storefront function in a dynamic fashion.

Often applications are divided into three components:

1. A first-tier, front-end. In this tier, a Web browser-based tool, such as IE, Netscape, or Opera, at a personal computer or workstation provides the graphical interface to the user.
2. A middle-tier, business logic application(s). This is a set of applications, accessible on a local area network, intranet server, reachable from the Internet.
3. A third-tier, back-end. This piece provides the database and transaction server, sometimes on a mainframe or large server such as an AS/400 (iSeries or I5)

As you would expect, AS/400 databases and transaction-oriented applications are part of the back end or third tier. The application

server (WebSphere in our case) is the dynamic middleman between browser-based front-ends and back-end databases and AS/400 systems.

In the Web context, the application server combines or works with a Web (Hypertext Transfer Protocol) server and is called a Web application server. With regard to Web applications, because the development community is always dealing with Web applications, they simply refer to this layer as an application server though it is more easily understood in its proper context as a Web application server.

The Web browser supports an easy-to-create HTML-based front-end for the user. The Web server (HTTP server) provides several different ways to forward a request to an application server and to forward back a modified or new Web page to the user. In the Web application server arena, the method used is called Java Server Pages though there are other ways to provide dynamic data through a browser interfaces. These are not the subject of this book, but you have probably heard of them already.

This list of other dynamic data approaches of course includes the Common Gateway Interface (CGI), FastCGI, and Microsoft's Active Server Page. In some cases, the Web application servers also support request "brokering" interfaces such as CORBA and Internet Inter-ORB Protocol (IIOP). Aren't you glad you don't have to know about all this stuff to work with a Web application server such as (WebSphere Express) that we cover in this book

Express vs. Other WebSphere Servers

Basically IBM is positioning its WebSphere Express application server as a capable, easy to use subset of something much larger. Anyone who has seen other versions and then works with Express learns that it is much easier to install and use than the base and enterprise versions of the product. Moreover, for most iSeries

shops, it's all you need to get on the Web with dynamic data and minimal pain. Most AS/400 shops will not need the big WebSphere versions – at least for a few years.

Why Is the Big WebSphere Such a Bear?

When IBM ported the WAS to the AS/400 in the late 1990's, they used the AS/400's inherent Unix capabilities with the Qshell interface. The WAS did not look or behave much differently than it did on other platforms since IBM kept WebSphere in its basic Unix flavor as they enabled it to run on the iSeries. Unfortunately, because OS/400 is a different operating system than Unix, the natural Unix interface to WebSphere was a foreign item for iSeries professionals. It was scarcely easier, if easier at all for an iSeries professional to work with it on the iSeries than if it were deployed on an actual Unix box. It was a big trick played on the AS/400 professional community. Unless you already had a clue about Unix concepts, you were going to have a rough ride with IBM's early entrees of WebSphere on the AS/400 and iSeries.

New Clothes for WebSphere

The good news is that with WAS Express, most of that confusion is over. It is not completely, but now it is sufficiently masked so that you have to care about it just as much as you care about what object code looks like. WebSphere Express is still that big, burly, ugly, Unix servlet server ported to the iSeries. However, IBM gave it a new set of clothes and dropped a few less needed functions. Unlike the Emperor's clothes, however, Express actually looks quite handsome--- even pretty in its new GUI. The part of Express that you see is so nice, you can tell that it was developed by Rochester instead of IBM's generic software group. When code comes from Rochester or even Toronto, it does not have the

patented “sow’s ear” look that the Unix crowd continues to endure.

The bad news is that it is still a Java servlet server and it’s not native to the operating system. You can get over that since you now do not have to code in Java to use its facilities. With WebFacing for example, as well as iSeries Access for the Web, HATS Limited Edition, and the scratch development facilities shipped with WAS Express and included in WDS Eclipse, IBM builds all the Java and you simply write in RPG or COBOL.

Still Java Oriented

That doesn’t mean it won’t hurt to learn Java terminology if you’re headed to the Web. It will help. But, you don’t have to learn how to program in Java, despite IBM’s insistence that it was necessary only a few short years ago. Even with Express, once you start deploying your applications, you will be trapped in an environment where freedom only comes when you understand the terms. You can’t get really good at this stuff unless you have a clue about the terms used on the other side – servlets, classes, Java Server Pages (JSPs), JavaScript, XML, etc. Like it or not, the more you have a feel for these things, the better your overall WebSphere experience will be.

OK, So What is WebSphere?

Now that we know what an application server is and that Express is an easier to use subset of an application server called WebSphere, what should we know next? What is a WebSphere, and why does it have value?

In a nutshell, WebSphere is a Web application server. We gave the three-dollar definition of an application server above. Now, let’s have a look at the quarter version. A Web application server is a bolt-on to an HTTP server. An HTTP server is also called a Web Server and it

enables a host system to serve documents to browsers. Functionally, an HTTP server is a bolt-on to an operating system. When you attach a Web application server to an HTTP server, your browser cannot only fetch documents; it can also fetch dynamic data from an application. So, HTTP expands an operating system so that it's able to support document fetch and delivery to browsers, whereas applications servers expand the HTTP server to support real business applications with dynamic data.

WebSphere Summary

To summarize, WebSphere is a servlet server. Servlets are Java classes (programs) that are designed to work on the Web through a Web applications server. WebSphere is a Web application server. Without a Web application server, the most common TCP/IP way of delivering dynamic data on the Web has been CGI or the Common Gateway Interface, an enhancement to HTTP server that enables well coded applications to be called from HTTP servers.

Without a CGI facility or a Web Application Server, an HTTP server is only equipped to provide static (unchanging) html documents, such as home pages, from the server. To illustrate, an HTTP server without CGI or WAS could provide a basic html document, but could not provide accounts receivable data. In recent years, HTTP servers have been enhanced to support Web Application servers such as IBM's WebSphere and the Apache organization's Jakarta Tomcat, to enable companies to more fully participate in the dynamics of e-business.

WebSphere Express Makes It Easier

Now, WebSphere Express provides the power of WebSphere for the iSeries developer. Express for iSeries provides a tight integration with the iSeries HTTP server through an easy-to-use Web-based Admin GUI. Wizards allow for easy configuration of multiple application

servers and deployment of applications. The wizards allow for easy configuration of both HTTP servers and access to iSeries databases.

Based on the latest Java and Web services standards, WebSphere Application Server - Express for iSeries lets you convert static Web sites into dynamic Web sites by permitting you to view or update live data from the corporate database through your web browser. Though WebSphere itself has been built with Java in mind, the new tooling provided for AD gives iSeries developers the opportunity to make it all happen using RPG and COBOL as your primary Web languages.

A Few “Unnecessary” Pieces Missing

Though IBM offers that its WebSphere Application Server - Express for iSeries is based on the latest Java and Web Services standards, the company offers a light caveat that if more advanced development and deployment capabilities are needed, you might look at the Base or Enterprise editions. For iSeries, I suspect that Express will be all you need for some time. WebSphere Application Server - Express now offers an affordable (\$0.00) starter servlet server to other, substantially pricier WebSphere Application Servers.

Chapter 4

What is the IBM Business Solutions Pack (5722-BZ1)

In its attempt to attract iSeries shops to WebSphere, IBM has packaged a few really nice applications with the WAS Express entitlement for i5/OS V5R3. With V5R2, IBM shipped a limited telephone directory application and this time around with WAS 5.1 Express and WAS 6.x Express, the company has upped the ante with its Home Page application and its new online survey creator. All of these applications are quite appealing. Hopefully, IBM will ship some mini applications with WebSphere in the future such as the ones that the company help build for Bob Cancilla's recently released book published by IBM Press

For your convenience, I have included the true names of these new IBM Business Solutions immediately below:

1. IBM Survey Creator v1.0
2. IBM Telephone Directory v5.2
3. IBM Welcome Page v1.1

IBM Survey Creator

As you would expect, the IBM Survey Creator application is brought to us by the same folks who gave us the IBM Telephone Directory with v5.0 of WAS Express. This new, free, Java-based app lets us create, deploy, and manage online surveys from our Web sites, with options for nine different types of question and answer layouts.

As part of the application, you get to specify who can create surveys and who can take them. All the survey data – properties, question and answer text, layouts, and results - are stored in the iSeries' DB2 Universal Database, from which the results can be “mined” to create custom reports and charts.

As noted above, this writer's perspective of the purpose of the free applications is to get iSeries shops in the WebSphere frame of mind by giving them a working application that they can touch, feel, and dissect. Unlike the IBM Telephone Directory application from WAS 5.0 Express, which continues to be included at the newer Express versions, IBM is doing more than just giving us code to use. They have upped the ante considerably.

With the Survey Creator, IBM is making an attempt to help iSeries developers (often known as programmers in their own shops) know how to put Web applications together. Not to be negative here but it's about time IBM helped its loyal AS/400 shops a bit. So, it is good. The IBM company is even supplying the source code for Survey Creator and, of course it is free to all WAS Express iSeries shops with who are current with the software. Considering that all iSeries shops on V5R3 began to get WAS 5.1 for free, the whole deal smacks like the iSeries (i5 or AS/400) constituency is getting some nice attention from the corporation.

As noted previously, for the curious iSeries developer, IBM has even given the source code away for this application. Thus, iSeries shops can customize it for their own use as well as take a hard look

at its inner workings to see what makes the whole thing tick. There's no better way to learn short of an IBM internship.

Survey Creator runs on all versions of WAS V5R1 and higher. To get it, you need the latest PTF for IBM Business Solutions or just the WAS 5.1 package.

More Goods on Survey Creator

In its most simple definition, IBM Survey Creator is a Web-based business application used to create and conduct online surveys. As an application administrator in your own iSeries shop, you will be the one who decides the people in the company who are allowed to own and conduct surveys. It will be your job or someone else in your shop's job to manage a possibly large list of names taken from the registry of users within your enterprise.

Not everybody in an organization is typically at the same trust level. Thus, the Survey Creator program permits you to identify those people you choose as survey owners. In this way, they can create and conduct their own personal surveys from your enterprise web site. Survey owners then decide who is allowed (or not allowed) to take their surveys, what questions are asked, and how the answers are laid out.

The Survey Creator provides nine different question-types and answer layouts. Some are very simple formats but there are also layouts for multiple answer choices, open-ended text answers, numerical value answers, and data block answers. The package also provides for company surveys to be set up for anonymous polling. Of course it can also accommodate requiring participants to log in before taking the survey.

The registry of users is stored securely in a shared directory provided by the same iSeries lightweight directory access protocol (LDAP) server that runs the IBM Telephone Directory application. However, it does not have to be. It may actually be located anywhere within your network. The best news for an iSeries developer is that all of the artifacts of the survey, from properties,

to question and answer text, to layouts, and results are stored in the same iSeries' DB2 Universal Database that we have been work with for years.

Just as the surveys that you have taken online, the program provides a summary of answers given by survey participants. Just as you would expect, it may be viewed online or filed locally to allow even desktop applications to process them. Results may also be accessed directly from the iSeries database using SQL or Query or other programs to create your own custom reports, charts, and graphs.

Note from IBM :

The IBM Survey Creator Web application was added to the IBM Business Solutions product after initial product packaging. It is made available by applying an "add function" program fix (PTF) after the product is installed. Refer to the PTFs & Support page for information regarding program fixes (PTFs) for IBM Business Solutions.

Application Specific Features:

If you're like me, it is tough to imagine all the actual work involved in giving and processing a survey. If you have no need for such an application, then it has little value, other than educating you in how to get things done with Java on the Web. However, if you actually need to process surveys, the design piece alone can save you many months. Not only is the application well designed for a starter app, but it is also programmed and ready to go when you are ready to go.

The Function Short List

- ✓ Conduct online surveys from your web site
- ✓ Surveys can be anonymous or restricted
- ✓ Survey owners run the shows
- ✓ Names of survey owners and participants are saved

- ✓ Additional owners may be assigned for collaboration
- ✓ Questions may be required or optional
- ✓ “Nine” different question types and answer layouts
- ✓ A preview display is provided
- ✓ Owners see summary results... while survey occurs
- ✓ Summary results exported to common format

IBM Telephone Directory Application

The newest version of the IBM Telephone Directory at v5.2 is included in the WAS Express V5.1 set of enterprise Web applications provided by the new IBM Business Solutions package. At its core, IBM Telephone Directory is a Web-based business application ready to deliver a valuable service for any organization. It provides an alphabetical listings of names, telephone numbers, e-mail addresses, and job responsibilities of personnel within an organization.

And that’s not all. Other information may be listed in the directory as well. Information, such as postal addresses, fax numbers, pager information, department and office codes, and even pictures fit naturally within the IBM Telephone Directory environment. As an organization-wide application, it gives authorized administrators the ability to manage the information that is listed.

Optionally, the package can be tailored to enable additional features including the ability to allow non-administrators to add new directory entries (referred to as "open enrollment") or to login to search, update, and remove existing entries.

Advanced features include the ability for you and your colleagues to initiate chat sessions and online meetings using links found on directory listings, and to manage your identity relationships for a single-signon environment when updating your directory entries. This facility however requires a link to some Lotus ware.

Personnel information for the application is stored securely in a shared directory provided by a lightweight directory access protocol (LDAP) server that may be located anywhere within your network.

In summary, the special features of the IBM Telephone Directory application include the following:

Telephone Directory Features:

- Search & view personnel information
- Obtain directory listings based on criteria
- Sort listings by name, telephone number, etc.
- Views include a photo of the individual
- Add chat links to directory (Sametime) IM
- Allow users to add names to Sametime buddy lists¹
- Enable EIM registration and user identity management for your Network's single-signon environment²

Administrator Directory Management:

- Search, view & manage directory
- Add new entries, update & remove existing entries
- Set/change passwords³
- Configure application properties
- Enable optional features using GUI
- Anonymous directory entry ("open enrollment")

Non-Administrators Directory Mgt

- Manage directory information
- Update & remove entries
- Set/change passwords

IBM Welcome Page v1.1

The IBM Welcome Page v1.1 package is also included in the set of enterprise Web applications provided by IBM Business Solutions.

This is a Web-based business application designed to work in conjunction with the other business applications that you bring to the Web. Its purpose is twofold.

- It provides a common starting point for you and your colleagues to go to find links to other applications on your site - a simple welcome page specifically for business solutions.
- It provides administrative interfaces used for set up and configuration of all applications provided by IBM Business Solutions.

I am not suggesting that the Welcome page is a crackerjack beat-all and end all application. I might not even call it an application at all. It is not a page creator. It is just a page. However it does provide a way for IBM to tie its two applications (Survey Creator and Telephone Directory) together in a nice Web page setting.

Chapter 5

WebFacing Gets You to the Web

Shipped with WDSc

As a special feature for application development, IBM ships the WebSphere Development Studio Client (WDSc) for iSeries as an entitlement with 5722-WDS. The company has positioned the WDSc and its Eclipse WorkBench Environment as its next generation of WebSphere Development Tools for iSeries. It has consolidated key workstation development tools in both the client server and e-business application development areas. The company has standardized across all platforms on the Eclipse-based Integrated Development Environment (IDE) WebSphere Studio Workbench. WDSc is the embodiment of Eclipse on the iSeries.

The WDSc package provides the development tools necessary to build Web-based applications that can be used with the Apache HTTP Server and the WebSphere Application Server - Express for iSeries.

Overview

The WebSphere Development Studio Client for iSeries (a.k.a. WDS*c*) is here to stay. Though it may be a few more years until the majority of iSeries developers accept the notion, this tooling is really something. I suspect that it will eventually sustain the day; even from a PDM programmer's point of view, but not necessarily in this release. WDS*c* is closing in on a full integration with the Eclipse WorkBench, and IBM says it is the iSeries programmer development tool, for now and for the future.

For many who have used it, including the WDS*c* blog at http://www.iseriesnetwork.com/isnblogs/wdsc_survivor/, WDS*c* and all it does is not the easiest thing in the world with which to deal. But, that is somebody else's story. For WebFacing, WDS*c* is absolutely needed, and overall, it does a very nice job.

Many AS/400 and iSeries developers are aware that CODE/400 has been available for over ten years under different PC operating systems, from OS2 to the many flavors of Windows. This PC-based PDM-like combination green screen development manager and GUI Editor is now just about completely integrated into the WDS*c* Eclipse packaging. Eventually CODE itself as an entity will be phased out, as all of its functions become integrated within the WDS*c* Eclipse Remote Systems Explorer.

VisualAge RPG, another component shipped with WDS*c*, may be next in line for full integration with Eclipse. But, it's anybody's guess how or whether IBM plans to pull that off. Right now, VARPG is shipped in the same package as WDS*c* Eclipse, but it is implemented as an add-on product to the integrated WorkBench..

Besides these two super client server tools, WDS*c* Eclipse on its own, introduced two hot new facilities to the iSeries development environment – WebSphere Studio and WebFacing. The Studio-based Web Interaction Wizard provides the ability to develop custom Web applications driven by new RPG and COBOL programs. Recently the Studio functions have been rolled into the WDS*c* Web Perspective and so the Eclipse WorkBench is now the tool used by Java developers to build their applications. VA Java is no more.

WebFacing Is the Star!

There are few who have been working with IBM's new Eclipse-based tools who do not think they are quite the act. Though probably not quite ready to overtake PDM, this tooling is very impressive. As impressive as all of this tooling is, WebFacing is clearly the star of the show. And, it is getting better and better with each iteration. In essence, this new capability provides a migration path for the developer from DDS to Web pages.

To understand what WebFacing does functionally, assume that IBM just built a new option in the DDS Display file compiler so that in addition to the display file object, the compiler also built Web pages and all the other artifacts necessary to launch them from a Web browser – without you coding anything. This is exactly what WebFacing does for you. However, it does not work in conjunction with the DDS compiler. WDS*c* is the product that makes Web pages from your display file DDS.

Additionally, it builds a framework around those Web pages so that your interactive RPG or COBOL programs can use their natural display file interface (Write / Read) to interact with the Web pages as if they were display files. It's so transparent that your programs have no idea that they are driving Web pages.

With the same interface as the display file, the very same RPG or COBOL program that drives your green screen panels drives the WebFaced pages. If it sounds like it can't be that good, you may have been right two years ago. In its early stages it was not really that good. Now, it's reasonably mature and it is now worth the accolades.

The rough edges of the early versions of the product, known as the WebSphere Development Tools are mostly gone. Those left are rapidly disappearing. WDS*c* Version 4 was IBM's first Eclipse release with WebFacing, and though much better, it continued to speak of a work in process. However, when Version 5 of the Workstation Development Studio Client (WDS*c*) became available in April 2003, it took most of the "work-in-process notion" out of the iSeries AD strategy. Though not quite perfect, and perhaps not quite ready to

take on PDM, with WDS_c V5.1 and V5.1.2 in the summer 2004, WebFacing has arrived.

iSeries and AS/400 AD Mindshare

If the iSeries (i5) “problem” is not technology, then what is it? Quite simply, it is mindshare! The big question is, “Can IBM’s new gallant efforts in AD convince its loyal constituency to restart the client server and Web war against the often younger and more determined technicians in the company? Is the war to regain mindshare worth fighting? I really don’t know. But, one thing is for sure, the platform neutral advertising that makes the best system in IBM appear to be an also-ran, does not help mindshare.

Is there a good reason for an iSeries person to invest the time to learn all these new AD tools when IBM does not give them a clear signal that their system knowledge matters as much as the knowledge of the company’s other children. That is definitely a problem and a concern that many have about my favorite system.

Can We Help?

In April 2003, Lets Go Publish finished its first book series that covers this phenomenally large product set (5722-WDS). The second book in the two book series is called [Getting Started With WebSphere Development Studio Client \(WDS_c\)](#). It is available from IT Jungle, iSeries Network, and MC Press.

The book starts with a simple “Advanced Hello World” program and takes that one program that includes a display format write and read as well as a database lookup and works it through all of the major RPG and COBOL tooling in the 5722-WDS and the WDS_c packages.

The application is first manipulated using the CODE/400 GUI tool to teach the reader how to work with CODE development. In tutorial

fashion, the RPG application is migrated to VARPG, and the converted application is completely redone with buttons, bells and whistles to show the similarities and differences of RPG and VARPG. The application is also migrated to Java for applet deployment.

Moving from client server to Web development, the Advanced Hello World code is reworked as a CGI program for those still interested in this technology. This sets the RPG benchmark for lines of code for CGI vs. normal green screen and WebFaced applications.

Following the CGI work, the same application is then WebFaced, showing how a skinny RPG or COBOL program can drive a Web application, while the RPG or COBOL program still thinks it is working with a display file. As a final step, to show how you can customize Web applications and still use RPG and/or COBOL, the front end of Advanced Hello World is re-built from scratch and the RPG code is modified to support the constructs created by the Web Interaction Wizard facility in the WebSphere Studio.

The book is the second part (GUI) of a two-book series that gets you into the 5722-WDS product set. The first book deals with the green screen tools in PDM. Additionally, this book, titled Getting Started with the WebSphere Development Tools teaches the notion of user interface (UI) and Logic separation. Six different methods are explored for practice. The same Advanced Hello World program is deployed in six different green screen environments, in which the UI is separated from the Logic portion of the program.

In addition to teaching the notion of UI/Logic separation, which is a pre-requisite for scratch development for the Web, as a side benefit, the book shows how to gain back interactive cycles by switching heavy interactive workload to batch. Those using the techniques can save many interactive cycles by transferring most of the processing load in a program to batch. If you do not have an Enterprise Edition machine, this is a good way of maximizing your interactive CPU resources.

The book that you are now reading uses this same simple application to teach you about WebFacing. It has been written to work hand in glove with the new and exciting WebSphere Express and the Apache HTTP server. Instead of teaching all about these products and all of

their functions as we did in the other books, this work gets right to the meat of the matter. It shows how to install everything from the server to the applications, get them PTF'd and configured, and then it shows you how to use them to deploy your green screen applications on the Web

WDS*c* is shipped with 5722-WDS

WebFacing is shipped as a component of WDS*c* and WDS*c* is shipped as an entitlement to the 5722-WDS product that originated with V5R1.

The 5722-WDS components include the following:

Program Development Manager (PDM)

- Screen Design Aid
- Source Entry Utility
- Data File Utility
- Other host based development tools

Compilers

- RPG
- COBOL
- C, C++
- Java

Client Server Tools

- WDS*c* has replaced the following:
 - CODE/400 – maintain traditional apps
(now part of Remote Systems Explorer)
 - Visual Age Java – client server and Web Java
(now included in WDS*c* Web /Java perspective)
 - VisualAge RPG – client server and Web RPG
Shipped as separate component

Web Tools

Studio for iSeries
WebFacing

The Full Bundle

As you can see in the above list, IBM began to supply all they had in Application Development (AD) with the exclusion of SQL, for just one price. In addition to the host tools and compilers, and the client server RPG and Java programming and the GUI development for green screen applications, the company surprised its loyal constituency by addressing two major Web development areas that formerly had been taboo for iSeries developers due to the absence of good tools.

These two areas are repeated below for emphasis:

1. Scratch Web Development in RPG / COBOL (Studio)
2. Re-facing existing 5250 application for the Web (WebFacing)

WebSphere Studio

Within the first WebSphere Development Tools package, IBM supplied the tooling necessary for traditional RPG and COBOL programmers to develop Web application from scratch using features added to IBM's Studio package. The features were referred to as AS/400 affinity. These have been continued into the Web Perspective of the WDS versions 4.0 and 5.0 and up. The most predominant feature is the *Web Interaction Wizard*.

Web Interaction Wizard

The WDS Web Interaction Wizard guides you with a series of interrogatories and asks enough questions to define a full Web interaction. Based on the input fields and the output fields that you have defined, you then code an RPG program that contains the

equivalent of an entry parameter list. The parameter list is the only knowledge that the RPG or COBOL code has that there is an external source (browser) that will be bringing in the data and displaying the responses.

The Web Interaction Wizard creates the JSPs and other objects necessary to communicate to the RPG program through the parameter list. Though it is a more difficult process than using SDA, it makes fully customized applications possible on iSeries with the COBOL and RPG languages in the background serving up the data. It's a great idea and it works well in practice.

The Better Idea

WebFacing, however, is now a far better idea. It is more natural for the RPG developer. The product is stable and it requires minimal knowledge about the Web. Because of this, the scratch development tools as provided by the Web Interaction Wizard have been “eclipsed” by all of the hoopla surrounding this far more exciting notion.

WebFacing: The Undisputed Champion!

Though providing a package for RPG scratch Web development was a first for IBM, with its Studio offering, it was not as eye catching an announcement as WebFacing. We will continue with more about WebFacing and the WDS environment and the WebSphere environment, as soon as we introduce two newer tools (iSeries Access for the Web and HATS/LE) and after we describe how to get all of the components installed, PTF'd and set up for use.

No Interactive Penalty

With V4 and V5 of WDS, major enhancements were made to WebFacing. Probably the ultimate signal by IBM that WebFacing

really matters to the iSeries Division is that, with V5R2, the refreshed OS/400 no longer charges WebFaced applications at the Interactive CPW rate. Though your WebFaced program still uses a WORKSTN device, and your program does either an EXFMT in RPG, or a Write / Read WorkStation operation in COBOL, WebFaced applications no longer pay an interactive penalty. That is a big time concession from IBM. This is the biggest indicator of how serious the company now is in helping you bring your time-tested green screen applications to the Web.

Chapter 6

What is iSeries Access for Web?

Dynamic 5250 Capability and More

iSeries Access for Web (iAWeb) is part of the broader iSeries Access family of products known as 5722-WY1. The iSeries Access family was originally introduced in the 1980s as “IBM PC Support.” It was rechristened in 1994 as Client Access/400 and again in V5R1 as iSeries Access. The main component of the family has always been its Windows / 5250 support. Most of us are familiar with the name iSeries Access for Windows since it is available on just about every PC today that communicates with an AS/400 type machine.

The iSeries Access for Web product depends on the existence of a WebSphere server and thus, that alone explains why it has not been used extensively until now. WebSphere Express is now free with V5R3 of OS/400 and therefore products, such as iSeries Access for the Web and HATS/LE now have a free place to live when they are installed on an AS/400.

iAWeb does not have all of the facility of the iSeries Access for Windows product but it is coming on board quickly with more and

more function. Today it offers browser-based access to iSeries servers. IBM likes to suggest that the product enables end users to leverage business information, applications, and resources across an enterprise by extending the iSeries resources to the client desktop through a web browser.

The iAWeb product is server based and, as most Web server based facilities today, it depends on Java Servlet technology. As such it uses industry standard protocols - HTTP, HTTPS and HTML, IBM has taken care to make the product capable of providing very light and thin client facilities. For iSeries shops it is one of the first products that require only a browser on the client to access the AS/400 world.

In addition to 5250 emulation, the client provides the ability to run batch commands and it provides access to the iSeries database, integrated file system, printers, printer output, and messages. The 5250 interface is smart looking and easy to achieve. Probably the best news is that unlike prior versions when the product was gasping for function, no other products are required to give full access to your AS/400. You may recall the need for Host Publisher in the past. This is gone and that is good. Understanding Host Publisher and all of its wonderful but difficult to achieve functions was a major impediment to iAWeb implementation.

For some reason, IBM felt compelled to add another product to the mix, however, to replace Host Publisher. Despite throwing it in since V5R2 and V4 of HATS, the HATS/LE product has no dependency on iSeries Access for Web. In fact, if I were telling IBM how to package HATS/LE, it would be with WebSphere, not with iAWeb. iAWeb is not needed for HATS/LE and HATS/LE is not needed for iAWeb.

Telnet in a Web Browser

The 5250 piece of iAWeb is much like a dressed up Telnet running in a Web browser, whereas HATS/LE is like a fancy WebFacing application in that its panels are much more attractive than simple

Telnet. Unlike WebFacing, however, iAWeb and HATS/LE run ad hoc, on the fly, and therefore burn precious interactive cycles. By all means do not conclude that you can run your standard edition i5's using these two products. At a minimum you would need the i5 Express Edition to get some interactive and ideally, you would need the pricy Enterprise editions, depending on how much Web interactive work you need done. As you may know from Chapter 5, WebFaced applications have no interactive penalty.

Though the emphasis in this book is the WAS Express version, iAWeb and HATS/LE also run on all the new iSeries WebSphere versions, (V5.0 and 5.1) including the base and Network Deployment versions.

Chapter 7

What is Host Application Transform Server Limited Edition (HATS/LE)?

HATS/LE is Ready to Go

Host Application Transform Server Limited Edition (HATS/LE) is the newest member of the iSeries Access Family. As noted in Chapter 6, it has no real relationship with iAWeb and I speculate that it is shipped with the iAWeb product more as a replacement for the Host Publisher entitlement than because iAWeb needs HATS/LE.

The major claim to fame of HATS Limited Edition is that it dynamically transforms 5250 screens with a point-and-click Web interface that is fairly attractive. Screens are converted on the fly and delivered as HTML to the end user's Web browser (Internet Explorer or Netscape). The screens do not particularly look like they are 5250 terminal oriented and thus, that is the major differentiator between HATS/LE and 5250 component of the iAWeb product.

O June 30, 2003, IBM began to include HATS/LE with new shipments of V5R2 iSeries Access Family. For those customers

who had acquired V5R2 iSeries Access Family prior to this time, IBM makes HATS/LE available by ordering an iSeries Access Family Refresh. This information is covered in detail in Section II Ordering, Installing, Fixing, and Configuring.

Load and Go

A major difference between HATS/LE and WebFaced applications is that no front-end work is required with HATS/LE. In fact, HATS/LE does not need access to the display file DDS in order to do its “on the fly” conversion magic. It works directly with the applications’ pre-compiled display files in much the same fashion as Client Access for Windows.

A major differentiator is that HATS/LE driven 5250 applications can be spruced up in much the same fashion as WebFaced applications without all the work. Just like WebFacing, however, there is no need for programming skills while the product rejuvenates 5250 applications with a nice Web-like look and feel. It does this with a number of stock HTML templates that shape the data stream into a non Telnet look and feel. These templates can be tailored if you choose giving far greater capabilities than the iAWeb Telnet like capability. For example, you can change the company name and logo in the template to spruce up the look of the applications you use on the Web.

Of course, you can’t do any HATS/LE activity without WebSphere. Though IBM says it integrates with WebSphere Application Server, in actuality HATS/LE is an application that runs under WebSphere. Just as iAWeb, it runs on many different iSeries WebSpheres besides Express, including the following and more:

- WAS - Express for iSeries (5722-IWE) Express 5.0
- WAS - Express for iSeries (5722-E51) Express 5.1
- WAS V5 for iSeries (5733-WS5)
- WAS V4 plus latest PTFs Advanced Edition (AE) for iSeries (5733-WA4)
- WAS V4 AE Single Server for iSeries (5733-WS4)

IBM has focused HATS as a limited edition for the AS/400 but it has just about all you would want in an AS/400 shop. From my analysis the major missing piece is that HATS/LE does not support 3270 data streams and Host Publisher multiple server sessions but for those of us in the 5250 world, I actually feel like saying, “Who cares?”

Section II

Prerequisites, Ordering, Installing, Fixing (PTFs) for IBM's Web Software for iSeries---

Apache HTTP,
WAS Express,
IBM Business Solutions,
WebFacing,
iSeries Access for Web, HATS/LE.

Chapter 8

Apache, WAS Express, IBM Business Applications, Prerequisites, Ordering, Installing, Fixing

The Apache HTTP Server

WAS Express requires the Apache HTTP server to send its Web pages. The original IBM CERN server does not work at all after V5R2. So, there is every reason to move to the more functional and more solid Apache server if you have not already done so.

In this section of the book, we are outlining the prerequisites, ordering information, product installation and fixes for the products necessary to support the IBM Business Solutions, WebFacing, iSeries Access for Web, and HATS/LE. Both the Apache HTTP server, 5722-DG1 and WebSphere Express for iSeries (5722-IWE --Express 5.0 or 5722-E51-- Express V5.1 and now 6.X,) are the major prerequisites for all of this action.

Unless you do something unusual, these free products are shipped with the operating system if you have a client access entitlement since V5R3 in spring 2004. Thus, they eventually will be on just about everybody's machine. If you execute a **GO LICPGM** and then you page through the products, you should find 5722-DG1, which is the Apache HTTP server. If you do not find it, I would recommend calling IBM support to find out in which distribution CD the product is contained.

However, when you page through the products, you will not find WAS Express unless you have previously installed one of the versions,

Apache Should Already Be There

In order to install Apache (5722-DG1) if it is not on your machine, go back to your original installation media for your software release – V5R1 or better. Then, place the first CD in your reader and restore the product using option 11 from the LICPGM menu. Scroll through the products looking for 5722-DG1 and select it for installation. If it is not in the list, call IBM for support and ask what the best approach would be to get Apache back on your system.

If, on the other hand, you are like most of us, you will have 5722-DG1 loaded already on your system. There are no special prerequisites. If you pass the WAS Express prerequisites test, then you will pass Apache's with flying colors. When you apply the Group PTFs necessary for WAS Express, you will have applied the fixes for Apache.

Now, that was easy.

Hardware Requirements

Unless you have received your WAS Express with a new 8XX series machine packaged with the Enterprise edition, or a new i5 of any edition, you've got to check both hardware and software prerequisites to make sure that you have what it takes to run

WebSphere on your iSeries. Of course, if you are already running WAS Standard Edition or Advanced edition from any point in the past, you can count on WAS Express running on your old hardware. It may not run any better, but it will run.

IBM wants you to know that WebSphere burns a lot of CPU without you thinking of it in those terms. That's because they don't want you getting mad at them when you find out it does not run as well as your green screen applications on similar hardware. So, when they give the minimum hardware suggestions, the Big Blue Company is not precluding limited use on systems that do not meet the minimum hardware prerequisites. They just don't want you to get upset if you go below minimum and your performance is not good or not up to your expectations. Even with IBM's minimum recommendations, WebSphere performance has not made many of my customers feel that it offered good performance. The moral of the story is – more CPW is best..

You'll find IBM sneaking in some caveats as you read about WebSphere Express in IBM documentation. For example, IBM writes: “Systems that do not meet the recommended minimums may be used in environments that support a limited number of users and where longer server initialization times can be tolerated.” Just don't expect too much from a slow system if that's what you've got and do expect to be burning CPU on your fast system if that's what you've got.

For raw power, the IBM guidelines suggest at least 300 CPW. For memory, the absolute minimum is 512MB. The IBM recommended minimum server models are as follows:

- iSeries Model 270 with processor feature #2250
- iSeries Model 820 with processor feature #2395
- iSeries Model 800 with processor feature #2463
- Any i5 processor has at least 500 CPW

Since the minimum 810, 825, 830, larger 800 models, ad all i5 models are well beyond the 300 CPW limitation, they are by default all supported platforms for WAS Express V5.X and V6.X.

Disk Requirements:

When IBM first put its hardware needs together, the company chose to tell us about how much space the products took on disk. For example, the amount of disk space that the various pieces of the WAS Express product requires at the V5.0 level are as follows:

5 MB	*BASE: WAS Express,	5 MB
780 MB	Option 2: WAS Express V5 ,	780 MB
4 MB	Option 3: IBM Telephone V5.1	4 MB

So, for V5.0, you can count on needing 800 MB or more of disk storage to install WAS Express and its accoutrements. For WAS Express V5.1, you'll need a little more for the Survey Creator. Overall, WAS Express takes little space but the business applications that are shipped 5722-BZ1 with V5.1 of WAS Express do take more space than V5.0. Over time, as IBM packs even more usable applications in this BZ1 product, you can expect even more space to be needed. Watch the future releases as the mini applications hopefully will get much slicker.

Other Required Hardware

There is one more required hardware feature for you to have a complete WAS Express system that will function. You need at least One Communications adapter, such as an Ethernet Adapter or Token Ring adapter. The adapter must support TCP/IP, which is the base protocol for WebSphere.

If you are actually planning to implement a new application without much opportunity for seeing how the Express server performs, it is a good idea and it is recommended by IBM that you use some sort of estimator to figure out your workload. IBM suggests their AS/400 Workload Estimator for help with sizing all system configurations. Of course, you may want to engage a business partner since it is highly

likely that a major application effort on WAS Express would require a hardware purchase. Your business partner can be a competent ally in evaluating your hardware needs for your planned Express applications.

IBM is very aware and very sensitive of the substantial power required to run WebSphere. The company emphasizes that the above hardware requirements represent the recommended **minimum** requirements. If your implementation is to support many users or requires rapid response times, you should consider additional hardware resources. Again, smaller configurations may be possible for very light applications, or those applications in which longer response times are fully acceptable.

WAS Express V5R0 iSeries Software Requirements

The operating system level and the required software components necessary to run WAS Express V5R0 on your AS/400 or iSeries are as follows:

- WebSphere Application Server - Express for iSeries
- OS/400 V5R1, or later (in an unrestricted state)
- IBM Developer Kit for Java(TM) (5722-JV1 option 5) V1.3
- OS/400 Qshell Interpreter (5722-SS1 option 30)
- IBM TCP/IP Connectivity Utilities for iSeries (5722-TC1)
- HTTP server for iSeries (5722-DG1)

*Note – Lotus Domino HTTP servers are supported. On new versions (V5R3 and above), the original HTTP server (CERN) is not supported

WAS Express V5R1 iSeries Software Requirements

The operating system level and the required software components necessary to run WAS Express V5R1 on your AS/400 or iSeries are as follows:

- WebSphere Application Server - Express for iSeries V5R1
- OS/400 V5R1,V5R2 or i5/OS V5R3, or later (in an unrestricted state)
- IBM Developer Kit for Java(TM) (5722-JV1 option 5) V1.4
- OS/400 Qshell Interpreter (5722-SS1 option 30)
- IBM TCP/IP Connectivity Utilities for iSeries (5722-TC1)
- HTTP server for iSeries (5722-DG1) *Note – Lotus Domino HTTP servers are supported. On new versions (V5R3 and above), the original HTTP server (CERN) is not supported
- Cryptographic Access Provider 128-bit for iSeries (5722-AC3)

Some optional iSeries and AS/400 software products that you might consider include the following:

- 1. OS/400 Digital Certificate Manager (5722-SS1 option 34). This is required for running if you choose to use SSL.
- 2 A. iSeries Access for Windows (V5R1) 5722-XW1
- 2 B. iSeries Access for Windows (V5R2) 5722-XW1 iSeries access for Web is an entitlement (5722-XH2) HATS/LE is also an entitlement
- 2 C. iSeries Access for Windows (V5R3) 5722-XW1 iSeries access for Web is an entitlement (5722-XH2) HATS/LE is also an entitlement

Note – iSeries Access V5R3 is recommended.

WorkStation Requirements

WAS Express for iSeries is managed by an HTTP Administration Graphical User Interface (Admin GUI). Certain functions (very few) require the use of the WebSphere HTTP browser-based Administrative Console. Please note that this is not the same as the Admin GUI but it also works from a browser. Because the interface is a browser, the PC workstation requirements to manage the WAS Express must include the ability to support a Web Browser.

With the WAS Express package as well as the Application Development tools (5722-WDS), IBM also ships the WorkStation Development Studio Client (WDS_c). Because it is shipped with 5722-WDS, you may already have it since it is a workstation entitlement. 5722-WDS AD product also includes all the compilers and workstation tools. IBM ships WDS_c with Express for a number of reasons, including the following:

- Express works best with J2EE applications as produced by WDS_c.
- WDS_c is the cross platform solution for developing WebSphere Applications.
- Without WDS_c, or a like product, deploying applications to WAS Express would be very difficult.

Ordering WAS Express

I recommend using WAS Express V5.1 (5722-E51) or V6.x or later with the latest fixes. V5.1 is substantially better than WAS Express V5.0 and it became free with V5R3 of i5/OS. As a point of note, the full IBM Business Solutions 5722-BZ1 including the IBM Telephone Directory Application are shipped only with the WAS Express V5.1 and later packages. Telephone Directory alone is shipped with V5.0 of WAS Express for iSeries.

Ordering V5.0 WAS Express

Though I recommend WAS 5.1 or greater, many shops may already have WAS 5.0 (5722-IWE) from the deals that IBM had in place when the company discontinued WAS 3.5 Standard Edition. WAS Express 5.0 sells for \$2,000 for an unlimited license and it also is available for \$500.00 for a twenty-user license with a \$25.00 fee for each user license over 20. IBM's recommended approach for getting WAS Express 5.0 is to use one of the following

- ✓ Your IBM representative
- ✓ An IBM Business Partner
- ✓ IBM Americas Call Centers at 800-IBM-CALL
(Reference: AE001)
- ✓ The Shop IBM site on the Web

Ordering V5.1 or V6.X WAS Express

The best way to get WAS Express V5.1 is to order your i5/OS V5R3 upgrade package. It is included with your order. Other than for V5R3, the product is available via Passport Advantage but at prior levels of the OS to V5R3, it is a licensed product with a charge. It is not available as shrink-wrap per se for iSeries. It is ordered as PID 5724-D06, but installs as PID 5722-E51 on the iSeries system.

You can use the same sources as above to acquire the package. Again. My recommendation s to order the V5R3 media to get WAS Express. In fact, it would be most propitious to actually install V5R3 but this is not a requirement to ordering or receiving WAS Express V5.1 for iSeries.

Installing WAS Express for iSeries

One of your first considerations as noted above in the software prerequisites is to determine which version of the iSeries operating system you are going to use.

Migrating Applications

Your best bet under normal circumstances, though not in all circumstances, is to install WAS Express on a V5R2 or V5R3 or greater system. Unless you have applications to migrate, I would recommend going to i5/OS V5R3 first. However, if you are migrating any applications from WAS 3.5 or other WAS servers, you may want to run both Express and the other server on the same system for a period of time on the same OS/400 V5R1 AS/400 or iSeries.

Considering that WAS 3.5 cannot run on V5R2 or higher, going to V5R2 with live WAS 3.5 applications that must be converted to WAS Express is sure “not a good idea.” The best approach, especially for WAS 3.5 systems in this case, is to first get WAS Express running on your production Web AS/400.

Migrating WebFaced Applications

If you have WebFaced applications that run on an older WAS version, do not plan to migrate them. You should go back to the latest version of WDS*c* V5.0.2 at least and redo the WebFacing with the new client as supplied by IBM. Then load the re-converted WebFaced application back up to the new WAS Express to assure it works. Do not migrate it. Many DDS keywords that did not work in prior version now actually work and the WebFacing process is now much smoother than in past releases.

IBM has made WebFacing quite handsome and powerful. The same migration notes apply to code built with the Web Interaction Wizard and Studio. Moreover, if you’ve got Java applications to migrate, first bring them into WDS*c*, test them until they run right, and then bring them to WAS Express V5.1.2 or better on the AS/400.

Migration Help

To help your migration if that is a concern, IBM has a number of Redbooks at www.redbooks.ibm.com. As another source for your migration education IT Jungle, iSeries Network, and MC Press all

distribute a book that I can recommend since I wrote it. It is called: [Migrating to WebSphere Express for iSeries](#) .

WAS Express 5.0 Installation Instructions

1. Stop all running HTTP servers by issuing the following command

```
ENDTCPSVR SERVER(*HTTP)
HTTPSVR(*ALL)
```

Instructions for OS/400 V5R2 and WAS Express 5.0

Skip this section for V5R3

2. For V5R2, insert the CD that contains IBM WebSphere Application Server - Express for iSeries, V5.0 product (5722-IWE) in the iSeries CD-ROM drive. Then enter the following command to install the base product (WebSphere Application Server - Express):

```
RSTLICPGM LICPGM(5722IWE)
DEV(OPT01)
```

3. Install option 2 of IBM WebSphere Application Server Express V5. It is on the same CD.

**RSTLICPGM LICPGM(5722IWE)
DEV(OPT01) OPTION(2)**

D. Install option 3 (IBM Telephone Directory). To gain familiarity with the notion of WAS Express and how applications can be used, I recommend that you install this application:

**RSTLICPGM LICPGM(5722IWE)
DEV(OPT01) OPTION(3)**

We recommend installing the Telephone Directory application following the instructions above and in the Configuration Section to make the application functional. This gives a great perspective on the WAS Express and what it is all about. Plus, you may like the application. We show you how to use the application in Chapter 19.

Please note that for the above restores:

** *OPT01* is your CD-ROM or DVD-ROM Drive. Change the name as appropriate. A confirmation message appears when the *Base option is successfully restored. The *Base option and Option 2 are both required to run WebSphere Application Server - Express V5.0. Option 3 is required for the Telephone Directory application.

Additional information is available in the following Redbook:

**[http://publib.boulder.ibm.com/series/
v5r2/ic2924/info/rzatz/51/
install/install.pdf](http://publib.boulder.ibm.com/series/v5r2/ic2924/info/rzatz/51/install/install.pdf)**

**Instructions for I5/OS V5R3 and WAS
Express V5.1**

As WAS Express 6.x matures, its instructions can be found on IBM's Web site.

V5R3 with V5.1 is a more natural install than V5.2 with V5.0 of Express and it does not require all of the checking for prerequisites as do the other flavors of installation.

1. Insert CD #1 with WAS 5.1 Express product and type:

Go LICPGM

2. Pick Option 1 (Install All), press Enter

3. When the installation of WAS Express 5.0 completes, remove the CD and mount the IBM Business Solutions CD (5722-BZ1). The ebusiness applications, including the Telephone Directory are shipped with WAS Express but are not part of the natural WAS Express installation. They get installed separately by removing the WAS 5.1 CD, placing the ebusiness CD in the CD drive, and entering the following:

```
RSTLICPGM LICPGM(5722BZ1)
DEV(OPT01)
```

Fixing (PTFS)

My recommendation is that you install all the products before you begin to apply the PTFs. The only exception to this would be if the installation of a product is buggy and needs the cumulative pack or another PTF group to assure a proper installation.

Having said that, to keep the presentation of all these facts in a logical sequence, I will repeat the PTF process for each specific product in its Fix section. In this way, the information will be easier to find.

Ordering PTFs for WAS Express, IBM Bizapps for V5RX

The first Fix action is to order your cumulative PTF package close to the time that you are going to perform your software installation (all software products). You order cumulative PTFS simply on a command line by entering the following command:

```
SNDPTFORD PTFID ((SF99VRM))
```

Where V = Version

Where R = Release

Where M = Modification level -

So, the command for ordering all the cumulative PTFs for V5R3 is as follows:

```
SNDPTFORD PTFID ((SF99530))
```

For V5R2, the command is the following:

```
SNDPTFORD PTFID ((SF99520))
```

When you perform the SNDPTFORD, press F4 and you will get the following panel as shown in Figure 8-1:

Figure 8-1

```

Send PTF Order (SNDPTFORD)

Type choices, press Enter.

PTF description:
PTF identifier . . . . . > SF99530          Character value
Product . . . . .          *ONLYPRD       F4 for list
Release . . . . .          *ONLYRLS       *ONLYRLS, VxRxMx
                               + for more values
PTF parts . . . . .          *ALL          *ALL, *CVRLTR
Remote control point . . . . . *IBMSRV    Name, *IBMSRV, SELECT
Remote network identifier . . . *NETATR   Name, *NETATR

Additional Parameters

Delivery method . . . . . *LINKONLY  *ANY, *LINKONLY  ***
Order . . . . .          *REQUIRED  *REQUIRED, *PTFID
Reorder . . . . .          *NO        *NO, *YES
Check PTF . . . . .          *NO        *NO, *YES

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
Bottom

```

From here, just press Enter and fill in the appropriate panels to cause the cumulative PTF pack to be ordered. You do not have to worry about delivery method for these PTFs since the SNDPTFORD command will cause them to be sent on CDs via the mail.

Applying WAS Express V5.1 PTFS on i5/OS V5R3

1. Once you have the product installed and you have received the cumulative PTF CDs from IBM, you can apply the cumulative PTF pack by typing the following:

GO PTF

2. From the Main PTF display, take

option 8 and press ENTER

3. You will see a panel such as that below in Figure 8-2.

Figure 8-2

```

Install Options for Program Temporary Fixes
System: HELLO
Type choices, press Enter.
Device . . . . . OPT01      Name, *SERVICE, *NONE
Automatic IPL . . . . . N      Y=Yes
                                N=No
Prompt for media . . . . . 1    1=Single PTF volume set
                                2=Multiple PTF volume sets
                                3=Multiple volume sets and *SERVICE
Restart type . . . . . *SYS    *SYS, *FULL
Other options . . . . . N      Y=Yes
                                N=No
F3=Exit  F12=Cancel

```

From here, type in the device name for your CD-ROM (OPT01 in this case and then you have to decide whether you can IPL or not. If you can IPL immediately, then type Y in Automatic IPL. If not, type N and you can load the cumulative fix pack for later application. Leave all the other options at default and press ENTER. The next time you do a powerdown, the PTFs will actually be applied.

Note: Because there are many PTFs in the groups that supersede the LIC cumulative PTFs (5722-999 - MFxxxxx), with V5R3 it is not recommended that you load additional PTFs from the groups until you apply the cumulative PTFs **with an IPL**.

The Group PTFs for Express on iSeries

Since there are two versions of WAS Express available (5.0, and 5.1) and there are three versions of the OS with which these various versions work, there is a combination of six options of Group PTF from which you must choose. Be careful that you order the right Group PTF for your WAS Express / OS/400 combination. The six combinations are as follows:

SF99270 for WAS Express 5.0 on OS/400 V5R1
SF99271 for WAS Express 5.0 on OS/400 V5R2
SF99272 for WAS Express 5.0 on OS/400 V5R3
SF99273 for WAS Express 5.1 on OS/400 V5R1
SF99274 for WAS Express 5.1 on OS/400 V5R2
SF99275 for WAS Express 5.1 on OS/400 V5R3

If you are installing WAS Express V6 as it becomes mature, these instructions are also available on the IBM Web site.

When I began the iSeries WAS 5.1 Express (5722-E51) installation on my V5R3 system, and I called up the IBM Web Documentation for installing the WebSphere Application Server Express (WAS Express) for iSeries, it clearly stated that I needed Group PTF # SF99275. I have a good habit of checking cumulative PTF levels whenever I order a group PTF. So, by habit, I ordered the cumulative CDs at the same time. As I read the documentation for the **SF99275** Group PTF for WAS Express 5.1 on V5R3, I came across a few interesting tidbits. The tidbit lines are as follows:

V5R3, Express 5.1 Group PTF Tidbits

APPLYING THIS GROUP PTF WILL ALSO
APPLY THESE GROUP PTFs:
SF99503 DB2 UDB for AS/400
(Database) Group
SF99269 JAVA Group
SF99099 IBM HTTP Server for AS/400
Group
SF99173 IBM Business Solutions

****WARNING****

THE 5722E51 PTFs IN THIS GROUP
PTF MUST ALL BE APPLIED TOGETHER.
IF IT IS NECESSARY TO REMOVE THE
5722E51 PTFs, REMOVE ALL THE
5722E51 PTFs.

****WARNING****

Cumulative PTF package must be
applied.
Apply cumulative PTF package
C4153530 or higher.

In essence, this one group PTF for WAS Express V5.1 also includes the machine procedures that apply the non-cumulative Group PTFs listed in the Tidbits above. Therefore, you do not have to order these Group PTFs separately or install them separately. When you order the WAS Express group, it automatically triggers an order for the PTFs listed above. Notice above that there is also a PTF for the new **5722-BZ1 (IBM Business Solutions)**. This is different from the PTF information tidbits for V5.0 of WAS Express for iSeries.

V5R2, Express 5.0 Group PTF Tidbits

For those installing 5722-IWE (WAS Express V5.0.0) on V5R2 of OS/400, there are a similar set of tidbits for Group PTF # **SF99271**:

THIS GROUP PTF WILL ALSO APPLY THESE GROUP PTFs.

SF99098 IBM HTTP Server for AS/400 Group

SF99169 JAVA Group

SF99502 DB2 UDB for AS/400 (Database) Group

****WARNING****

A number of 5722IWE PTFs are concurrent PTFs. These PTFs have CONCURRENT in the title.

Removing a PTF with a label of concurrent requires removing the other concurrent PTFs. Unpredictable operation may result in doing differently.

****WARNING****

Cumulative PTF package must be applied.

Apply cumulative PTF package C2344520 or higher.

Planned Update Schedule: Approximately once a month

Again, this one group PTF for V5.0 of WAS Express also includes the machine procedures that apply the non-cumulative Group PTFs listed in the Tidbits above. Therefore, you do not have to order these Group PTFs separately or install them separately. Notice above that there is no PTF for the new 5722-BZ1 (IBM Business Solutions) since it is not a component of V5.0 WAS Express for iSeries. This is different from the PTF information tidbits for V5.1 of WAS Express for iSeries.

Two valid combinations not discussed above include the following:

Installing 5.0.x WAS Express on V5R3 of I5/OS **Group PTF # SF99272**

Installing 5.1.x WAS Express on V5R2 of OS/400 **Group PTF # SF99274**

Of course all PTF information is fleeting (only current when it is current) so your best bet is to visit IBM's iSeries Web site at

www.as400.ibm.com

To get the specifics regarding Group PTFs for your version of i5/OS or OS/400 and your version of WAS Express, take the option for **Support** in the left frame. Then take the option for **Fixes** under **Popular Links** in the right frame. From here, take the option for **Group PTFs** in the right frame. Unless IBM changes its Web site again, your next choice will be the version and release of OS/400 or i5/OS. From here, you will be presented with a list of all Group PTFs for the version you are checking. Find the version of WAS Express you are using and the group PTF # will be next to it. Order the appropriate Group PTF. While you are there, click on the link so that you can read the tidbits for your circumstances as I have presented for the two options above.

Installing Express on OS/400 V5R1

By the way, For the V5R1 folks installing V5.0 of WAS Express, there is a similar tidbit. Again, you get to see those tidbits by taking the following link

**<http://www.iseries.ibm.com> >>
Support >> Fixes >> PSP >> All
Group PTFs by Release >> V5R1
(etc.)**

If you must stay on OS/400 V5R1, these are your choices:

**SF99270 for WAS Express 5.0 on
OS/400 V5R1**

**SF99273 for WAS Express 5.1 on
OS/400 V5R1**

The Group PTF Process

I would recommend that you install all products first, and then install all PTFs. The Group PTF application process is very similar to the cumulative PTF process. However, in almost all cases, unless advised otherwise by IBM, after applying the cumulative pack and performing an IPL, you can load all of the group PTFs on the system (for all products) and, only on the last group would you chose the IPL option. Of course, you can also load all the PTFs and select no for IPL each time including the last time. Then, at your convenience, you can perform a power down and the PTFs will be applied on the restart.

Of course, just like the cumulative PTFs, you must order the Group PTFs. To do this, enter the SNDPTFORD command and press F4. Fill out the panel in Figure 8-3 for the particular group that you are ordering. In the case below, we are ordering the iSeries WAS 5.1 Express Group PTF for i5/OS V5R3

Figure 8-3

```

Send PTF Order (SNDPTFORD)

Type choices, press Enter.

PTF description:
PTF identifier . . . . . > SF99275          Character value
Product . . . . .          *ONLYPRD      F4 for list
Release . . . . .          *ONLYRRLS     *ONLYRRLS, VxRxMx
                               + for more values
PTF parts . . . . .          *ALL          *ALL, *CVRLTR
Remote control point . . . . . *IBMSRV   Name, *IBMSRV, *SELECT
Remote network identifier . . . *NETATR   Name, *NETATR

Additional Parameters

Delivery method . . . . . *ANY          *ANY, *LINKONLY   ***
Order . . . . .          *REQUIRED     *REQUIRED, *PTFID
Reorder . . . . .          *NO           *NO, *YES
Check PTF . . . . .          *NO           *NO, *YES

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
Bottom

```

When ordering Group PTFs, always be careful to press F10 so that you can change the delivery method to *ANY. The SNDPTFORD command is not as smart with Group PTFs as it is with cumulative PTF orders. If you do not specify *ANY for the delivery method, the command will fail because the Group PTFs are not available for downloading via this command.

Applying the Received Cumulative and Group PTFs from CD

The cumulative / group PTF process for PTF application is as follows:

1. Type **GO PTF**
2. From the Main PTF display, take option 8 and press ENTER
3. You will see a panel such as that below in Figure 8-4.

Figure 8-4

```

Install Options for Program Temporary Fixes
System: HELLO

Type choices, press Enter.

Device . . . . . OPT01      Name, *SERVICE, *NONE
Automatic IPL . . . . . N      Y=Yes
                                   N=No

Prompt for media . . . . . 1      1=Single PTF volume set
                                   2=Multiple PTF volume sets
                                   3=Multiple volume sets and *SERVICE

Restart type . . . . . *SYS      *SYS, *FULL

Other options . . . . . N      Y=Yes
                                   N=No

F3=Exit  F12=Cancel

```

From here, type in the device name for your CD-ROM (OPT01 in this case and then you have to decide whether you can IPL or not. Do not type in the Y for IPL until you have installed the cumulative PTFs and all of the group PTFs for all of the installed products, including iAWeb.

If you can IPL immediately after the last group, then type Y in Automatic IPL. If not, type N and you can apply all of the groups at once at a later time. Leave all the other options at default and press ENTER. The next time you do a power down, if you had selected the N option for IPL in the cumulative and Group PTF panels such as Figure 8-4, all the group PTFs will actually be applied.

PTF Information

From the information above, in addition to applying the cumulative PTF with the above method, by applying all the Group PTFs, you also apply the group PTFs for HTTP, JAVA, DB2, and for V5R3, IBM's business applications, 5722-BZ1, including the IBM Telephone

Directory Application. (See Tidbits section above) That's good to know, since ordering these groups after you order the WAS Express Group PTF would be redundant.

Two other tidbits in the documentation I show are that cumulative PTF # C2344520 is a prerequisite for the WAS Express V5.0 group PTF for OS/400 V5R2, and cumulative PTF # C4153530 at V5R3 is a prerequisite cumulative PTF for the WAS Express 5.1 Group PTF for i5/OS.

For example, if you are at C2211520 at the time you are ready for a WAS Express 5.0 installation on V5R2 of OS/400, you would know that you need to apply the cumulative tape and IPL for a full apply, before you can install the WAS Group PTFs. When you are not sure, a good rule of thumb is to read the PTF documentation. However, I admit that sometimes this still does not help. In these cases, call your friendly IBM support folks and ask them your questions. They are very helpful.

Caution: Make sure that your IBM Support contract includes support for WebSphere Express. At the time of this writing, this was available only for those customers who purchased a Passport Advantage contract. Call your IBM Business partner to assure that WAS Express is covered for support.

The Overall PTF Strategy

In a nutshell, my advice is to install all of the products you are interested in. These include iSeries Access and iSeries Access for the Web and of course WebSphere Express, including the 5722-BZ1 business applications or the V5.0 Telephone Directory CD. Use the Chapters 9 and 10 to order, install and Fix the rest of the products in this Section following this chapter to handle WebFacing and iSeries Access, and HATS/LE.

When you have all the products installed, then apply the cumulative PTF with a full IPL to assure that the LIC PTFs are all applied before you begin to install the Group PTFs. After the cumulative PTF installation is complete, install the groups as outlined above. Also, be

sure to apply the client access PTFs as noted below. This will also take care of the bulk of the PTFs needed for iSeries Access for the Web.

Note: At the time of this writing there were no PTFs for HATS/LE (Chapter 10). However, one would expect that there will be some PTFs. Since HATS/LE is shipped as an EAR file (explained in context in the WebFacing Chapters beginning with Chapter 20), I would expect that this would be made available for download when IBM chooses to fix HATSLE. You may want to call IBM at some point in the future to ask about how you would get these fixes since such documentation is not now available and IBM support reps are unaware as to how this will be done.

Chapter 9

WebFacing Prerequisites, Ordering, Installation, Fixing

Hardware Prerequisites

To support WebFacing on your PC, you need the IBM WebSphere Development Studio Client. My recommendation is that you assure that you have ordered the latest version of WDS (V5.1.2 as of December, 2004) for your PC Development WorkStation, IBM recommends the following minimum hardware and software:

1. Intel(R) Pentium(R) II processor (600 MHz), or higher, recommended
2. 512 MB minimum RAM needed for testing. Recommend 1GB or better. Integral WAS Test Environment in WDS requires 512 MB RAM
3. For CODE or VisualAge RPG, a minimum of 128 MB RAM is required.
4. 2 GB required hard drive space:
5. 600 MB of temporary hard drive space is needed during product installation
6. VGA graphics card (800 x 600, or higher, recommended and 256 colors)

7. DVD drive preferred – CDROM drive will work
8. Mouse or pointing device

Software Prerequisites

The software requirements for your PC workstation are as follows:

Web browser that supports HTML and CSS. TCP/IP must be installed, configured, and functional on your LAN based workstation. The Requirements for WebSphere Development Studio Client for iSeries include the following:

For Windows 98, ME and XP:

Microsoft Internet Explorer 5.5, SP1, or later

I do not recommend Windows 98 or Windows ME. Do not use these if you want to have an installation that does not “hang” frequently.

For Windows NT:

Windows NT Server or Workgroup V4.0, SP6a, or later
Microsoft Internet Explorer 5.5, SP1, or later

For Windows 2000:

Windows 2000 Professional, Server, or Advanced
Server, SP1, or later
Microsoft Internet Explorer 5.5, SP1, or later

My favorite types of machine for WDS Sc is a 1 or 2 GB Memory Pentium IV class machine with a 10000 RPM disk running Windows 2000 Workstation or XP professional. It’s like the saying in the movie **Love Story**. Just as “love means never having to say you’re sorry,” having lots of power and memory and disk speed for WDS Sc means never having to say you’re sorry.

Ordering Information

WebFacing is a component of the Eclipse based WebSphere Development Studio Client (WDS), which is shipped with the base 5722-WDS product as an entitlement. Therefore, you do not have to order it. It comes with products that every development shop already has.

WDS Installation and Fixing (PTFs)

As described in this book, WebSphere Development Studio Client (WDS) is the tool under which WebFacing works. It runs on a very healthy Intel Workstation. The purpose of this book is not to teach or even describe WDS but to ignore its existence so that we can better understand WebFacing. There is no prevarication intended in this section. However, if we assume that once WDS is installed, our purpose is WebFacing and not all the wonderment of WDS itself, our mission is very much more accomplishable.

When you receive your 5722-WDS products, there will be several numbered CDs and just one DVD. The one DVD contains the same information as the multiple CDs. Depending on your workstation hardware, you can use either media to install WDS. If you have a choice, select the DVD option for it will make the installation much easier and less prone to error.

When you receive your packet, you will note that the DVD is the first medium in the packet. There are also several numbered CD-ROM disks.

To install WebSphere Development Studio Client for iSeries, regardless of whether you use DVD or CD, the instructions are the same. However, you will change disks about five times in the CD version.

1. Insert disk 1 into the **CD/DVD-ROM/RAM** drive.
2. If installation does not start automatically:
 - a. Click **Start**, and then click **Run**.
 - b. Type **x:setup**, where **x** is the **CD-ROM** drive letter.
 - c. Press **Enter**.
3. Follow the on screen instructions for **WebSphere Development Studio Client for iSeries** as shown in the following panels

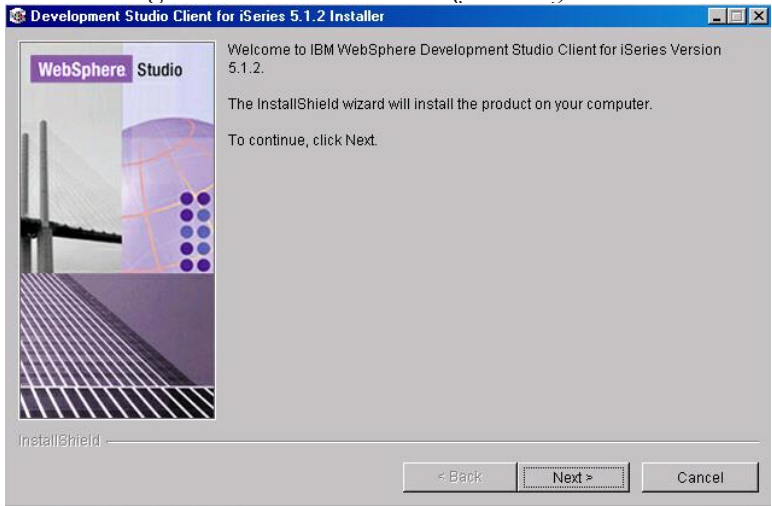
Not all the panels that you see are shown in the sample installation below. The installation is very logical, however. Pick what you want and then press **Next** to continue. The CD installation is far more arduous than the DVD installation. You will have to observe several different zippers and several different start/restarts over five CDs, vs. one DVD run and go.

When the installation starts, you will see a panel as shown in Figure 9-1 and after a while, it changes to the panel as shown in Figure 9-2.

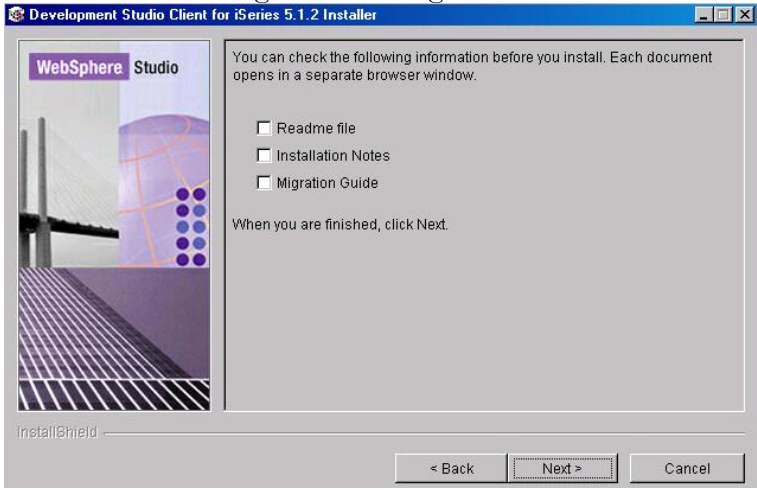
Figure 9-1 Wizard is Initializing



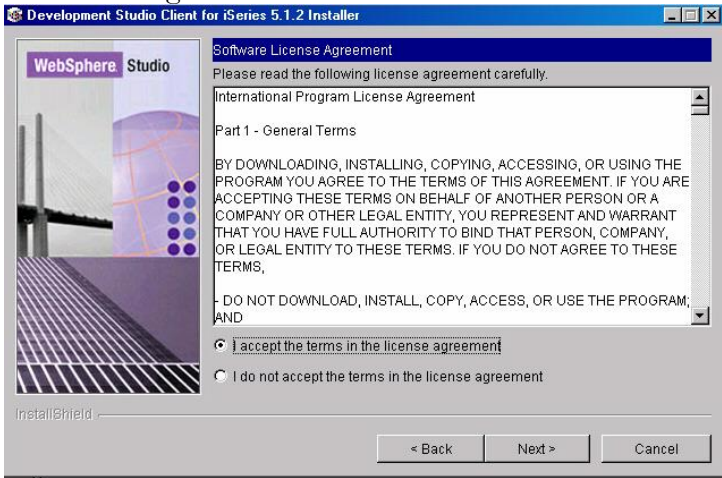
Figure 9-2 WDS Sc Install Wizard Ready To Go



To get the process moving, click on the Next button as shown in Figure 9-2

Figure 9-3 Reading Choices

If you choose to do some helpful reading along the way, select the three options as shown in Figure 9-3, and in all cases click Next to proceed.

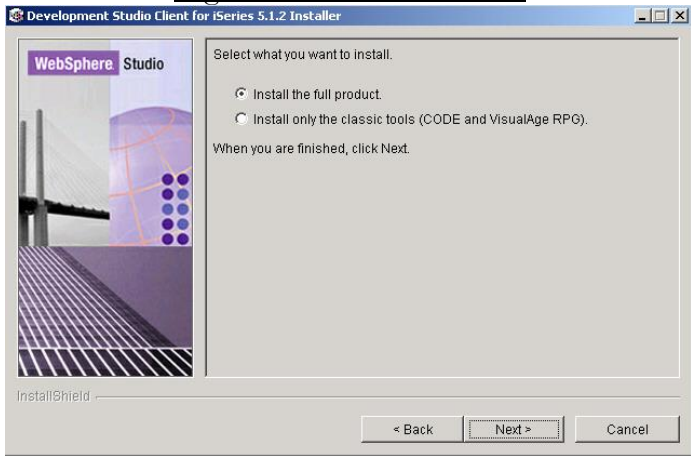
Figure 9-4 IBM's Terms and Conditions

You must accept the terms in Figure 9-4 by clicking Next or your installation will abort. The installation process then begins to give you information about what it is doing as shown in the panel in Figure 9-5. Then it's got some more questions as shown in Figure 9-6.

Figure 9-5. Extracting Installation Code

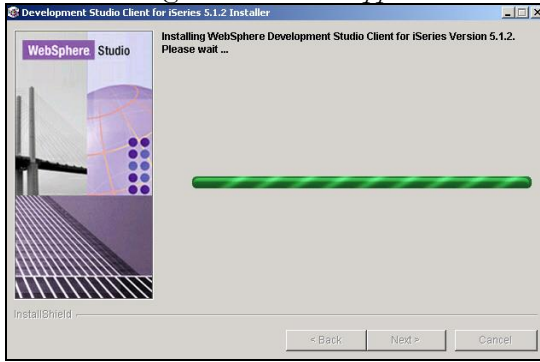


Figure 9-6 Installation Selection



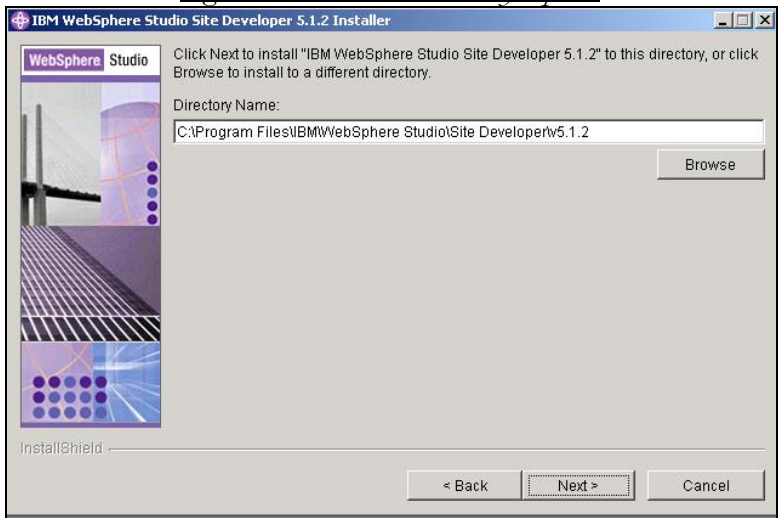
For WebFacing to be installed, select the Full product radial option as shown in Figure 9-6

Figure 9-7 Status Zipper



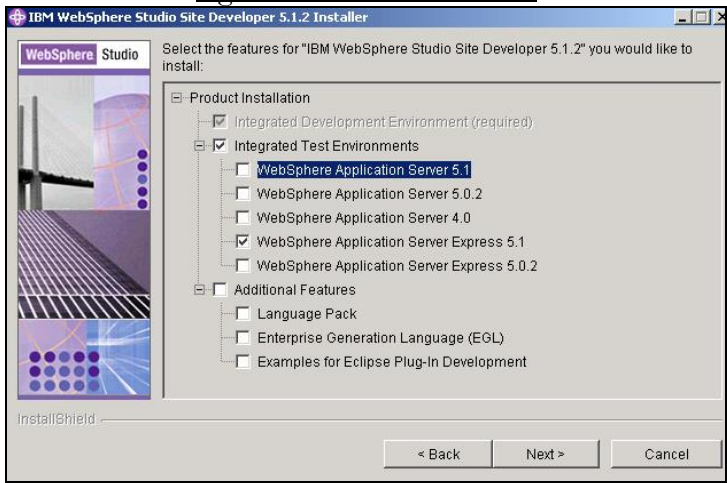
Watch that there is not a message hiding under this panel as it shows that the product is being installed. Grab the blue line and move it if you get no more messages. One may be hiding under the Window shown in Figure 9-7. The next panel you see is shown in Figure 9-8.

Figure 9-8 Install-To Directory Option



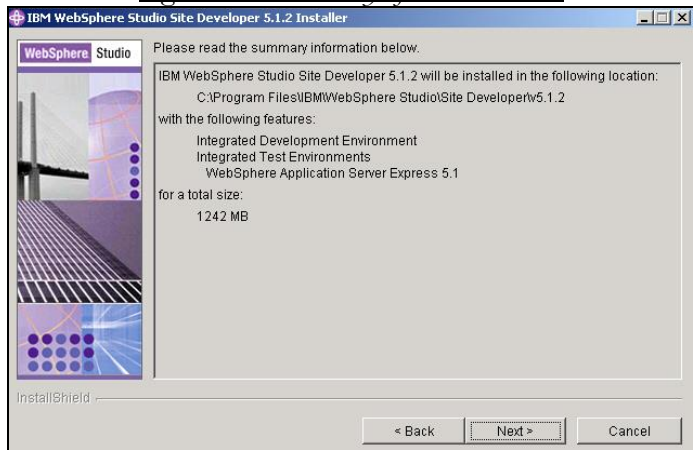
Take defaults as shown in the panel in Figure 9-8.

Figure 9-9 Installation Choices



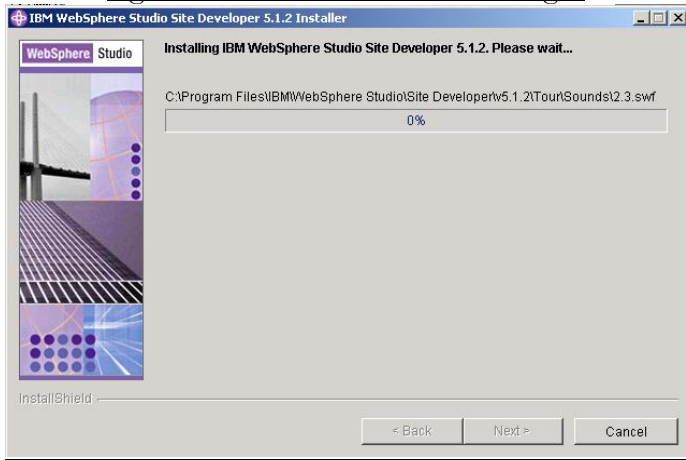
Select the option to install the WAS Express Test Environment at your level (5.1 for this book) as shown in Figure 9-9. Click Next to continue with the panel in Figure 9-10.

Figure 9-10 Summary of Choices Made



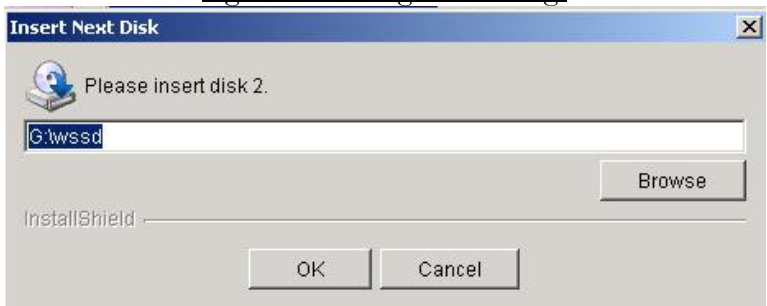
The panel shown in Figure 9-10 gives you information about the installation options you have selected. Click Next to continue.

Figure 9-11 Product Installation Has Begun

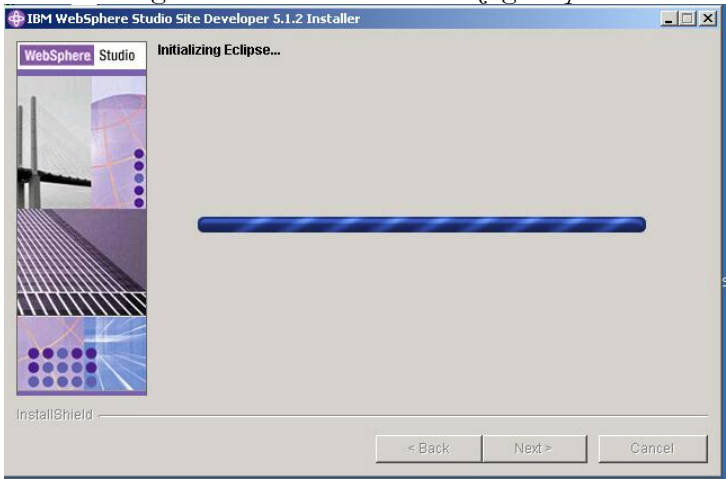


You're off and running with the installation as you can see in the panel shown in Figure 9-11

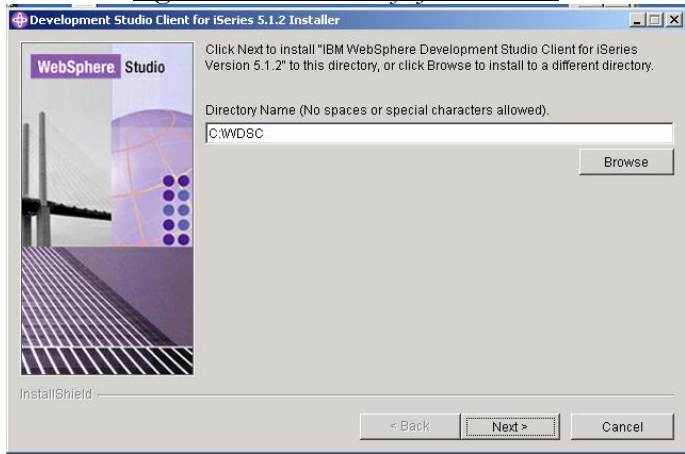
Figure 9-12 Change CD Message



You will see this message in the panel shown in Figure 9-12 four more times until you have inserted the right CD each time. Change CDs and say OK when you get the message as shown in Figure 9-12.

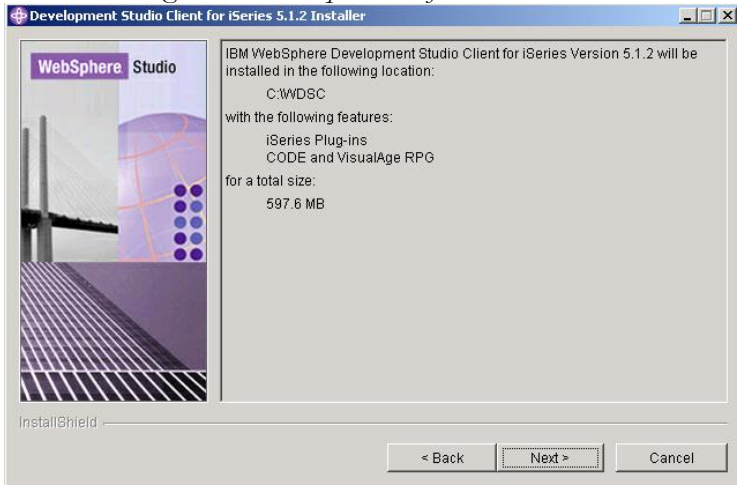
Figure 9-13 New Phase Initializing Eclipse

After several CDs, you'll see the message displayed in Figure 9-13 as it begins to install the WDS component after it has completed the Site Developer portion of the installation. As you can tell by these messages, WDS packaging consists of a number of major components.

Figure 9-14 Second Half of Installation

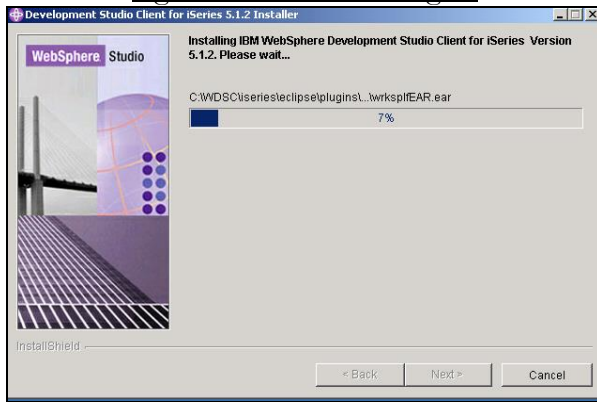
No, you're not starting over at this point as shown on the panel in Figure 9-14. You are, however, moving to a different installation phase. Press Next to continue

Figure 9-15 Completion Information Panel

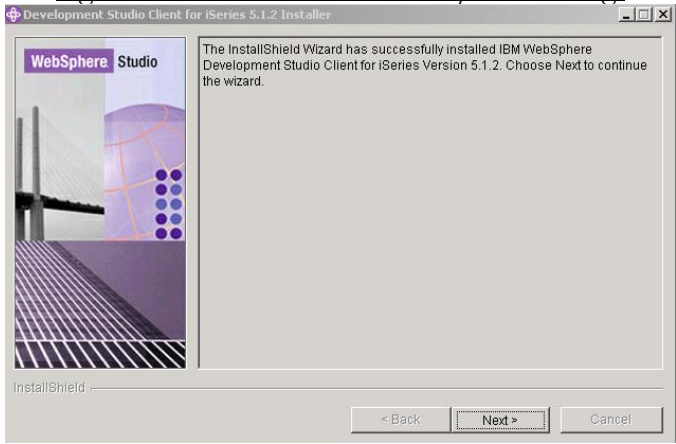


As you can see in the panel shown in Figure 9-15, the Wizard wants you to know how much space it's taking. Click **Next** to continue.

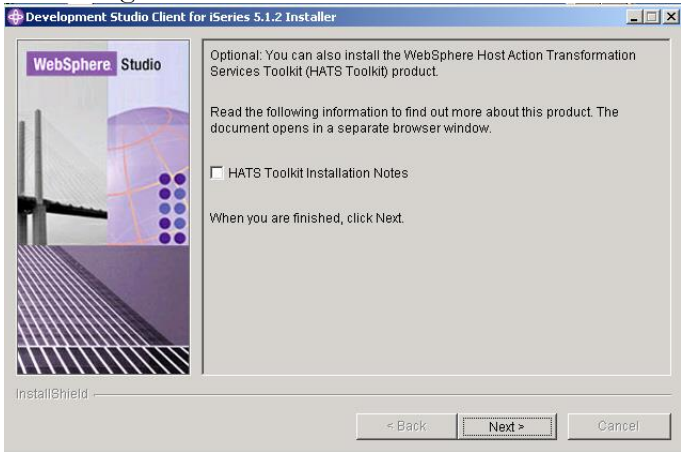
Figure 9-16 Installation Progress



Good news, as shown in the panel in Figure 9-16, the installation is continuing

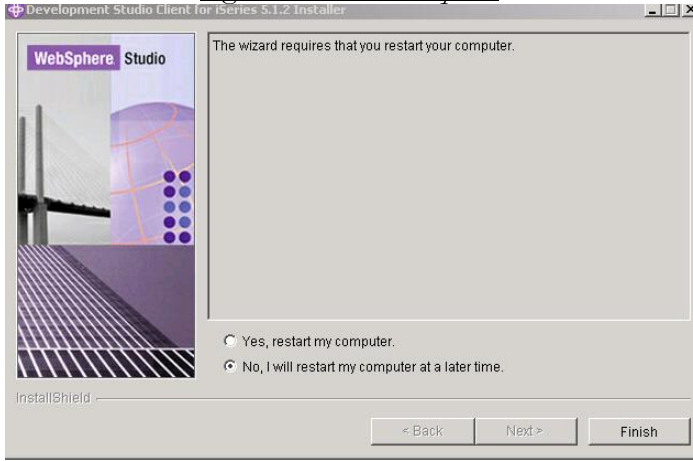
Figure 9-17 WDSc Installation Completion Message

WDSc Eclipse installation is almost over as you can see in the panel in Figure 9-17. Click **Next** to continue. You will be taken to a panel similar to that shown in Figure 9-18.

Figure 9-18 HATS Installation – Not Now

Take the defaults in Figure 9-18 to not install HATS (from WDS) and click the ENTER key. You will then see the panel shown in Figure 9-19

Figure 9-19 Restart Option



We did not take the restart option in the panel as shown in Figure 9-19 immediately. Instead we took the “No” Option and pattered around a bit checking out the CD. Shortly afterwards, however, we restarted the computer. Press Finish to continue from this panel.

4. Reboot the workstation.

When your product is initially installed, it is a good idea to restart your PC so that you can use WDS for PTF acquisition.

Fixing (PTFS)

5. To make sure that prerequisite fixes have been applied to your iSeries system take the following clicks from your newly rebooted PC workstation.

>>Start >> Programs >> IBM WebSphere Studio >> Development Studio Client for iSeries 5.1.2

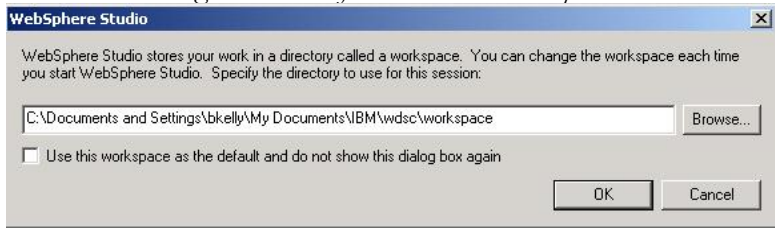
The clicks are shown below in Figure 9-20:

Figure 9-20 Launch WDS for PTFs



You get to answer a question before the package launches.

Figure 9-21 Default WDS WorkSpace



In the panel shown in Figure 9-21, the WDS software automatically selects a workspace path for this WDS session. This can be a path in the C:\ drive root directory or any path that you choose. WDS will build this workspace in whatever path you give as a response to this prompt. My recommendation is that you let it create a directory in your root directory such as the following

C:\workspaces\workspace1

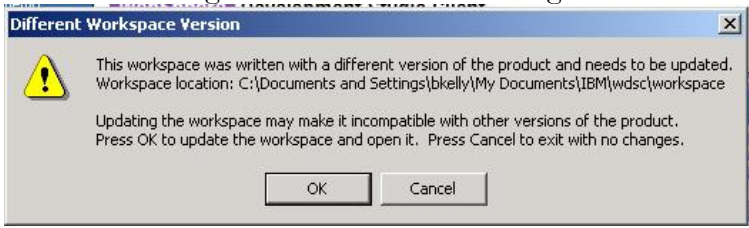
Name the first workspace *workspace1* or something representative and as you create additional workspaces during your application development sessions, you can number them consecutively or give them meaningful names. As you learn more about WDS, you will note that it is sometimes easier to use a separate workspace for each project rather than have multiple projects within the same workspace.

From this panel when you finally click OK, you will see the launch startup panel (Figure 9-22 for WAS Express 5.0, similarly descriptive panel for WAS 5.1 and greater) to help you feel good that you launched the correct program from your PC.

Figure 9-22 WDS Eclipse Launch Panel

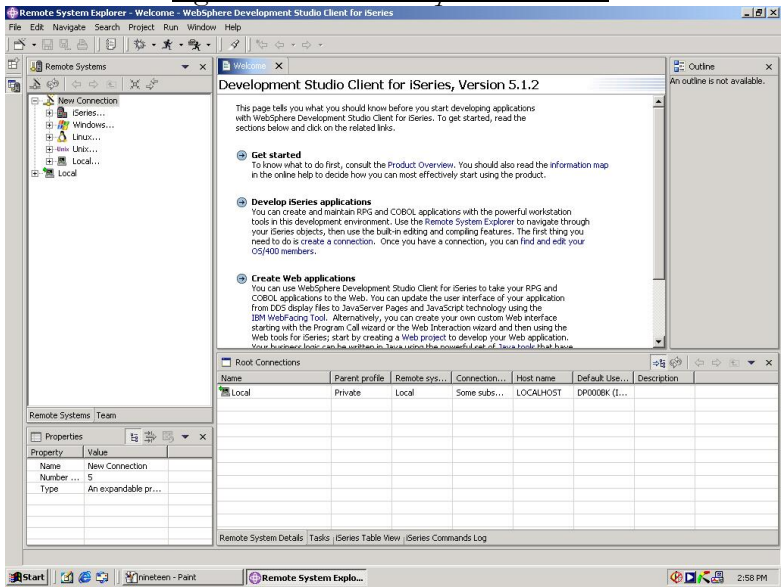


If you had ever used the workspace for a prior version of WDS on this PC, you will get the following message as shown in Figure 9-23.

Figure 9-23 New Version Message

Unless you plan to use that old workspace as a migration source, or you want to keep it in tact for some reason, click OK in Figure 9-23 to reformat the workspace. Otherwise, CANCEL the operation and you will go back to the panel as shown in Figure 9-21.

When all your workstation decisions are out of the way, and you can enter the package in your workspace of choice, you will be taken to the WDS Eclipse WorkBench as shown in Figure 9-24 (WDS 5.1.2).

Figure 9-24 WDS Eclipse WorkBench

Getting Fixes

Since we do not want to use the WorkBench yet, but rather, we want to bring down fixes for the WDS code, let's get that process rolling. To bring down fixes to your workstation, click HELP and then select the Drop down for Software Updates as shown in Figure 9-25.

Figure 9-25 Software Updates (Fixes)



When you let go from the panel in Figure 9-25, your PC goes out to IBM's Web Site looking for WDS code fixes. Unless you want to cancel the operation, let it run until it comes back. While it is searching, you will see a panel such as that shown in Figure 9-26. The process is very intuitive. Some of the panels you will see are shown in Figures 9-26 to 9-28. Fill in the panels to select all the fixes you need and then click on the Next button.

Figure 9-26 Searching For Fixes

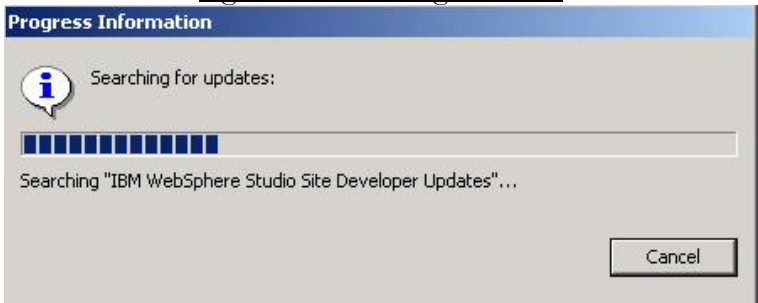


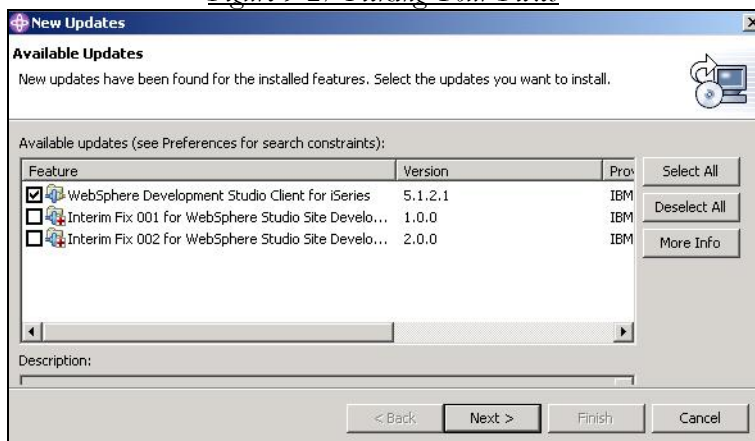
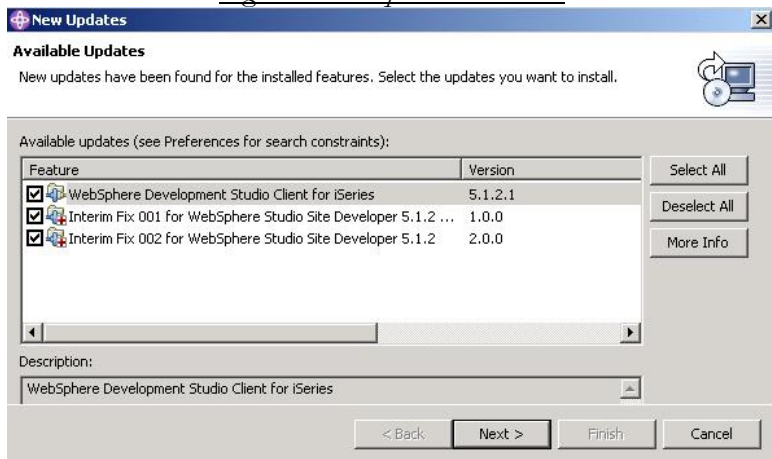
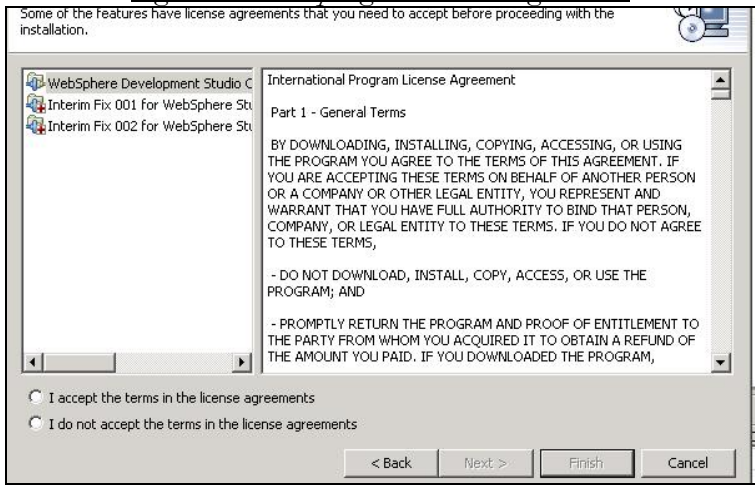
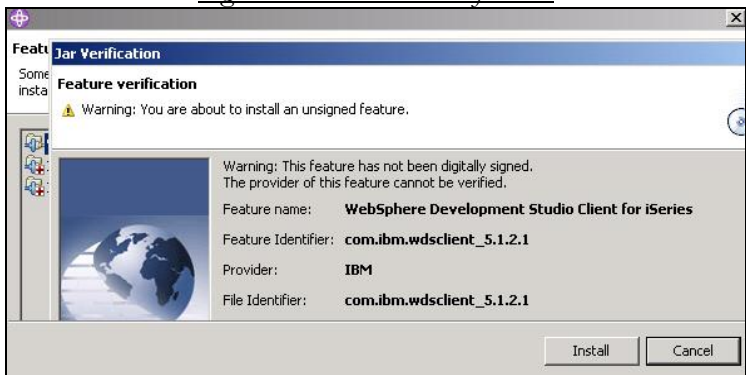
Figure 9-27 Picking Your Fixes*Figure 9-28 Updates as Picked*

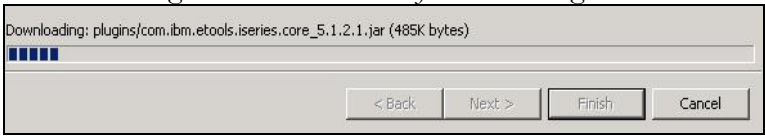
Figure 9-28 Accepting the License Agreements

Select Finish in Figure 9-28 to accept the PTF License Agreement.

Figure 9-29 Feature Verification

To verify that you have the correct feature to install IBM gives you messages for each Fix. Select Install as in Figure 9-29 to put on the fixes that you need.

The next two panels (Figure 9-30 and 9-31) show the Download of the fixes in process.

Figure 9-30 Download of Fixes In Progress*Figure 9-31 Download of Fixes Continuing*

Restart WorkBench After Download

Eventually, the process stops and you are asked to restart the WDS Eclipse WorkBench. You do not have to reboot your PC. Once you do this, it is good to check to make sure that you have the HOST PTFs for WDS that are part of the AS/400 Fix pack.

It would be unusual indeed for you to have all of the prerequisites for WAS Express on your PC and not have the prerequisite host PTFs for 5722-WDS of which WDS must communicate. These HOST PTFs are shipped with the cumulative PTF pack so unless there was a mistake, you should be OK. However, it is good to check and WDS gives you a means to do just that.

WDS Remote Systems Explorer

In Chapter 14, we show how you set up WDS for use with WebFacing and Remote Systems Explorer. You may want to wait until that chapter to perform the HOST PTF test for WebFacing (WDS). If you choose to do it now, you can click on the iSeries

option as shown in Figure 9-24. Pick a new connection and follow the panels of which the one shown in Figure 9-32 shall be one of the first. Once you have a connection to the AS/400 (iSeries), you can then take the Check PTF option to make sure you have all the HOST PTFs.

Figure 9-32 Remote Systems Explorer iSeries Connection



Check Host PTFs

The Create new connection Wizard in the Remote Systems Explorer sets up a connection to your iSeries server. As a part of the setup, there is an option that you will see that is labeled “Check PTFs.” This verifies that the AS/400 server has the required License Program Products and Program Temporary Fixes (PTFs). We recommend that you do this immediately after your WDS product is installed.

However, as noted earlier in this discussion, the HOST PTFs for WDS are provided in the cumulative PTF pack. So as long as your cumulative PTF Fix pack is on your AS/400, WDS should be OK. However, as I repeat throughout the installation of all products, the system is not really ready for application development until you have installed the cumulative pack, all the Group PTFs, and the Client

Access PTFs. Please keep this in mind. The information to do this is given in Chapter 8.

The WDS Sc workstation code itself needs to be checked for fixes also and this is accomplished in the process that we just completed. To repeat, we clicked on HELP, and took the Update links. IBM then checked the PC for the proper PTF level and sent us on a process to update the code as necessary.

If you are not on version 5.1.2. or higher of WDS Sc, you owe it to yourself to get those CDs or DVD from IBM. In our case, they came with V5R3 of OS/400 (i5/OS) so they were quite available. If you don't have them or you are installing on a prior release of OS/400, contact IBM or your business partner to order these CDs.

When I worked on previous versions of WDS Sc, the refresh PTFs for WDS Sc often did not work. I needed special routines and support from IBM to be able to get the fixes. It is much easier to use the V5.1.2 or greater CDs so that you don't require such a detailed level of attention from IBM's wonderful support group – including Tosh and Melinda.

WebFacing PTF

Having said that, the very first time that I ran WDS Sc WebFacing, V5.1.2 with all the requisite PTFs at the time, it failed. Please note that I have been WebFacing for several years and had not failed for some time until V5.1.2. It is probably not V5.1.2's fault but there is a definite problem in trying to exit from a WebFacing application without the PTF that was being built as I was experiencing the problem. The PTF is SI14623. According to IBM, the error situation is described in APAR SE16308 and is corrected by the PTF. You probably should get it and put it on.

Post Install

Once your code is up to date, you can use the integrated server and test environment for the development and testing stages of WebFacing. These are discussed in detail in Part III. To deploy and run your WebFaced e-business applications in a production environment, you will 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. That's why you bought this book. It's all in here.

Eventually, you will deploy your WebFaced applications to the WebSphere Application Server – Express for iSeries. This is the “free” entry level server in the WebSphere Application Server family for iSeries, providing support for static content, servlets, and JSP files.

Most iSeries shops have an RPG compiler and have SEU et al. Therefore, you will also have the 5722-WDS product in which these are contained. In addition to host development tools, the 5722-WDS product also contains the necessary pieces to provide the handshaking with the PC based WDS Eclipse tools.

If, however, you purchased Development Studio Client as a workstation-only product, and you do not have 5722-WDS on your host system, then you need to ensure that WebSphere Development Studio is installed on any iSeries server where you choose to do the host portion of your development work – even if it is not your main system. WebSphere Development Studio (5722-WDS) however, is not required on the iSeries server you use for WebFacing application deployment.

Chapter 10

ISeries Access for Web, HATS/LE Prerequisites, Ordering, Installing, Fixing

Hardware Prerequisite

The biggest hog for processor and disk hardware will definitely be WebSphere Express. If you can fit WAS Express on your system and you have about 400MB of extra disk storage, then you will definitely be able to run iSeries Access for the Web and HATS/LE.

However, if you are planning to really use this facility, it is appropriate that you fully understand the load that it will place on your system. Applications that run through WebSphere do not run as well as native green screen applications, and that may be a gross understatement of the issues you may face regarding performance. My suggestion is to try it and pull back if it hurts. OK, so that's not a scientific approach but it's what we do with everything else in our AS/400 shops and it works. If you promise somebody that you can use this and you have a deadline, then I would suggest that you get IBM's Global Service team involved to help you understand just how much of a load you can put on this guy.

If you want to try it yourself, rather than the trial and error method (heuristic reasoning for those who like expensive names for simple things), IBM has a workload estimator available. For help with sizing all system configurations, IBM recommends using the IBM Workload Estimator for iSeries, located under **Tools** on IBM iSeries Support (<http://www.ibm.com/eserver/iseries/support/>).

Major Hardware Concern

Unlike WebFacing, which has been tuned to operate using client server (batch) cycles, iSeries Access for the Web uses precious interactive cycles. Therefore, if you are using the Standard Edition of the operating system, you simply cannot use this product. If you are using an older box or a model 800 or the Value or Express versions of i5, you can use it to the extent that there is enough interactive CPW (30 or 60 CPW on model i5 Express).

Considering that the poorest batch performing box that IBM has announced in the last several years clocks in at 500 CPW, 30 or 60 CPW for the interactive component of iSeries Access for the Web and/or HATS/LE does not give a lot of playing room. How much does it give? I would say that you have to try it to see. If it doesn't work, you are a WebFacing candidate. Even with WebFacing, however, since it is being driven through WebSphere, it still will not be as slick as 5250 transactions even though there is lots of CPW behind it.

Enterprise Edition

If you purchased an Enterprise Edition machine (iSeries or i5), you will be able to run these two products better than on any other versions. Again, it won't be as good as 5250 terminal emulation via Telnet but it should be reasonable. How reasonable, of course, depends on your transaction volume.

WorkStation Hardware

iSeries Access for the Web is called from a browser to be able to run. Therefore, you must have browsers that are running at compatible levels such as IE 5.5 or better in order to run the iAWeb and/or HATS/LE programs.

Software Prerequisite

If your iSeries or AS/400 machine has the prerequisites to run WebSphere Express, then you have the software necessary to run iSeries Access for the Web and HATS/LE.

iSeries Access for the Web / HATS/LE

Do not try to install an older iSeries Access for the Web if you are just now trying to get this done. WebSphere is far too complex an environment compared to our normal interactive programming model to not give every advantage to yourself. If you have come this far, I would suggest that you not limit your chances for success by installing a less than current iSeries Access for Web version. In fact, I would recommend getting the V5R3 versions of the operating system and the most current version of WAS Express (V5.1.X at this writing). You can run iSeries Access for the Web V5R3 with WAS Express 5.0 or 5.1 and it will work fine but WAS 5.1 Express and i5/OS is your best underlying combination for success. To that, add V5R3 of iSeries Access for the Web.

Ordering iSeries Access for the Web and HATS/LE

V5R3 iSeries Access for the Web (5722-XH2) and HATS/LE are both shipped in the iSeries Access CD package with your new V5R3 release upgrade material. If you are installing to V5R2, to get both components, you must order the full V5R3 iSeries Access refresh package from your business partner. iSeries Access for the Web itself is downloadable from the iSeries Access main Web site. However, as of this writing, HATS/LE is available only on media.

www.as400.ibm.com/clientaccess

The best way to get the software for a V5R2 system is to ask your business partner to order a 5722XW1 (iSeries Access Family) refresh. Software subscription is required. You can tell your business partner or your IBM representative that the refresh feature is **2647**.

This code will also work for i5/OS systems at V5R3 and greater. However, it should not have to be ordered special since it is part of the standard V5R3 upgrade media.

If you do not have a relationship with an IBM Authorized Business Partner and are in the U.S. or Canada, call 1-800-IBM-CALL or go to “**How to Buy**” at:

<http://www-1.ibm.com/servers/eserver/series/buy/marketing/>

Installing iSeries Access for Web i5/OS (V5R3)

The easiest and best type of install as noted above is a V5R3 iSeries Access version on a V5R3 i5/OS version. If you choose to upgrade your iSeries software to V5R3 and you are entitled to the iSeries Access Family (5722-XW1), you should receive what is needed for iSeries Access without having to order anything extra.

iSeries Access for Windows (5722-XE1, V5R3) is shipped on your V5R3 iSeries installation media with the standard set of products for your shop. It is worthy to note that just the 5250 emulation and Data Transfer components of iSeries Access for Windows require you to purchasing a license for iSeries Access Family (5722-XW1). All other functions of the iSeries Access product proper do not require 5722-XW1 (iSeries Access Family) licenses. However, HATS/LE and iSeries Access for the Web do require 5722-XW1 licenses

iSeries Access for Web (5722-XH2, V5R3): is available on your Release Upgrade iSeries media set with the normal keyed set of products or as noted previously, it can be ordered as a special feature. All functions of iSeries Access for Web require you to first have the required 5722-XW1 licenses. So, there is a gotcha if you don't have the right licenses.

WebSphere Host Access Transformation Services Limited Edition (HATS/LE) V5 comes on its own separate CD-ROM, SK3T-8162 in the same package. Again, the use of HATS/LE requires the 5722-XW1 license.

Though we are not focusing on this product in this book, the iSeries Access for Wireless (5722-XP1, V5R3) breaks these rules. It is also available on your iSeries media with the standard set of products. However, this product **does not** require the purchase of iSeries Access Family licenses.

For Linux Client Buffs

If you are dabbling with Linux, you should be aware that iSeries Access for Linux is available as a web download. Refer to the iSeries Access for Linux home page at:

**[http://www-1.ibm.com/servers/
eserver/series/access/linux/](http://www-1.ibm.com/servers/eserver/series/access/linux/)**

Please note that there is no apparent restriction for iSeries Access for the Web clients to be run from Linux desktops. As long as the client has a compliant browser, iSeries Access Web facilities should be available from a Linux or Mac browser after the software is installed on the iSeries.

iSeries Access for Windows Installation(V5RX)

You should not have to install V5R3 iSeries Access for the Web to an i5/OS system. It should already be there from your release upgrade from V5R2. If you have a new system, it should be there from your preloaded software. However, if it is not, you would install to V5R3 the same way you would install the media to V5R2.

To install to V5R2, perform the following tasks

0. Assure that WebSphere Server is already installed
1. Insert the iSeries access media into the CD drive
2. Type **GO LICPGM** and press ENTER from a command line
3. Use option 1 to install all of Client Access

Fixing iSeries Access for Web (PTFs)

If you are at V5R3 and you have judiciously applied the V5R3 iSeries Access PTFs, more than likely, you already have the PTFs that you need for iSeries Access for the Web. The iAWeb PTFs are included with the iSeries Access PTFs. To verify that you have the right level of iSeries Access PTFs, you can check the PTF table at the iSeries Access site shown below. First, find out the PTF level on your iSeries Access product by performing the following

DSPPTF 5722-XE1

Press **ENTER**

You will see a chain of PTFs for Client Access. Write down the top PTF #. This is the number that you will compare against the online information.

For example, suppose the top # is SI13587

To see if you are current with iA/400 PTFs, go to the main iSeries Access page at:

www.as400.ibm.com/clientaccess

From the iSeries Access Home page in the left frame, pick

Service Packs

Review the service pack information.

At the time of this writing as you can see in Figure 10-1, the following service packs were available

V5 R3 SI15176
V5 R2 SI14294
V5 R1 SI11806

Figure 10-1 iSeries Access PTF Levels

Latest Service Packs for Supported Releases					
<ul style="list-style-type: none"> To see service packs for every release, refer to Service Pack History for All Releases. For problems downloading a service pack, click here. For support information on iSeries Access and the Microsoft Windows operating systems, click here. 					
iSeries Access for Windows (Client Access Express) (5722-XE1) Release Level	Latest Service Pack PTF Number	Server Maintenance	Date PTF was Available	Installed File Date	Target for Next Service Pack
V5R3M0	SI15176	Refer to Server Notes	September 3, 2004	09/02/2004	December 10, 2004
V5R2M0	SI14294	Refer to Server Notes	June 23, 2004	08/08/02	November 19, 2004
V5R1M0	SI11806	Refer to Server Notes	April 30, 2004	05/08/01	Not in plan
Problems downloading a Service Pack?					

In this particular example, the V5R3 i5/OS, iSeries Access product, 5722-XE1 at V5R3 is at the most current PTF level – SI13587.

If you are not current and you must apply the PTF to become current, you can the SNDPTFPRD or the Internet option to get the Client Access Group PTF. This includes the iAWeb PTFs From here, you can apply the PTF to the host AS/400.

By far, from my experience, the easiest way to get the PTF is to use the **SNDPTFORD** command to order the PTF (product 5722XE1) and install it on your iSeries. Since the iSeries access service pack PTFs generally exceed the size limit to be sent electronically, change the "Delivery Method" (DELIVERY) parameter on SNDPTFORD to *ANY as shown in Figure 10-2. In this way, you will receive the PTF package on media (probably

CD). As you may already know, the SNDPTFORD command defaults to *LINKONLY. Therefore, if you do not change the parameter to *ANY, the operation will fail.

Figure 10-2

```

Send PTF Order (SNDPTFORD)

Type choices, press Enter.

PTF description:
PTF identifier . . . . . > SI13587      Character value
Product . . . . . *ONLYPRD           F4 for list
Release . . . . . *ONLYRLS           *ONLYRLS, VxRxMx
                                     + for more values
PTF parts . . . . . *ALL              *ALL, *CVRLTR
Remote control point . . . . . *IBMSRV  Name, *IBMSRV, *SELECT
Remote network identifier . . . . *NETATR  Name, *NETATR

Additional Parameters

Delivery method . . . . . *ANY          *ANY, *LINKONLY   ***
Order . . . . . *REQUIRED            *REQUIRED, *PTFID
Reorder . . . . . *NO                 *NO, *YES
Check PTF . . . . . *NO               *NO, *YES

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
Bottom

```

You can also use the special Internet PTF Delivery (iPTF) method. To find out about this service and the requirements, go to

<http://www.ibm.com/eserver/series/support/>

From the left menu, select "**Fixes**".

Note: In V5R3, I discovered that for the first PTF package after moving to the new release, I could not load all of the various PTFS at once (cum pack, group PTFS and iSeries Access) as I had become accustomed. Instead, after I had installed all of the host products, including iSeries Access, I first had to fully apply the cumulative PTFS with an IPL. Without this, I could not install all of the other group PTFS. So, it will take at least two IPLs to get up to date with V5R3 and the Web products outlined. The issue is that some of the PTFS in the WebSphere groups supersede the LIC PTFS in the cumulative pack. Therefore, they cannot be installed until the cumulative pack is fully applied.

Installing HATS/LE

I am going to save the big part of the HATS/LE setup for the next section. HATS/LE is shipped as a WebSphere application, not as a product per se. WebSphere applications get installed in their own special way but there is always the same first step. The application, in the form of what Sun calls an Enterprise Archive File (EAR) is first copied to an IFS directory.

In each WebSphere instance, IBM builds what it calls an *installable applications* directory within the instance structure on the IFS. This is the directory in which IBM would like you to load your EAR files for later WAS installation. I don't like that convention.

Therefore, I create a first level AS/400 IFS directory called /INCOMING and with iSeries Navigator and NetServer, I create a share to that directory. I call the share "INCOMING" because I do not like to forget things. Then, I map a drive to the INCOMING share and for each application that I create using Studio or Java or WebFacing, I copy it (COPY and Paste) to the mapped drive on the INCOMING Share. This positions it to be installed in WebSphere after the requisite WebSphere instances are created for HATS/LE.

Copying HATS5LE.ear file to INCOMING

To install HATS/LE, you will use the file /HA/HATS5LE/HATS5LE.ear file on the installation CD that comes with V5R3 of the iSeries Access refresh. HA is the volume identifier. The first task required to install HATS/LE is to copy this file from the CD to your iSeries server.

To make this easy, do the following:

0. Put the CD in your PC WorkStation

1. Create an /INCOMING directory on the iSeries IFS in the root directory.

MD /INCOMING

2. Use iSeries Navigator to create a share called INCOMING for the directory named /INCOMING .

3. Map a drive, say the “H” drive to the INCOMING share

4. Use the COPY and PASTE Windows facility by right clicking on the HATS5LE.ear file in the HA directory on the CD and pasting it to the H: drive.

5. HATS/LE is now on your iSeries waiting to be installed

6. The installation instructions that will be followed in the Setup section all assume that this ear file is on your iSeries server and that you can locate it using a Browser

Additional HATS/LE Notes

You can either place the HATS5LE.ear file in the InstallableApps directory for your WebSphere Application Server instance, or create a new directory as we did above. Since we have yet to create a WebSphere instance this is not an option for us at this time. If we had created an instance, we could create a share against the installable applications directory in a structure such as:

QIBM/UserData/WebASE/ASE5/<instance>/InstallableApps

In this case, the <instance> is the name of the application server instance on which you are installing HATS/LE. As an alternative, as we did in the instructions above, you could create a temporary directory (INCOMING) that will be easier to browse to, and copy HATS5LE.ear to that INCOMING directory.

Then as noted above, you can map a Windows network drive to your iSeries machine. This facilitates copying files to and from your iSeries machine. You may want to refer to <http://www.ibm.com/servers/eserver/iseries/netserver/> for information on how to start the NetServer and create the shares.

Regardless of how you do it, you must copy the HATS5LE.ear file to your iSeries server and remember where it is. Later in the Setup section IV, we will show you how to take this .ear file (dot EAR file) and make it a WebSphere application.

Section III

Setup and Configuration for IBM's Web Software for iSeries---

Apache HTTP,
WAS Express,
IBM Business Solutions,
WebFacing,
iSeries Access for Web, HATS/LE

Section III is designed to have multiple chapters for some of the highlighted products:

Chapter 11

Apache HTTP Server and WAS Express Setup and Configuration General Steps

Apache, WAS Express Are a Team

Once you have the iSeries WAS Express installed along with the WDS, IBM Business Solutions, iAWeb, and HATS/LE as shown in Section II, as the song goes, “you have only just begun.” For example, it is possible, though highly unlikely that you did not install the free IBM HTTP Server, powered by Apache. It is unlikely especially because in the HTTP installation chapter, we discussed what you needed to do to reinstall this vital item if it is not present in your original distribution.

Unfortunately, if you have no Apache Server on your system at this point, after you install it, you must also apply its PTFs. Re-applying the group PTF for WAS Express after HTTP Apache is installed will also accomplish the same thing.

Not Yet Apache?"

If you have yet to install Apache on your AS/400 or iSeries, then, you need to take a trip to the following URL to find out from IBM how to get this done:

```
http://publib.boulder.ibm.com/series/v5r2/ic2924/index.htm?info/rzaie/rzaiegetstarted.htm
```

Apache Is the Way to Go

Once you have the WAS Express and the Apache HTTP server installed, all you have to do is configure them to work. Of course, if it were only so easy, there would be no need for this book.

Over the years, I had come to favor the simple and intuitive format of the IBM Original HTTP server and its directives (server rules). It was based on the original HTTP server built by Tim Berners Lee, who many regard as the founder of the World Wide Web.

Very Brief HTTP History

The CERN group has not maintained its HTTP Server (<http://www.w3.org/Daemon/>) since 1996. Considering that IBM had just adopted the CERN server for use on the AS/400 in 1995, when CERN shut off all new updates, one might joke that the IBM Company provided a bit of bad luck for the CERN Web Server project.

Tim Berners Lee

As a bit of background, The historical Tim Berners Lee wrote the first World Wide Web server, "*httpd*", and the first client called, "*World Wide Web*." This was a "what-you-see-is-what-you-get" hypertext

browser/editor which ran in Steve Jobs' NeXTStep environment. This work was started in October 1990, and the program "WorldWideWeb" first made available within CERN in December, and on the Internet at large in the summer of 1991. A replacement project called JigSaw was begun in 1996, but the CERN server idea has since been eclipsed since 1996 by the Apache Server.

Rob McCool & Marc Andreesson

Another historical figure, Rob McCool began working with the Web in 1993, at the University of Illinois on the Mosaic Project. McCool later worked with Marc Andreessen, the creator of the Mosaic browser. McCool built the NCSA httpd server (Web server like CERN). Eventually, Andreessen, then twenty-two years old, left the NCSA and formed a company called Mosaic Communications that later evolved into Netscape. McCool's work on NCSA httpd (better performing than CERN) led to Web server functionality that is taken for granted today, including CGI. His work at NCSA formed the basis of the Apache Project.

NCSA Becomes Apache

Over time the NCSA project died and the few remaining developers saw their improvement efforts as "a patchy" version of NCSA httpd 1.3. Eventually the developers formed the open software group known today as the Apache Software Foundation, and their HTTP daemon is now the most widely used Web server in the world. Thus, IBM was wise to switch, rather than fight. However, for me, its directives just are not as clean and simple as CERN. But, I'll survive.

If you can afford to start from scratch with HTTP, rather than having to convert an existing CERN HTTP server on your AS/400 to Apache, the job is actually cake. You will see this as we build our WAS Express server and HTTP server together with the same Wizard.

For example, when you are using the Wizards that creates the WAS Express, it asks if you would like a companion HTTP server built along with the WAS Express server. If you say yes, you can more than likely serve your whole dynamic Web application from this HTTP

server without having to change a single directive. That was our entire experience. Having so little to do is about as easy as it gets.

WebSphere Needs Web Server

You may already know that Apache has its own servlet server, known as TomCat. In fact, for awhile, it looked like it would be IBM's iSeries answer for free servlet serving, before the iSeries Division finally became convinced that WebSphere was the brand they needed to push.

I would bet a dollar to a donut that IBM would have continued with TomCat if the company were able to change its name on iSeries to something with WebSphere in it. There is little wrong with TomCat as a Web Servlet server and the migration from WAS 3.5 (if needed in your shop) is much simpler than moving to WAS Express. It is just speculation on my part that the Apache Foundation would not permit them to rename a TomCat version to "WebSphere Express" just for the iSeries.

Anyway, as you may know, IBM introduced the Express version of WebSphere for iSeries in early 2003. Make no bones about it. It is IBM's strategic vehicle for small and medium sized businesses to move into the e-business sphere. For most iSeries customers who have obtained it, it came as a free upgrade to the free Standard Edition V3.5 WebSphere. For others, it came when they ordered the Enterprise Edition of a new iSeries Server model. For the minority of users, it came as a result of paying a \$2,000 unlimited use fee, or a \$25.00 per user (\$500 minimum) variable fee.

On May 4, 2004, after three years of no solution, IBM relented and chose to distribute WAS Express at no charge with its new i5/OS V5R3.

Do You Need WAS Express?

The terms Web Server and HTTP Server are synonymous. You may ask, "Isn't an HTTP server enough to supply static and dynamic

content on the Web?” The answer to that is “yes,” but only if you choose to build your applications using the outdated Common Gateway Interface (CGI) method. However, if you want to take advantage of the area in which all of the industry development dollars are being spent, and you want to be able to perform the next latest greatest function on the Web, you would do well not to look at CGI at all. Go right to servlet serving.

Before IBM had a standard offering for the AS/400 regarding servlet servers (WebSphere Express), CGI was getting a hard look by those AS/400 shops needing to get something dynamic on the Web. In fact, because I saw the demand for CGI increasing at the time, I wrote a book, titled Getting Started with WebSphere Development Studio Client (WDS_c) in 2002. Along with a lot of other things, the book demonstrates how to take a simple RPG application using a 5250 panel, and transform it into a CGI program.

In this book, however, because IBM finally standardized its Web direction on WebSphere Express I am back to writing about using WebSphere to get your applications to the Web using RPG and servlets.

On its own, a Web Server, a.k.a. an http server or http daemon, such as Apache, cannot process servlets. Servlets are java programs (classes) that run on Web application servers. Servlets do not run on clients. Servlets are part of the thin client notion that suggests a browser is the client and the java code is executed on the server. Web application servers (servlet servers) are bolt-ons to HTTP servers, such as Apache. WAS Express and TomCat are examples of servlet servers. If you want to stay within IBM's strategic Web offerings for the iSeries, then your servlet server choice should be WAS Express or a full WebSphere server product..

Web Server and Servlet Server

Said another way, in order for a Web Server (HTTP Server) to support servlets, it must work together with a servlet server, which is also called a Web Application Server (WAS). IBM's WAS is called the WebSphere Application Server (WAS Express V5.X in this case) and there have been and are lots of versions of the IBM WAS. The

version we concentrate on in this book is the Express V5.1 version, though we often make reference the WAS Express V5.0 version.

Just as you can add function to a browser via a plug-in, you can add function to an HTTP server via big plug-ins or bolt-ons, as I like to call them. Thus, you can logically look at a WAS as a big bolt-on to an HTTP Daemon (Web Server). Without the bolt-on, the Web Server can provide static content (pages) and dynamic content (database and program access) only via the Common Gateway Interface (CGI). However, it cannot provide servlet serving. Since most modern server based applications use servlets, Web Servers therefore must be equipped with a WAS plug-in.

Use Apache

TomCat, IBM's WebSphere, Bea's WebLogic, Caucho Technology's Resin, Sun One and the plethora of servlet servers available today all operate as plug-ins to Apache and other HTTP servers such as the original CERN Web Server. However, the iSeries WAS Express works only with the iSeries based Apache HTTP server. The message is that if you've got a CERN server driving your iSeries Web pages now, and you want to use the same server functions for WAS Express, you've got an HTTP server migration to work through (from CERN to APACHE). Besides in V5R3, CERN is no longer supported on iSeries. My suggestion therefore, would be for you to migrate your CERN server as soon as possible so that an Apache server can continue to serve your static Web Pages and any CGI you might choose to convert. Use a separate APACHE HTTP server instance, built along with WAS Express, to serve the pages for your servlet server.

Build WAS Express and HTTP Server Together

The HTTP relationship may not be obvious at first to the new WAS Express user. Just because you have an Apache HTTP server up and running, and you have the WAS Express server installed and fully PTFed, the one server does not immediately work with the other

server. Having them installed does not mean that either knows about the other. If they don't know about each other, then the plug-in and the plug-ee are basically unplugged. So, long after your HTTP server is serving static pages and CGI pages, in order for it to serve servlets, a relationship must be established between it and the particular servlet server that you choose to use. In this book, our object of study is the WebSphere Express V5.X servlet server.

The reverse notion, however, is not true. When you have installed and configured WebSphere Express and you have configured a servlet server instance, it does not follow that this particular WAS Express server instance can receive servlet requests and send program results or data through an existing HTTP server. Of course you can set it up to point to an HTTP server of your choice but that is optional.

When you create a WAS Express server instance through the Web-based Administrative Graphical User Interface (Admin GUI), which we are about to use in this chapter, you must tell it which HTTP server instance you will be using. Unless you point it to the proper HTTP server instance at the time you create it, it is not so easy to have an already created WAS server talk to a different HTTP server. Thus as noted above, this reverse idea is not true.

However, to say one more time, as noted above, it is very easy to make an existing HTTP server talk to a new WAS server. It's the vice versa that does not work. Even though you can link the two, it is still more clean, when you have made your transition to WAS Express, for you to build your WAS Express servers along with your new HTTP instance. The Wizard then automatically makes the HTTP server instance a companion to the new WAS server instance. During the process, the Wizard tailors the new WAS instance and builds the necessary directives into the new HTTP server instance so the two server instances (WAS and Apache HTTP) can work as partners in the WebSphere experience. .

What Is an Instance?

Right now, if you have any experience with WebSphere or the original HTTP server, the term may be confusing. For our purposes, regarding the WAS Express, let us consider an instance to be a distinct and separately configurable named environment of the WAS. Each

WAS server that you create has one instance associated with it. For our purposes, we call the whole notion an instance.

For those of you comfortable with AS/400 Work Management, the server/instance will appear in your AS/400 as a job under the QASE5 (WAS 5.0 Express) or QASE51 (WAS 5.1 Express) subsystem. For HTTP, let us consider an instance to be a distinct and separately configurable named environment of the APACHE HTTP server.

Let us also consider that for each unique WAS instance (say *default*, *production*, *test*, *internal*, and *playground*) there should be a unique “named instance” of the HTTP server. For clarity, it would help if your HTTP instances had the same names as your WAS instances (say *default*, *production*, *test*, *internal*, and *playground*). It would also help that a particular named instance of the WAS worked with and was paired with a similarly named instance of HTTP. For example, the *internal* instance of the WAS should work with the *internal* instance of HTTP, and so on. However, there is really nothing, other than sanity, that prevents you from using whatever name you choose for either. For clarity and sanity, we do not recommend mixing and matching server names.

Though all WAS versions prior to WAS Express used the term *instance* to describe a unique environment that was unaffected by other instances, IBM notes that technically, the WAS Express ships with just one instance per created server. The rationale behind using multiple instances in prior WAS versions was to prevent an errant program (perhaps in testing) to bring down the whole WebSphere server. So, now, instead, you run multiple servers and each one has one instance. By running production and test in separate server instances, the implementer can still protect against a problem in one instance affecting any other instance.

Build Separate Application Servers

With WAS Express, the term instance does not mean the same thing as it did in other WAS versions. However, the notion of an instance as we defined above is not gone. You can build separate “application servers” that provide a similar degree of reliability and function separation as instances did in prior versions of WebSphere. Since the

servers operate as separate jobs in the QASE5 or QASE51 subsystems, they are separate and distinct from one another.

Therefore, the notion of separate software application servlet servers provides some of the positive reliability characteristics of the other WAS server instances. In my experience, so far, I have yet to see a problem with one server bringing down another server. Now, that I may have confused matters a bit, for education purposes, and since it is a good term to help build understanding, I will continue to use the term instance for the named HTTP server and for the named WAS server. However, when I use this term, I am speaking of a separately named server, and not the full notion of instance as implemented in prior WAS environments.

Link Between Web Server and WebSphere

As you will also note, the link between the browser and the instances (WAS and HTTP) is the HTTP port that is used. The HTTP Port is the bolt and the “WAS” default host port is the “nut.” This enables the WAS and the HTTP server to function as one server, permitting both static and dynamic page capability using Java servlets. In fact, the server pairs created by the Wizard enable even your static pages within your WAS applications to be deployed without having to code any additional directives for Apache.

Management Through HTTP Admin GUI

The Apache HTTP server is clearly the chicken here. The browser based HTTP management tool that I have been using over the last several years to create HTTP instances, has been getting stronger with each new release. It has really peaked in terms of usability with WAS Express. The management of the Express Server is incorporated into the very fiber of the HTTP Administration Graphical User Interface.

This HTTP Administration Graphical User Interface (Admin GUI) is used to create new WAS “instances and new HTTP instances. It is also the vehicle used to manage and administer these “instances.” It is a very intuitive tool and it almost makes WebSphere easy.

When you create a WAS Express server instance, for example, you can tell the Admin GUI to also create a named HTTP server instance to associate with the new WAS. I recommend building both HTTP and WAS “instances” together during the WAS Express application server creation process. Again you should name them both the same for clarity purposes. There will be more of this later.

Hopefully, you are getting the right message: This ain’t your Father’s WebSphere! It is highly functional, slick, reasonably easy to implement, and reasonably easy to manage. It is WebSphere Express, managed by some slick servlets deployed within the Apache HTTP Admin GUI.

HTTP Role / Config Choices

Most HTTP servers default to port 80. When a Web (HTTP) server runs on port 80, a URL coming in from a browser for a particular Web document does need a port number in the URL to make the request. Eighty (80) is the assumed default port for all HTTP servers. However, when you use an HTTP instance that sits on a port other than port 80, such as port 1101, the port number must be included with the URL, following the host and domain name and a colon. It is often followed by a slash and a particular file name that you would like served.

An example is shown immediately below:

```
http://myhost.mydomain.com:1101/new  
wddirectory/filename
```

Automatic Directives

For the WAS to talk to the HTTP server, there are a few HTTP server directives necessary. Again, with the creation wizard, you do not have to add them. They are automatically built for you. You may know that directives are rules that are pre-specified to the HTTP server. The HTTP server uses these rules when it starts to understand what to do. Through directives, you tell the HTTP server its capabilities and limitations for your shop. Among other things, you tell it what documents it can serve, and what documents it cannot serve.

When the hosts are specified for the WAS Express, in order to create a link between HTTP and WAS, the HTTP port must also be specified. Along with some other items, that is how the WAS sever knows that it is the instance that is supposed to deal with servlet requests from a particular Web server.

The HTTP Admin GUI extensions provided with Express make setting up the servers for initial work as easy as pie. Really! You do not have to have any clue about Apache Directives. So, if you are coming from the original CERN server to APACHE, as we have noted previously but bears repeating, the easiest approach is to create a WAS and HTTP server together and observe the linkage directives that are built within the APACHE instance.

Firing Up the HTTP Admin GUI

With all WAS servers, before you set up WebSphere, you must have a functional HTTP Server. When you install your HTTP server (5722-DG1 – typically installed at release update time), there is a special instance known as *ADMIN. You connect your browser to this instance by typing the following into the URL:

`http://myhost.myserver.com:2001`

Funny as it may seem, in order to set up an Apache or a WAS instance, you actually need to use the Apache Web server

administrative GUI from port 2001. Moreover, IBM has already set up an instance of Apache on every AS/400 and iSeries that has the HTTP server installed. To handle the dynamic interactions of the GUI interface, IBM set up an invisible servlet server that is used as the engine behind the Admin GUI.

Starting the *ADMIN HTTP Server on the iSeries

So, then how do you use the Admin GUI Configurator? It is very easy. You use the special *ADMIN instance of the HTTP server that IBM built for you. This is how you configure your own APACHE instances and your WAS Express instances.

By the way, since it is not a good security idea to have the *ADMIN instance available at all times, IBM does not automatically start this server for you when you install HTTP. To get your *ADMIN instance going, you have two options. You can use the iSeries (Operations) Navigator, or you can go to a command line and type in the following:

```
STRTCPSVR SERVER (*HTTP)
HTTPSVR (*ADMIN)
```

You should then assure yourself that the ADMIN HTTP instance is up by looking at a WRKACTJOB display and paging down until you see a few lines in Figure 11-1.

Figure 11-1 ADMIN HTTP Server Alive on Your AS/400

Opt	Subsystem/Job	User	Type	CPU %	Function	Status
—	QHTTSPVR	QSYS	SBS	.0		DEQW
—	ADMIN	QTMHHTTP	BCH	.0	PGM-QZHBHTTP	SIGW
—	ADMIN	QTMHHTTP	BCI	.0	PGM-QZSRLOG	SIGW
—	ADMIN	QTMHHTTP	BCI	.0	PGM-QZSRHTTP	SIGW
—	ADMIN	QTMHHTTP	BCI	.0	PGM-QYUNLANG	TIMW

Your ADMIN Server Is Running

Now that you are sure that the ADMIN Web server instance is up and running at port 2001, you can get on with the process of configuring the WAS Express instance using the tool. Then, or at the same time, you can configure the corresponding HTTP server using the same tool.

Setting up the Apache Web Server for WAS Express

You start a browser session with the Admin HTTP server instance by typing in a URL in your browser that is shaped like either of the following:

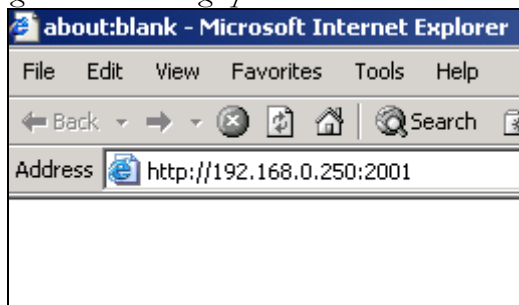
```
http://yourhostname.yourdomainname:2001
```

```
http://yourIPAddress:2001
```

In our case, since the IP address of the HELLO2 server that we are using is a private IP address (not an Internet address) within the “192.168 block,” for simplicity, we will use the IP address to access the Admin server as follows:

```
http://192.168.0.250:2001
```

When typed into the browser, this looks as shown in Figure 11-2.

Figure 11-2 Calling up the HTTP Admin Server

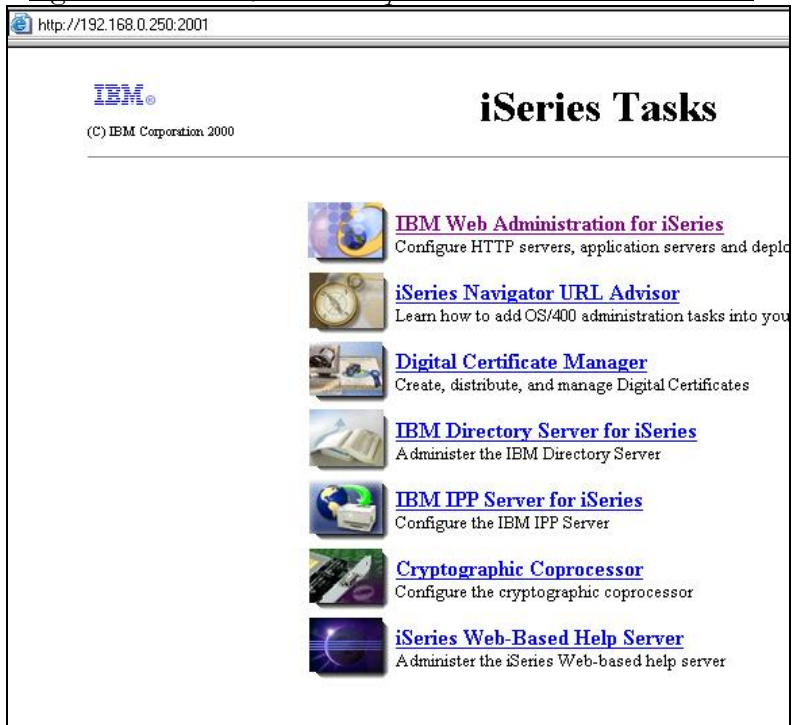
When you press ENTER, you will get the first panel of the HTTP GUI Admin dialog as shown in Figures 11-3 and 11-4.

Figure 11-3 Older HTTP Admin Menu

With the advent of WAS Express, this was your entrée' to HTTP tasks and WAS Express tasks. Regardless of which of the top two options you chose, IBM would graciously take you to the same panel -- the Admin GUI.

Today, in i5/OS V5.3, there is only one option at the top as you can see in the next panel, Figure 11-4..

Figure 11-4 V5R3 / WAS Express HTTP Admin Main Menu



Admin GUI Look and Feel Is a Moving Target

Not only has it changed again, but these changes seem to be continuous. I have written a number of books on the topic and it seems that IBM keeps changing the front door to the Admin GUI to make it different. So if your panel does not look exactly like this for V5R3, or if it is the V5R2 panel in Figure 11-3, do not panic. Work with it. When you pick the first option, you are going to be

taken to the next panel and though it too may not look exactly as the one shown in the Figure, it should be enough to get you going OK.

Clearly, with the Admin GUI, IBM has created some powerful synergy between the Apache Server and the WAS Express. So, to use the Admin GUI, on either release, take the top option of the panel in Figure 11-4, and the Admin GUI Welcome / Create Wizard panel will look very much like that shown in Figure 11-5.

Figure 11-5 Create Wizard Panel



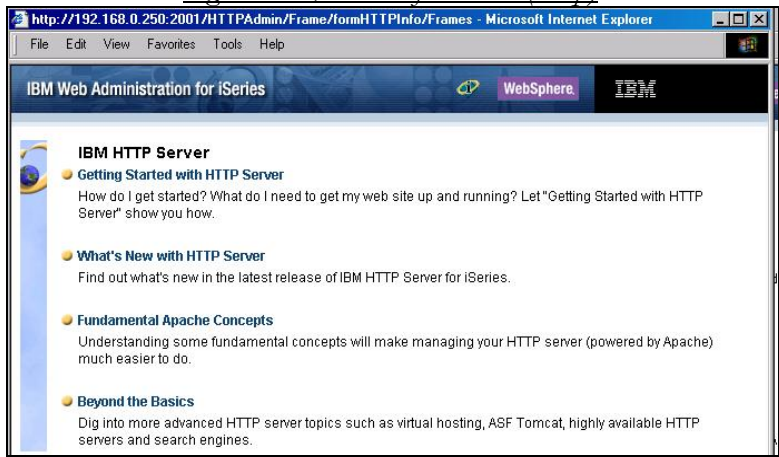
Running the HTTP Admin GUI

Depending on whether it is your first run through or not, you may find a welcome panel to help you. However, you may also find the first panel of the HTTP or WAS Create Wizard as shown above in Figure 11-5

On the right side of the panel, you can see the link for more information. If you click the link for HTTP or WAS Express, you will get Help information in the form of major instructional points for either the HTTP server or the WAS Express server. . This Help Text is first rate.

If you received the Help panel from V5R2, then you can immediately go through the Help text to learn more about the product. However, if you were at the Create panel as in Figure 11-5, to learn more about the HTTP server, you would click the More Information link and be taken to the panel in Figure 11-6. The same approach would be used on the lower link to get Help text for the WAS Express.

Figure 11-6, More Information (Help)



The Admin GUI will quickly become your friend. In the next three chapters, you will use the Admin GUI to create WebSphere Express server instances as well as companion HTTP server instances. This book shows you how to set up the WAS servers and then shows you how to run your applications in those server instances. As you will see in the next Chapter, we are about to create a number of new instances using the Admin GUI. These are as follows:

- ✓ **WASEONE** – WebFacing Applications and the IBM Business Applications
- ✓ **iwa51exp** iSeries Access for Web
- ✓ **HATSLE**—HATS Limited Edition
- ✓ **TELDIR** – IBM Telephone Directory Applications
- ✓ **SURVEYC** – IBM Survey Creator

The steps used for WAS creation are as follows:

Create a New Application Server

In the next several chapters as noted above, we will be creating WAS Express Server Instances. An application server instance provides the runtime environment for your applications. An instance of WebSphere Application Server - Express consists of a single application server, which connects to an HTTP server instance to receive client requests. The application server performs administrative functions and provides services that your application uses to process its client requests.

The general steps to create a new application server with Admin GUI (HTTP Server Administration interface) are as follows:

Please note that these are very general steps and the intention is not so create or start these servers but to note the basic steps necessary to do so. In the next chapter, we will create a real set of servers and fire them up.

To start and test a new application server or to test a new application, follow these steps:

1. To start the WAS Creation Wizard, call up the Admin GUI and sign on. Type the following to start the process:

http://yourdomainor ipaddress:2001

2. When you get to the Man Admin GUI panel take the option for IBM Web Administration for iSeries
3. Take the Setup Tab or the Manage Tab from the Admin GUI Panel in which you arrive.
4. Select the Create Application Server option to invoke the CREATE WAS Express Wizard.
5. Click Next to move past the HELP Panel
6. Name the server and provide a description of the WAS Express Server. Then click the Next button.

Again, our recommendation is that you name it the same as the HTTP server that you plan to use with it.

7. Select the radial button option in the Create WAS Express Wizard to create a new HTTP server (powered by Apache).

Please note that the Original IBM CERN Server is not supported with WAS Express or V5R3. Thus, it is not one of the options when picking an HTTP server to work hand in glove with WAS Express.

8. Run NETSTAT. Go to an AS/400 Command Line and type **NETSTAT**, then take option 3, Work with TCP/IP connection status. Press command key 14 to see the open ports. Look for the Local Port numbers in column 3 of the NETSTAT report and roll down. The ports are in sequence. When you are asked for an HTTP port number in the WAS Express Create Wizard, be prepared to enter a value that is not used (not in the screen report). For the WAS Express ports, make sure that there are thirteen ports available from the starting port number since WAS Express needs thirteen ports. Also, make note of these ports since your Firewall will need to be updated to permit traffic through these ports.

9. Supply a name for the HTTP server that is the same as the WAS Express Server instance. Use all IP addresses and set the HTTP

port to an available value from your NETSTAT work. Click Next to continue.

10 Either use NETSTAT again or while you are in the NETSTAT panels on your AS/400 or iSeries, pick another port number from which the system can allocate thirteen additional consecutive ports (14 in all) for other WAS Express activity. You will need this value when you complete the next Wizard panel. Again avoid conflicts with other IP servers on your iSeries. Provide this port information to your firewall expert if you are expecting this new server to be reached from the Internet.

11. Provide the first Port in the WAS Express range of 13 ports for this instance, and click the Next button..

12. In the next panel, you will be asked which IBM Business application to select. If you plan to run the IBM Telephone Directory Application or the IBM Survey Creator Application, click the appropriate radial buttons. Additionally, choose the IBM Samples so that the instance you create can be verified as being functional. Then click on the Next button to continue.

13. If you have selected any of the IBM Solutions, the CREATE WAS Wizard will turn control over to the LDAP Wizard so that it can collect some information about the Lightweight Directory Access protocol that is needed to run the IBM applications. The first LDAP panel is shown in Figure 11-7.

Figure 11-7 LDAP Page 1 Wizard Auto-Invoked

Business application install - Configuration for LDAP

LDAP server location and port

LDAP server host name: e.g. "hostname.domain.c

LDAP port:

Business applications administrative group and administrator information

Administrative group name: e.g. "Administrators"

Administrator name: e.g. "Administrator"

Password: Important - Keep this for future referenc

14. Change the application Administrator Name and Password and remember them. The first time through, you might want to use bizapps as a user ID and bizapps as a password. From this panel, please note that you are setting the application user id and password, you are not asked on this panel to supply your main LDAP password.

To repeat, so that you can keep your IBM applications separate and managed independently, while you are figuring out how to manage LDAP in your shop, I would recommend changing the Administrator name as shown to the name of the WAS instance or something easy for you to remember (bizapps and bizapps). In Chapter 12, for example, we build a server called WASEONE that has LDAP capability.

Thus, I would suggest that in the panel shown in Figure 11-7, you should write over Administrator on this page with the name bizapps. Again, for now, until you understand LDAP, I would make the password and the password confirmation panel equal to the name bizapps. So, bizapps as an example would appear three times in this panel when you have completed it. Once for the user id; once for the password; and once for the password confirmation.

If you have selected the IBM Telephone Directory application, you will probably want to scroll down further on this panel and select the option for open enrollment so you can test the ITD application without security issues.

Click next when you have the information filled out properly and you will see a panel such as that shown in Figure 11-8.

Figure 11-8 Creating Server WASEONE with Wizard

Create WebSphere Application Server - Express, V5.1
LDAP Authentication

The wizard will connect to your LDAP server and attempt to gather the proper configuration information. The wizard cannot continue without a valid LDAP Administrator user and password for this server.

Specify LDAP user and password: ?

LDAP administrator DN: e.g. cn=admin

LDAP administrator password:

Back **Next** **Cancel**

15. This panel wants you to supply the main LDAP password so that the Wizard knows that it is OK for you to add names to the LDAP directory (bizapps in prior panel). If you do not know your LDAP password (highly likely), you can use iSeries (Operations) Navigator to change the main LDAP password (just once) and remember it.

In Figure 11-8 above, you must use the LDAP Administrator name so I would not touch this panel. It is important to understand that this is not the application user as the first panel (Figure 11-7) represented. This Wizard panel wants the Big LDAP password to prove that you are able to authorize the creation of the user id from the prior page.

Changing the Big LDAP Password

If you do not know your LDAP password, you can go to iSeries Navigator. Navigate to TCP/IP Servers and look for the Directory application Servers, Right click and you can change the LDAP password. Then make sure you remember it for the future.

Again, the first user id and password combos are for the application. The second set as shown above in Figure 11-8 are for the LDAP authorization. I know it's a little hokey and one day IBM may make it better but for today, we live with it.

Type in the LDAP Administrator password and press click Next to continue. You'll see the third PANEL similar to that shown in Figure 11-9 below

Figure 11-9 Default LDAP Parameters – Panel 3

Create WebSphere Application Server - Express, V5.1

LDAP Configuration Parameters

Information describing user entries ?

Parent DN: Browse

Naming attribute: cn

Object class: inetOrgPerson

Information describing the administrative group entry ?

Parent DN: Browse

Naming attribute: cn

Object class: groupOfUniqueNames

Member attribute: uniqueMember

16. Review the Configuration parameters on the panel shown in Figure 11-9, and click Next. On the next panel, click FINISH to create the new WAS Express and HTTP servers.

Starting Your WAS and HTTP Servers

17. Start the WAS Server and the HTTP Server for the instance just created using the Admin GUI

18. Type in the HTTP URL in the following form

```
http://yourdomainname:yourhttpport  
#
```

```
http:192.168.0.250:1600
```

1600 is a sample port in this example. Your port number would be determined by that assigned in the Admin GUI Create WAS Server and Create HTTP server processes.

Press ENTER on your browser and you will get the sample home page for the instance that you have selected (port #).

19. Run Snoop (IBM Test Sample Program) to make sure your servers are working. Use the following URL:

```
http://yourdomainname:yourhttpport  
#/snoop
```

```
http:192.168.0.250:1600/snoop
```

20. Check the snoop report. If it ran, your WAS is OK.

That's all there is to create a WAS instance. In the next two chapters the panels for the WASEONE and iwa51exp WAS and HTTP servers will be shown in greater detail so that you will know cold how to create WAS instances.

Chapter 12

WAS / HTTP Server for IBM Bizapps and WebFacing

The HTTP Server Wizard

None of us in this book need to know anything about setting up an HTTP server for static pages. If you are reading this book to learn how to do HTML and create Home Pages and the like, I regret to say that you would be better off finding another book or Redbook. Don't worry! There are plenty if that is what you want to do. In fact, there are many more than there are for WebSphere Express on iSeries using the Apache Server.

What we are doing in this book is far more complicated and far more sophisticated than creating static pages to be delivered from a static Web server for a Web browser to view.

HTTP Apache Migration Considerations

The Admin GUI that we first fired up in Chapter 11 permits us to create an HTTP server to serve as a launching pad for static pages (like

home pages) using the same Wizard interface that is used to create a WAS Express server. Additionally, the Admin GUI permits us to convert our old CERN Web environment that we may be running and convert it to the Apache format.

If you need to convert HTTP servers, make that part of another project prior to your WebSphere project. This book assumes that type of work is already completed. Feel free to try the conversion Wizards on your own to help make your CERN to APACHE move much easier than it would otherwise be.

For the exercises in this book, there are no original HTTP instances that we need to deal with so we will create our Apache HTTP server instance from scratch while we create the corresponding WAS Express instance. Yes, both will be created with the Admin GUI using the one pass with the Create WAS Instance Wizard.

Starting the Admin GUI

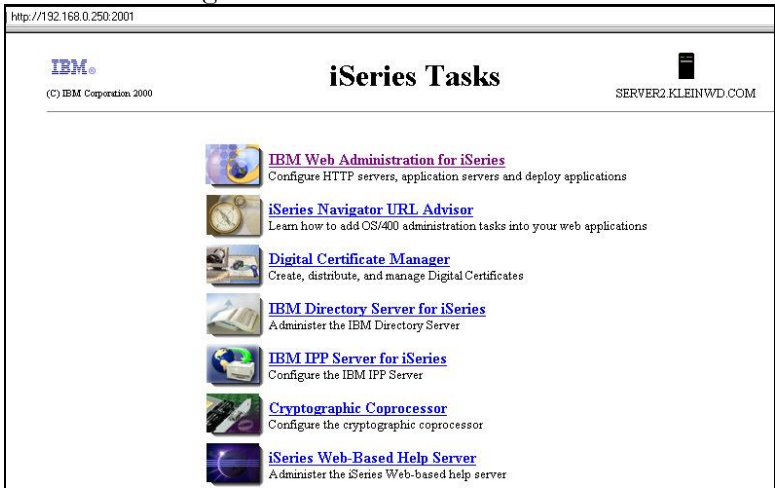
As you may recall, you begin the Web Admin GUI process by opening a browser window and typing in a URL in this form

http://yourdomainoripaddress:2001

The Admin GUI fires up and you get asked for a user ID and password as shown in Figure 12-1

Figure 12-1 Admin GUI Sign-On Panel

Once you sign on, you will be taken to the Main Admin GUI panel as shown in Figure 12-2

Figure 12-2 Main Admin GUI Panel

If you've been here before in another OS/400 release, you will notice that there are different options and names on the panel. However, just like the other looks of the Admin GUI, the top line gets you where you want to go. A close up is shown in Figure 12-3.

Figure 12-3 WAS/HTTP Admin GUI



Click on the top line to move to the functional part of the Admin GUI manager. As soon as you get into the WAS Express / HTTP GUI Manager, take the Setup Tab as shown in Figure 12-4

Figure 12-4 Create Wizard Panel

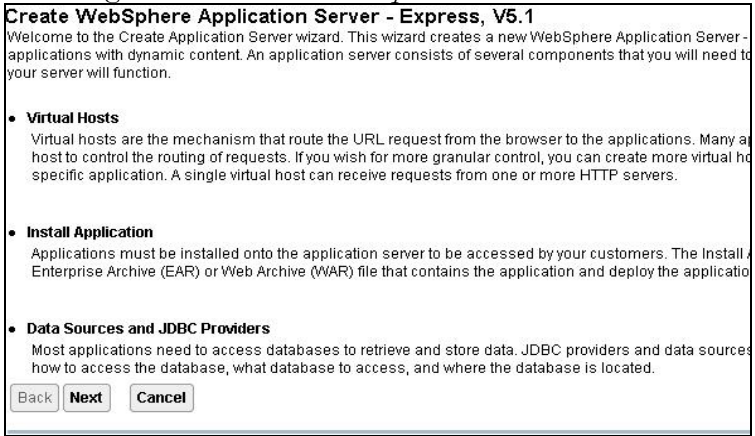


Create New Express Server

When you click *Create Application Server* or *Create a New WebSphere Application Server* from the left or right panes of Figure 12-4, you will have started the Create WAS Instance Wizard.

The first panel is shown in Figure 12-6. Before we move to describe the Wizard’s functional panels, let’s stop for a second and examine the helpful text in Figure 12-5 and talk about what it means to you.

Figure 12-5 Create New Express Server – Intro Panel



The Create Server Wizard

You are now in the *Create New Express Server Wizard*. The first panel of course doesn’t do much other than provide some reinforcement that you have pressed the correct button and provide a little more information about the GUI Admin tool for WAS Express. This wizard is about to create a new WebSphere Application Server - Express “instance” to run your iSeries Access for the Web applications with dynamic servlet content.

An *application server* is not as discrete but is very similar to the notion of server instances as discussed earlier in this text. Just as a WAS server instance works with a separate HTTP server, a WAS Express Application Server is very similar in that it too works with its very own HTTP instance.

The WAS Express server consists of several pieces that you will need to consider and configure before your server will fully

function. These are as noted in the Help Text in Figure 13-2 and are as follows:

Virtual Hosts

Virtual hosts are the mechanism that route the URL request from the browser to the applications. Many application servers use one virtual host to control the routing of all requests. If you wish for more granular control, you can create more virtual hosts to route a particular URL to a specific application. A single virtual host can receive requests from one or more HTTP servers.

As you will soon see, the Wizard creates two virtual hosts for your server. One is for administrative purposes and the other is for you to run your applications. So, unless you want to do something fancy, and the first time through is not the right time for that, you merely need to acknowledge the presence of a virtual host. You do not have to fully understand it or configure it.

Install Application

When you get an application ready for installation on your WAS, it just doesn't happen by loading it to some directories as you may be accustomed in other WAS versions or TomCat. Applications must be formally "installed" within the application server (instance) that provides for their access over the Web. The Install Application Wizard works by taking an Enterprise Archive (EAR) or Web Archive (WAR) file that contains your application and it deploys the application on the WAS Express in the instance within which the Wizard was launched. Obviously for this to be true, there must be an Install wizard in each named application servlet server (instance) and there is.

When you get your HTTP Administration GUI into action, you will note that you add WAS Express applications such as iSeries Access for the Web under a menu item that you access via the following clicks:

>Applications >> Manage Installed Applications

This takes you to the Manage Installed Applications Wizard. From the Wizard, you can install new applications or delete (uninstall) applications. It is worthy of note at this point for those with WAS 3.5 experience, that this is all done without the use of any complicated WAS Console. There is no WAS driven WAS Console needed with Express. There is a browser-based Console but for most functions, this is not needed. Moreover, the browser-based Console for iSeries WAS Express is not needed to install any applications. The HTTP Admin GUI does the job fine. This is a major difference from WAS 3.5. This is a change that makes your life much easier.

Data Sources and JDBC Providers

Most (Java) applications need to access databases to retrieve and store data. JDBC providers and data sources are the vehicle to tell the server how to access the database, what database to access, and where the database is located. Of course if you are using the Java toolkit (JT.JAR), you do not necessarily have to set up these sources and providers. If you have never worked with these, and your purpose is WebFacing or light HLL Web development, you don't have to know about this stuff.

Naming the Server

Now that you know the options that you will be specifying, let's continue the Create Server process. You are now ready to create a WAS Express Server and a corresponding HTTP Server (built to work together). When you take click on NEXT from the panel in Figure 13-5, you will come to the first functional panel of the Create Wizard as shown in the panel in Figure 13-6.

Creating WAS Server for IBM Bizapps and WebFacing Project

For this book, and it is good for you to do this in your own shops, we have created functional WAS servers / instances for the type of work that we are going to perform in WebSphere instance. For example, in the next Chapter, we create a WAS and HTTP instances specifically for IBM's Bizapps. In Chapter 14 we create a pair of servers for WebFacing. In Chapter 15, we create servers for iSeries Access for Web, and our last HTTP and WAS Express Server pair get created in Chapter 16 for HATS/LE

In this Chapter, we are going to create an instance (WAS and HTTP) that can be used for both IBM Business Applications and for WebFacing. Additionally, we will enable the IBM Examples to run so we can prove that the instance works before we try to use it for any of these purposes. In WAS Express V5R1, IBM provides a Survey Creator application and a Telephone Directory Application in its Bizapps product. You will see that these are built at the time the WAS instance is created. You will see in this chapter that there is an additional Wizard dialogue invoked for the Lightweight Directory Access Protocol (LDAP). This is called to provide directory information to enable the IBM applications to run.

Create the WASEONE Server

This of course is the first WAS Express instance that we are creating. The next step in creating the WASEONE server instances is to take the Next option on the panel that we presented way back in Figure 12-5, and this will move you directly to a panel that looks very similar to that shown in Figure 12-6.

Figure 12-6 WASEONE is a WAS 5.1 Express Server

Create WebSphere Application Server - Express, V5.1

Specify Application Server Name

Specify a unique name for the application server.

Application server name:

Server description:

Type in the name WASEONE and give it a description. Then, click NEXT to continue.

Figure 12-7 Create Companion HTTP Server WASEONE

Create WebSphere Application Server - Express, V5.1

Select HTTP Server Type

The application server requires an association with a HTTP server. This HTTP server will server.

Choose the HTTP server type:

Create a new HTTP server (powered by Apache)

Select an existing HTTP server (powered by Apache)

As noted previously, we want to create an HTTP server right along with the WAS Express server so that the two have links to each other and work hand in glove. You merely need to click the radial button as in Figure 12-7 and the Create HTTP Server panel will appear as in Figure 12-8.

Note:

The Original IBM CERN Server is not supported with WAS Express or V5R3. Thus, it is not one of the options when picking an HTTP server to work hand in glove with WAS Express.

Figure 12-8 Creating HTTP Companion for WASEONE

Create WebSphere Application Server - Express, V5.1

Create a new HTTP server (powered by Apache)

A new HTTP server (powered by Apache) will be created and configured to be used by this application.

HTTP server name:

HTTP server description:

Your HTTP server may listen for requests on a specific IP address or on all IP addresses of the system.
On which IP address and TCP port would you like your HTTP server to listen?

IP address:

Port:

Note: Most browsers make requests to port 80 by default.

Type in the HTTP Server name (WASEONE) and a description. Additionally, choose All IP addresses and for this instance, set HTTP to listen on port 1111 as shown in Figure 12-8. Then, click NEXT to continue with the Wizard. You will see a panel similar to that shown in Figure 12-9.

Name the HTTP Server

Though we have already specified WASEONE as the name of the WAS Express “instance,” and we have asked the Wizard to build an HTTP server at the same time, there is no magic to the names being the same. Nothing enforces this. For simplicity purposes, again we recommend that you name your WAS and HTTP instance the same, but, as you can see in the panel in Figure 12-8, you can name them

anything you like. Of course, to demonstrate our recommendation, we again choose the name WASEONE. This time, however, we are naming the HTTP instance, not the WAS instance.

TCP/IP Ports for HTTP

TCP/IP keeps its applications separate by assigning a five-digit port number. Think of a port as an opening by which an application can talk to and from TCP/IP. If you envisioned TCP/IP as a physical entity, it would have about 99,999 different openings by which it could reach the outside. For a TCP/IP application to communicate with a user, the user must establish a connection via one of these numbered openings or ports.

For Web servers using the HTTP protocol, the default port was established long ago as port 80. Through port 80, browsers can naturally talk to Web servers. Moreover, more than one browser can talk through port 80 at the “same” time. So, many different users can be sending and receiving from port 80 to/from a given Web server over the Internet or intranet at the same time.

However, there are many ports in TCP/IP that are not assigned to specific applications. Moreover, a given TCP/IP server system can support many different “instances” of HTTP servers. In order for TCP/IP to keep the “instances” straight, each must communicate on a different port.

IP Addresses

In Figure 12-8, the first piece of information that is requested by the Wizard is the IP address(es) by which this

HTTP server is to listen. The AS/400 and iSeries TCP/IP supports permits multi-homing. Thus, one AS/400 can be assigned a number of IP addresses. This would enable one Web server on the iSeries to cover the incoming traffic from multiple Web Addresses.

The second question asks which of those IP addresses is to be used with the HTTP server. For educational purposes (simplicity), we have just one IP address assigned per AS/400 server. Thus, you can leave this option at its default of:

All IP addresses

The third question on this panel asks which port should be used for HTTP services for this instance. Since port 80 is usually taken by Domino or a main home page application in a “default” HTTP instance, we select a port # that is not used on our system. In the case as shown in Figure 13-5, it is port 1111. Your port # can be anything you like, as long as it is not used by another TCP/IP process on the same IP address.

In most AS/400 shops, on the main AS/400 IP address, port 80 is used for something – even if it is a test Web Server. Since one of our objectives is to avoid confusion, we recommend using a port other than port 80. The WAS Express configuration works with the HTTP configuration and thus, through this Wizard, the link is made at port 1111.

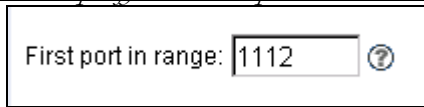
Assigning Ports for Other WAS Functions

To find out what ports are available on what interfaces on your iSeries, from the AS/400 command line, issue the following command:

NETSTAT

Take option 3 from NETSTAT and then hit F14 to see what ports are available. Pick one for the HTTP server. After you have completed the panel in Figure 12-8, click on the NEXT button and you will then see a panel similar to that shown in Figure 12-9.

Figure 12-9 Specify WAS Express Server Port Range



First port in range: ?

All you need do on this panel (Figure 12-9) is put in the port number range for WAS Express. It needs twelve consecutive ports. In our case, we have already used NETSTAT on an AS/400 command line and took the Ports option so we know that the next twelve ports after 1112 are available. If you need to examine your ports, feel free to use the NETSTAT command to help you find your way.

Tip:

To run NETSTAT, you would Go to an AS/400 Command Line and type NETSTAT, then take option 3, Work with

TCP/IP connection status. The press command key 14 to see the open port numbers. Look for the Local Port numbers in column 3 of the NETSTAT report and roll down. The ports are in sequence. When you are asked for an HTTP or other port number in the WAS Express Create Wizard, be prepared to enter a value that is not used (not in the screen report). For the WAS Express ports, make sure that there are twelve ports available from the starting port number since WAS Express needs twelve ports. Also, make note of these ports since your Firewall will need to be updated to permit traffic through these ports.

The port range is usually the last real piece of information that you normally give the Wizard. However, in V5R1 of WAS Express, IBM enhanced the procedure to implement its Telephone Directory Application and it added the Survey Creator Application to the mix. These are bundled under 5722-BZ1 now and are called the IBM Business Applications.

In the next panel, we get to choose the applications that we want and then IBM invokes the LDAP Wizard, necessary for these applications, to assure that we get LDAP configured properly with minimal pain. In prior versions of Express, the Telephone Directory application required a Qshell script to be executed to set up LDAP prior to selecting the application. It's much easier now.

Though you cannot see it in Figure 12-9, at the bottom of the panel is a Next button. Click on this button and you will proceed with the Wizard to the panel as shown in Figure 12-10.

Do You Want Some Free IBM Applications?

IBM ships a sample program called Snoop with the package in the Express Samples application. I would suggest selecting this option for all servers that you create. Big Blue also packs its nicely done Telephone Directory application along with WAS Express. We will study this in Chapter 19. In each server that you build, you decide

whether you want to use IBM's applications during the Wizard creation. As you can see in Figure 12-10, with no further work, we chose the samples.

When you click the NEXT button from Figure 12-9, you come to the panel in Figure 12-10. This is where the Wizard asks you about the IBM Telephone Directory or other applications that you may wish to install while the WAS Express and HTTP servers are being built for you.

Figure 12-10 Create IBM Business Applications

Select which business applications to install:

Note: Installing an IBM business application automatically installs the IBM Welcome Page.

IBM Telephone Directory - An online phone book, providing powerful search capabilities

IBM Survey Creator - An online survey tool used to create, configure, and manage surveys

Select which sample applications to install:

ExpressSamples - A set of samples, including SnoopServlet, that can be used to demonstrate the capabilities of the WebSphere Express server

DB2 Web Services - A set of samples that demonstrate using the Web Services Client API

Back Next Cancel

In this case, you can see that there are four different items that you can select to have installed for you. These are as follows:

- IBM Telephone Directory Application
- IBM Survey Creator
- WAS Express Sample Applications
- DB2 Web Services Sample Applications

The last item, DB2 Web Services Sample Applications, is not covered in this book. However, so that you are not left

with a goosey feeling about Web Services (in case you have yet to be introduced to the notion), we borrow below, a semi-formal definition of a Web service from IBM's tutorial on the topic:

Web Services: Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

Yes, we would need a lot more space to fully describe Web services, and a lot more space to make sense out of the examples. Web services are fairly new and the notion comes with its own set of protocols to enable all of the wonderful things promised in the definition.

Unfortunately, the notion of Web services is far too large for a simple explanation. If you are more interested in this advanced topic, I would suggest that you pull out your favorite search engine and type in “What are Web Services?” You may be amazed at all the good things that have been written about this topic.

Sample Applications

The third application on this panel is the WAS Express Sample Applications. Under WAS V3.5, this group of applications was far more extensive. However, there is good news. The savior application, Snoop as noted above, has been retained from the WAS 3.5 samples pack. This is the application that I have used for the past several years to prove that I have built a functional WAS and corresponding functional HTTP server. The use of the Snoop proves your

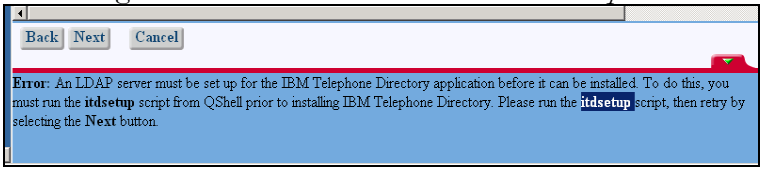
work. It proves that the HTTP server and the WAS server are working together. This is very reassuring, particularly when you are first beginning. You will see Snoop and how it works with WAS Express after you build the WASEONE HTTP and WAS Express servers with the Wizard.

WAS 5.0 LDAP Shows Its Need

The second application on this list is the IBM Survey Creator and the very first application on this panel is the IBM Telephone Directory. As a point of note, both the IBM Telephone Directory and the IBM Survey Creator applications require the Lightweight Directory Access Protocol (LDAP) to be configured and operational on your AS/400. In prior releases, you had to perform this operation separately using a script, and it had to be done prior to getting to this point in the Wizard.

Since LDAP itself requires thinking that is independent of the WAS Express, if you are using WAS Express V5.0, I would suggest that you defer the LDAP installation until your first WAS Express server is fully functional. Then come back to this point.

If you choose to go ahead and install the Telephone Application without LDAP having been installed in WAS Express 5.0, IBM does not treat you very well. The WAS Express Creation Wizard will greet you with an error message similar to that shown in Figure 12-11.

Figure 12-11 LDAP on WAS 5.0 Not Set Up Yet

If you could actually read the message in Figure 12-11, you might think that you could just run the *itdsetup* and have this thing (LDAP) enabled. In WAS Express V5.0, however, it is not quite so easy. You must go through all of the work of setting up LDAP, and then you could run the referenced *itdsetup* script. In WAS Express V5.0, this will complete the prerequisites for the Telephone Directory application. In WAS 5.1 Express, the LDAP script setup step is not needed.

In WAS 5.1, IBM enhanced the Create WAS Express Wizard to include a new Wizard to create LDAP parameters on the fly without having to run an *itdsetup* at all. If you happen to get the above message while installing WAS 5.1, call IBM and get a new set of disks—you've got WAS 5.0.

In this chapter, as we are creating the WASEONE WAS Express Server, as you will soon see, you will enable LDAP. Since this prerequisite to IBM's business applications will be fulfilled by the LDAP Wizard, the Telephone Directory application as well as the IBM Survey Creator application will be created automatically when the WASEONE server is created.

From Figure 12-10, therefore, you would click on the Next button and the LDAP Wizard will commence as shown in Figure 12-11.

Figure 12-11 LDAP Page 1 Wizard Auto-Invoked

Business application install - Configuration for LDAP

LDAP server location and port

LDAP server host name: e.g. "hostname.domain.c

LDAP port:

Business applications administrative group and administrator information

Administrative group name: e.g. "Administrators"

Administrator name: e.g. "Administrator"

Password: Important - Keep this for future referenc

As noted in Chapter 11, change the Administrator name to something like bizapps for these test rounds so you can remember it. Write them down at this point so that you don't forget them (especially your password).

When you fill the password in twice on this panel, click on the Next button so you can continue with the Wizard.


Figure 12-12 LDAP Wizard – Scroll, Re-Enter Password

Business applications administrative group and administrator information

Administrative group name: e.g. "Administrators"

Administrator name: e.g. "Administrator"

Password: Important - Keep this for future reference

Error: A value is required for this field. 

In Figure 12-12, you may notice that there is a message saying confirm password. If this panel were in color, you would notice that it is in red. When you enter the password for LDAP, enter it twice. You

can't see it in Figure 12-11, but if you scroll down there is a spot for the password confirmation. Confirm the password, and click NEXT again. You will be taken to the panel in Figure 12-13.

Figure 12-13 Creating Server WASEONE with Wizard

Create WebSphere Application Server - Express, V5.1
LDAP Authentication

The wizard will connect to your LDAP server and attempt to gather the proper co cannot continue without a valid LDAP Administrator user and password for this

Specify LDAP user and password: ?

LDAP administrator DN: e.g. cn=admin

LDAP administrator password:

Back **Next** **Cancel**

Type in the big LDAP Administrator password and click Next to continue.

Figure 12-14 Default LDAP Parameters – Panel 3

Create WebSphere Application Server - Express, V5.1

LDAP Configuration Parameters

Information describing user entries ?

Parent DN:

Naming attribute: cn

Object class: inetOrgPerson

Information describing the administrative group entry ?

Parent DN:

Naming attribute: cn

Object class: groupOfUniqueNames

Member attribute: uniqueMember

Unless you are changing your host names, the next panel shown in Figure 12-14 above can be accepted as is. Click the Next button and you will get the LDAP wrap-up sheet as shown in Figure 12-15 below.

Figure 12-15 WASEONE Summary Information

Create WebSphere Application Server - Express, V5.1

Application server name: WASEONE

Server description: WAS Express V5.1

Internal port range: 1112 - 1124

Virtual host: default_host

Business applications:

Application name	URL to administer application
IBM Welcome Page	SERVER2:1111/ibm-bizApps/welcome/admin.do
IBM Telephone Directory	
IBM Survey Creator	

Sample applications:

Application name	URL to access application
ExpressSamples	SERVER2:1111/snoop

Click Finish to continue and you will see the panel in Figure 12-16.

Figure 12-16 WASEONE Server Bring Created

In Figure 12-16, you can see the yellow button on the top left pane and the word *Creating* next to it. This is very good news. This means that at the time of the “Creating” status, the WASEONE servers were on their way to be created – just several minutes away. The next panel depicts what you would see on the right side of the panel with the Manage Tab selected. It will look similar to the panel shown in Figure 12-18.

AS/400 Jobs for Create Servers

During the building of the WAS Express Server, you can see a few jobs doing the internal work if you care to take a look at the active jobs (WRKACTJOB) display on your AS/400 or iSeries. You will see a panel similar to that shown in Figure 12-17.

Figure 12-17 WRKACTJOB - CRTWASINST

Work with Active Jobs							SERVER2
						08/28/04	12:51:42
CPU %:	34.5	Elapsed time:	00:00:03	Active jobs:	160		
Type options, press Enter.							
2=Change		3=Hold	4=End	5=Work with	6=Release	7=Display message	
8=Work with spooled files		13=Disconnect		...			
Opt	Subsystem/Job	User	Type	CPU %	Function	Status	
	QJVAEXEC	QTMHHTTP	BCI	.0	PGM-QZSHSH	EVTW	
	QPOZSPWT	QTMHHTTP	BCI	32.2	PGM- CRTWASINST	RUN	
	QINTER	QSYS	SBS	.0		DEQW	
	QPADEV0003	RODS	INT	.0	CMD-STRSQL	DSPW	
	TESTB1	BKELLY	INT	.8	CMD-WRKACTJOB	RUN	
	QSERVER	QSYS	SBS	.0		DEQW	
	QPWFSEVSD	QUSER	BCH	.0		SELW	
	QPWFSEVSO	QUSER	PJ	.0		DEQW	
	QPWFSEVSO	QUSER	PJ	.0		DEQW	
							More...
Parameters or command							
==>							
F3=Exit		F5=Refresh		F7=Find		F10=Restart statistics	
F11=Display elapsed data		F12=Cancel		F23=More options		F24=More keys	

Notice in Figure 12-17 that the program *CRTWASINST* is being used to build the WAS Application Server. The program is running in the QHTTP subsystem. This happens to be the same command that is used in WAS V3.5 and V5.0 to create additional WAS instances. In WAS 3.5, however, you had to invoke this command using the Qshell Unix-like command facility. As you may recall this *CRTWASINST* was launched by the Admin GUI Create Wizard.

When the server is built, and you hit the Refresh button, the status changes to “Stopped,” as shown in Figure 12-19.

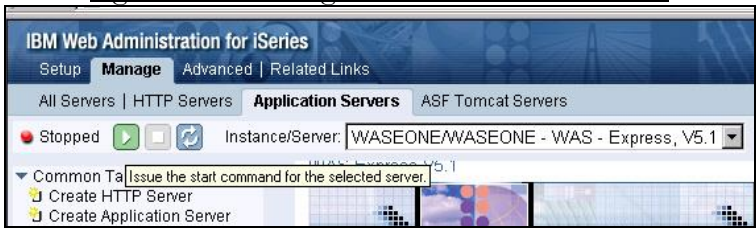
Figure 12-18 IBM Business Apps Created, Stopped State

Current Configuration for Server - WASEONE/WASEONE		
Manage Virtual Hosts	Manage Installed Applications	Manage Data Sources
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> default_host <input checked="" type="checkbox"/> admin_host 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> ExpressSamples <input checked="" type="checkbox"/> IBM Welcome Page v1.1 <input checked="" type="checkbox"/> IBM Telephone Directory v5.2 <input checked="" type="checkbox"/> adminconsole <input checked="" type="checkbox"/> IBM Survey Creator v1.0 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Default Datasource

WAS, HTTP Created, Stopped

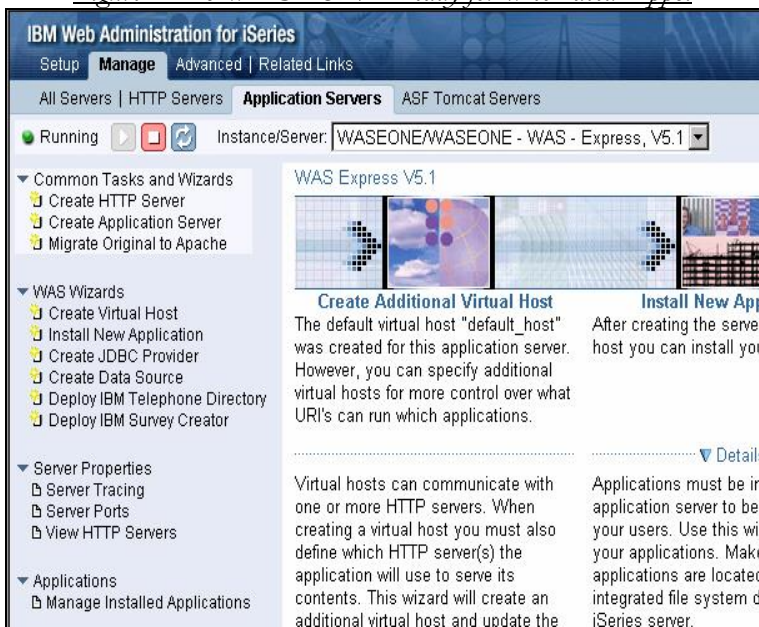
A few minutes after clicking the “Create” button, the WAS Express and HTTP server instances would be created. Another while longer and the applications would be created as you can see in the center of the panel shown in Figure 12-18. The red button to the left of the applications means that they are in a stopped condition and are basically not open for business.

Figure 12-19 Clicking Start Server / Instance Button



Starting WASEONE Server

To get both the new WASEONE WAS Express Server instance and the applications started so that they can be used, the next step would be to click the green and white arrow button at the top of the page as shown in the left pane of Figure 12-19. Several minutes later the WAS Express instance WASEONE and the applications would be ready to go.

Figure 12-20 WASEONE Ready for WebFaced Apps.

As you can see in Figure 12-20, the WASEONE WAS Express server instance is running. Look across to the right pane and you would see if the applications are also ready. Figure 12-21 shows what you would see when the start process is completed.

Figure 12-21 IBM Applications Ready for Use

As you can see, almost all of the applications are all started. The only one not started is the adminconsole and that is because we have yet to enable it. The “Console” is needed so infrequently with WAS Express that to simplify this book, we have not included any information about it. For example I needed it for nothing that we do in this work.

At this point therefore, we are just about done. The WASEONE WAS Express server instance is started as are the IBM Business Applications that we selected, including the Express Samples, IBM Telephone Directory, and the IBM Survey Creator Application. There’s only one more thing at this point that would have to be done for the WASEONE server to be able to accept requests for Snoop or the other applications.

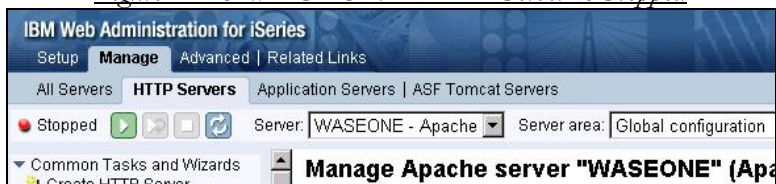
Can you guess what it is? Well, we have the WASEONE WAS Express server and the HTTP server and the applications all created, and almost all of them are started. Besides the adminconsole, an item not started that we need is the WASEONE HTTP server. It does not start when you start the WASEONE Express server instance.

For Was Express to run this HTTP instance must be started. So, to get this guy started, you must first go to the top left pane of the Admin GUI as shown again in Figure 12-22.

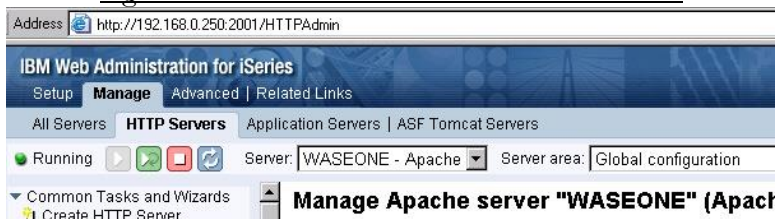
Figure 12-22 Switch to HTTP Tab



Note that the Application Server Sub Tab of the Manage tab is selected and as you can see the WASEONE server is running. You must change this to the HTTP Servers Sub Tab so that you can work with the new WASEONE HTTP server. So, when you click on HTTP Servers, you will see a panel very similar to that shown in Figure 12-23.

Figure 12-23 WASEONE HTTP Server Is Stopped

As you can see in Figure 12-23, the HTTP server named WASEONE is stopped. The WAS Express Server named WASEONE cannot receive input from a browser until its companion HTTP server instance WASEONE is started. To get this HTTP server started, click on the green and white right pointing arrow shown in the left pane of Figure 12-23.

Figure 12-24 WASEONE HTTP Server is Started

Now, as shown in the panel in Figure 12-24, the HTTP WASEONE HTTP server has been started. The WAS Express instance WASEONE is now ready for business. In Section IV, we will give this instance some business. But, first, we have a few more instances to create.

Chapter 13

Setting Up HTTP and WAS Express Server-- IBM Business Solutions

Create SURVEYC instance for the IBM Survey Creator

The steps to create a new application server SURVEYC with Admin GUI (HTTP Server Administration interface) were shown in general form in Chapter 11. The very specific steps to create the all-everything WASEONE server were shown in Chapter 12. This time the objective is to create a specific server for the IBM Survey Application. However, the steps to do this will not be shown in the detail that they were in Chapter 12. The difference between WASEONE and SURVEYC instances is that SURVEYC will not enable the IBM Telephone Directory application and it will not be used (though it could) for WebFacing applications. The steps to create this WAS Express instance and companion HTTP server are as follows:

To start and test a new application server or to test a new application, follow these steps:

1. To start the WAS Creation Wizard, call up the Admin GUI and sign on.

http://yourdomainoripaddress:2001

2. When you get to the Main Admin GUI panel take the option for IBM Web Administration for iSeries

3. Take the Setup Tab or the Manage Tab from the Admin GUI Panel in which you arrive.

4. Select the Create Application Server option to invoke the CREATE WAS Express Wizard.

5. Click Next to move past the HELP Panel

6. Name the server SURVEYC and provide a description of the WAS Express Server. Then click the Next button.

Again, our recommendation is that you name it the same as the HTTP server that you plan to use with it.

7. Select the radial button option in the Create WAS Express Wizard to create a new HTTP server (powered by Apache).

8. Run NETSTAT. Go to an AS/400 Command Line and type **NETSTAT**, then take option 3, Work with TCP/IP connection status. Press command key 14 to see the open ports. Look for the Local Port numbers in column 3 of the NETSTAT report and roll down. The ports are in sequence. When you are asked for an HTTP port number in the WAS Express Create Wizard for SURVEYC, be prepared to enter a value that is not used (not in the screen report). For the WAS Express ports, make sure that there are thirteen ports available from the starting port number since WAS Express needs thirteen ports. Also, make note of these ports since your Firewall will need to be updated to permit traffic through these ports.

9. Supply a name for the HTTP server (SURVEYC) that is the same as the WAS Express Server instance. Use all IP addresses and set the HTTP port to 1600 or an available port if 1600 is taken on your system. Click Next to continue.

10 Either use NETSTAT again or while you are in the NETSTAT panels on your AS/400 or iSeries in Step 8, pick another port number from which the system can allocate thirteen additional consecutive ports (14 in all) for other WAS Express activity. You will need this value when you complete the next Wizard panel. Again avoid conflicts with other IP servers on your iSeries. Provide this port information to your firewall expert if you are expecting this new server to be reached from the Internet.

11. Provide the first Port in the WAS Express range of 12 ports for this instance, (port # 1601) and click the Next button..

12. In the next panel, you will be asked which IBM Business application to select. Pick IBM Survey Creator. Then click on the Next button to continue.

13. If you have selected any of the IBM Solutions, as we have with Survey Creator in Step 12, the CREATE WAS Wizard will turn control over to the LDAP Wizard so that it can collect some information about the Lightweight Directory Access protocol that is needed to run the IBM applications. The first LDAP panel is shown in Figure 13-1.

Figure 13-1 LDAP Page 1 Wizard Auto-Invoked

Business application install - Configuration for LDAP

LDAP server location and port

LDAP server host name: e.g. "hostname.domain.c

LDAP port:

Business applications administrative group and administrator information

Administrative group name: e.g. "Administrators"

Administrator name: e.g. "Administrator"

Password: Important - Keep this for future referenc

14. In prior examples, we changed the Administrator name to bizapps. This time, let's change the application Administrator Name to SURVEYC and Password to SURVEYC and again let's write them down (remember them). Click Next when you have the information filled out properly and you will see Figure 13-2.

Figure 13-2 Creating Server HATSLE with Wizard

Create WebSphere Application Server - Express, V5.1

LDAP Authentication

The wizard will connect to your LDAP server and attempt to gather the proper co
cannot continue without a valid LDAP Administrator user and password for this

Specify LDAP user and password: ?

LDAP administrator DN: e.g. cn=admin

LDAP administrator password:

15. Plug in the big LDAP password.

In Figure 13-2 above, leave the Administrator name alone and plug in your LDAP password. If you do not know your LDAP password, you can go to iSeries Navigator, TCP/IP Servers and look for the Directory application Servers, Right click and you can change the LDAP password. Then make sure you remember it for the future. The first user id and password combos are for the application. The second set as shown above in Figure 13-2 are for the LDAP authorization.

Click Next to continue. You'll see the third PANEL similar to that shown in Figure 13-3 below

Figure 13-3 Default LDAP Parameters – Panel 3

Create WebSphere Application Server - Express, V5.1
 LDAP Configuration Parameters

Information describing user entries ?

Parent DN:

Naming attribute: cn

Object class: inetOrgPerson

Information describing the administrative group entry ?

Parent DN:

Naming attribute: cn

Object class: groupOfUniqueNames

Member attribute: uniqueMember

16. Review the Configuration parameters on the panel in Figure 13-3 and click the Next Button. Then, on the next panel, click FINISH to create the new WAS Express and HTTP servers.

17. Start the WAS Server and the HTTP Server for the instance just created

18. Type in the HTTP URL in the following form

```
http://yourdomainname:yourhttpport  
#
```

```
http:192.168.0.250:1600
```

Press ENTER on your browser and you will get the sample home page for the instance that you have selected (port #).

19. Run Snoop with the following URL

```
http://yourdomainname:yourhttpport  
#/snoop
```

```
http:192.168.0.250:1600/snoop
```

20. Check the snoop report. If it ran, your WAS for SURVEYC was built correctly and it is OK.

That's all there is to create the SURVEYC WAS and HTTP instances.

Create TELDIR Instances for the IBM Telephone Directory

The steps to create a new application server TELDIR with Admin GUI (HTTP Server Administration interface) are exactly the same as the SURVEY C except we will pick the IBM Telephone Directory instead of the IBM Survey Creator. Start the process as follows:

To start and test a new application server or to test a new application, follow these steps:

1. To start the WAS Creation Wizard, call up the Admin GUI and sign on at

http://yourdomainor ipaddress:2001

2. When you get to the Man Admin GUI panel take the option for IBM Web Administration for iSeries

3. Take the Setup Tab or the Manage Tab from the Admin GUI Panel in which you arrive.

4. Select the Create Application Server option to invoke the CREATE WAS Express Wizard.

5. Click Next to move past the HELP Panel

6. Name the server TELDIR and provide a description of the WAS Express Server. Then click the Next button.

Again, our recommendation is that you name it the same as the HTTP server that you plan to use with it.

7. Select the radial button option in the Create WAS Express Wizard to create a new HTTP server (powered by Apache).

8. Run NETSTAT. Go to an AS/400 Command Line and type **NETSTAT**, then take option 3, Work with TCP/IP connection status. Press command key 14 to see the open ports. Look for the Local Port numbers in column 3 of the NETSTAT report and roll down. The ports are in sequence. When you are asked for an HTTP port number in the WAS Express Create Wizard, be prepared to enter a value that is not used (not in the screen report). For the WAS Express ports, make sure that there are twelve ports available from the starting port number since WAS Express needs

twelve ports. Also, make note of these ports since your Firewall will need to be updated to permit traffic through these ports.

9. Supply a name for the HTTP server (TELDIR) that is the same as the WAS Express Server instance. Use all IP addresses and set the HTTP port to 1500. Click Next to continue.

10 Either use NETSTAT again or while you are in the NETSTAT panels on your AS/400 or iSeries, pick another port number from which the system can allocate thirteen additional consecutive ports (14 in all) for other WAS Express activity. You will need this value when you complete the next Wizard panel. Again avoid conflicts with other IP servers on your iSeries. Provide this port information to your firewall expert if you are expecting this new server to be reached from the Internet.

11. Provide the first Port in the WAS Express range of 12 ports for this WAS instance, (port # 1501) and click the Next button..

12. In the next panel, you will be asked which IBM Business application to select. Pick the IBM Telephone Directory Application. Then click on the Next button to continue.

13. If you have selected any of the IBM Solutions, such as the Telephone Directory, the Create WAS Wizard will turn control over to the LDAP Wizard so that it can collect some information about the Lightweight Directory Access protocol that is needed to run the IBM applications. The first LDAP panel is shown in Figure 13-1.

Figure 13-1 LDAP Page 1 Wizard Auto-Invoked

Business application install - Configuration for LDAP

LDAP server location and port

LDAP server host name: e.g. "hostname.domain.c

LDAP port:

Business applications administrative group and administrator information

Administrative group name: e.g. "Administrators"

Administrator name: e.g. "Administrator"

Password: Important - Keep this for future referenc

Back **Next** **Cancel**

14. Change the application Administrator Name to TELDIR and Password to TELDIR and remember them. Click Next when you have the information filled out properly and you will see a panel similar to that shown in Figure 13-2.

Figure 13-2 Creating Server TELDIR with Wizard

Create WebSphere Application Server - Express, V5.1

LDAP Authentication

The wizard will connect to your LDAP server and attempt to gather the proper co
cannot continue without a valid LDAP Administrator user and password for this

Specify LDAP user and password: ?

LDAP administrator DN: e.g. cn=admin

LDAP administrator password:

Back **Next** **Cancel**

15. Plug in the big LDAP Password.

In Figure 13-2 above, leave the Administrator name alone and plug in your LDAP password. If you do not know your LDAP password, you can go to iSeries Navigator, TCP/IP Servers and look for the Directory application Servers, Right click and you can change the LDAP password. Then make sure you remember it for the future. The first user id and password combos are for the application (Telephone Directory). The second set as shown above in Figure 13-2 are for the LDAP authorization.

Type in the LDAP Administrator password and click Next to continue. You'll see the third PANEL similar to that shown in Figure 13-3 below

Figure 13-3 Default LDAP Parameters – Panel 3

Create WebSphere Application Server - Express, V5.1

LDAP Configuration Parameters

Information describing user entries ?

Parent DN:

Naming attribute: cn

Object class: inetOrgPerson

Information describing the administrative group entry ?

Parent DN:

Naming attribute: cn

Object class: groupOfUniqueNames

Member attribute: uniqueMember

16. Click Next on the panel shown in Figure 13-3. Then, review the configuration parameters on the next panel and when you are satisfied, click the Finish button to create the new WAS Express and HTTP servers for the IBM Telephone Directory application.

17. Start the WAS Server and the HTTP Server for the instance just created

18. Type in the HTTP URL in the following form

```
http://yourdomainname:yourhttpport  
#
```

```
http:192.168.0.250:1500
```

Press ENTER on your browser and you will get the sample home page for the instance that you have selected (port #).

19. Run Snoop with the following URL

```
http://yourdomainname:yourhttpport  
#/snoop
```

```
http:192.168.0.250:1500/snoop
```

20. Check the snoop report. If it ran, your WAS was built correctly and it is functioning OK.

That's all there is to create the TELDIR WAS and HTTP instances.

Chapter Summary

In this chapter, we walked through the general directions to create two new WAS and two new HTTP instances to support the IBM Survey Creator and the IBM Telephone Directory applications respectively. If you have any issues following these directions, feel free to go back

to Chapter 12 where the directions are shown in greater detail. In this case, you would substitute the name of the instances you are creating for the name WASEONE, and pick the proper IBM application.

Your journey should be successful. But, we're not finished creating WAS instances and HTTP instances just yet.

Chapter 14

Setting up for WebFacing

Preparing to Use V5.1.2 WDS Eclipse for WebFacing

In this chapter we will explore the Eclipse Workbench supplied via the WDS packaging and when we have learned enough of that to handle WebFacing, we will move our attention in that direction in Section IV.

The AD Tool for the Millennium

WDS Eclipse is IBM's AD tool for the new millennium. Make no mistake about that. Whether it fits or not, IBM will do its best to place any new AD capabilities in this new workstation based tool set.

You still may not care to use it for maintaining your green screen code because SEU and SDA work well enough, and you may not choose to use it for VisualAge RPG because client server is dead. However, chances are you will select WDS Eclipse for its WebFacing facilities if you hope to dress up your code to create a new image on the Web. You won't find a green screen version of WebFacing like the one we are about to show you.

IBM is very serious about having you switch to WDS to use the new Remote Systems Explorer (RSE) facility as your new PDM. Within the new RSE, IBM has integrated the old CODE Editor so that when you pick an AS/400 member for editing, you get what was once CODE to

perform the function. With RSE taking over for both PDM and the old “PDM-like” CODE Organizer, there is not much need for CODE, per se in WDS. However, IBM is not interested in annoying those who have adopted CODE in their shops so the company also packs the latest version of CODE 400 within WDS as a separate item – in addition to its use as the integrated editor for RSE.

You can get a sense of IBM’s intentions by doing the infamous STRPDM command from a command line. After all this time, in V5R3, IBM has chosen to provide a commercial for the WDS / Eclipse tooling that the company has provided “free of charge” with your 5722-WDS AD package. Once you issue the command, you’ll see the commercial as shown below in Figures 15-1 and 15-2

Figure 14-1 The New PDM Startup Advertising Message

```

AS/400 Programming Development Manager (PDM)

Select one of the following:

    1. Work with libraries
    2. Work with objects
    3. Work with members

    9. Work with user-defined options

Information about new tools - press F1 for details

Selection or command
====>

-----
F3=Exit      F4=Prompt    F9=Retrieve   F10=Command entry
F12=Cancel   F18=Change defaults

```

The operative advertising words in this panel are as follows:

**Information about new tools -
press F1 for details**

Figure 14-2 Work With Members Advertising Message

```

Work with Members Using PDM                                     S105PV4M

File . . . . . QDDSSRC
Library . . . . HELLO                               Position to . . . . .

Type options, press Enter.
2=Edit          3=Copy  4=Delete 5=Display    6=Print    7=Rename
8=Display description 9=Save 13=Change text 14=Compile 15=Create
module...

Opt Member      Type      Text
JOBINFO        PF        LANGUAGE File For Hello World
LANGUAGE       PF        LANGUAGE File For Hello World
PANEL         DSPF      Display File Panel For Advanced Hello World
PANEL1        DSPF      Display File Panel For Advanced Hello World
RCVDATA       PF        Receive Data From CGI
VIEWDATA      PF        Receive Data From CGI

Parameters or command                                         Bottom
===>
F3=Exit        F4=Prompt      F5=Refresh     F6=Create
F9=Retrieve    F10=Command entry F23=More options F24=More keys
Have you tried the modern alternative to PDM? Press F1 for more details.

```

The operative advertising lure in this message is contained on the last line of the panel shown in Figure 14-2, and is as follows:

**Have you tried the modern
alternative to PDM? Press F1 for
more details.**

Then, when you press the infamous F1 at the bottom of the page, the context sensitive help shows the following message:

Figure 14-3 The Actual Advertising Message

```

Additional Message Information
Message ID . . . . . : PDM0760      Severity . . . . . : 00
Message type . . . . . : Diagnostic
Date sent . . . . . : 08/17/04      Time sent . . . . . : 10:26:09

Message . . . . . : Have you tried the modern alternative to PDM? Press F1 for
more details.
Cause . . . . . : WebSphere Development Studio Client is part of the tool
set that you already own. It's a full set of workstation tools for iSeries
programmers. It includes things like Web tools, Java tools, WebFacing, and a
modern tool for traditional RPG/Cobol development called Remote Systems
Explorer (RSE). Give the new tools a try. Once you've taken the time to get
familiar with RSE, you'll find you can be even more productive than with the
old text-based iSeries tools. You can find a no-charge distance learning
course on WebSphere Development Studio Client at:
http://www7b.software.ibm.com/wsgd/library/tutorials/dl/swd73/
Recovery . . . : You can disable this message display from PDM Defaults.
                                                    Bottom

Press Enter to continue.

F3=Exit   F6=Print   F9=Display message details
F10=Display messages in job log   F12=Cancel   F21=Select assistance level

```

Of course, as you can read, you do have the option of turning off the PDM advertising from the defaults panel. While I was with IBM and since my retirement, I cannot remember a product such as PDM being altered to provide a directional message such as this. Though WDS is slick, it is still bulky and buggy. (Sorry IBM). The last time I installed it, there were bugs. And when I installed V5.1.2, there were bugs. Oh, IBM will fix them for you, but for the record, I don't remember the last bug I saw in SEU, SDA, or PDM.

The WebFacing Hosting Platform Decision

So that you have the WebFacing picture right, please note that it is my humble opinion that the tool would be better hosted on the AS/400 than on the development workstation. In fact, the interface described immediately below – the one that looks a lot like PDM would be a nice place to start. Considering that client server is dead, IBM's use of a heavy development client server approach for application development is suspect in 2005. Fat clients just are too much work to deploy. Having said that, please know that I think that the WebFacing tool, no matter whether it is implemented on the client or the server, is a neat product and it is fun to use. Though I would like it to be a little

more like what I have outlined below, I am pleased to take it as given.

The Standard WebFacing / PDM Interface Wish

Without using WDS_c however, as you would expect, you can WebFace your applications using a simple native green screen interface. This may be the better choice if you do not have Workstation or Web AD experience. The first step in the PDM-based WebFacing process as you would expect is to set the options for program calls and to select the style sheets to govern the overall look and feel of your converted display panels. IBM has a number of predefined styles, such as “gradient” and “avenue” or, of course, you can have an artist design your own style sheet from scratch. The styles define the overall look and feel of your site so their selection is not a trivial matter. You may use the WebFacing browser interface or the PDM green screen interface to make this selection prior to running the PDM-like WebFacing manager.

You start WebFacing on the AS/400 host similarly to PDM by using the Start WebFacing Conversion Manager command as follows:

STRWFM

The panels you see after you type this command look almost exactly as PDM panels. See Figures 14-4 and Figure 14-5.

Figure 14-4 The Main WebFacing Panel

```

AS/400 WebFacing Conversion Manager (WFM)

Select one of the following:

    1. WebFace libraries
    2. WebFace objects
    3. WebFace members

    9. Work with WebFacing style options and program calls

Information about new tools - press F1 for details

Selection or command
====> _____

F3=Exit      F4=Prompt      F9=Retrieve      F10=Command entry
F12=Cancel   F18=Change defaults

```

You can walk choose to walk through your libraries and pick the source files just as you would select source files with PDM. The WebFacing Conversion Manager is smart enough to convert just the Display File source. You can also walk through all the members of your source file to select those DSPF DDS members that you would like to convert. This option is shown in Figure 14-5 below:

Figure 14-5 Work With DSPF Source File DDS Members

```

Work with Display File Members Using WFM                                HELLO

File . . . . . QDDSSRC
Library . . . . HELLO                                     Position to . . . .

Type options, press Enter.
 1 Select For WebFacing  2=Edit  3=Copy  4=Delete  5=Display  6=Print  7=Rename
 8=Display description  9=Save   13=Change text  14=Compile  15=Create module...

Opt  Member      Type      Text
---  ---
 1   PANEL        DSPF      Display FILE Panel For Advanced Hello World
 1   PANEL1       DSPF      Display FILE Panel For Advanced Hello World
---  ---
    RCVDATA      PF        Receive Data From CGI
    VIEWDATA     PF        Receive Data From CGI

Bottom

Parameters or command
====> _____

F3=Exit      F4=Prompt      F5=Refresh      F6=Create
F9=Retrieve   F10=Command entry  F23=More options  F24=More keys

Have you tried the modern alternative to WFM? Press F1 for more details.

```


Besides the slick and very familiar interface to WebFacing as shown in the almost PDM-like panels above, there is even more good news. On the CRTDSPF command itself, IBM has added the option to WebFace the panel immediately. So, while it is creating a 5250 interface for your program, it also creates a Web interface at the same time and publishes the results as JSPS and XML artifacts to the proper directories of WebSphere Express for immediate Web execution. You don't have to know anything about the new technology to use it.

So, as you can see, there is no real reason why the new SDA cannot serve as your Java Server Page creator without ever having to know anything about JSPs or WebSphere itself for that matter. Since the display file is what gets converted and the interface is so intuitive for AS/400 developers, it could not be any easier.

Sorry, Only Kidding.

I really am sorry that I am kidding about the above. From the first time that I observed the WebFacing process, I was surprised that it was PC driven. I always felt that if the AS/400 were the target for execution under WebSphere, and all the source were already on the AS/400, it did not make much sense to pull it down to a big fat PC client to do the conversion, and then bring it back again to be installed under WebSphere.

Please forgive this trick I played. I wanted you to see how easy it could have been and how easy one day IBM can make it if it chooses.

The display files are the only piece of the action, which the WebFacing process converts. The RPG program with its EXFMT or the COBOL program with its READ/WRITE interface to display files does not change at all with WebFacing. In fact, the same compiled program can drive both 5250 panels and the WebFaced panels at the same time. When the display file DDS is translated into the Web artifacts, the program uses these to send out the display file as a Web page. Thus, the Web artifacts must be loaded to WebSphere on the AS/400 in order to work with your browser.

There is a WebFacing Server that is very much like the virtual terminal support that we have in iSeries Access. This server must be started before WebFacing can be run:

STRTCPSVR *WEBFACING

This whole section of the book is dedicated to showing you all that you must do for WebFacing to happen. There are literally tons of steps for something that could have been so simple if it were kept on the AS/400. However, overall the WebFacing process on the PC under WDS Sc Eclipse is reasonably easy to learn and it is actually fun to perform. Besides that, once your workstation code is settled, it also works!

Launching the WorkBench

Now, let's shift to the PC environment and start working with the WDS Sc we installed and fixed in Chapter 9. To get it going take the following clicks

**Start>> Programs>> IBM WebSphere
Studio >> Development Studio
Client for iSeries**

The clicks will look very much like the panel as shown in Figure 14-6.

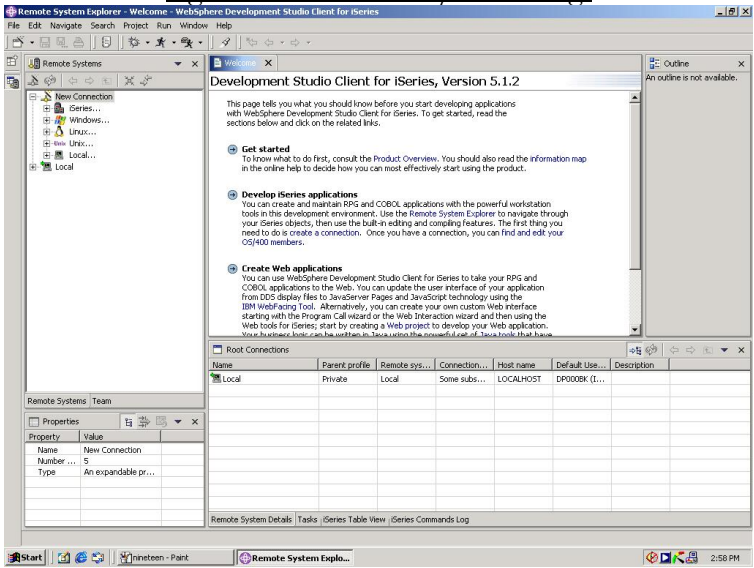
Figure 14-6 Invoking Eclipse

Once you release the mouse, you will see the WDS*c* Eclipse intelligent development environment (IDE) being launched. Do not get impatient. It takes quite a while to load this tool. The V5.0 or V5.1 or V5.1.2 page in Figure 14-7 appears for what should be briefly, but in reality it may be as much as 30 seconds.

*Figure 14-7 WDS*c* Version 5.0 Launching*

The first real page you see is shown in Figure 14-8. This is known as the default perspective. It is also known as the Remote Systems Explorer (RSE) perspective.

Figure 14-8 WDSc Eclipse First Page



When the Startup Page disappears, you will find yourself in what seems to be strange territory as seen above in Figure 14-8. If you are looking for a friendly face, you aren't going to find one here. When they say that Eclipse is as its plugins are, "they ain't kidding." Unfortunately, the panel is too big and full of information to make it readable in this size book. Don't worry, you'll see enough of it to know what is on this page.

At the first startup, before you have a chance to create any entrails, it is an empty Workbench – a vast expanse with lots of space just waiting for something to show.

The Eclipse Perspective

Now you know what I mean by non-intuitive. When you first arrive in Eclipse, you are shown what is called the Default Perspective, which happens to be the continually enhanced component, known as

the Remote Systems Explorer. It was first built when IBM chose to change from its original toolset (WDT) and move to the Eclipse Workbench.

The Eclipse packaging has gotten much better than in version 4. To use the Help panels in the 4.X release of the WDS, for example, if you have been an Eclipse user, you may recall that you had to change your “perspective” to the Help Perspective. We will define the notion of a *perspective* shortly, after we show more of the Help Text that will help you in your early days with the product.

In Release 5.1.2, as an example of a modern version of WDS, the Help Text emerges in its own window, independent of a WDS perspective. In this way, unlike the first release of WDS at version 4.x, you can use it to help you do your job, while you are doing your job. With V4.x, because HELP was in its own perspective, it covered your work while being displayed. Therefore, you would have to print or copy the HELP perspective first so that you had it to reference while you were doing what it instructed. The newer V5.1.2 and subsequent releases of WDS Eclipse permit you to read Help text while working on tasks that are explained in the Help text.

Great Help Overall

The WDS HELP is very awesome and very powerful, and consequently, the first time through Help, it is also very overwhelming. Considering that WDS does lots more than any one person can think of ever learning, the Help is necessarily huge, but it covers many areas that you may never need.

Now that we have spoken about perspectives, namely the Help Perspective in V4.X, let’s look at some of the terms, such as “perspective” that have particular meaning when you work with WDS Eclipse in any version.

What is a Perspective?

Before we walk you through the Help provided with V5.x, let's define the term "perspective," and discuss a few of the "views" that you'll find within. A *perspective* can be thought of as a collection of different views. When you eventually get to see the WebFacing perspective, you will see that it has its windows (views) arranged differently from the other perspective that you will have seen – Remote Systems Explorer (default).

The various views in perspectives are common throughout all perspectives in the WorkBench (the WDS Sc Eclipse product). To add more views, you can select the following from the menu bar:

V4.x -- Perspective > Show View

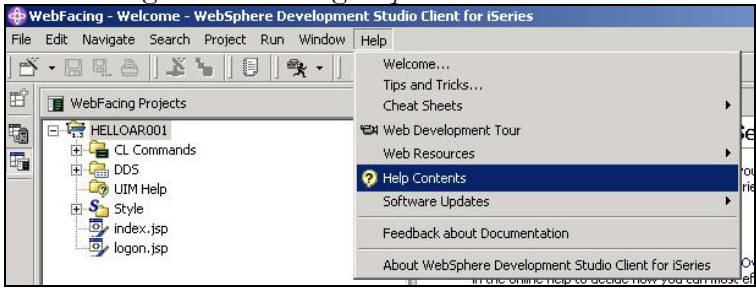
V5.x -- Window > Show View

You can close, resize, or move any of the views. Okay, now let's move on to examine what was once the HELP Perspective..

What Do We Do Now? Help!

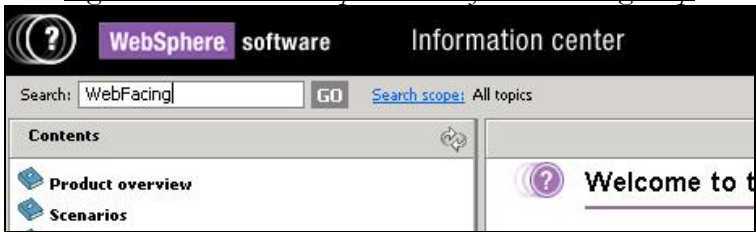
With WDS Sc V5.1.2, there is no HELP perspective so you get your help in the traditional way through the Windows Help Menu item as shown in Figure 14-9 below:

Figure 14-9 Getting Help with WDS Sc V5.1.2



The first time I sought help, I used the Windows Help option shown as the top option (not selected) of Figure 14-9. The Welcome panel is very nice and it may be what you need to get started. However, if you are looking for specific WebFacing information, select Web Contents as shown in Figure 14-9, and you will be taken to a panel that lets you search for WebFacing. This is shown in Figure 14-10

Figure 14-10 WDS Sc Help – Search for WebFacing Help



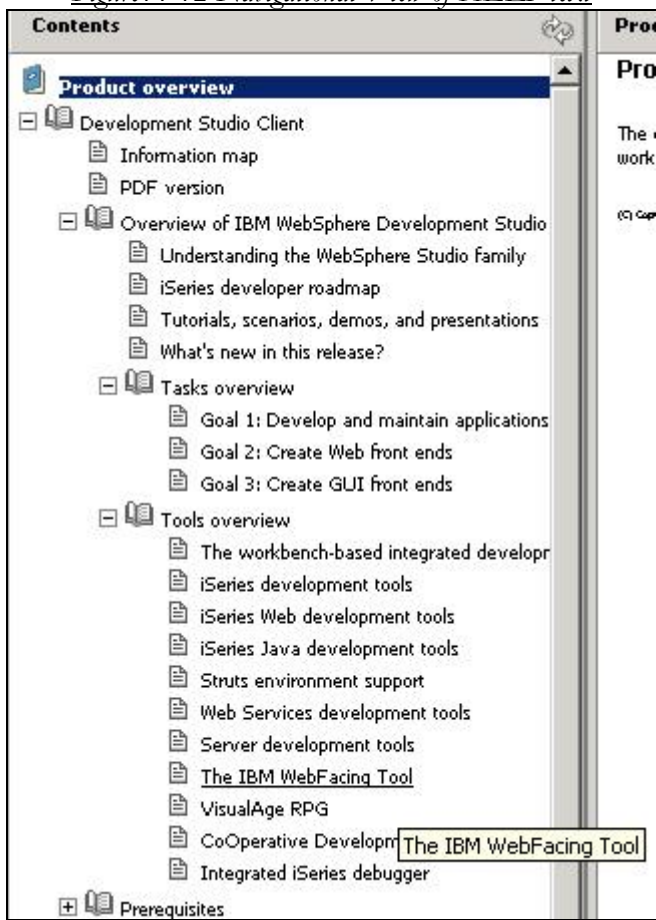
When you type in WebFacing in the search Windows and click the GO button in this case, you will get a ton of hits for WebFacing. A sample panel of hits is shown in Figure 14-11.

Figure 15-11 Sample Hits for WebFacing

The screenshot shows a search interface with the following elements:

- Search bar:** Contains the text "WebFacing" and a "GO" button.
- Search scope:** A link labeled "Search scope: All topics".
- Search Results:** A list of search results with percentages and titles:
 - 100% Starting and stopping the WebFacing runtime server on y...
 - 98% Opening the WebFacing perspective
 - 91% Configuring the WebFacing server for multiple interactive
 - 91% Creating a WebFacing portlet project
 - 88% Changing the port used by your WebFacing applications
 - 85% Customizing your WebFacing applications
 - 84% Creating custom JavaScript library files
 - 81% Developing your WebFacing services with a Web diagram
 - 79% Creating a WebFacing Web project
 - 79% Choosing a Web style
- Right-hand panel:** Contains a "Welcome to" section with a question mark icon, followed by "This information" and "More product-rel" (partially visible). Below this is a "Getting started" section with a question mark icon.

If you would rather search for HELP via the navigation method, you can take the Help Route as shown in Figure 14-12.

Figure 14-12 Navigational View of HELP text

As you can see in Figure 14-12, you can navigate from the Help Topics shown when you initially get into HELP from the Help Topics Launch as shown in Figure 14-10. If you keep opening up the items in the list, you will unearth the WebFacing topical area from which you can then explore all of WebFacing Help Text in a more logical fashion than via the search route.

No Startup Wizard

Unlike the original WDT WebFacing, WDS*c* Eclipse WebFacing does not first walk you through creating a project. This does make the process much less intuitive. Unless you navigate through the bulk of the Help Text (outside of the Welcome Panel), or you take my word for it, you would not naturally know what to do next in WDS*c*.

To help you position yourself to get the most Help that you can in your Eclipse sessions, before you create your first WebFacing project, take a trip down HELP lane as shown above so that you can be more equipped to deal with any issues as they arise.

After you read the Help material and you are ready to roll, you begin the process with a few mouse clicks. If you have had enough HELP and are ready to click, move on to Chapter 15.

But, make sure you've got your mouse in your hand!

We'll continue with WDS*c* in Section IV, Chapter 20 as we use it to WebFace an application. As you will see this process takes us eight chapters to complete and, is in fact a book within an integrated book.

For now, it's time to set up the HTTP and WAS server for iSeries Access for Web and while we are at it we will also set up and configure the new iSeries Access for Web program in the new WAS server – iwa51exp that we create in Chapter 15.

Chapter 15

Setting Up HTTP and WAS Express Servers for iSeries Access for Web.

Creating WAS Server for Use with iSeries Access for Web

As you have been observing in this book for the last two chapters, we create functional WAS servers / instances for the type of work that we are going to perform in a particular WebSphere / HTTP instance combination. It is good for you to do this in your own shops. In Chapter 12, for example we created a WAS and HTTP instance needed for WebFacing that could also be used to run the IBM Business Applications such as Telephone Directory and Survey Creator. As you have seen will see in Chapter 13, however, our plan is to not run the IBM Business Applications in the WASEONE server but rather to create server instances for each. Actually, we can run Bizapps in both.

In this chapter, we carry the notion of server instances for major applications further by creating a specific set of instances (HTTP and WAS Express) for iSeries Access for Web.

In addition to iSeries Access for Web, we create these instances to permit the IBM Examples to run. In this way, without having to launch iSeries Access for Web, we can prove that the WAS Express / HTTP combo works by running the infamous Snoop program.

This of course is the fourth WAS Express instance that we are creating. It will be named iwa51exp. WASEONE was created in Chapter 12. TELDIR and SURVEYC were created in Chapter 13, and HATSLE will be created in Chapter 16. This separation of function helps us in many ways:

1. Permits one set of server instances (HTTP and WAS Express) to represent one major Web function.
2. Permits us to isolate problems to one instance. If a problem is so severe that it brings down WebSphere or HTTP, only one instance is affected.
3. It gives us practice creating WAS Express and HTTP instance pairs so this important function will not be foreign to us.

Since we have been through a most inclusive Create Wizard exercise in Chapter 12, and we followed the instructions for two more in Chapter 13, we should now be somewhat adept at creating WAS instances.

Using the WAS Creation Wizard

To start the WAS Creation Wizard, as you have become accustomed in Chapter 12, call up the Admin GUI and sign on. Select the Setup Tab and start the process. The panel from which you invoke the Admin GUI is shown in Figure 15-1, followed by the sequence of panels necessary to get the WAS Server created.

Note:

There is minimal text for these panels just as there were few figures in Chapter 13. For the most part, we have created an instance similar to this once before. WASEONE in Chapter 12 is a superset of this instance since WASEONE also supported the Bizapps. For more detail on the process of creating an Instance, feel free to use the information in Chapter 12.

1. To start the WAS Creation Wizard, call up the Admin GUI and sign on.

http://yourdomainor ipaddress:2001

Figure 15-1 WAS/HTTP Admin GUI

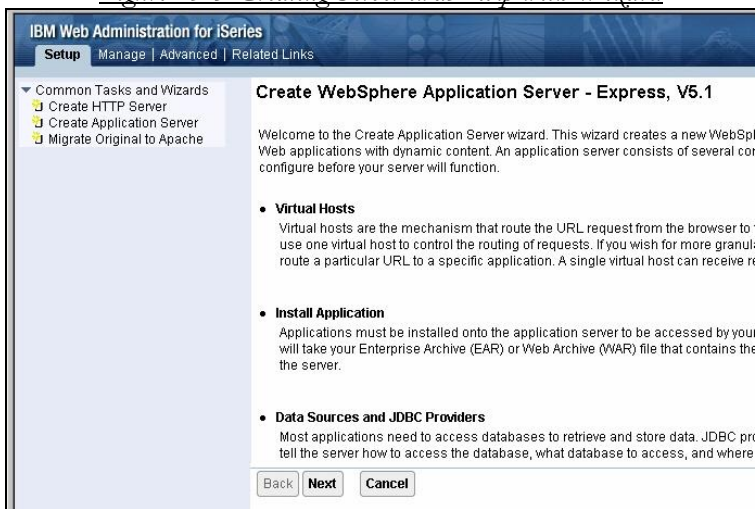


2. When you get to the Man Admin GUI panel take the option for IBM Web Administration for iSeries as shown in the panel in Figure 15-1.
3. Take the Setup Tab from the Admin GUI Panel to which you arrive. You will see a panel similar to that shown in Figure 15-2.

Figure 15-2 WAS Creation Wizard for iwa51exp Server



4. Select the Create Application Server option to invoke the CREATE WAS Express Wizard.

Figure 15-3 Creating Server iwa51exp with Wizard

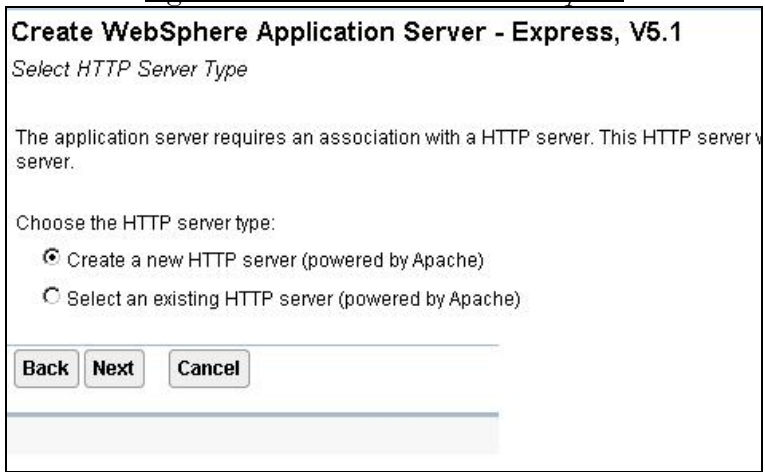
5. Click Next from Figure 15-3 to move past the HELP Panel

Figure 15-4 Giving Express Application Server a Name

6. In the panel shown in Figure 15-4, name the WAS Server `iwa51exp` and provide a description of the WAS Express Server. Then click the Next button.

Again, our recommendation is that you name it the same as the HTTP server that you plan to use with it. In this case, as is our normal procedure, we are going to have the Wizard create both an HTTP server and a WAS Express server at the same time. `iwa51exp` is the name that you use for both.

Figure 15-5 Select the HTTP Server Option



7. Select the radial button option to Create a new HTTP server (powered by Apache). You will then see a panel similar to that shown in Figure 15-6.

Please note that the Original IBM CERN Server is not supported with WAS Express or V5R3. Thus, it is not one

of the options when picking an HTTP server to work hand in glove with WAS Express.

8. Run NETSTAT. Go to an AS/400 Command Line and type NETSTAT, then take option 3, Work with TCP/IP connection status. Press command key 14 to see the open ports. Look for the Local Port numbers in column 3 of the NETSTAT report and roll down. The ports are in sequence. When you are asked for an HTTP port number in the WAS Express Create Wizard, be prepared to enter a value that is not used (not in the screen report). For the WAS Express ports, make sure that there are twelve ports available from the starting port number since WAS Express needs twelve ports. Also, make note of these ports since your Firewall will need to be updated to permit traffic through these ports.

Figure 15-6 HTTP Provide Server Specifics

Create WebSphere Application Server - Express, V5.1

Create a new HTTP server (powered by Apache)

A new HTTP server (powered by Apache) will be created and configured to be used by the

HTTP server name:

HTTP server description:

Your HTTP server may listen for requests on a specific IP address or on all IP addresses

On which IP address and TCP port would you like your HTTP server to listen?

IP address:

Port:

Note: Most browsers make requests to port 80 by default.

Back
Next
Cancel

9. Supply a name for the HTTP server that is the same `iwa51exp`, as the WAS Express Server instance. Use all IP addresses and have the HTTP port set to 2042. Click Next to continue. You will see a panel similar to that shown in Figure 15-7.

10. Either use NETSTAT again or while you are in the NETSTAT panels on your AS/400 or iSeries, pick another port number from which the system can allocate thirteen additional consecutive ports (14 in all) for other WAS Express activity. You will need this value when you complete the next Wizard panel. Again avoid conflicts with other IP servers on your iSeries. Provide this port information to your firewall expert if you are expecting this new server to be reached from the Internet.

Figure 15-7 Specify Beginning Internal Port #

Create WebSphere Application Server - Express, V5.
Specify Internal Ports Used by the Application Server

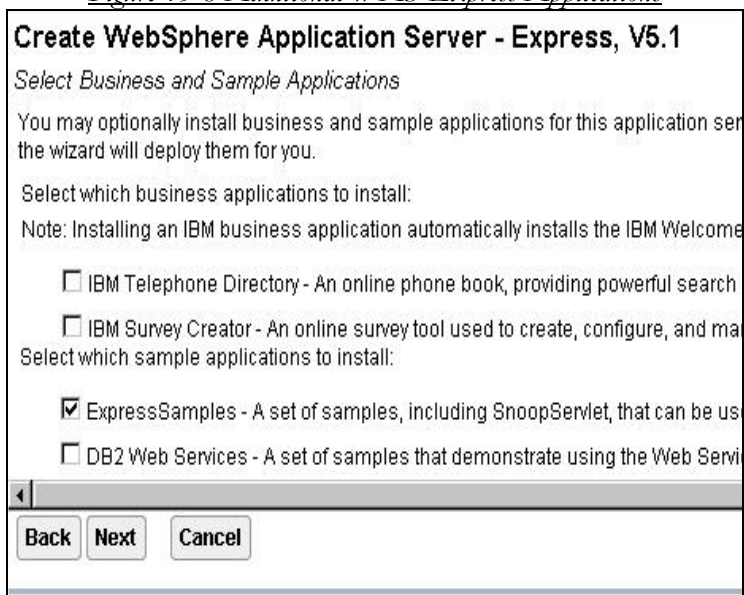
The WebSphere application server uses several internal services such as internal HTTP transport service, Simple Object Access Protocol (SOAP) service, name service, and several other services to perform its processing. In order for these services to be configured, you must provide a block of 12 consecutive ports that are currently not in use on your system. Specify the first TCP port number in the range and the wizard will assign the ports that are to be used by each internal service. For example, if 1312 is entered as the first port in the range, then ports 1312 to 1323 will be configured.

First port in range: ?

The full text for this panel is shown as follows:

Help Panel Text: The Application Server - Express, uses several internal services such as internal HTTP transport service, Simple Object Access Protocol (SOAP) service, name service, and several other services to perform its processing. In order for these services to be configured, you must provide a block of 12 consecutive ports that are currently not in use on your system. Specify the first TCP port number in the range and the wizard will assign the ports that are to be used by each internal service. For example, if 1312 is entered as the first port in the range, then ports 1312 to 1323 will be configured.

11. Provide the first Port in the WAS Express range of 13 ports for this instance, and click the Next button and you will be taken to the panel in Figure 15-8.

Figure 15-8 Additional WAS Express Applications

Continuing with the WAS Creation process

12. In Figure 15-8, do not select any of the Business applications but do click on the ExpressSamples option to assure that we can run Snoop. Then click on the Next button to continue with the panel shown in Figure 15-9.

Because there is no LDAP in this instance, skip Generic Steps 13 through 15 as outlined in Chapter 11.

Figure 15-9 Review Creation Parameters

Create WebSphere Application Server - Express, V5.1

Summary
When you click **Finish** this WebSphere application server will be created.

Application Server **HTTP Server**

Application server name: iwa51exp
Server description: iSeries Access for Web instance
Internal port range: 2043 - 2055
Virtual host: default_host
Business applications: None
Sample applications:

Application name	URL to access application
ExpressSamples	SERVER2:2042/snoop

Note: To access the installed application(s), start the application server and HTTP server, then window to access the application.

Back **Finish** **Cancel**

Save all of the pending changes for this task.

Review the Creation Parameters

16. When we build this WAS (iwa51exp), as noted in the summary in Figure 15-9, we ask the Wizard to enable the WAS Samples. The sample program we care about is called Snoop. This application has been around since the early WAS versions and is very handy in proving that the WAS server instance has been built correctly, can be started, and is capable of delivering applications.

Make a note of the URL in Figure 15-9

SERVER2 : 2042 / snoop

Or in IP form

192.168.0.250:2042/snoop

This will launch the Snoop application when the instance is created and started. Write it down so you won't forget it after all your installation work is done. In our case, as you can see, to execute snoop, we would type the following into our browser:

http://server2:2042/snoop.

Again, this reflects the system name in the PC host table, the HTTP port number, and the name of the Snoop application that is called.

You can substitute your domain or your IP address for server2.

The Admin GUI process is smart enough to go to the TCP/IP host name in your PC host table in this example to resolve the name server2 to the proper IP address. Soon after you hit the Finish button in Figure 15-9, you will have a new set of servers built, which can be started with the Admin GUI. Once the HTTP and WAS Express instances are started, you can run the Snoop application with the above URL (or similar URL to represent your domain name or IP address) to prove that your new *iwa51exp* instance is functional.

When you have digested the items that you have previously specified as shown in Figure 15-9, and all seems well, click

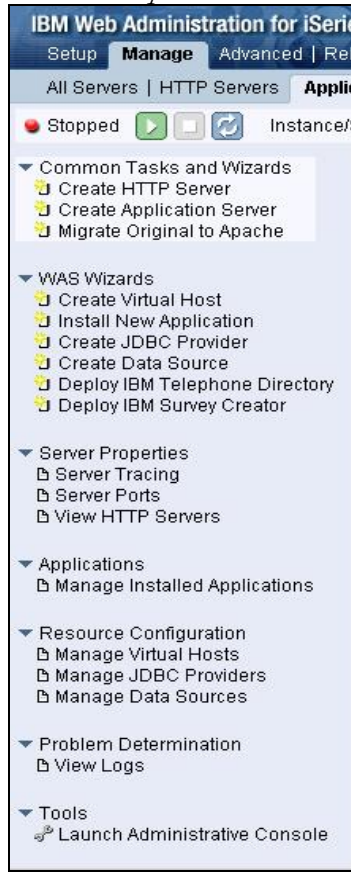
on the Finish button to create both your WAS Express and HTTP servers. While the servers are in the process of being created, the left frame of the HTTP Admin GUI Admin display lights up the “Creating” message and its little yellow button (to its left) as noted in Figure 15-10.

Figure 15-10 Creating WAS Express and HTTP Servers



WAS and HTTP Being Created

With WAS Express 5.1 the fact that the instance is being created is shown in the left pane as you can see above. When the create process has finished, the left pane of the panel you will see will be similar to that shown in Figure 15-11.

Figure 15-11 WAS Express Server Created / Stopped

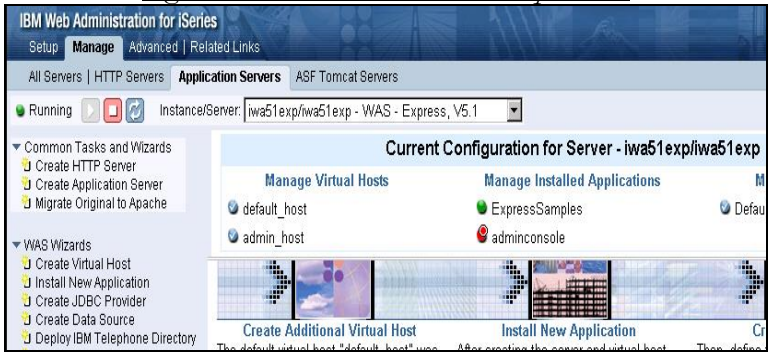
At this point, our iwa51exp WAS Express 5.1 server and our Apache HTTP Server have been created. Now, it's time to start this new WAS instance to assure that it works.

17. Start the WAS Server and the HTTP Server for the instance just created using the Admin GUI

Just as with WASEONE, you click the bright green and white right arrow button at the top in the left pane (next to the word stopped) in Figure 15-11 to start the newly created iwa51exp server. The “stopped” status will change to “starting” almost immediately after

you click the START button. After several minutes the status will change again to “Running” as shown in Figure 15-12.”

Figure 15-12 WAS Server iwa51exp started



The next step would be to switch to the HTTP Server Tab and start the Apache HTTP server instance called iwa51exp before running Snoop.

18. Type in the HTTP URL in the following form to start the Snoop Program

**http://yourdomainname:yourhttpport
#**

http:192.168.0.250:2042

2042 is the HTTP port for iAWeb as told to the Wizard in this chapter. Your port number would be determined by that assigned in the Admin GUI Create WAS Server and Create HTTP server processes.

Press ENTER on your browser and you will get the sample home page for the instance that you have selected (port #).

19. Run Snoop (IBM Test Sample Program) to make sure your servers are working. Use the following URL:

```
http://yourdomainname:yourhttpport  
#/snoop
```

```
http:192.168.0.250:2042/snoop
```

20. Check the snoop report. If it ran, your WAS is OK.

That's all there is to create the iwa51exp WAS instance. .

In the last four chapters, we have created four Apache HTTP Server instances as well as four WAS Express server instances. We will use the WASEONE server for WebFacing; the SURVEYC instances for the IBM Survey Creator, the TELDIR instances for IBM Telephone Directory; and the iwa51exp instances for iSeries Access for Web. That's a lot of work, but we still are not done.

In the next chapter, we will describe what needs to be done to create just one more WAS instances that we will dedicate for the HATS/LE environment

iSeries Access for Web Additional Config and Setup.

Once iSeries Access for the Web is installed and you have applied the PTFs, your iAWeb installation work has just begun. No, it's not really that bad but it is unusual work shall we say for the novice WebSphere user.

If you can think back that far, in this chapter, we already created a WebSphere Express 5.1 instance that we named iaw51exp. As we told you way back then, this WAS instance is where we will run the iSeries

Access for the Web application. We also created a companion HTTP instance and likewise we named it `iaw51exp`. If you were using WAS Express 5.0, your instance creation activities would be quite similar.

iAWeb is an application that runs under the control of WebSphere. It is a WebSphere application just like the WebFacing application that we will complete starting in Chapter 20. However, instead of providing a specific application, such as we will do with Advanced Hello World for WebFacing, the iAWeb application under WebSphere provides a means for us to run all of our other applications.

It thereby provides a reasonably complete Telnet-like command level interface to the AS/400. The thing that makes it much different than Telnet is that you do not have to install anything on your PC. It runs from your browser. Just as if you were running jobs from a terminal, you can fire up your Web browser and use iAWeb for typical green screen facilities from your AS/400 system.

If you look at your Licensed Programs display after performing the installation tasks already described in this book, you would not be able to tell that the iAWeb program 5722-XH2 is not functional at this time. Oh, sure, it's there but it is not ready to go to work for you yet.

The GO LICPGM install process was unable to complete the installation of 5722-XH2 because it is not sure of which version of WebSphere you might be using and it is not sure that you had even set up or installed WebSphere in the first place. This testifies to the lack of full integration of WebSphere and the Web enablers on the iSeries.

As an aside, if the AS/400 (iSeries or i5) were treated as an integrated platform today, this would not be the case. There would be one servlet server on the system and, just as AS/400 journaling; it would not have its own identity. It would merely serve servlets when called upon by the Operating System. But, I digress.

Running the iAWeb Setup Script

Though iAWeb is not completely set up at this point of the process if you are following this book, the IBM Raleigh developers do provide a nice script for you to run with whichever version of WebSphere you

have chosen. Once you pick the script and run it, it's as if iAWeb were there on your AS/400 from the time you did the last release update. For our installation, we have chosen to use WAS Express 5.1 and i5/OS V5R3 since that is the major basis for this book. However, since I have done a V5R2 installation with iSeries Access Version 5.3, I know that the script is the same. Moreover, iAWeb for WAS 6.x should be more of the same. So, if you are on V5R2, just follow along.

The script requires a facility called Qshell because it was built in a language WebSphere understands quite well from its Unix heritage. It is not a CL script. It is a Java script, which can be processed by the Unix interpreter on the system, namely the Qshell interpreter.

To get to the Qshell interpreter so that you can install iAWeb, go to an AS/400 green screen command line and type in the following three letters followed by a hearty depression of the ENTER key:

QSH

From here, to find the script, you have to change directories. The process is very much like an old DOS PC or a Unix box. To find your way into the directory that has the iAWeb installation script, you have to change directories. The script is stored in your integrated file system within the following directory in the root file system:

```
/QIBM/ProdData/Access/Web2/install
```

If you were to open a command prompt on another green screen session or a terminal, you would see what is in the install directory by using IBM's WRKLNK command. WRKLNK is very much like a directory command (DIR) in DOS in that it shows you the contents of a directory. (folder in Windows parlance).

So, do not abandon your QSH session. Start a new command level session or use a different terminal to “walk through” the directories with the WRKLNK command to find this script before you run it. Start on an AS/400 command line by typing

WRKLNK

Then, press ENTER.

Roll until you find QIBM. Then take an option 5 next to it and press ENTER. Then find ProdData and put a 5 next to it and press ENTER. Then find Access and put a 5 by it and press ENTER. Then, find Web2 and place a 5 next to it and press ENTER. Finally find the *install* directory and again place a 5 next to it and hit ENTER. You will then see the contents of the install directory. Look for the following script:

cfgaccweb2

If you want to at this point, you can view the script contents once you navigate to *cfgaccweb2* it by placing a “5” next to

it and hitting ENTER, just as you did to walk through the directory tree. A built in editor will show you its contents.

At this point, you can believe that you have found the script and it is not a mystery any longer. Now, you can go back to the QSH session that you already started. To assure you are starting at the root directory, you can issue the Unix change directory command, Just like DOS, it is CD but, unlike DOS, the CD command to get to the root directory of the IFS is followed by a forward slash (not back slash). That command would look as follows:

From CL command line

```
CD /
```

If you perform this operation in QSHELL, you will be in the root directory. However, you are in a Unix environment so your CL won't help now one bit to get to the actual directory where the script resides, you can type in the full path change directory command all at once:

```
CD /QIBM/ProdData/Access/Web2/install
```

When you are in Qshell and you are about ready to change your directory, your screen will look similar to the top part of the panel shown in Figure 15-13

Figure 15-13 Qshell Change Directory

```

QSH Command Entry

$
> cd /QIBM/ProdData/Access/Web2/install
$

==> cfgaccweb2 -appsvrtype *WAS51EXP -wasinst iwa51exp

F3=Exit   F6=Print  F9=Retrieve F12=Disconnect
F13=Clear F17=Top   F18=Bottom F21=CL command entry

```

From here, as you can see in Figure 15-13, you can type in the script followed by some important parameters:

```
cfgaccweb2 -appsvrtype *WAS51EXP
-wasinst iwa51exp
```

Notice the format of the command. All parameters to script commands start with minus signs as you can see in Figure 15-13. First, the parameter prompt “*-appsvrtype*” defines the type of server that iAWeb will be deployed within. This alerts the script as to the specific structure of the directories for that WAS type. *WAS51EXP is the descriptor for the WAS Express version 5.1 that we are using. The next prompt is *-wasinst*. This asks for the name of the WebSphere Express instance that we have previously set up in the first half of this Chapter. By now, we all know that this is **iwa51exp**.

When you press ENTER in QSH, the shell executes and prepares iAWeb to function in the iwa51exp instance. While this is occurring, you will see the types of messages on your Qshell panel as shown in Figure 15-14

Figure 15-14 Script Message-- iAWeb WebSphere Install

```
QSH Command Entry
$
> cd /QIBM/ProdData/Access/Web2/install
$
> cfgaccweb2 -appsvrtype *WAS51EXP -wasinst iwa51exp

Configuring iSeries Access for Web

Preparing to perform the configuration changes.

Calling WebSphere to perform the configuration changes.

iSeries Access for Web command has completed.

The WebSphere instance application server must be stopped and then started to
enable the configuration changes.

$
===>
```

When you are finished with Qshell and you have your success message, the iAWeb is fully installed on your system. Go to Part IV of this book to see how to run the iSeries Access for Web program. Look for this excitement in Chapter 28.

Chapter 16

Setting Up HTTP and WAS Express Server for HATS/LE; Configuring HATS/LE

Introduction

As you may recall from Chapter 10, we “installed” the HATS/LE code to the AS/400 by mapping a drive and copying the code from the installation CD to the IFS /INCOMING directory of the AS/400 or iSeries server. To complete the setup, configuration and actual WebSphere installation of HATS/LE, you must use the file that starts out as /HA/HATS5LE/HATS5LE.ear on the HATS CD, now residing in the /INCOMING directory of your AS/400.

You may also recall that in Chapter 12 thru 15, we created a number of WebSphere Express instances. This list of servers (instances) , includes *iwa51exp* that we built in Chapter 15 and we used to install iSeries Access for Web.

We could install HATS/LE in any of these other WAS instances but to keep things clean, again we will create a

new WAS instance for HATS/LE. To keep it really clean and understandable, let's call the instance HATSLE and let's create a companion HTTP instance for it with the same name. Let the port for Apache HTTP start at 18000 and the port range for WAS Express start at 18001.

The instructions to create the HATSLE instances look just like the generic instructions described in Chapter 11 and used for the IBM Bizapps other than one thing. Just like iwa51exp, The HATSLE instance does not need the IBM Business Solutions installed.

Creating WAS and HTTP Servers for HATS/LE

As you now know from all the instances we have created, an application server instance provides the runtime environment for your applications. An instance of WebSphere Application Server - Express consists of a single application server, which connects to an HTTP server instance to receive client (browser) requests. The application server performs administrative functions and provides services that your application uses to process its client requests.

The general steps to create a new application server for HATS/LE with the Admin GUI (HTTP Server Administration interface) are as follows:

1. To start the WAS Creation Wizard, call up the Admin GUI and sign on.

http://yourdomainor ipaddress:2001

2. When you get to the Man Admin GUI panel take the option for IBM Web Administration for iSeries

3. Take the Setup Tab or the Manage Tab from the Admin GUI Panel in which you arrive.

4. Select the Create Application Server option to invoke the CREATE WAS Express Wizard.

5. Click Next to move past the HELP Panel

6. Name the server HATSLE and provide a description of the WAS Express Server. Then click the Next button.

Again, our recommendation is that you name it the same as the HTTP server that you plan to use with it.

7. Select the radial button option in the Create WAS Express Wizard to create a new HTTP server (powered by Apache).

8. Run NETSTAT. Go to an AS/400 Command Line and type **NETSTAT**, then take option 3, Work with TCP/IP connection status. Press command key 14 to see the open ports. Look for the Local Port numbers in column 3 of the NETSTAT report and roll down. The ports are in sequence. When you are asked for an HTTP port number in the WAS Express Create Wizard, be prepared to enter a value that is not used (not in the screen report). For the WAS Express ports, make sure that there are twelve ports available from the starting port number since WAS Express needs twelve ports. Also, make note of these ports since your Firewall will need to be updated to permit traffic through these ports.

9. Supply a name for the HTTP server (HATSLE) that is the same as the WAS Express Server instance. Use all IP addresses and set the HTTP port to 18000. Click Next to continue.

10 Either use NETSTAT again or while you are in the NETSTAT panels on your AS/400 or iSeries, pick another port number from which the system can allocate thirteen additional consecutive ports (14 in all) for other WAS Express activity. You will need this value

when you complete the next Wizard panel. Again avoid conflicts with other IP servers on your iSeries. Provide this port information to your firewall expert if you are expecting this new server to be reached from the Internet.

11. Provide the first Port in the WAS Express range of 13 ports for this instance, (port # 18001) and click the Next button..

12. In the next panel, you will be asked which IBM Business application to select. Pick just the IBM ExpressSamples so that we can invoke Snoop to insure that the instance works. Then click on the Next button to continue.

13. Skip steps (generic steps) 13 to 15.

16. Review the Configuration parameters on the Next panel and when you are satisfied, click FINISH to create the new WAS Express and HTTP servers.

17. Start the WAS Server and the HTTP Server for the instance just created

18. Type in the HTTP URL in the following form

http://yourdomainname:yourhttpport#

http:192.168.0.250:18000

Press ENTER on your browser and you will get the sample home page for the instance that you have selected (port #).

19. Run Snoop with the following URL

```
http://yourdomainname:yourhttpport  
#/snoop
```

```
http:192.168.0.250:18000/snoop
```

20. Check the snoop report. If it ran, your WAS is OK.

That's all there is to create the HATSLE WAS and HTTP instances.

Set Up HATS/LE in HATSLE

HATS/LE is a separate product bundled with 5722-XH2 iSeries Access for Web. You may recall that we “installed” HATS/LE in Chapter 10 by copying the EAR File from the HATS CD to the IFS /INCOMING Directory. Prior to the WAS Express and HTTP server pair being built for it, we could not fully set up HATS to run on our AS/400. Now, that we have created the HATSLE server, we can install the HATS/LE EAR file to this WAS instance.

The mechanism to set up HATS/LE in this new server instance that we created is the same method used to install any application (other than Bizapps) to a WAS Express server. From a WAS perspective, you will be installing the HATS5LE.ear file to the HATSLE WAS server instance. The installation is like any other EAR file to WAS Express. To install HATS/LE, you must use the file that starts out as /HA/HATS5LE/HATS5LE.ear on the HATS CD, now residing in the /INCOMING directory.

The panels to create the HATSLE WAS Express and HTTP instances are shown in the beginning of this Chapter. Now, it's time to install the product into this instance.

Installing HATS5LE EAR File

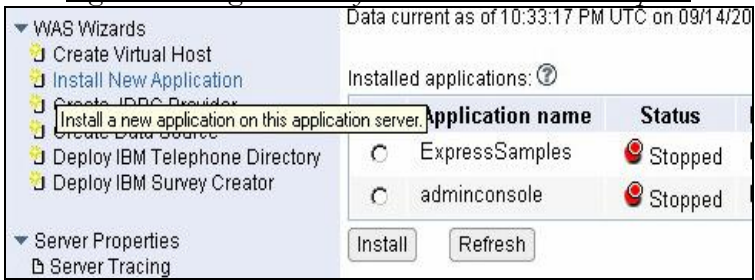
An EAR file is a super compressed directory with all of the necessary information for a Web project plus metadata describing the project itself. In the J2EE world, EAR files are the mechanism by which applications are ported from the development workstation to the server. As noted previously, the HATS/LE EAR file is already on our AS/400 as we begin the application installation process.

To get things going for the HATS/LE EAR file installation to WebSphere, start by signing-on to the Admin GUI and immediately select the Manage Tab. Midway down the panel you will see a menu item under the word “Applications” that says

Manage Installed Applications

Click on it and a panel similar to the panel in Figure 16-1 will appear.

Figure 16-1 Begin Install of EAR File to WAS Express

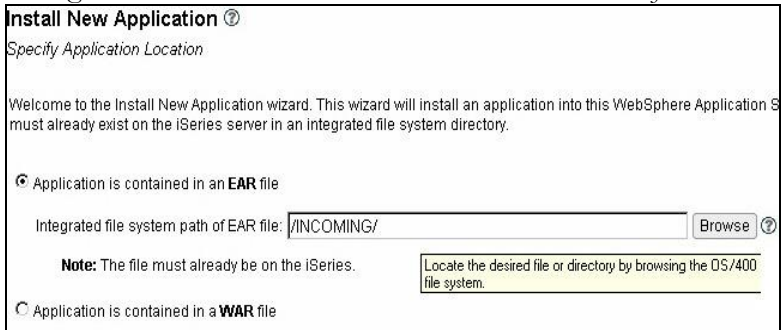


Under the Manage Tab and WAS Wizards, as shown in the panel in Figure 16-1, you will see a menu item called Install New Application. Start the installation process by clicking on this item and you will be taken to a panel similar to that shown in Figure 16-2.

Tip:

Since you had previously clicked the Manage Installed Applications, the right pane in Figure 16-1 includes a look at the installed applications. At the bottom of this pane, you can see an Install button and a Refresh button. Clicking the Install Button from this pane has the same effect as taking the Install New Application menu item from the left pane.

Figure 16-2 EAR File Is in /INCOMING – Browse for File



In 16-2, you get to tell the WAS Application Install Wizard where the ear file exists for the application you wish to install. From this panel, take the Browse option to find the IFS directory and the HATS5LE.ear file. You will then see a panel similar to that shown in Figure 16-3.

Figure 16-3 Found HATS5LE.ear File in /INCOMING



You may have to navigate to the root to find the /INCOMING directory since the Wizard will default to the installableapps directory in the HATSLE WAS instance IFS structure. When you get to the /INCOMING directory, select the HATS5LE.ear file by clicking on it, and then click OK. The Browse function will bring the path of the file back to the Wizard as shown in Figure 16-4.

Figure 16-4 Confirm HATS5LE EAR File to Install Wizard

Install New Application ?

Specify Application Location

Welcome to the Install New Application wizard. This wizard will install an application into the iSeries server. The application must already exist on the iSeries server in an integrated file system directory.

Application is contained in an **EAR** file

Integrated file system path of EAR file: ?

Note: The file must already be on the iSeries.

Application is contained in a **WAR** file

From Figure 16-4, all you have to do to continue is confirm the location of the HATS5LE.ear file by clicking Next. You will come to the next panel in the Install Wizard shown in Figure 16-5.

Figure 16-5 Additional Options for Installation

Install New Application ?

Provide Options to Perform the Install

Specify application deployment options

Application name: ?

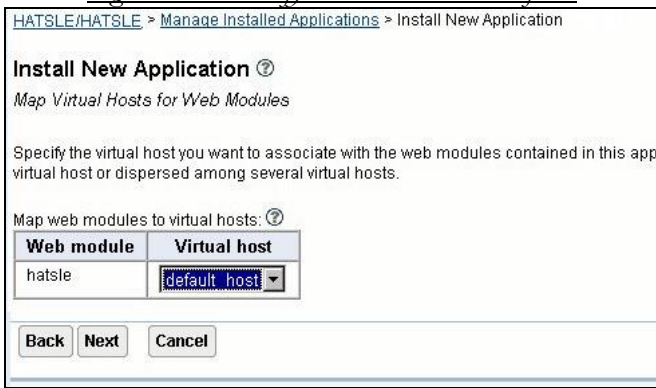
Directory application installed to:

Pre-compile JSPs ?

Note: Pre-compiling JSPs can have significant performance impacts. When enabled, installation time, causing the application install to take longer. When disabled, the application is not pre-compiled, causing the first use of the application to take longer.

In Figure 16-5, under the window containing the name of the application to be installed (HATS5LE.ear) the Wizard shows the directory to which HATS/LE will be installed. Permit this to default. It also provides the name of the application from the name of the ear file. Select the “Pre-compile JSPs” option by clicking the box in the panel. This assures that the HATS/LE program will operate as quickly as possible when it is started. Click Next to continue with the WAS Express Application Install Wizard and you will get the next panel as shown in Figure 16-6.

Figure 16-6 Verify Virtual Host as Default



Take the default for Virtual Host and click the Next button to continue with the Summary panel shown in Figure 16-7.

Figure 16-7 Summary of HATS5LE Installation Params

HATSLE/HATSLE > Manage Installed Applications > Install New Application

Install New Application ⓘ

Summary

When you click **Finish** the installation will be started for the following application.

Integrated file system path of EAR file: /INCOMING/HATS5LE.ear

Application name: HATS5LE.ear

Directory application installed to: /QIBM/UserData/WebASE51/ASE/HATS5LE/installedApps

Pre-compile JSPs: Enabled

Map web modules to virtual hosts:

Web module	Virtual host
hatsle	default_host

Review the install parameters and assure that they are correct. Then, click **Finish** to have the HATS/LE application installed within the HATS/LE WAS Express server instance. You can see in the panel shown in Figure 16-8, that the HATS5LE.ear application is in the process of installing. When it is completely installed, as shown in Figure 16-9, the radial button will turn to red, the status will change to *Stopped* from *Installing* and the Enablement will change to *Enabled*. Then, the application can be started and used.

Yes, that's all there is to installing a WAS application, even if that WAS application happens to be something as big as HATS/LE. .

Figure 16-8 HATS/LE Being Installed to HATSLE Instance

HATSLE /HATSLE > Manage Installed Applications

Manage Installed Applications ?

Data current as of 10:41:27 PM UTC on 09/14/2004

Installed applications: ?

	Application name	Status	Enablement
<input type="radio"/>	ExpressSamples	Stopped	Enabled
<input type="radio"/>	adminconsole	Stopped	Disabled
<input type="radio"/>	HATS5LE.ear	Installing	Disabled

Install Refresh

Figure 16-9 HATS/LE Successfully Created in HATSLE

IBM Web Administration for iSeries

Setup **Manage** Advanced | Related Links

All Servers | HTTP Servers **Application Servers** ASF Tomcat Servers

Stopped Instance/Server: HATSLE\HATSLE - WAS - Express, V5.1

HATSLE /HATSLE > Manage Installed Applications

Manage Installed Applications ?

Data current as of 10:58:20 PM UTC on 09/14/2004

Installed applications: ?

	Application name	Status	Enablement
<input type="radio"/>	ExpressSamples	Stopped	Enabled
<input type="radio"/>	adminconsole	Stopped	Disabled
<input type="radio"/>	HATS5LE.ear	Stopped	Enabled

Install Refresh

Starting the HATSLE Instances and HATS/LE Application

In order to be able to run HATS/LE from the HATSLE instance, three things must be set properly:

1. HATSLE WAS Instance Started
2. HATSLE HTTP Instance Started
3. HATS5LE.ear application started.

As you can see in Figure 16-10, the HATSLE WAS Instance is in a stopped state and the HATS5LE.ear application is also stopped. Since the HTTP instance was just created, for this exercise, you can also assume that it is stopped. In order to use HATS/LE, the product, therefore, you just start all three of these items.

From the panel in Figure 16-9, click on the green and white Start Button in the left pane of Figure 16-9. This action will start the new WAS Express instance for HATSLE and by starting the instance, the process will also start the enabled applications, of which HATS5LE.ear is one. While the instance is starting, the next panel will look very similar to that shown in Figure 16-10.

Figure 16-10 HATSLE WAS and HATS App Starting



When the HATSLE WAS Server and the HATS5LE application are started and ready, the panel will look similar to the one shown in Figure 16-11.

Figure 16-11 HATSLE WAS and HATS App Running OK



HATS/LE Tailoring

At this point, HATS/LE is installed in the HATSLE WAS Express instance. We'll show you how to run HATS in Section IV. Now it's time to set the dials so that HATS/LE is tailored and enabled to run in the HATSLE WAS Express instance.

HATS/LE and iSeries Access for Web

HATS/LE and iSeries Access for Web's 5250 facilities are very similar but different. With HATS/LE you can quickly

transform your 5250 host application into a more attractive Web application and make it available to your users through their browsers. You can also have iSeries terminal function with iSeries Access for Web 5250 support. The difference is that HATS affords you customization options that are not available in iSeries Access for Web.

More than likely, from what I have seen with HATS/LE, there is little need for the product as a separate piece of the iSeries Web Action. When IBM begins to focus more on integration for the iSeries, hopefully this will be addressed. This function should all be part of iSeries Access for Web which should be part of the overall iSeries standard Web enablement set. Ideally, there would be one less product to install and one less thing to get the non-WebSphere savvy iSeries community confused about. It would help IBM offering more pain free Web access facility.

Investment Protection

Both HATS/LE and iSeries Access for Web help you protect your investment in legacy applications while presenting them to your end users—customers, suppliers, and internal staff—as Web applications accessible from their Internet or LAN based workstations. Both transform the screens of a 5250 host application into Web pages that are part of a WebSphere application running on a Web server.

Both enable end users to access the pages from their Web browsers and use the pages to send and receive data from the host application. The end users might not even be aware that they are interacting with a host application. From a user perspective, everything is taking place on the Web server.

The difference between HATS/LE and iSeries Access for Web 5250 support is that HATS/LE is more tailor-able. It has a number of style selection facilities that can be invoked via a

configuration Wizard or and / or an Administrative Wizard. Thus, you can have attractive banners and styles and button or link options for Function keys. This lets you can take HATS/LE one step further in design than you can with iSeries Access for Web, and so it may very well be a more compatible player with the design of the rest of your Web site. iSeries Access for Web may appear a little more obvious as a Telnet-like emulator than the more tailor-able and more pretty-able HATS/LE.

Do I Need Full HATS?

Moreover, the IBM choices cause even additional confusion in that the Host Access Transformation Services Limited Edition is a functional subset of the full product, Host Access Transformation Services (HATS). The full HATS is a toolkit for IBM WebSphere Studio that enables you to customize thoroughly the Web applications that transform your legacy host applications and make them available on your users' workstations. You can upgrade your HATS/LE application to a full HATS application, adding further customization and abilities to produce the function and ease of use that you want.

Full HATS' rules-based transformation of host screens and support for iterative development mean you can add features, as you are ready. With HATS you can also extend 5250 and 3270 applications to Web browsers to Deliver HTML directly to the desktop. One obvious advantage of this product is the same as iSeries Access for Web as it requires zero-footprint and zero-download of emulation ware.

With full HATS, you can also exploit the security and scalability of industry-leading WebSphere Application Server applications using a HOST Publisher approach that can combine information from multiple host screens into one Web page. This latter facility is not available with iSeries Access for Web

nor is it available on the HATS/LE product. The objective of the big HATS product is that you are enabled to incorporate your own business logic and thus provide legacy applications within enterprise portals through integration with WebSphere Portal.

The bottom line is that just as with iSeries Access for Web and HATS/LE, Full HATS enables you in this way to get on the Web quickly and then it provides more tools so that you can further customize at your own pace.

Too Many HATS?

My personal commentary regarding HATS and Full HATS is that having a free subset product and a pay-for complete product comes off as a bait and switch. Moreover, it further aggravates the confusion that exists among the AS/400 faithful regarding the existence of a real IBM iSeries Web strategy.

From my point of view, two HATS gives the impression that IBM's Software Division is trying to nibble some revenue for its products from AS/400 shops. Because the Software Division gets no revenue from integrated function, its motivation logically would be to keep products separate, rather than permit IBM to integrate them into the AS/400 product as in days of yesteryear. This apparent gamesmanship from the Software Division causes confusion and damages the prospects of the AS/400 being able to be a Web player with fully integrated function for the long term.

Hopefully one day, Rochester will figure out how to pay off the Software Division so that the AS/400 can be as integrated on the Web as it is with 5250 transaction processing. Why in these days of the Web Browser as the truly Universal Client would IBM try to exact a toll from those few AS/400 constituents who dare to move their shops towards WebSphere? HATS/LE and full HATS are a manifestation of IBM not willing to continue

the integration of AS/400 where the Web is concerned. Perhaps if they hear from us on that subject, they may get their thinking clear once again.

After all, the AS/400 is IBM's integrated system and, when you pay for OS/400, you pay a hefty price for that integration. So, why have piece parts cluttering up the walkway. Just as PDM and the Programmers menu before it formed the basic development toolkit for years, iSeries shops are looking for a standard replacement set of tools. The more IBM muddies the water trying to whet your appetite for half packages and limited packages and semi functional tools and for-fee full tool sets, the more confusing it is, and the less integrated it becomes.

Get Help from IBM

In this Chapter, now that the commentary is behind us, we focus on HATS/LE V5.0. At the time that I wrote this book, however, there was little published information on the Web for HATS/LE V5. IBM has made some inroads since my last trip but it is still disjointed. I think that the reason it is not as crisp as other Rochester offerings is because it is not a Rochester product and it is not integrated into the V5R3 documentation. Hoping that you can get some good up-to-date documentation about HATS and HATS/LE, I would suggest that you try the following IBM Web site.

www.ibm.com/software/webservers/hats

There is also a HATS Information Center at

www.ibm.com/software/webservers/hats/library/version5/infocenter

Because HATS runs on other platforms, this documentation has not been tuned for the AS/400 reader in my visits. However, IBM still has an opportunity to fix this.

The natural way to get HATS/LE documentation would be for you to visit IBM's Information Center. Though there are no major manuals like you will find with other products, there are links to standard HATS and HATS/LE documentation. To get to the HATS/LE material in the iSeries Information Center, take the following links from your browser.

```
>>www.as400.ibm.com  
>>Library - Left frame  
>>iSeries Information Center  
>>North America  
>>English V5R3  
>>Connecting to iSeries  
>>iSeries Access  
>>iSeries Access for Web  
>>Install and Setup  
>>WebSphere HATS/LE
```

Take the link in the right panel to get to the HATS/LE for iSeries Web Site

Right now, the shortcut is at the following URL but if IBM enhances and integrates its documentation, and changes this, you may miss out on some new documentation by not going through the Information Center.

**[http://www-1.ibm.com/servers/
eserver/series/access/hatsle/](http://www-1.ibm.com/servers/eserver/series/access/hatsle/)**

On the right side, there are a number of other opportunities for information:

**Getting Started Manual
Read Me
ISeries Access for Web and
HATS/LE Redbook
Other good information**

As noted previously, there is no direct link from the AS/400 information center. Thus, in some ways, because HATS/LE is like an orphan product its documentation is not in the form that you would expect from Rochester. Moreover, this makes the product a little more difficult to understand prior to actually working with it.

The documentation in the IBM Web Sites describes the functions of HATS/LE and how to install and configure your HATS/LE application for WebSphere. Of course, at this part of the book, we have already done this. The HATS5LE.ear file is already installed to WebSphere. However, there are many setup options for HATS/LE. In fact, there are far more than this simple chapter can cover.

Before installing, IBM suggests that you review the HATS/LE Readme file, which contains late-breaking information. It's a good idea. As we have already demonstrated, you can install HATS/LE in fairly short time. Now that HATS/LE is installed, however, we do have some final configuring and tailoring to do in order to activate the HATS5LE.ear application

for use on your system. As you will see, we can do this in a very short time. Here are the steps you'll follow to set up HATS/LE to run from your AS/400.

Configuring HATS/LE

Unlike iSeries Access for Web, HATS/LE is not ready to go after it is installed. You must pick your options for how you want HATS/LE to pretty up your green screen panels. Therefore, prior to running a HATS/LE 5250 session to your iSeries, you must go through a tailoring process. In our examples, we take mostly defaults and we leave it up to the reader to delve further into the mysteries of HATS/LE options to give your work more than a default face.

To configure HATS/LE, you follow the easy-to-use browser-based wizard. With the Wizard, you can configure default and specific profile settings for your HATS/LE application. You can use any supported browser on your local PC to perform the final tailoring. The WAS Express application server must be running when you configure your HATS/LE application since the HATS/LE product is actually used to configure the HATS/LE run environment for your shop.

To start the HATS/LE Configuration wizard, enter the following URL in your Web browser:

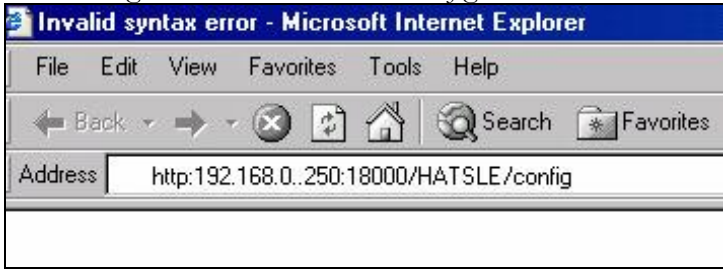
http://<servername>/HATSLE/config

http://192.168.0.250/HATSLE/config

In the above URL, <servername> can be either the name or the IP address of the WAS Express server machine where you installed HATS/LE. So, the browser window when you enter

this URL should look similar to the panel as shown in Figure 16-12

Figure 16-12 HATS/LE Configuration URL



You will be taken to a sign-on panel similar to that shown in Figure 16-13.

Figure 16-13 HATS/LE Signon Panel

Log In		Help ?
Enter your user name and password. Click OK to continue.		
Host name:	192.168.0.250	
User name:	<input type="text" value="bkelly"/>	
Password:	<input type="password" value="*****"/>	
Language:	English <input type="button" value="v"/>	
		<input type="button" value="OK"/>
IBM WebSphere HATS Limited Edition © Copyright 2003, 2004 IBM		

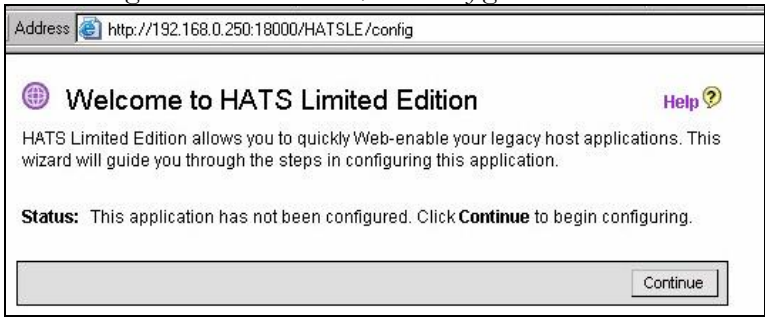
You must log in with more than just a valid user name. You must have at least *SECADM authority and you must supply a password for your profile. On this panel, you can also specify the language in which you wish to use the Configuration wizard.

As you use the wizard, feel free to click Help on any panel for more information. This is a good way to learn HATS and its facilities.

As you will see later in this section, the Configuration wizard can also be launched through the HATS Administrative Console by selecting the Configure menu item on the left-hand side as shown in Figure 16-24.

Click OK to continue from Figure 16-13. You will be taken to the Welcome panel as shown in Figure 16-14.

Figure 16-14 HATSL/LE Config Welcome Panel



When you log in to the Configuration wizard, you will see the Welcome page. This page shows the status of your HATS/LE application. If you have already configured and activated this application, there could be users connected to the application. These users will not be affected while you are reconfiguring your application. Obviously, this is our first time through HATS/LE at this point so the only status information we get is that HATS/LE has not been configured. So, to continue with our Configuration Wizard, click the **Continue** button.

Note:

If users are connected to your HATS/LE application, later when you click Finish to save your configuration setup, the

users will remain connected to the host to which they connected when they started your HATS/LE application. If you change connection settings, these users will use the new settings the next time they start your HATS/LE application. If you make changes to the template or transformation settings, connected users will see the changes when their browser is refreshed.

Caution:

This page also shows whether any other users are currently configuring this application. If you and another administrator configure your application at the same time, the administrator who saves configuration changes last will overwrite the changes saved by the other administrator.

Figure 16-15 HATS/LE Connection Settings

Address http://192.168.0.250:18000/HATSLE/config

Connection Settings Help ?

Configure your host connection settings. Click **Finish** if you are done configuring your application, or **Next** to continue to the next step. For more information, click the Help icon above.

Connection:	<input type="text" value="192.168.0.250"/>
Description:	<input type="text" value="Main Connection to iSeries"/>
Host name:	<input type="text" value="192.168.0.250"/>
Port:	<input type="text" value="23"/>
Code page:	<input type="text" value="037 United States"/>
	<input checked="" type="checkbox"/> Enable screen reverse (if applicable) <input type="checkbox"/> Enable Unicode data streaming (if applicable)
Workstation ID:	<input checked="" type="radio"/> Server assigned <input type="radio"/> Set to value (wildcards allowed): <input type="text"/> <input type="radio"/> Set from HTTP session variable: <input type="text"/> <input type="radio"/> Prompt user
Security:	<input type="checkbox"/> Enable SSL (Secure Sockets Layer) Certificate file: <input type="text"/> <input type="button" value="Upload..."/>

In the Connection Settings page as shown in Figure 16-15, you get to provide information about your host connection. Please note that the use of IP address for connection name as we have provided above, is not permitted but it is OK for the host name. When you believe you have your connection information set, click the OK button to continue. If you have specified an IP address for a name as we did, you will see a panel similar to that shown in Figure 16-16.

Tip:

The Connection Settings page displays a table listing all of your connections. In this case, however, we are just creating our first. HATS/LE allows for multiple connections giving you access to different hosts. Once the configuration has been made for the main connection and the first time configuration of the application is completed as shown in Figure 1-4, you can then configure multiple connections. After multiple connections are configured and active, you can choose which connection to use. The table shows connection names, descriptions of the connections, host names as well as what connections are active to your application.

Figure 16-16 HATS/LE Connection Name Error

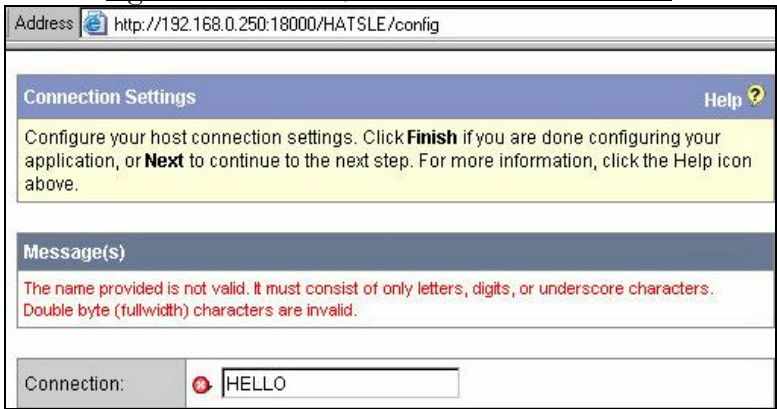


Figure 16-16 shows the error message section is you don't do everything right. In this case, the Wizard tells us that we cannot have an IP address as a name and asks us to change it. So, as you can see we replace the IP Address with a connection name of HELLO to the HELLO System. When you click the OK button this time, for your first connection (default) you will be taken to the connection settings panel, which shows all of your connections. This is given in Figure 16-17.

Figure 16-17 Connection Settings

Address <http://192.168.0.250:18000/HATSLE/config>

Connection Settings Help ?

Configure your host connection settings. Click **Finish** if you are done configuring your application, or **Next** to continue to the next step. For more information, click the Help icon above.

Select Connection	Active	Connection Name	Description	Host Name
<input checked="" type="radio"/> (default)	<input checked="" type="checkbox"/>	HELLO	Main Connection to iSeries	192.168.0.250

Enable user connection list
When the above option is disabled, the default connection in the list will be used to connect to the host, regardless of activation status.

Allow workstation ID parameter override in URL

In Figure 16-17, if you are tire kicking as we were with HATS/LE, you would make the one connection that we are creating active and set it as a default configuration so that while learning about HATS/LE, you need not be concerned about other HATS/LE connections to other systems.

If there were other connections as there may be some time in the future for your shop, there would be more than one line in the table in Figure 16-17. You would make one of the table entries (connections) active by placing a check mark in the Active box next to the desired connection in the table.

As you can see with all the buttons on this panel, there are a number of other options that you can take. You can add additional connections by selecting the Add button, or edit an existing one by selecting Edit. Additionally, you can click Remove to delete a specific connection. If you select the Add or Edit buttons, you will be asked to supply the following information:

- ✓ A name for your connection
- ✓ A description of the connection
- ✓ The hostname or IP address of the host machine
- ✓ (default value of “localhost” is pre-filled)
- ✓ The port on which your HATS/LE application will
- ✓ access the host machine
- ✓ (default value of “23” is pre-filled)
- ✓ The code page your host application uses
- ✓ How you want workstation IDs to be assigned v
- ✓ Whether you want to use Secure Sockets Layer (SSL)

You would click OK when finished or Cancel to return to the Connection Settings page.

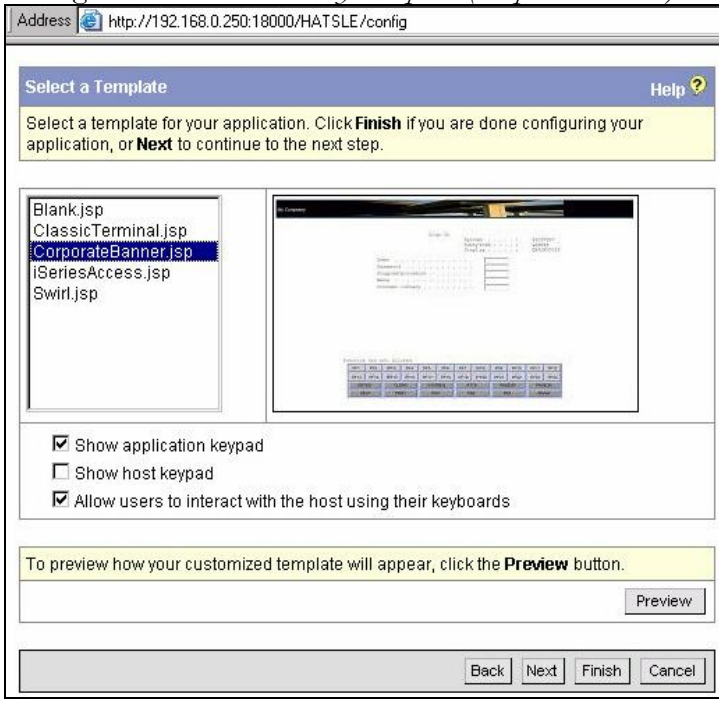
Tip:

When you have multiple connections, they can be placed in order of priority followed by descending order of importance. You can use the Move Up or Move Down buttons to perform this operation. You can make a connection your default as we did in the table by clicking the Select Connection radio button and then selecting the Default button. The default connection will then be highlighted in your connections settings table.

If you use a connection list, make sure you check the Enable user connection list check box. After you have successfully configured your application, a connection list will be added to the index page of your application. You can then select which connection to use from the list.

The connection parameters can also be edited at any time by going to the Administrative Console and selecting the Advanced panel. If you enabled the connection list by selecting the Enable user connection list check box on the Connection Settings page, you can modify how the connection list will look using the Connection List Display Editor. To launch the editor, you must be in the Administrative Console (shown in Figure 16-24 below) and click on the Advanced link on the left-hand side. Click the Edit button to launch the editor

Click Next to continue with the panel in Figure 16-18.

Figure 16-18 Pick a Pretty Template (Corporate Banner)

Using a Template to Control Your Application's Appearance

As you can see in Figure 16-18, you now have the opportunity to pick a template to make your HATS/LE panels look more attractive. This facility is not available in iSeries Access for Web. Templates enable you to control the appearance of your HATS/LE application. A template is a JSP file with an area reserved for the transformed host screen. The template can contain company logos, information, and links to other Web pages. A template also defines the background color for the area where the transformed host screen appears. Templates

contain HTML and JSP code to include some or all of the following design capabilities:

- ✓ At least one stylesheet (.css) file v
- ✓ A banner area across the top of the page
- ✓ An area for the transformed host screen
- ✓ Host and application keypads

HATS/LE supplies a few basic templates as you can see in Figure 16-18. You can use these in your projects. You can see the names of these templates on the template page of the configuration wizard as shown in Figure 16-18. When you click the name of a template, a small preview appears on the right of the panel. You can click Preview to see a larger view. As you can see in Figure 16-18, we have selected the “Corporate Banner” look and feel for our template.

Using the Keyboard and Keypads

Users frequently interact with host applications using special keys on the physical keyboard, such as F1, Attn, and Clear. There are two different ways in which the end users of your HATS/LE projects can send keystrokes to the host: The most straightforward approach is by pressing keys on the physical keyboard. If you want to enable your users to use these function keys on the keyboard, check “*Allow users to interact with the host using function keys on their keyboards.*” As we have done. The keys will then be passed directly to the host application.

Another way of denoting “keys” to the HATS/LE user is to use a keypad, which is a graphical table of HTML buttons that represent keys on the physical keyboard. You would click on the desired key in the keypad to send that host key to the host. It would be as if you pressed the key on a terminal. There are two keypads that you can add to your template: One is called the “host keypad,” with keys (such as F1, Attn, and Clear) that

represent host keys. These keys control functions on the host. The other is the “application keypad,” with keys that represent application-level functions. These keys control functions within the HATS/LE application. As you can see in this example, we have selected to use the application keypad and not the host keypad.

Note:

You can perform more advanced configuration by adding a new template, editing an existing template, or exporting a template to your local system so that you can edit it with your favorite workstation editor.

As you can see in Figure 16-18, you have the opportunity to preview the overall look and feel of the resultant HATS/LE panels before proceeding. When you are ready to move on, you would click the Next button to be able to select specific style sheets and to type text into the banner that will appear in your finished panels. This is shown in Figure 16-19.

Figure 16-19 HATS/LE Corporate Banner Setup

Address http://192.168.0.250:18000/HATSLE/config

Configure Template : CorporateBanner.jsp Help ?

Configure this template to look more like your corporate Web site. Click **Finish** if you are done configuring your application, or **Next** to continue to the next step.

Window title:	<input type="text" value="The HELLO Template"/>		
Banner text:	<input type="text" value="Welcome to the HELLO Company"/>		
Banner background:	<input type="text" value="Corporate1top.gif"/> ▼	<input type="button" value="Upload..."/>	<input type="button" value="Preview"/>
Style sheet:	<input type="text" value="monochrometheme.css"/> ▼	<input type="button" value="Upload..."/>	

To preview how your customized template will appear, click the **Preview** button.

In this example, shown in Figure 16-19, we typed: “Welcome to the HELLO Company” as our banner text and we used IBM’s defaults for background and style sheet. If you have a schooled Web staff that creates style sheets and background panels, you may choose to upload them at this point in the Wizard. Also, on this panel, you have the opportunity to preview the look and feel again before proceeding to the next panel in the Wizard. When you are ready, click on the Next button to continue with the tailoring process.

Figure 16-20 The Look of the Banner

We could not resist the temptation to hit the Preview button and so we got the panel shown in Figure 16-20. When we came back after “X”ing out of the panel in Figure 16-20, we took the Next button to get to the panel shown in Figure 16-21.

Figure 16-21 Transformation Settings for Display Panels

Address http://192.168.0.250:18000/HATSLE/config

Transformation Settings Help ?

Select how to render your transformed host screens. Click **Finish** if you are done configuring your application, or **Next** to continue to the next step.

<input checked="" type="checkbox"/> Detect function keys	Render as: <input type="radio"/> Button <input checked="" type="radio"/> Link	Settings...
<input checked="" type="checkbox"/> Detect selection lists	Render as: <input type="radio"/> Button <input type="radio"/> Button table <input checked="" type="radio"/> Link <input type="radio"/> Dropdown list <input type="radio"/> Option list	Settings...
<input checked="" type="checkbox"/> Detect tables		Settings...
<input checked="" type="checkbox"/> Detect subfiles		Settings...
<input checked="" type="checkbox"/> Detect fields	<i>Fields are always detected.</i>	Settings...

Back Next Finish Cancel

The Wizard page shown in Figure 16-21 gives you another opportunity to tailor the look and feel of the HATS/LE display for your shop. Each choice on this page tells HATS/LE how to handle various functions that appear on green screen panels, such as function keys, selection lists, tables, subfiles, etc. You have choices as to whether you would like to see these items in button form or link form or in drop-down-lists, etc. In fact, there are so many choices to know exactly what you want; it is my humble opinion that you will have to play with HATS/LE, altering these choices to see which look and feel you like best. The choices we took are shown in Figure 16-21. The next step is to click the Next button to get the summary parameters for this configuration session.

Figure 16-22 Summary of HATS/LE Design Choices

Address <http://192.168.0.250:18000/HATSLE/config>

Summary Help ?

Confirm the following settings are correct. Click **Finish** to commit your changes.

Connection Settings

Connection:	Host name:	Port:	Code page:	Workstation ID:	SSL enabled:
HELLO (default)	192.168.0.250	23	037 United States	Server assigned	No

Template

Template:	CorporateBanner.jsp
Show application keypad:	Yes
Show host keypad:	No
Keyboard support enabled:	Yes

Template Configuration

Window title: The HELLO Template, Banner text: Welcome to the HELLO Company, Banner background: Corporate1top.gif, Style sheet: monochrometherme.css

Transformation Settings

Automatically detect function keys, selection lists, tables, subfiles, fields.

Activate application

Back Finish Cancel

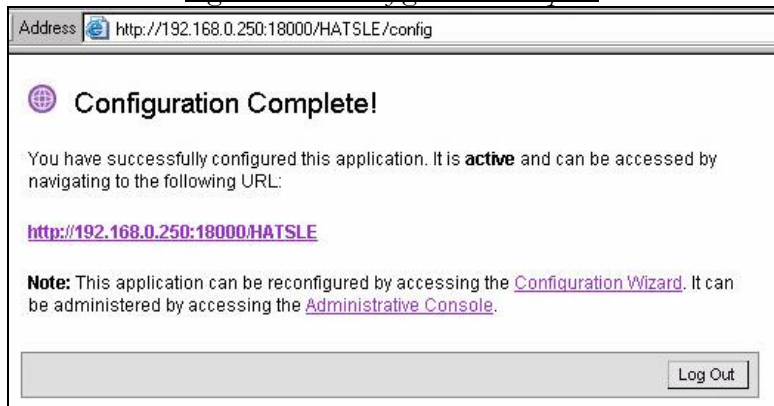
Just like all good Wizards, the HATS/LE configuration Wizard shows you a summary of your choices before it permits you to create the application. In the panel shown in Figure 16-22, as you can see you can back up to change something, proceed to completion by clicking the Finish button, or you can cancel this session because it is nothing like what you want. IN this case,

we selected the Finish button, and after a short while, we are greeted with the happy display shown in Figure 16-23.

Tip:

The first time you configure your HATS/LE application, you will see an Activate application check box. By default this box is checked as you can see at the bottom of Figure 1-11, so that your HATS/LE application will be activated when you click Finish. The Configuration wizard remembers the status of this check box and uses it when you click Finish from an earlier panel. This means that if you have left the box checked, your application will be activated when you click Finish from any panel; if you cleared the check box, your application will not be activated when you click Finish.

Figure 16-23 Configuration Complete



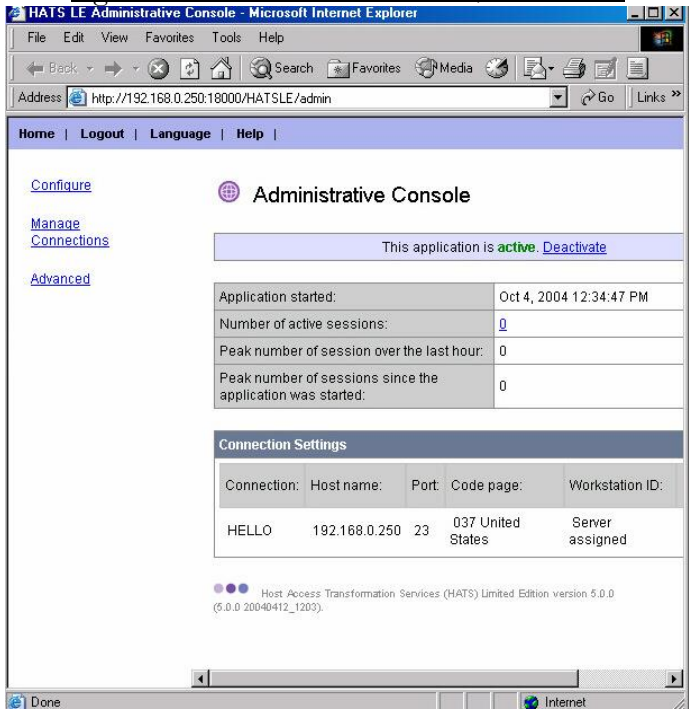
Tip:

You may want to perform iterative configurations. In other words, when you have finished configuring your HATS/LE application, the Configuration wizard will provide a link to your application. You can click this link to start the application, log in to your host system, and review how the host screens are transformed. If you want to make changes, you can restart the Configuration wizard and change the settings. Also, when you reconfigure your HATS/LE application, you might not need to revisit every panel in the Configuration wizard, so you can hit the Finish button much sooner in the process.

This panel as shown in Figure 16-23 is more than just a congratulations memo. There are three important links on this page. The first is one you should write down for sure since it is the URL that you would use to access the HATS/LE application from a browser for ad hoc AS/400 access. That, of course is the purpose of all this work. The next link is the configuration wizard, which we just completed, and the third link is for the Administrative Console used to manage HATS/LE after configuration. The link in our shop to access the Admin Console directly is

`http://192.168.0.250:18000/HATSLE/admin`

You can either access the Administrative Console via the Link from Figure 16-23, or you can type the link above. In either case, you will see a panel similar to that shown in Figure 16-24.

Figure 16-24 A Look at the HATS/LE Console

The HATS/LE Admin Console ?

When you have just one definition and one connection, HATS/LE can be fairly simple to manage. However, as your installation needs for HATS/LE increase and your mix becomes more varied, you will be glad that IBM has built a special console Web Interface for you to manage your HATS/LE connections and configurations. Figure 16-24 shows the console and the various links that you can select.

Note:

The HATS/LE Console is not the same console as the WAS Express console.

Section IV

Managing and Using IBM's Web Software for iSeries---

Apache HTTP

WAS Express

IBM Business Solutions

IBM Telephone Directory

IBM Survey Creator

WebFacing,

iSeries Access for Web

HATS/LE.

Chapter 17

Managing & Using HTTP and WAS Express Servers

The Admin GUI Is a Good Friend

The tool used for the management of the APACHE HTTP server and for the WAS Express Server is the good old Admin GUI that you just used to create your servers (instances). For those who have worked with the GUI of other WAS versions, please note that this is not the WAS Console and for that matter, it is not the HATS/LE Console.. When you install the WAS Express, the Admin GUI of the HTTP server gets upgraded to become the Admin GUI for both HTTP servers and for WAS Express servers.

Thus, the new Admin GUI can be thought of as the upgraded Web interface to the HTTP ADMIN Server. If you have ever worked with the HTTP Admin GUI prior to WAS Express, you surely will notice the major differences. Additionally, the Admin GUI is different in V5R3 from how it was in V5R2. Yet, it is similar enough that if you know one the other is easy to follow:

Overall, the WAS Express integration through the Admin GUI is a wonderful piece of work on IBM's part. It is so naturally wonderful, that from this person's view, the WAS Express and the HTTP server should be available on every AS/400. And, with V5R3 (i5/OS), it now is.

Lots of New Stuff

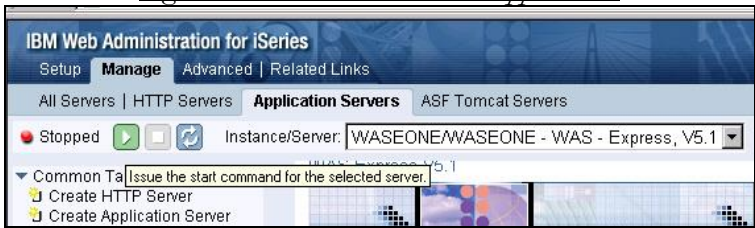
With the Admin GUI, you get to learn a new interface to an old facility (HTTP) as well as a new facility with a new interface (WAS Express). Unlike change for change's sake, however, the Admin GUI is a well-done blend for managing both WAS Express along with the Apache HTTP. It is extremely functional and reasonably intuitive. That's enough for me to say, unequivocally. I like it a lot.

Additionally, many of the functions that had to be performed by the "nasty" WAS Console in the past, are now performed using the Admin GUI itself. Overall, these changes to WAS management make the WAS Express seem more like a close sibling rather than an unwanted in-law.

For this chapter, most of the screen snaps will be from WAS 5.1 and V5R3. If you are on a different combination, the tasks are similar but not exactly the same.

Using the Manage Facilities of the Admin GUI

Figure 17-1, shows the V5.1 WASEONE WAS Express Server in a stopped state immediately after Wizard creation. You may recall that we created this server in Chapter 12. This figure also shows a blow-up of the functions available through the Admin GUI Manage tab. If you look from left to right, you will see a Welcome tab, a Setup tab, and a piece of the Manage tab. After creating the WAS and the HTTP server for you, the Admin GUI Create WAS Wizard immediately switches you to the Manage Tab.

Figure 17-1 WASEONE in Stopped State

If we were to dissect the capabilities of the Admin GUI regarding the management of the WAS Express environment we would find the list below reasonably complete:

The categories of functions performed / managed by the Admin GUI are as follows:

Tasks and Wizards
Server Properties
Applications
Resource Configuration
Problem Determination
Tools

Select Your Server

When you see the Manage Tab in the Admin GUI as shown in Figure 17-1 above, you can select your server by pulling down the GUI Window (List box) at the top right. The default server name that you get in the window, after you use the Create Wizard, is the name of the WAS server that you just created. In this case, we've got *WASEONE* and the Wizard adds the terms "*- WAS - Express V5.1*" to the name so that we know it is a WAS vs. an HTTP "instance." We also know that is at the 5.1 level and not 5.0 or 5.2. It displays the complete name in the pull down List Box window.

The Manage Tab facilities of GUI Admin also gives you the opportunity to switch the iSeries WAS Express server name in the list box window. This comes in handy when you want to manage a different server than is currently displaying. However, the most common reason would be to manage any server long after all servers are created. The GUI Admin Manage Tab gives you the flexibility of selecting the server that you want to manage upon entering the tool and at any point after completing a function.

Select All Servers

Besides being able to pick a particular HTTP or WAS Server to manage, you can also pull down an "All servers" option, which will give you a nice display that shows many if not all of the servers you have created segregated by server type tabs (HTTP, Application, Tomcat). It will show you them all, but not all at the same time. Depending on how many servers you have, you may have to scroll and as you can see on the right side of the panel in Figure 17-2, you do have to change Tabs to move from server type to server type. This new set of Tabs was added sometime before V5R3 in a PTF update. If you are accustomed to seeing all the servers of all kinds in this display, you have yet to apply these PTFs.

Figure 17-2 Navigating to WASEONE Express Server

IBM Web Administration for iSeries

Setup **Manage** Advanced | Related Links

All Servers HTTP Servers | Application Servers | ASF Tomcat Servers

Common Tasks and Wizards

- Create HTTP Server
- Create Application Server
- Migrate Original to Apache

Manage All Servers ?

All HTTP Servers All Application Servers All ASF Tomcat Servers

Data current as of 03:43:07 PM UTC on 08/18/2004

	Server ▲	Version	Status	
<input checked="" type="radio"/>	iwa51exp.iwa51exp	WAS - Express, V5.1	Running	*:20
<input type="radio"/>	WASEONE.WASEONE	WAS - Express, V5.1	Running	*:11

Figure 17-3 Close-up of WASEONE Server Status

Manage All Servers ?

All HTTP Servers All Application Servers All ASF Tomcat Servers

Data current as of 03:50:48 PM UTC on 08/18/2004

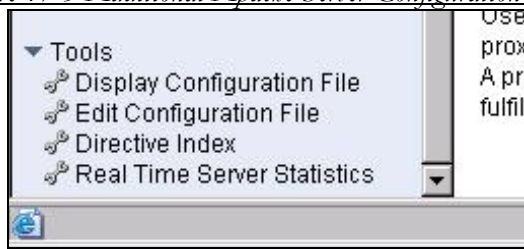
	Server ▲	Version	Status	Address:Port	Associated WAS Instance
<input type="radio"/>	ADMIN	Apache	Running	*:2001	None
<input type="radio"/>	IWA51EXP	Apache	Stopped	*:2042	iwa51exp, V5.1 - Express
<input checked="" type="radio"/>	WASEONE	Apache	Running	*:1111	WASEONE, V5.1 - Express

Figure 17-4 Managing Companion Apache HTTP Server

Working With Apache

When you use the Window as shown in Figure 17-1 and you click the HTTP server tab, the all servers view is shown in Figure 17-3. You can change the item in Window to *WASEONE Apache* and then you will be looking just at the *WASEONE Server* but in greater detail as shown in Figure 17-4. This gets you to a series of items in the left frame that look very much like those from the old Admin GUI for HTTP only. If you have set up an Apache Web Server with the GUI in the past, the panel in Figure 17-4 should look very familiar to you.

You would use the panel in Figure 17-4 to maintain the configuration of your Apache HTTP Server. However, if you want to see all the options, and perhaps even look at the Apache Configuration that was built during this process, you would have to scroll down a few lines. When you scroll, you would see the additional options as shown in Figure 17-5.

Figure 17-5 Additional Apache Server Configuration Tools

Apache Configuration Directives

Surely, all the technical-at-heart want to see this configuration right now. So, we will not delay. To do this, from the panel in Figure 17-5, select the Display Configuration File option. This will take you to a panel similar to that shown in Figure 17-6.

Figure 17-6 Apache Configuration

```

Selected file: /www/waseone/conf/httpd.conf

1 LoadModule ibm_jdap_module /QSYS.LIB/QHTTPSVRLIB/QZSRVLDAP.SRVPGM
2 WebSpherePluginConfig /QIBM/UserData/WebASE51/ASE/WASEONE/config/cells/plugin-cfg.xml
3 LoadModule ibm_app_server_http_module /QSYS.LIB/QASE51.LIB/QSVTIHSAH.SRVPGM
4 Listen *:1111
5 DocumentRoot /www/waseone/htdocs
6 Options -ExecCGI -FollowSymLinks -SymLinksIfOwnerMatch -Includes -IncludesNoExec -Indexes -MultiViews
7 ServerRoot /www/waseone
8 LogFormat "%(Cookie)n %r" "%t" cookie
9 LogFormat "%(User-agent)i" agent
10 LogFormat "%(Referer)i -> %U" referer
11 LogFormat "%h %l %u %t %r" "%s %b" common
12 LogFormat "%h %l %u %t %r" "%s %b %i" "(Referer)i" "(User-Agent)i" combined
13 CustomLog logs/access_log combined
14 SetEnvIf "User-Agent" "JDK/1.0" force-response-1.0
15 SetEnvIf "User-Agent" "Java/1.0" force-response-1.0
16 SetEnvIf "User-Agent" "RealPlayer 4.0" force-response-1.0
17 SetEnvIf "User-Agent" "MSIE 4.0b2;" nokeepalive
18 SetEnvIf "User-Agent" "MSIE 4.0b2;" force-response-1.0
19 SetEnvIf "User-Agent" "Mozilla/2" nokeepalive
20 <Directory /www/waseone/htdocs>
21     Order Allow,Deny
22     Allow From all
23 </Directory>
24 <Directory />
25     Order Deny,Allow
26     Deny From all
27 </Directory>
28 Include /QIBM/UserData/WebASE51/ASE/WASEONE/ibs/WASEONE/apache20.inc
29 # HTTP server (powered by Apache) configuration

```

Is That All There Is?

If I were reading your mind right now, I would not expect to be hearing, “Is that all there is?” Actually, that is pretty OK. Unlike the CERN HTTP configurator, which creates mostly commented directives, this Admin GUI gives you 29 live and hot directives that permit this Apache HTTP “instance” to work immediately with your new WAS Express “instance.”

Moreover, there are enablers for Java (lines 14 & 15), and WebSphere (lines 1 & 2). Additionally, there is a specific enabler on line 20 for all documents that are referenced from the *htdocs* directory within the “WASEONE” directory within the “www” directory. This is where you can load up those static Web pages such as your home page and other Web pages. IBM sticks a sample home page in a directory within each HTTP instance.

The automatic creation of a fully functional Apache HTTP server is a very nice feature of the WAS build process. For CERN (Original HTTP) server users who have gotten used to a completely different directive mechanism, having this facility removes Apache education from the pre-installation activities. You can use Apache with WAS Express without knowing anything about it.

Throw Away CERN

Though I hate to give up the original (CERN) directives, since they are substantially easier to me and I know them well, having a fully functional Apache instance to examine more than compensates for anything lost in this regard. It is of immense assistance to any implementer trying to make the transition from the Original to the Apache server.

Apache is a prerequisite task for anybody choosing to replace a WAS 3.5 server or TomCat server with WAS Express, as noted above and repeated here for emphasis. However, if you choose never to go to i5/OS V5.3, you can leave your Original CERN server alive and just change the URLs of your dynamic files and servers to point to your new WAS Express. Let the static pages be if you don’t have to change them. When you have time, work the whole shop into Apache. Of

course I would recommend that you move to i5/OS V5.3 rather than mess with such possibilities.

CERN Migration

If you don't like that idea of having old servers hanging around, for your own reasons, you can convert your existing Original (CERN) instance to an Apache "instance." The Admin GUI offers a migration aid for this conversion. It has a facility that enables you to create a new Apache HTTP server based on the directives of an existing CERN original server. When you choose this option, you may also choose to bind your WAS Express to the converted HTTP server.

Thus, when you build your WAS Express server to work with this migrated HTTP server, you would not necessarily build the new HTTP server at the same time. Then again, you might go ahead and build it to capture the specific directives and default host entries necessary for the two servers to commingle.

Regardless of your approach, these tools, plus the IBM documentation within the product and the documentation available on IBM's Web site and other Internet sites, will help you better understand the Apache server directives so that you can begin to use the WAS express in a far shorter amount of time than otherwise would be needed.

Save Cutover Work

The overall idea is that even though IBM provides htdocs and directories within the WAS structure depositories for your new static Web documents, if you are not on i5/OS V5R3, you may not want to move your existing pages into this new directory structure. Why should you complicate things more than necessary? If you can get by without an HTTP conversion, that's the ideal. But, you eventually will have to make the transition to Apache for your static Web documents.

You might even consider not converting your original HTTP server at all unless you are currently using it for WAS 3.5 servlet serving. Just move to Apache. Again, if you use an original HTTP server for static Web pages, you can still use it for the same purpose until you move to i5/OS V5R3. You can put your links for dynamic servlet serving

within the existing CERN directives documents and use your new WAS and Apache HTTP server pair for dynamic servlet serving.

If you have few pages on your current CERN server, and you would be pleased to eliminate the original server instance, you can modify your Web documents with links as necessary after you set up your WAS Express server. You can also create directives in your Apache server that enable the existing links in your documents to function as they always have – once you shut down your old CERN HTTP instance. Overall, you accomplish the conversion in place mission by adding Apache Directory directives that “Allow” vs. “Deny” access to these existing directories.

Deleting a WAS Server

Now that you have an idea of how to use the tools to manage WAS Express Servers and Apache HTTP servers, it would help to know how to do a few other things with the Manage Tab. For example, what if you want to delete a WAS Server or an HTTP Server? It’s just as easy. However, you won’t find the option on any of the major Admin GUI menus. There is not one option to delete a server of any kind.

So, how would you delete a WAS? Start by taking a look at the WASEONE Admin GUI as shown in Figure 17-7. Here, you can see a hint of what is needed. Instead of navigating away from the All servers item as we did to get to the panel in Figure 17-2 to reach the WASEONE server, this time select the `iwa51exp` server by finding its name as presented in the panel shown in Figure 17-7. You can see all the application servers since the All Application servers Tab is selected.

Figure 17-7 Manage All Servers (Delete WASEONE)

Manage All Servers ?

All HTTP Servers **All Application Servers** **All ASF Tomcat Servers**

Data current as of 04:13:11 PM UTC on 08/18/2004

	Server ▲	Version	Status	Address:Port
<input checked="" type="radio"/>	iwa51exp/iwa51exp	WAS - Express, V5.1	● Running	*:2043,2048,2052,2053,2054,2055
<input type="radio"/>	WASEONE/WASEONE	WAS - Express, V5.1	● Running	*:1112,1117,1121,1122,1123,1124

From this menu, as shown in Figure 17-7, you can start, stop or delete any of the applications servers. By changing the Tab to HTTP, you can start, stop, or delete any of the HTTP servers. So also for Tomcat if you have chosen to use it. This is indeed a handy panel.

In the case shown in Figure 17-7, notice that the servers are running. In order to delete either one, you must first select it. `iwa51exp` is selected in the example. After selecting it, you would have to stop it. When it is fully stopped, you would then have the ability to delete it. Only when it is stopped would the delete button be enabled.

WAS Express Factors for Deletion

There are a few things to notice in Figure 17-7 that enable you to perform the Delete Function:

1. The `iwa51exp` Express V5 server has been created.
2. The `iwa51ex` Express V5 server is selected.
3. The `iwa51ex` Express V5 server is in a stopped condition.

Note that the radial button for the `iwa51ex` Express server in the panel shown in Figure 17-7 is already selected. If you were to click the DELETE button after stopping this server, the delete process would begin and you would see a panel noting that the `iwa51exp` server was deleting.

Since we have been working with WASEONE, to get back to be able to delete this server, we would click the radial button by it. The delete panel sequence for WASEONE WAS Express server is shown below in Figures 17-8, 17-9, 17-10. Note first in the panel in Figure 17-8 that the WASEONE server is stopped.

Figure 17-8 Delete Server First Panel

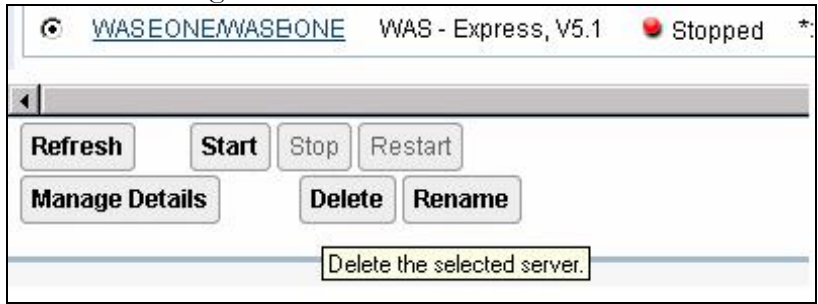


Figure 17-9 Delete Server Second panel

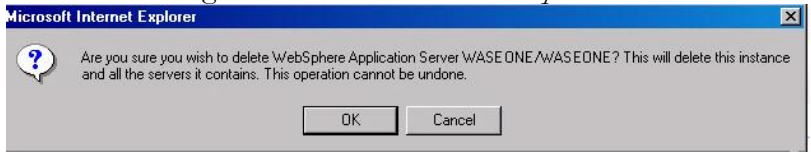


Figure 17-10 Delete Server Third Panel



Check It Out on the AS/400

It is always a good idea as you are learning the WAS Express to revert back to your familiar green screen interface and to perform a WRKACTJOB from the command line. If you quickly navigate to the QASE5 (V5.0 WAS) or the WASE51 (V5.1 WAS) subsystem, you would see the WASEONE server “instance” running and burning a bit of CPU as it is being deleted. If you continue to roll, to the HTTP server subsystem, you would see the WASEONE Apache instance with at least three jobs. One of the jobs would be burning CPU cycles as it is the managing job for the Admin GUI that is overseeing the deletion of the WAS Express server. That’s it for the Delete Function.

“A Little Bird Dun Tole Me!”

For now, let’s pretend we did not delete the WAS Express server named WASEONE or even the iwa51exp server. Or, if you are not good at pretending, assume that we did delete them and then we quickly rebuilt them exactly as they were. Thus, the servers would be in the “stopped” status.

Starting a WAS Express Instance

The next step would be to start the WASEONE “instance.” Depending on the view that you are in, you can start the WAS Express in one of two ways. From the Manage WASEONE WAS Express V5 view, just press the START button (Green & White) to start the WAS Express server. The other approach is to select the All Servers view as in Figure 17-7. Then click the radial button for WASEONE and click the Start Button at the bottom of the panel. Either approach will work. But, please not so quickly!

A Helpful Tip

Yes, starting the WAS server should be the very next thing to do. However, until IBM fixes a little bug in WAS Express, there’s one

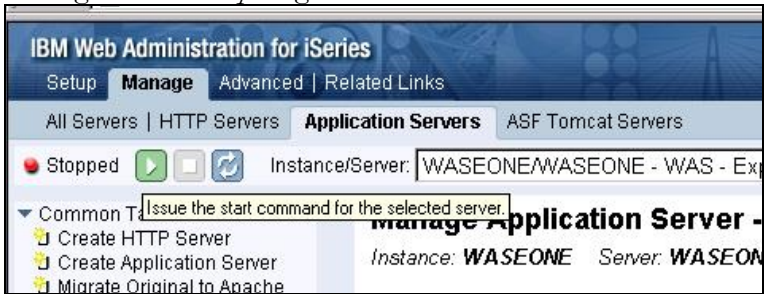
additional step required. Until then, you must do something that I will call activating the QEJBSVR user profile. Without performing this innocuous yet important activity, your efforts may be in vain. I discovered this after three or four days of Support Line activity. Later, after solving it through Support Line, I noticed later that the iSeries Network had an alert on it. Needless to say, I was just glad to get it solved. I would say that you could expect a few more bugs than this before you are live. But, IBM will get them fixed for you.

Here Is the Workaround:

Using whatever method you find acceptable, change the password for the QEJBSVR user profile to something that you will remember. Then sign on using green screen with the profile and password. Then, sign off and change the password back to *NONE. This process activates the profile for use with WAS Express.

You can now press the little green and white Start Button as shown in the panel in Figure 17-11 or you can start the WAS Express WASEONE server using the All servers panel.

Figure 17-11 Preparing to Start WASEONE WAS Server



While the WAS Express server is starting, if you use the first method, you will notice that your management panel changes to look similar to the panel as shown in Figure 17-12

Figure 17-12 WAS Express Starting Up



The key item on this panel is the information by the yellow button – *Starting*. If you switch over to a WRKACTJOB panel, you will see that there is a lot of CPU being expended to get the WAS Express server up and running. A panel similar to what you will see is shown in Figure 17-13.

Figure 17-13 QASE51 Subsystem WASEONE Starting

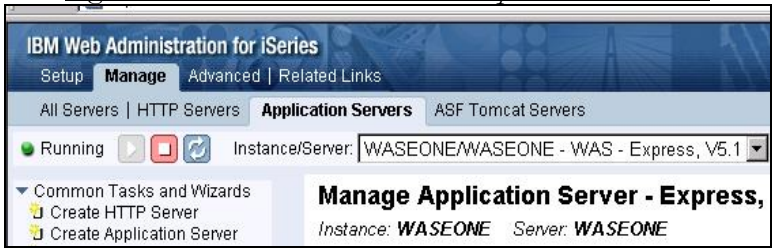
Opt	Subsystem/Job	User	Type	CPU %	Function	Status
	Work with Active Jobs				HELLO	
					07/08/04 00:55:43	
	CPU %:	89.0	Elapsed time:	00:00:01	Active jobs:	173
	Type options, press Enter.					
	2=Change	3=Hold	4=End	5=Work with	6=Release	7=Display Msg
	8=Work with spooled files		13=Disconnect	...		
	WASEONE	QEJBSVR	BCH	82.9	PGM-QASESTRSVR	RUN
	QBATCH	QSYS	SBS	.0		DEQW
	QCMN	QSYS	SBS	.0		DEQW
	QCTL	QSYS	SBS	.0		DEQW
	QSYSSCD	QPGMR	BCH	.0	PGM-QEZSCNEP	EVTW
	QHTTPSVR	QSYS	SBS	.0		DEQW
	ADMIN	QTMHHTTP	BCH	.0	PGM-QZHBHTTP	SIGW
	ADMIN	QTMHHTTP	BCI	.0	PGM-QZSRLOG	SIGW
	Parameters or command					
	===>					
	F3=Exit	F5=Refresh	F7=Find	F10=Restart statistics		
	F11=Display elapsed data	F12=Cancel	F23=More options F24=More keys			

Startup Burns CPU

Eventually, the CPU use will die down and the WASEONE WAS Express Application Server will be started. At that time, after you hit the Refresh button to the right of the word “Starting” in Figure 7-12,

“Starting” eventually changes to “Running” and the Admin GUI looks like the panel as shown in Figure 17-14.

Figure 17-14 WASEONE WAS Express Server Started



Where is WASEONE?

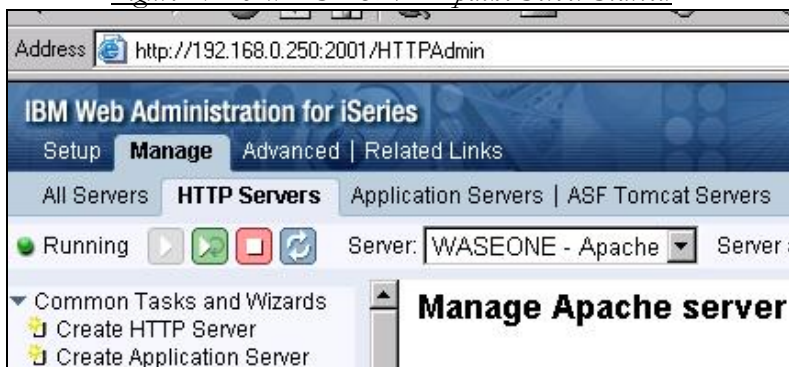
If you look hard at Figure 17-13, you will wonder why there is no WASEONE under the HTTP Server subsystem. It is true that no WAS can serve anything on the Web unless its companion HTTP server is available and running. The companion WASEONE HTTP server would be on the next page if you were to scroll the WRKACTJOB panel and if the Apache server were running. But, it actually isn't. That's because you have not started it. It does not start when the WAS Express starts. You need to start it separately either through the Admin GUI or via a trusty iSeries command line. There is an option, however, in the HTTP server to start the companion WAS server when you start it.

Starting WASEONE Apache

For now, let's start it with the Admin GUI. Just take your Admin GUI as shown in Figure 17-4, and pull down the Server List Box until you see the WASEONE Apache Server as shown in Figure 17-4. Select it and then you will be taken to a panel similar to that shown in Figure 17-15.

Figure 17-15 Starting Apache HTTP Server for WASEONE

Notice in the panel in Figure 17-15, that the WASEONE Apache server is stopped. From the panel in Figure 17-15, to start this server, just click on the green (start) button at the top of the menu item list in the left frame right next to the word “Stopped.” When you click on the HTTP server start button, things happen quickly. Without even having to hit Refresh, the Admin GUI will come back to you in a brief instant with the good news as shown in Figure 17-16.

Figure 17-16 WASEONE Apache Server Started

Running the Snoop Servlet

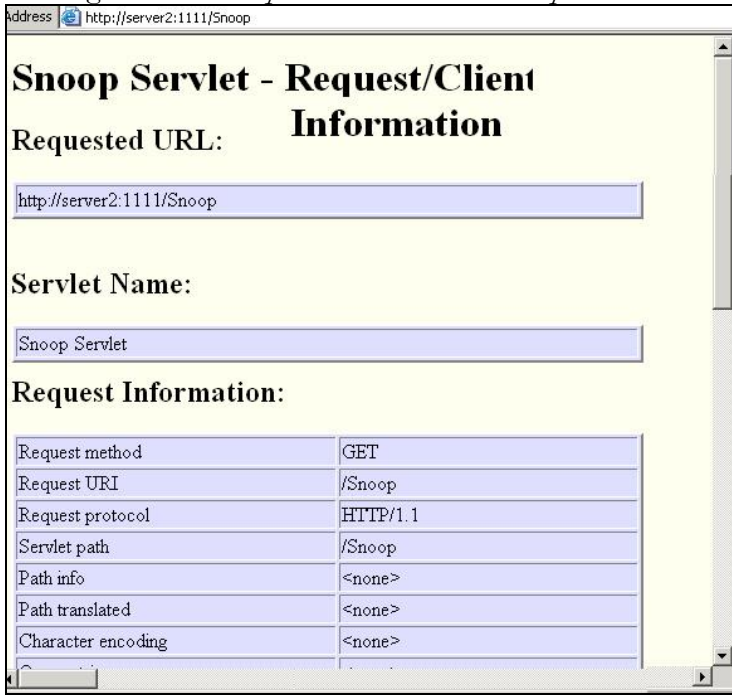
Now, the WASEONE Server is ready for business. Both the WASEONE HTTP server and the WASEONE Express server are ready for action. Since they are ready for business, let's give them some business. Since you already installed the sample applications, including Snoop, from IBM for this WAS "instance," you have a nice way of testing to see whether this new Apache HTTP server and the WAS Express application server have been built properly and are ready to work together effectively.

From our work in Chapter 12, you may recall that the URL for accessing the Snoop test application in WASEONE was given as the following:

`http://SERVER2:1111/snoop`

`http://192.168.0.250:1111/snoop`

Assuming that you have a host entry on your PC for SERVER2 that points to your AS/400, the first URL will work. Substitute your IP address in the URL on the second line and you can also run snoop from WASEONE. Just place this in the URL area of your browser and launch the URL. The servlet Snoop should run fine for you and it will produce a multi-page output set that begins with the page shown in Figure 17-17.

Figure 17-17 Snoop Works With WAS Express V5.1

Snoop Works

Of course, Snoop means nothing other than the fact that your installation and configuration of HTTP and WAS Express have been successful. Then again, that is pretty good for us so far since that is our objective. When Snoop comes up clean as in this picture, it is really time to declare victory and have a cocktail or a nice birch beer... even if no birch is available.

So Snoop matters in that after you are assured that it works, the other applications that you care about can be given an opportunity to run in this server "instance." In many ways, Snoop is like a food tester. When it runs, your servers are safe. When Snoop works with V3.5, there is ample reason for joy, because it probably took the

implementer several weeks or longer to get to that point. With WAS Express, however, you should have Snoop working in a day or so.

The Technical Look Inside

Coming up you will learn how to set up the IBM Telephone Directory application and the Survey Creator. Before we move there, however, let's look at what the WASEONE Express server job looks like from an AS/400 / iSeries point of view. To do this, let's start by working with the active jobs by typing the following command on a command line:

WRKACTJOB

Press the ENTER key and you will see a panel similar to that shown in Figure 17-18

Figure 17-18 WRKACTJOB-- WASEONE Express Server

Work with Active Jobs						HELLO2
CPU %:	.0	Elapsed time:	00:00:00	Active jobs:	178	05/01/03 10:41:23
Type options, press Enter.						
2=Change		3=Hold		4=End		5=Work with
8=Work with spooled files		13=Disconnect		6=Release		7=Display Msg
Opt	Subsystem/Job	User	Type	CPU %	Function	Status
---	AUTOINVC	QSYS	SBS	.0		DEQW
---	DBATCH	QSYS	SBS	.0		DEQW
---	DINVC	QSYS	SBS	.0		DEQW
---	QASE51	QSYS	SBS	.0		DEQW
<u>5</u>	WASEONE	QEJBSVR	BCH	.0	PGM-QASESTRSVR	JVAW
	TELDIR	QEJBSVR	BCH	.0	PGM-QASESTRSVR	JVAW
---	QBATCH	QSYS	SBS	.0		DEQW
---	QCMN	QSYS	SBS	.0		DEQW
---	QCTL	QSYS	SBS	.0		DEQW
						More...
Parameters or command						
====>						
F3=Exit	F5=Refresh	F7=Find	F10=Restart statistics			
F11=Display elapsed data	F12=Cancel	F23=More options	F24=More keys			

Looking at the Jobs

Notice that there is another WAS Express server in the panel. This is the TELDIR server that you built in Chapter 13. All WAS Express servers operate under the control of either the QASE5 (WAS 5.0 Express) subsystem or the QASE51 (WAS 5.1 Express) subsystem.

To see the WASEONE Express Server Job in detail, place a 5 next to it as shown in Figure 17-18 and press ENTER.

Display Job Log

From here, you will see a panel with the various options for the job. Select option 10 by typing it on the command line. This option is to Display Job Log, if active or on job queue. When the Display Job Log panel appears, press F10 to see the detail.

Figure 17-19 Display WASEONE Job Log

```

Display All Messages
Job ...: WASEONE      User ...:  QEJBSVR      System:  HELLO2
Number ...:          :          :          :  009710

>> CALL PGM(QASE51/QASESTRSVR) PARM('-instance' '/QIBM/UserData/WebASE51/AS
WASEONE' '-server' 'WASEONE')
Server starting with user profile QEJBSVR and JDK 1.4.2.
WebSphere application server WASEONE ready.

Press Enter to continue._____ Bottom

F3=Exit  F5=Refresh  F12=Cancel  F17=Top  F18=Bottom

```

The next panel should have the startup call for the WASEONE server and not much else, unless there are some errors that you might need to investigate. The panel should look like that shown in Figure 17-19.

Is the Joblog OK?

If your WASEONE server does not come up shortly after installation, then, as noted previously, you may need to activate the QEJBSVR user profile. To do this, you merely sign on to it and then quickly sign off. To do this, you need to change the password to something you can remember, and then change it back to *NONE when you have activated it.

Server Ready for Business

Notice in the job log show in Figure 17-19 that the QAESTRSVR program is being called from the QASE51 library. As you can see the server instance to be called is WASEONE. QEJBSVR is shown as the user profile for the instance. The last entry in the log shows that the WASEONE Express server is ready for business

Checking Out The IFS Structure

As in all prior WAS versions, there is a ton of information about the WAS kept in the IFS directories. Though you don't have to become an expert in how the WAS is structured on your iSeries IFS, it does help to have a clue. This next set of panels shows the navigation through the green screen command panels to get to the directories in which the new WASEONE server information is stored.

Begin this process by typing the following command:

WRKLNK

The Work with Link (WRKLNK) command is the equivalent of a DIR command on a Bill Gates computer. It shows the directory path and it permits a number of commands to be executed against items that are stored in various directories and subdirectories.

The IFS Root Directory

The first WRKLNK panel displays the caption “Work with Object Links.” Depending on some configuration parameters on your iSeries, you will more than likely be positioned to the root directory “/” of the iSeries after issuing the WRKLNK command. If you are not, to get there, type in the change current directory command as follows:

```
CD DIR (/)
```

After you issue the CD command, perform another WRKLNK and you will see a panel that resembles the top part of the composite we built in Figure 17-20.

Figure 17-20 Navigating the IFS

Work with Object Links			
Directory	/		Root
<u>5</u>	QIBM	DIR	
<hr/>			
Directory	/QIBM		QIBM
<u>5</u>	UserData	DIR	
<hr/>			
Directory	/QIBM/UserData		UserData
<u>5</u>	WebASE51	DIR	
<hr/>			
Directory	/QIBM/UserData/WebASE51		WebASE51
<u>5</u>	ASE	DIR	

Moving to the WAS Express Directory

Take note that in the composite panel in Figure 17-20 that the first line tells you what directory you are in and the next line (with the “5” display option) says to open the directory and list its contents. We spare you the work of analyzing all of the panels and show how you might choose to navigate more deeply into the IFS directory structure of the WASEONE server instance.

In the last “5” option shown in Figure 17-15, you can see that the path stands at /QIBM/UserData/WebASE, and we are entering the ASE directory. When we get into the ASE directory, the path becomes the following:

/QIBM/UserData/WebASE51/ASE

The full directory listing for WebASE51 shown in Figure 17-21

Figure 7-21 Checking Out the ASE Subdirectory

```

Work with Object Links

Directory . . . . : /QIBM/UserData/WebASE51

Type options, press Enter.
 2=Edit  3=Copy  4=Remove  5=Display  7=Rename  8=Display a
 11=Change current directory ...

Opt  Object link          Type  Attribute  Text
--  -
 5   ASE                  DIR

```

Bottom

```

Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position
F22=Display entire field          F23=More options

```

ASE With and Without WASEONE

If we had yet to create the WASEONE instance, the ASE directory listing would show as in Figure 17-22. By taking a 5 as in Figure 17-21, we get to look at the contents of ASE as shown in Figure 17-22. Notice that WASEONE is not there.

Figure 17-22 ASE Without WASEONE

```

Work with Object Links

Directory . . . . : /QIBM/UserData/WebASE51/ASE

Type options, press Enter.
 2=Edit  3=Copy  4=Remove  5=Display  7=Rename  8=Display attributes
 11=Change current directory ...

Opt  Object link          Type  Attribute  Text
--  -
   iwa51exp              DIR

```

Bottom

```

Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position to
F22=Display entire field          F23=More options

```

However, once we create the WASEONE Express instance, it would appear in the list of instances as shown in Figure 17-23.

Figure 17-23 ASE With WASEONE Built

```

Work with Object Links

Directory . . . . : /QIBM/UserData/WebASE51/ASE

Type options, press Enter.
 2=Edit   3=Copy   4=Remove  5=Display  7=Rename  8=Display attributes
 11=Change current directory ...

Opt  Object link      Type   Attribute  Text
---  -
 5   WASEONE         DIR    DIR

```

Bottom

```

Parameters or command
===>
F3=Exit  F4=Prompt F5=Refresh F9=Retrieve  F12=Cancel  F17=Position to
F22=Display entire field      F23=More options

```

The Inside Structure of a WAS Express Instance

In the panel shown in Figure 17-23, take a “5” (display) option on the WASEONE IFS directory. This will open the WASEONE “instance” directory and extend the path to the following:

/QIBM/UserData/WebASE51/ASE5/WASEONE

Figure 17-24 Subdirectories for a WAS Express Instance

```

Work with Object Links

Directory . . . . : /QIBM/UserData/WebASE51/ASE5/WASEONE

Type options, press Enter.
  2=Edit  3=Copy  4=Remove  5=Display  7=Rename  8=Display attributes
  11=Change current directory ...

Opt  Object link          Type   Attribute  Text
-----
    bin                   DIR
    config                DIR
    etc                   DIR
    installableApps      DIR
  5  installedApps      DIR
    lib                   DIR
    logs                  DIR
    properties            DIR
    temp                  DIR
                                     More...
    tranlog               DIR
    wstemp                DIR

Parameters or command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position to
F22=Display entire field  F23=More options

```

WAS Express Instance Sub Directories

Notice in Figure 17-24 that there is a “more...” command near the bottom on the right side of the screen. I tricked you. This panel is actually a conglomerate of two screens. The WASEONE subdirectories did not all fit so I “mushed” them together using my poetic license.

Tip:

You may note that directly above the `InstalledApps` directory is another directory named `InstallableApps`. When we installed HATS/LE, you may recall that we copied the `HATS5LE.ear` file to a directory called `/INCOMING` to make it easier for us. The natural WAS method is to go to the IFS `InstallableApps` directory and deposit the “ear” file. When we performed the install in Section II of this book, we could not do this because the WAS instance had not been created and thus, there was no structure for the WAS instance at that time.

Moving along, let's take a peak at what installed applications we have within the WASEONE instance by taking the 5 option from the panel shown in Figure 17-24.

Figure 17-25 Installed Appls Directory of WASEONE

```

Work with Object Links

Directory . . . . . : /QIBM/UserData/WebASE/ASE5/WASEONE/installedApps

Type options, press Enter.
  2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
  11=Change current directory ...

Opt  Object link          Type      Attribute  Text
  5_  HELLO2_WASEONE       DIR

Parameters or command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position to
F22=Display entire field  F23=More options
  
```

Notice the directory name HELLO2_WASEONE in Figure 17-25. As part of the WAS Express structure, the TCP/IP system name is appended to the WAS Express server name to form a directory container for the “instance” within the InstalledApps directory.

From Figure 17-25, pick this directory entry and place a 5 by it to open it. Now, the ever-expanding IFS path appears as follows:

**/QIBM/UserData/WebASE51/ASE/WASEONE
E/installedApps/HELLO2_WASEONE**

Figure 17-26 WASEONE Applications – Prior to WebFacing

```

Work with Object Links

Directory . . . . . : /QIBM/UserData/WebASE51/ASE/WASEONE/installedApps/SE >

Type options, press Enter.
  2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
  11=Change current directory ...
  
```

```

Opt   Object link           Type   Attribute   Text
      ExpressSamples.ear     DIR
      IBM Survey Creator >  DIR
      IBM Telephone Dire >  DIR
      IBM Welcome Page v >  DIR

Parameters or command
====>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel   F17=Position to
F22=Display entire field   F23=More options

```

Bottom

This directory shows the installed applications for this “instance.” Figure 17-26 shows what the SERVER2_WASEONE directory looks like before any user written application is installed. Notice that just the IBM Business Apps and the Sample apps are in the picture.

Figure 17-27 WebFaced App in Installed Applications Directory

```

Work with Object Links
Directory . . . . : /QIBM/UserData/WebASE51/ASE/WASEONE/installedApps/SE >
Type options, press Enter.
  2=Edit  3=Copy  4=Remove  5=Display  7=Rename  8=Display attributes
  11=Change current directory ...

Opt   Object link           Type   Attribute   Text
      ExpressSamples.ear     DIR
      IBM Survey Creator >  DIR
      IBM Telephone Dire >  DIR
      IBM Welcome Page v >  DIR
 5   EELLOAR001ear.ear >  DIR

Parameters or command
====>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel   F17=Position to
F22=Display entire field   F23=More options

```

Bottom

Figure 17-27 shows the directory after an application named HELLOAR001ear.ear was installed. The application was created using the WebFacing tool.

In WAS 5.0, these figures (17-26, 17-27) would show a link to the directory for the WAS Express Administration Console (adminconsole.ear). The Console application is installed by IBM in every WAS instance in V5.0 Express, one per instance. With V5.1 Express, the console is installed just in the /QIBM/ProdData directory and not in every instance.

By going into the EAR file for the WebFaced application we will be installing in Chapters 20 and on, the directory path we are examining now looks as follows:

```
/QIBM/UserData/WebASE/ASE5/WASEONE  
/installedApps/HELLO2_WASEONE/HELL  
OAR001ear.ear/
```

The WRKLNK looks similar to the panel as shown in Figure 17-28.

Figure 17-28 HELLOAR001 App Objects and Directories

```

Work with Object Links

Directory . . . . : /QIBM/UserData/WebASE/ASE5/WASEONE/installedApps/HEL >

Type options, press Enter.
 2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
11=Change current directory ...

Opt  Object link      Type   Attribute  Text
---  ---
---  index.html        STMF
---  logon.html        STMF
    styles           DIR
    theme            DIR
    ClientScript     DIR
    META-INF         DIR
    RecordJSPs       DIR
    UIMHelp          DIR

Parameters or command
====>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel   F17=Position to
F22=Display entire field   F23=More options

```

Figure 17-28 shows the files and directories within the HELLOAR001 Application Directory. At this point in our work, we have not yet shown how you create the HELLOAR001 application in the WASEONE instance. That's OK; we'll get to it in Chapter 20. It is not on the agenda until after we complete the setup and use of the IBM Business Solutions projects in Chapters 18 and 19.

Directory Lane

The little trip we just took down Directory lane should be somewhat helpful in understanding where in the iSeries IFS the Admin GUI puts all of the stuff that you create. If you are coming from WAS V3.5, this should be especially helpful since it proves that the two are very dissimilar in structure. One other point of note, there is no such SQL Library object in WAS Express known as the *Repository*. In V3.5 the repository was stored in an SQL catalog and one such catalog was built for each instance. I looked long and far and there is no such thing in the library structure. It has moved to the IFS with Express.

I have just one more path to talk to you about. As we hinted earlier in the chapter, the adminconsole is in a different place in 5.1 than it was in WAS 5.0. In WAS 5.0, IBM chose to place the adminconsole application which permits extra tailoring to WAS Express, in each WAS instance that was created. It showed up every time in the *installedApps* directory. In WAS 5.1, IBM has gotten much more

efficient. Now, the company places the *adminconsole.ear* file in the */QIBM/ProdData* section, rather than the */QIBM/UserData* section. Since the console is an IBM application, there was never a good reason to duplicate it in *QIBM/UserData*. So, now it exists in the *QIBM/ProdData* directory at the following path

`/QIBM/ProdData/WebASE51/ASE/installedApps/DefaultNode`

Chapter Summary

Well that's that. We used the Admin GUI tool in this Chapter to manage the WAS 5.1 Express environment. We had previously learned how to Create WAS instances in Chapters 11 through 16. In this Chapter we demonstrate some other WAS facility and how the Admin GUI is used to do everything from start and stop servers to create them and delete them. In Chapter 20 as we begin a WebFacing Project, we will look at some other management tools as we learn how to install a WebFaced application into the WAS Express environment. We will also learn how to delete an application if need be.

Following our good work with the Admin GUI in this chapter, we took a hard look at the structure of WAS Express in the AS/400 integrated file system (IFS). We took this notion one step further and showed the directory structure of a WAS instance and also a WAS application in EAR form.

All of this will come in handy one day, when you are on the phone with IBM support and they want you to send them some logs from the IFS so they can figure out what is not working.

Chapter 18

Using IBM Business Applications – Survey Creator

Another Give-Away Application

With the release of the WAS Express V5R1 came a new packaging for the software applications that IBM is “giving away” with Express for the iSeries. Now, instead of being a throw-in piece of code, the IBM Telephone Directory (ITD), covered in Chapter 19, now has a home in a specific product known as the IBM Business Solutions, 5722-BZ1. This new solutions package also contains the IBM Survey Creator software (ISC), including its source code. The Survey Creator is the subject of this chapter.

Getting IBM Help for BizApps

To find the IBM Business Solutions package documentation at IBM, you can take the following links:

```

>>www.as400.ibm.com
>>Library
>>iSeries Information Center
>>North America
>>>Your Language and V5R3
>>>Ebusiness and Web Serving
>>>Business Solutions

```

The IBM information Center panel that you will come to will look very much like that shown in Figure 18-1

Figure 18-1 Web Doc for IBM Business Solutions



If you are using IBM's documentation to install and use the Business Solutions package, the best instructions for installation of both IBM Telephone Directory (ITD) and the IBM Survey Creator (ISC) is in the IBM Welcome Page Link in Figure 18-1, so don't forget to take that link for ISC in this Chapter and for ITD in the next chapter.

As we discussed in Section I of this book, the IBM Business Solutions packaging provides a set of enterprise Web applications available for iSeries^(TM) servers. Each application solves a common, though not critical, business need.

As we are now in Section IV of this book, we have already gone through all of the prerequisites and preliminaries of installing, applying fixes, and setting up the IBM Survey Creator V1.0 (ISC) application in its own WAS Express instance, named SURVEYC. Additionally, we created a WAS Express instance called WASEONE that could support both the IBM Survey Creator and the IBM Telephone Directory application. However, we chose not to use examples of the Survey Creator from the WASEONE instance. Instead these will come from the SURVEYC instance in which we built the Survey Creator for demonstration purposes.

As a point of note, whenever you install the Survey Creator (ISC) or the Telephone Directory (ITD) application in any WAS server instance, the Welcome Page application is automatically installed and linked to those applications.

What is IBM Survey Creator V1.0?

In Section I, Chapter 4, we have already detailed the purpose and the capabilities of the ISC software. However, now that we are about ready to use the software, it would help to review its capabilities briefly before we begin to use the product.

You can use your IBM Survey Creator to build business surveys on the Web that your customers or employees may take at their leisure. You can then get the results of the surveys and you can manage surveys that you have created by adding or modifying questions. You may also delete questions and you can delete whole surveys when they are no longer needed for your business purposes

The ISC stores information in the AS/400 database and retrieves information through the lightweight directory access protocol (LDAP) server that comes standard with the operating system OS/400, or i5/OS.

There are actually two types of directory servers that are supported by the IBM Survey Creator (ISC) application. These are:

- iSeries Directory Server (LDAP) which comes standard with the system
- LDAP on Domino^(R) 6.0 for iSeries (Domino Directory services).

When the ISC application is installed on your AS/400 server, it can be deployed into instances of the following editions of WebSphere^(R) Application Server:

- WebSphere Application Server for iSeries V5
- WebSphere Application Server for iSeries V5.1
- WebSphere Application Server Network Deployment for iSeries V5
- WebSphere Application Server Network Deployment for iSeries V5.1
- WebSphere Application Server - Express for iSeries V5
- WebSphere Application Server - Express for iSeries V5.1
- This book is concerned only with deploying IBM Survey Creator with the WAS Express V5.1.

As WebSphere 6.xx is coming online, the Survey Creator via the IBM BizApps package is supported.

Survey Creator Admin Pages

As you would expect, you can use ISC to manage which of your users can create and which ones can take surveys. Through administrative pages, you grant users survey owner authority. In order for users to perform these functions, however, they must first have valid LDAP entries to obtain survey owner authority. Unfortunately most AS/400 experts are not adept at LDAP and view it as a foreign function on the AS/400 system. Though IBM has not really prepared its AS/400 constituency well for LDAP, these free applications do not require much knowledge of its facility.

The way it works is that a valid LDAP entry adds a user to list of authorized survey owners. The administrator can select users from a

list of valid LDAP entries to authorize them as survey owners. The administrator pages also allow users to be removed from owning a survey. To better understand how this new piece of code works (BTW, the source code comes with it) let's go through the work of setting up an administrator, create a survey, take a survey, and examine the survey results. That's about all anybody could want in a survey application.

To access the IBM Survey Creator administrator pages, perform these steps:

Specify one of the following URL types in your Web browser:

`http://your.server.name:port/ibm-bizApps/welcome/admin.do`

or

`http://your.server.name:port/ibm-bizApps/survey/manage/admin`

For our purposes, using IP addresses is the simplest way of accessing these facilities from the local LAN. Thus, we would type:

`http://192.168.0.250:1600/ibm-bizApps/welcome/admin.do`

`http://192.168.0.250:1600/ibm-bizApps/survey/manage/admin`

Your.server.name above is the name of the AS/400 server where IBM Survey Creator is installed and **port** is the Apache HTTP port number for this application. The second URL is used if you want to access the IBM Survey Creator administrator pages without using the IBM Welcome Page application. The second set shows the same notion-using IP addresses for domains.

Using the top URL with IP address option, you would be taken to the sign-on panel first as shown in Figure 18-2

Figure 18-2 Biz Apps Sign On



Once you get the sign-on screen, you are about to enter the IBM Welcome Page. First, however, you must sign on with the user id and password that you selected when you created the SURVEYC WAS instance (the LDAP Wizard). This is very important. The next panel you see will be the Welcome Page for the IBM Business Solutions for the SURVEYC WAS instance. This page will look different if you also have installed the ITD application. To continue, sign on and click the OK button and you will be taken to the main panel as shown in Figure 18-3.

Figure 18-3 IBM Welcome Page for Survey Creator

To move on with the IBM Survey Creator application, you would click on the IBM Survey Creator Link as shown in Figure 18-3. From here, you will be able to authorize users for surveys. After you authorize users for surveys, you will be able to use a different URL to create surveys and perform all of the tasks necessary for the IBM Survey application. All of these tasks are well documented at the IBM Web Site (Information Center) in the URL / links and navigation that we showed in the beginning of this Chapter. We hope you enjoy your IBM Survey Creator Experience.

Some Problems

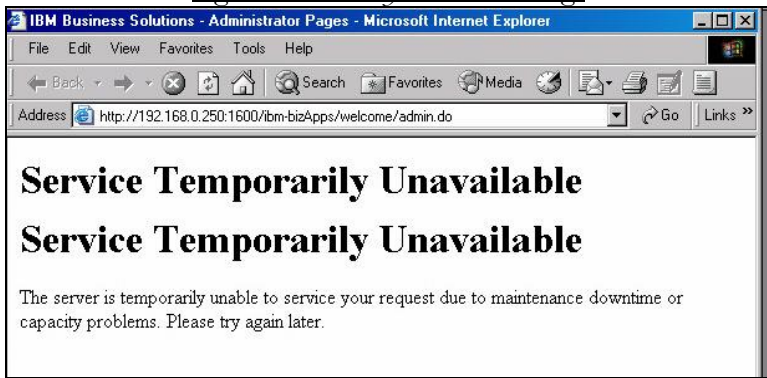
If you start having problems, try to work them through for a while but don't be afraid to call IBM Support for help. The 1-800-IBMSERV number can solve lots of WebSphere type maladies. If I had not worked through the problems that I had in installing and using Survey Creator, with IBM's help, I would have been blaming IBM for bad code. In my experience with the IBM Survey Creator, its implementation was rough, but almost all of that was my fault.

In the prerequisites section of this book (Section II), I noted that 5722-AC3, the 128-bit cryptographic feature is a requirement for V5.1 WAS Express. After three weeks of problems making Survey Creator work, I found the problem. We had V5R1 of 5722-AC3 installed and that was not good enough. Our Test Box lost a disk in 2004 while I was working through the applications with V5R2.

Without sharing all of the issues, suffice it to say that we restored our V5R1 backup and then made a two release jump to V5R3. We did not position the new 5722-AC3 in the V5R3 install deck at the time so after V5R3, the product continued to appear to be installed and it did not say back level. Just yesterday (at the time of this writing), I noticed that it was at V5R1 and IBM quickly helped me solve that and with that, all of the Survey Creator problems went away.

I guess I got used to IBM's GO LICPGM and Go PTF functions being so good that I did not have to think as much in going from new release to new release. The machine in question is a model 270 and is used just for testing. Life would not have been very fun if we had gone to V5R3 on the production box in the same way at this account. The moral of the story is to be careful during installation, and make sure that 5722-AC3 is on the machine at the V5R3 level. I sensed in a few areas, including PPP ECS installation that the V5R3 product set absolutely wants the 5722-AC3 to be there and be up-to-date.

I'll show you some of the error panels I got so if you see these, call IBM for help immediately. My very first encounter was when I typed in the initial URL and received this message as shown in Figure 18-4.

Figure 18-4 Survey Creator Message

Eventually, I was let in to get the sign on panel. However, using the Administrator id and the password that I had keyed in, I was unable to get on. At first, I got this message as shown in Figure 18-5. Then the message changed a little as shown in Figure 18-6, but the results were the same.

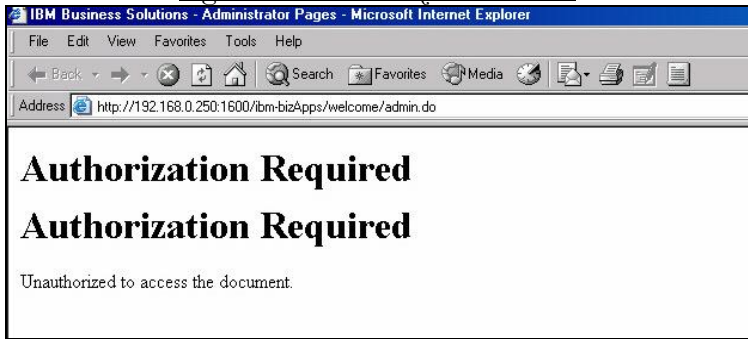
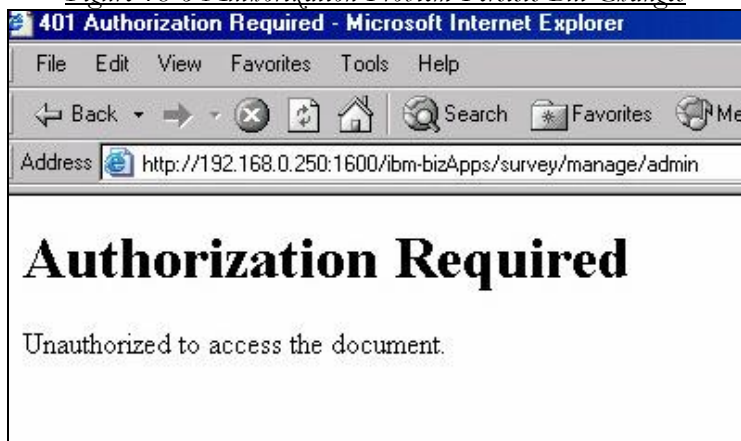
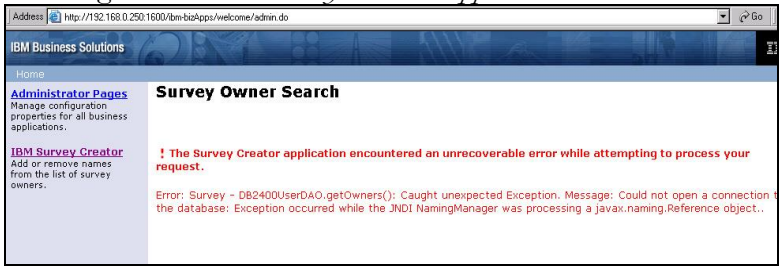
Figure 18-5 Authorization Problem

Figure 18-6 Authorization Problem Persists But Changes

Because IBM Support did initially know how to fix this, since it appeared that my 5722-AC3 was OK, I tried some things myself. I created another WAS instance for the Survey program called TSURVEY. When I got to the first LDAP panel, I put in the user id as TSURVEY and I put in the password as TSURVEY and confirmed it.

I fired up the TSURVEY instance and signed on as TSURVEY, TSURVEY. The Welcome Page came up and I received a second sign on. I signed on again and the sign-on panel went away and I was left with the panel as shown in Figure 18-3. From here, I clicked the IBM Survey Creator Link, thinking life was good and I got the message shown in the panel in Figure 18-7. Life did not get good until I got the right version of 5722-AC3 on this machine.

Figure 18-7 The Survey Creator Application Panel Error

I hope you see none of those errors but I show you them so that you hang in there, even if things do not look good at first.

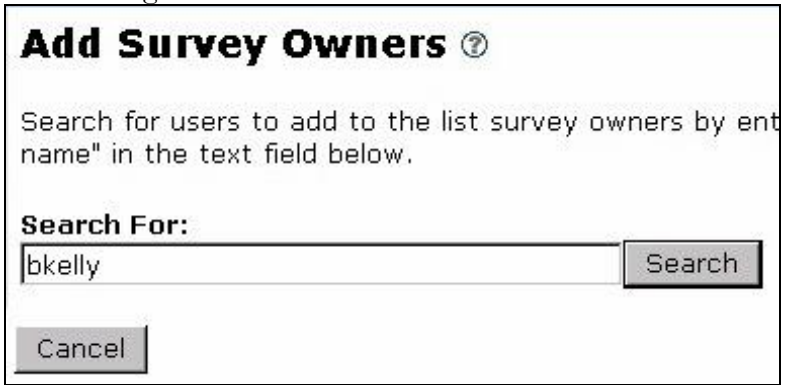
Step 1 Setting Up Survey Administrators

Let's get to work. We start by adding some survey administrators, a necessary step prior to being able to create a survey.

When you click on the Survey Creator link in the left pane of Figure 18-3, you are taken to a panel similar to that shown in Figure 18-8.

Figure 18-8 Create Survey Administrators

Click the Add button in Figure 1808 and you will be taken to a panel similar to that shown in Figure 18-9.

Figure 18-9 Look For Candidate Administrators

The candidate survey owners are the names that you used in the LDAP sub Wizard in the CREATE WAS for the SURVEY WAS Express instance. I typed my name in as you can see in Figure 18-9 but the system did not respond well because bkelly is not an LDAP

administrator for IBM's bizapps. So, I changed my approach by typing using an asterisk (*) for a search criterion as you can see in the panel shown in Figure 18-10

Figure 18-10 Candidate Administrator IDs

Search For:

*

Search Results

Select from the list of users below, then click the "OK" but

	Name	Phon
<input type="checkbox"/>	Administrator	
<input type="checkbox"/>	SURVEYC	
<input type="checkbox"/>	bizapps	

As you can see in Figure 18-10, bizapps is one of the users as is SURVEYC. These two users should be able to administer surveys in our mythical shop. So, to select them, I clicked on both buttons and they immediately filled with checkmarks as shown in Figure 18-11.

Figure 18-11 Select Administrators from Generic Results .

Search For:

Search Results

Select from the list of users below, then click the "OK" button.

	Name	Phone
<input type="checkbox"/>	Administrator	
<input checked="" type="checkbox"/>	SURVEYC	
<input checked="" type="checkbox"/>	bizapps	

Click OK to continue with these two users selected.

Create A Survey

Figure 18-12 Back to the Welcome Page

Address  http://192.168.0.250:1600/ibm-bizApps/welcome/

IBM Business Solutions

Home

Welcome

You are at the IBM Business Solutions home page. Select one of the following links:

 **IBM Survey Creator**
Create, configure, and manage Web-based surveys.

The only function that can be performed at the Survey Creator URL is to Add or Remove Survey users. Whenever it gets frustrating because

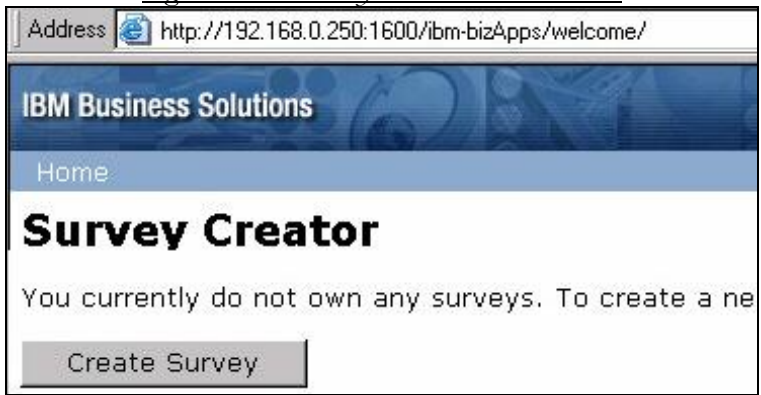
you'd rather have the program be intuitive than to look at the documentation, it's time to check the book.

From the Web documentation, I learned that I had to go back to the bizapps Welcome URL to proceed. In our shop, the URL is as follows:

http://192.168.0.250:1600/ibm-bizApps/welcome

So, when you type this URL into your browser, you get back to the Business Solutions welcome page as you can see in Figure 18-12. By then clicking on IBM Survey Creator, you advance to the Survey Creation panels starting with the page in Figure 18-13.

Figure 18-13 Survey Creator Launch Panel



Click the Create Survey button in the panel shown in Figure 18-13 to create the first survey.

Figure 18-14 Create a Survey

Address http://192.168.0.250:1600/ibm-bizApps/welcome/

IBM Business Solutions

Home

Create Survey

The following survey properties control how your survey appears to your participants below and click the "Create" button to create your survey. You can then begin adding questions.

Fields marked with an asterisk (*) are required.

Survey Elements

Hidden description:

Title:

Introduction:

Closing:

Destination URL upon survey completion:

Security

Additional Owners:

As you can see, I am initiating a survey to see what the folks think about the Advanced Hello World program definition as a valid test vehicle. Note where the various items are typed and then note how they appear on the survey taker's page. The second half of the Create Survey page is shown in Figure.18-15. You see this if you scroll down from the panel in Figure 18-14.

Figure 18-15

Security

Additional Owners:

Require participants to login before taking the survey

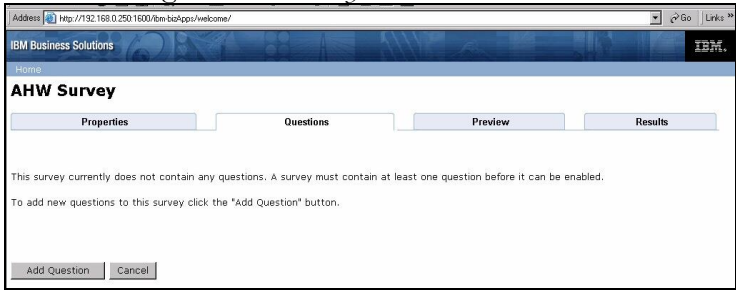
Restrict the following participants from taking my survey

Restricted participants:

Allow the following participants to take my survey

Allowed participants:

I did not pick any of the options from the second half of the Survey Creation page for this survey.

Figure 18-16 Survey Creation Main Panel

Add a Question to the Survey

From the page in Figure 18-16, you can now add a question to your survey. Take this option by clicking the Add Question button. You will arrive at a page similar to that shown in Figure 18-18.

Figure 18-18 Add Question to Survey

As you can see in this panel (Figure 18-18), I selected the option to make the survey choices for answers lined up in a vertical fashion.

Figure 18-19 Question Page with Choice Text Completed

Select One (Vertical) ▾

* How much do you like the Advanced Hello World Program? ▾

* Choice text:

Choices:

It is the best of the best
It is almost the best
It is almost better than less than almost the best
It is OK
It stinks

Buttons: Add>, Move Up, Move Down, Remove, Edit

Yes No

Yes No

The question is “How much do you like the Advanced Hello World Program?” You enter the choices for the answer one at a time in as shown in Figure 18-19, and when you finish each one, click the Add button under Choices. By the time I took this screen snap, I had entered all five choices.

The full panel is shown in Figure 18-21. As you can see, I made this a required question.

Figure 18-21 Full Question Panel

This question type allows your participants to select one answer from multiple choices using radio buttons or a menu. Enter the information below, then click the "Finish" button to create your question.

Fields marked with an asterisk (*) are required.

Format: Select One (Vertical) ▾

Question text: * How much do you like the Advanced Hello World Program? ▾

Answers: * Choice text:

Choices:

It is the best of the best
It is almost the best
It is almost better than less than almost the best
It is OK
It stinks

Buttons: Add>, Move Up, Move Down, Remove, Edit

Include "Other" choice: Yes No

Required question: Yes No

Include "Comments" textbox: Yes No

Passthrough HTML: ▾

Figure 18-22 Question Available for Selection

Order	Question	Type	Required?
<input type="radio"/>	1 How much do you like the Advanced Hello World Program?	Select One (Vertical)	Yes

When you finish the question, you have the opportunity to end the survey and to select the questions that will be available in the first iteration of the survey. In other words, after the fact, you can deselect questions from the survey. Since this survey has just one question, I clicked the radial button to select it as shown in Figure 18-23.

Figure 18-23 Question Selected

Order	Question	Type	Required?
<input checked="" type="radio"/>	1 How much do you like the Advanced Hello World Program?	Select One (Vertical)	Yes

When you have completed the survey, you can review the survey by clicking on the Preview Button as shown in Figure 18-16. You can see that the survey exists in the panel shown in Figure 18-24.

Preview the Survey

Figure 18-24 Survey Preview

AHW Survey	
Welcome to the Advanced Hello World Survey. I am sure you know how important you Pasttime.	
Fields marked with an asterisk (*) are required.	<input type="button" value="Preview"/>
<p>* 1. How much do you like the Advanced Hello World Program?</p> <ul style="list-style-type: none"> <input type="radio"/> It is the best of the best <input type="radio"/> It is almost the best <input type="radio"/> It is almost better than less than almost the best <input type="radio"/> It is OK <input type="radio"/> It stinks 	
Thank you for your participation in this survey. You are great and deserve araise from	

Figure 18-24 is the meat of the survey shown on one page. If you shift to the Properties view, you can see the URL that you must include in your Web pages so that people can take the survey. This is shown in Figure 18-25.

Check out Survey Properties

Figure 18-25 Survey Properties

AHW Survey ⓘ

Properties **Questions**

The following survey properties control how your survey appears to your participant. Save Changes button to update your survey properties.

Fields marked with an asterisk (*) are required.

Survey Elements

URL path and file name: /ibm-bizApps/survey/survey.jsp?id=1

Hidden description:

Title: *

Each time you create a survey, the Wizard will give you the URL that is needed to take the survey. This is shown as the first Survey Element in Figure 18-25 next to the prompt URL path and file name. For this first survey, the URL and file path to call this survey from the Web is as follows:

`/ibm-bizApps/survey.jsp?id=1`

Obviously, that's not enough for a URL. However, when you add your host and domain name and port # of the WAS Express SURVEYC instance that supports the application to this path, you have a complete URL that can be inserted into your pages. This URL looks as follows on our system using IP address format:

`http://192.168.0.250:1600/ibm-bizApps/survey.jsp?id=1`

Take the Survey

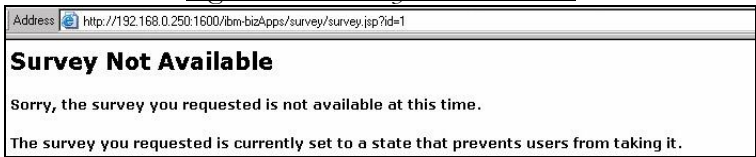
To Call the Survey, type in the above URL into a Web Browser as in Figure 18-26:

Figure 18-26 Full URL for First Survey



When you start the URL shown in Figure 18-26, you will get the Survey as shown in Figure 18-27. Keep your Survey Administration panels where they are and start the URL in Figure 18-26 in a new browser window.

Figure 18-27 Survey Not Available



Resolving Initial Survey Problem

We know that we created the survey and that it is available so why does it appear not to be available. Let's retrace some of the options. If we go back to the Survey Creator panel with the survey name as shown in Figure 18-28, you will notice that the status is disabled. So, we forgot to enable the survey so it can be taken. This is a good

safeguard and it permits you to create surveys ahead of time that you can enable and disable as appropriate.

Figure 18-28 Survey Status

Survey Creator ⓘ

Below is a list of your surveys, including other surveys to which you have been granted by the "Create Survey" button to create a new survey.

Survey Name	Status	Description
<input checked="" type="radio"/> AHW Survey	Disabled	This is the Survey about Advanced Hello World

Update Remove Enable Disable

Create Survey

To Enable the Survey as shown in Figure 18-28, all you've got to do is click the Enable button.

Go back to the Survey-taking URL as shown in Figure 18-26. This time, you will be able to take the survey as shown in Figure 18-29.

Figure 18-29 Taking the Survey

AHW Survey

Welcome to the Advanced Hello World Survey. I am sure you know how important your responses are Pasttime.

Fields marked with an asterisk (*) are required.

* 1. How much do you like the Advanced Hello World Program?

It is the best of the best

It is almost the best

It is almost better than less than almost the best

It is OK

It stinks

Thank you for your participation in this survey. You are great and deserve arise from your employer.

Submit Survey Reset Form

The page in Figure 18-30 shows that we have answered the question for you with the most favorable choice. We obviously love Advanced Hello World.

Figure 18-30

Fields marked with an asterisk (*) are required.

*** 1. How much do you like the Advanced Hello World Program?**

It is the best of the best

When you answer the question, close the survey and you get the handsome “Thank you” panel as shown in Figure 18-31. Our work in this browser session is now complete.

Figure 18-31 Thank You for Taking the Survey

Address  <http://192.168.0.250:1600/ibm-bizApps/survey/takerAnonymousMain>

Thank You!

Your survey information was received successfully.

Survey Completion

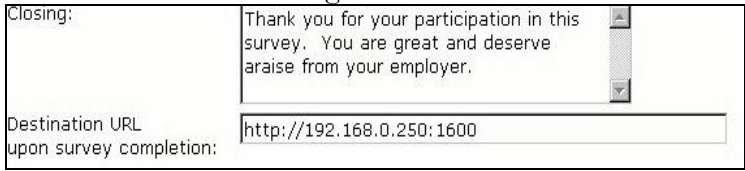
Close out the survey and this browser session by clicking on the Close button. When we first created this survey, as you may recall, we told the Wizard that when a person finished our survey, they were to go back to the Apache HTTP home page for the SYRVEYC instance. At least, that is what we thought we did. However, the home page did not come up.

To find out why, we need to take a look at the survey properties again in our other browser session. Notice in Figure 18-14 that we selected the following URL as our completion URL:

http://192.168.0.250

This is the default Apache Home page but it does not come up unless there is a default HTTP instance running at port 80. Since we run at port 1600 for SUREVEYC, we can change this in the properties as shown in Figure 18-32.

Figure 18-32



As you can see in Figure 18-32, the Destination URL has been reset to the following:

http://192.168.0.250:1600

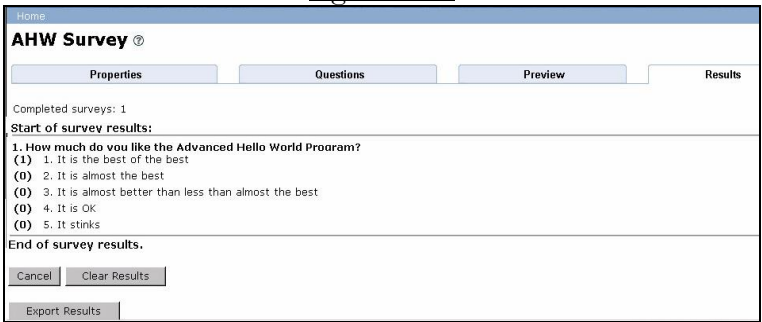
The port number is very important.

Moving back to the other browser, if we close the Survey application, after making the change to the properties of the survey as in Figure 18-32, we are taken to the SURVEYC home page as we had asked. Success has been ours in this survey undertaking as we can see in Figure 18-33

Figure 18-33

Checking Survey Results

There is really just one thing left to do at this point. Let's check our results. You and I both know that only one person took the survey and we know how well the answer actually was. To see the survey results, from the Survey Creator panel go to the fourth button on the far right at the top of the main panel and click Results. The results of the survey will be shown as seen in Figure 18-34.

Figure 18-34

So, we can see that there was one question answered and the Advanced Hello World program has been rated as outstanding. All the other responses have 0 responses. That's a pretty neat survey tool and a pretty good Advanced Hello World program.

What Else Can You Do?

There are actually lots more things that you can do with the data collected but you have to delve deeper into the program. The URL at the beginning of the Chapter will help you in this endeavor. For now, you have a functioning WebSphere application to pick apart and study and perhaps use as a model for your own scratch Web development.

BTW, IBM was also gracious enough to give you the source code for the IBM Survey Creator Package. Enjoy!

Chapter 19

Using IBM Business Applications – IBM Telephone Directory

Free Application # 2 from IBM

Let's go ahead and bring up the Free Telephone Directory (ITD) application now. IBM has graced us with the object code for the Company's Telephone Directory application. In Chapter 13, when we created the TELDIR WAS instance, we created the application on our system and we created and assigned an LDAP user to manage the application.

As you now know, the IBM Telephone Directory application is shipped with 5722-BZ1 along with the IBM Survey Creator (ISC) application. The 5722-BZ1 free product is shipped as an entitlement to WAS Express starting in V5.1.

It is not only shipped along with WAS Express V5.1 and later, the "Bizapps" are also shipped with WAS V5.0, though the packaging and installation is much different. In WAS V5.0, for example, you must run a script program called `itdsetup` after assuring LDAP is functional. A brief outline of the WAS 5.0 installation process is included as a Chapter Appendix.

With V5.1 of WAS Express, the IBM Labs put hooks in the WAS Express installation process to make it easier for you to bring it up as a functional application for your organization.

Installing the IBM Telephone Directory Application

The first major task in installing the Telephone Directory Application is to create a functional LDAP setup for ITD. You can't miss this one. When you create a WAS as we did for TELDIR in Chapter 13, with V5R1, you are automatically taken through an LDAP setup Wizard which accomplishes all of the tasks necessary to set up the LDAP for the ITD.

This is just a reminder that LDAP is the means that accomplishes the ITD application. The application does not use typical AS/400 database files. In WAS V5.0, if you had not run the ITD setup script, the WAS instance create for the ITD would fail. In V5.1, it walks you through the LDAP Wizard to assure that there will be no failings.

To use the ITD application, use your Admin GUI for WAS, and start the TELDIR Apache HTTP Server as well as the TELDIR WAS Express Server. Then, you're ready to roll. Once both servers are up, you can run the ITD application.

Running the IBM Telephone Directory Application on TELDIR Express server

The steps to invoke and run the application are shown below:

In V5.0, use the following URL format:

```
http://yourserveraddress:yourport#  
/bizApps/
```

a.k.a.

```
http://192.168.0.250:1500/bizApps
```

For V5.1, use the following URL format:

```
http://your.server.name:yourport#/  
ibm-bizApps/welcome/home.do
```

a.k.a.

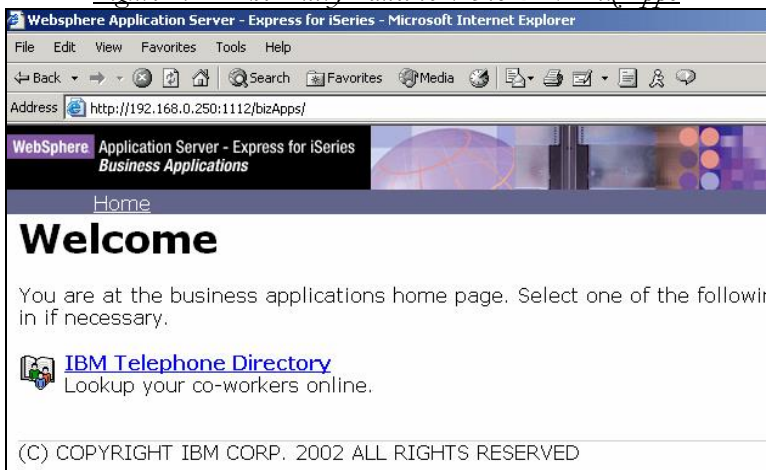
```
http://192.168.0.250:1500/ibm-  
bizApps/welcome/home.do
```

A. Start the IBM Business Applications

Depending on which version of WAS Express (V5.0 or V5.1) you are running, take one of the options above

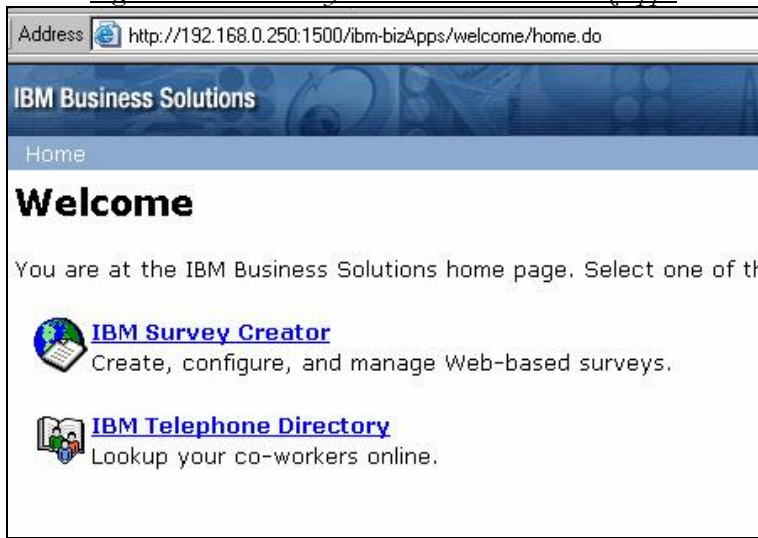
The next panel you will see in WAS V5.0 is similar to that shown in Figure 19-1 below.

Figure 19-1 The Entry Panel to V5.0 IBM Biz Apps



The next panel you will see in WAS V5.1 is similar to that shown in Figure 19-2.

Figure 19-2 The Entry Panel to V5.1 IBM Biz Apps



Note:

The Survey Creator Option would not be on the panel shown in Figure 19-13, unless you have also enabled the application in the TELDIR WAS server instance.

Business Applications Welcome Page

In this application server scenario, */bizApps* (V5.0) or */ibm-bizapps* (V5.1) is the context root for the Welcome Page that is used to launch the IBM Telephone Directory application. The Business Applications Welcome page appears as shown in Figure 19-1 or 19-2. IBM refers to this as the “business applications welcome page,” since it is the entry point for all IBM supplied business applications. At the time of this writing, Telephone Directory and Survey Creator are the only IBM applications supplied and enabled. As more become available, IBM will more than likely enhance its Welcome Page accordingly.

Many who have seen her at COMMON and other conferences know that Kelly Schmotzer, the former Domino Fireball inside the IBM company is now IBM’s Worldwide Marketing Manager for eServer SMB Initiatives which focuses heavily on the Express branded solutions like finding ways to better market WAS Express. In this role she single handedly convinced IBM to supply a real application (IBM Telephone Directory) for WAS Express installations. Take a look at it on an IBM Business Partner site, MORPHEUS out of the UK, in the next Chapter.

More to Come

With Kelly Schmotzer driving new ideas to the development teams at IBM, it shouldn’t be long before we see an exciting and interesting application that every customer can use appearing in this column. The only reason I think something might happen is Kelly tapped a few of her trusted friends and partners with some of her ideas, and I was lucky enough to be one of those “tapped.”

The new server applications, especially for Express servers, including iSeries, if plans work out, may be easier to use than to not use. I am not privy to share the details yet, but if Schmotzer's plan proceeds, these applications could help you get a Web site with real value applications running in less time than it now takes to install the Express Server itself. To that, on behalf of all iSeries Web affiliates, I say a premature but confident thank you to Kelly.

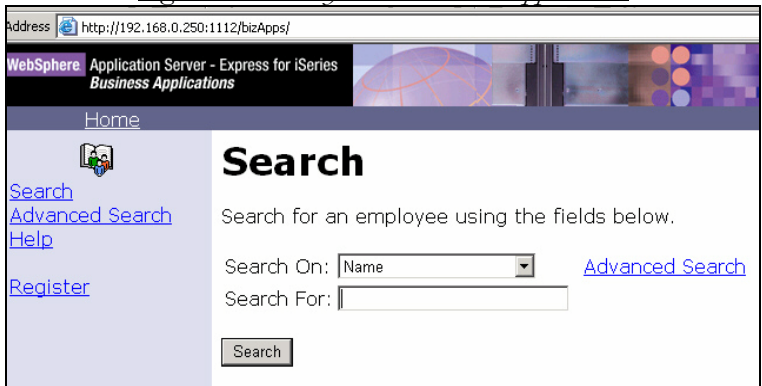
Let's get back on track now and continue with the startup of the Telephone Directory Application as follows:

Since the application is so similar in V5.1 as it is in V5.0, we use the V5.0 panels below to demonstrate how to use the ITD application.

B. Click Telephone Directory Link.

You are redirected to the page as shown in Figure 19-3.

Figure 19-3 Entry Point to ITD Application



Telephone Directory Application Search Page

The Search page appears next. However, since this is the first time through the application, we know that we have no directory entries to

show. The only two items that make sense at this point are the Help item in the left frame and the Register item. Since we have a deep desire to show what this application actually does, we know intuitively that we must enter names and other directory information in order to have a body of content with which to work. In this example run, the plan is to register one entry, and then use the generic search capabilities of the application to find that one LDAP record.

Register New User

So now, let's start by registering the infamous and fictitious Nipper Novakowski into our database. Click on **Register** in Figure 19-3 and you will be taken to a panel similar to that in Figure 19-4.

C. Register a new entry. Click the Register link.

The registration page appears as shown in Figure 19-4. In this example, you are directed straight to the registration page because we used the Open Registration option (-o) in the setup script. If the (-o) option were omitted, we would be required to sign on with administrative privileges to the LDAP directory using the iweAdministrator user id as discussed previously.

Figure 19-4 Telephone Directory Registration

Address http://192.168.0.250:1112/bizApps/

WebSphere Application Server - Express for iSeries
Business Applications

Home

[Search](#)
[Advanced Search](#)
[Help](#)

[Register](#)

Register

Registration Information

First Name: *

Middle Initial:

Last Name: *

Suffix:

User ID: *

Password: *

Confirm Password: *

D. Enter the Registration Details

This is shown in Figure 19-4.

Note that each user ID is created for a specific person in your organization or for individuals who may be affiliated in some way with your organization. The user ID is based on the first name, middle initial, last name, and suffix. You can override the user ID to whatever you may choose. In this application, we chose the name NIPPER NOVAKOWSKI as the registrant name and as the user ID. We also chose NIPPER as the password. Of course, in your application, you would probably not pick a first name as the password.

User Name Particulars

During the initial registration process, it is very important to select a user ID that you are satisfied with, since the IBM Telephone Directory application, as written, does not let you change it after you register a user. Similarly, the system creates what it refers to as a "Display Name". The display name is the last name, followed by

a comma, then the first name, then the middle initial, and title. This is the name format that you use to search the directory.

Again, you cannot override the display name. It is set to these prescribed values. If you don't really like these constraints, you can probably write the application using EXFMT and a Display file, WebFace it, and provide a link to the JPEG file so that the picture can be displayed. But, really, those constraints aren't that bad for such a neat application.

Model Application

Whether it is a use-as-is application for your company or not, the Telephone Directory application certainly shows what can be done. Moreover, if it isn't exactly what you want, you can use it as a basic part of an application design to write your own. Maybe IBM will even give you the source code.

Registration Details

IBM labels the display name in Figure 19-4 as immutable. You cannot update it after the registration is completed. Thus, you must exercise care using this application to enter user data. If you really need to alter the display name or the user ID, and you are not up for re-writing and WebFacing the application, then you must delete and recreate the entry.

E. Scroll Down Through the Available Fields.

At the bottom of the registration options (not shown), you can type information about related employees.

Tip:

With the notion of open enrollment (selected in the LDAP Wizard, or with the `-o` option in the `itdsetup` script as shown in this Chapter's Appendix), you must enter a user ID, also

known as the "user name," and corresponding password to update user details. If the directory is created with closed enrollment, only the directory administrator (user name - iweAdministrator, password - the value that was specified in V5.1 in the LDAP Wizard, or with V5.0, with the -a option while running the itdsetup script) and any user in the iweAdministrators group are allowed to change user details.

Related Employees and Image Selection

To locate a related employee, at the bottom of the registration panel (not shown in Figure) click one of the Find button. This function enables you to search the directory and locate another entry, such as an employee's manager. There are currently no additional entries in the database, so at this point of our run; this function can only be done later as other entries are added.

You can also enter a picture of the employee to display with your entry. This is really neat. We have chosen to enter "Nipper's" picture along with the text. This is a very handy process. The JPEG image can be located on your local hard drive. It is copied right to the LDAP directory when you submit the registration form. Just click the Submit button to register your entry.

F. If Problems Found, Error Message.

If everything is not OK, you'll get an error message telling you about the problem and suggesting that you correct the problem and resubmit.

Our first big issue came when we tried to have the system absorb Nipper's entire picture. It was too large. We got a bad message.

Caution:

The maximum size of the image is 15360 bytes, and the image must be in the JPEG format. If you have .BMP images, they cannot be used for two reasons:

1. They more than likely will be too big and
2. They are not JPEG.

So, you must convert the image to JPEG before submitting the form to LDAP. You can use an image of any width and height, but the image is always resized to 115 by 115 pixels. So, assure yourself that it looks as expected after uploading to LDAP with the SUBMIT.

Registration Completion

When the picture is accepted the open registration for Nipper Novakowski, famous fictitious character is completed. You may ask how did we get a picture for a fictitious character. One of my clients William Kepics a.k.a. Pierre Le Kep, from Plains, Pennsylvania, posed for the digital picture.

G. Enter Additional Entries.

We recommend that you enter one or two additional entries for testing purposes, so that you have something to search for other than good old Nipper. Though Nipper Novakowski may be an exquisite entry, you might want to include an entry for Cornelius (CON) Salwawski, and perhaps one for Jeremiah Jumpstart and another for Betty Beanyak. These are just a few test data suggestions provided at no additional charge. However, I don't think that Pierre Le Kep would agree to pose for all the pictures.

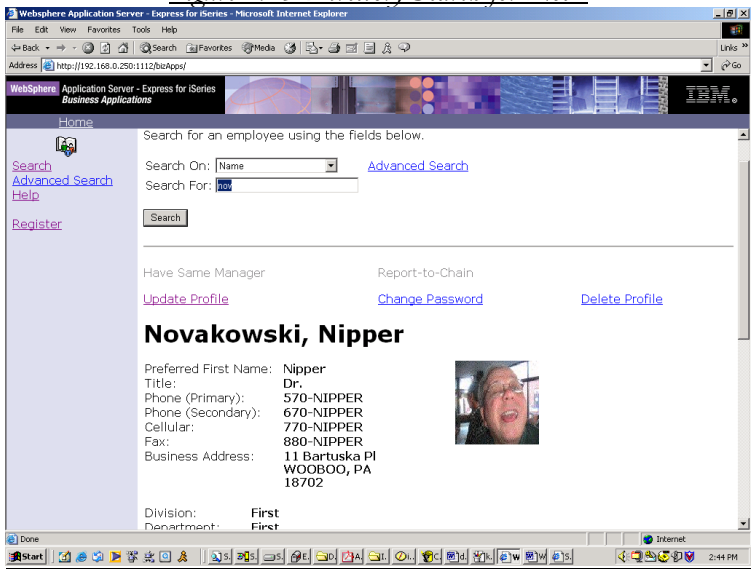
H. Start Searching for Entries

When you've got a few folks registered, you are ready to start searching for entries. Click the Search link on the upper left side of the page. This gives you access to basic searches. Then, click the drop-down list button to look at the basic search options. Searches can be performed using an "*" as a wildcard. For example, a search for "ha*" would match any word beginning with "ha," such as hat, hamster, happy, and Hawaii.

Try searching with a few different options. If the search produces more than one entry, a list of results appears. If there is only one entry the entry itself appears. Since we entered generic criteria, (nov*) we received one entry in response to our search. There is

no list. But, as you can see in the panel in Figure 19-5, the actor posing as Nipper Novakowski is in full regalia.

*Figure 19-5 Directory Search for Nov**



Good Work Pierre!

If you get a list back because multiple records match the search criteria, to view the details of the entries found, click the link that is provided. As noted, if only one entry is found as above, the details of that entry are displayed immediately as shown in Figure 19-5.

Tip:

You can choose to use the Advanced Search link too. However, there is a difference as to how the wildcard ('*') is used by each search option. This and a whole host of other opportunities to exercise the Telephone Directory application are given to you in the IBM WebSphere Express Redpiece referenced previously. If becoming a Telephone Directory Guru is in the cards for you, the Redpiece in the appendix of

this Chapter is a must read along with the Information Center information on ITD. Practice is surely your second best way.

Using Telephone Directory Application at Ignite400.org

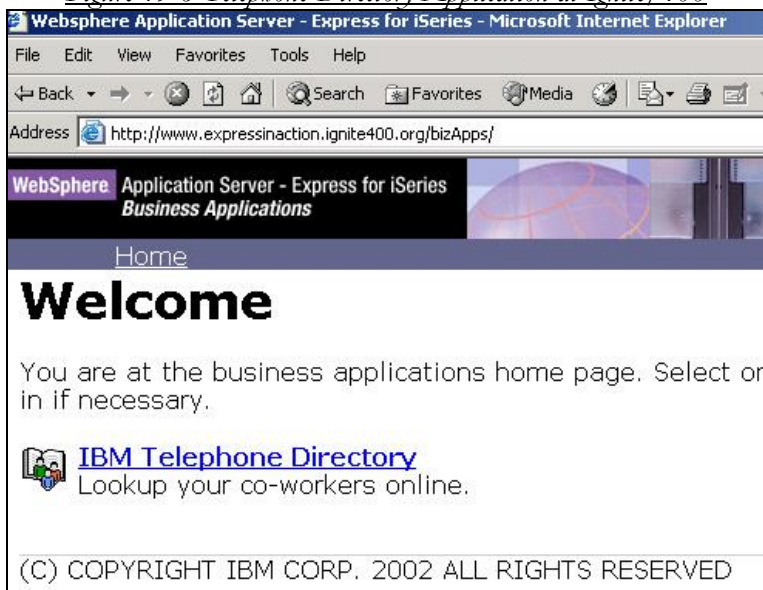
Now that we have shown how you would start this application in your own shops, we will digress to a company that has deployed the application publicly on the Web. The company is Ignite400 and is managed by Bob Cancilla and Bleddyn Williams. The company is owned by an IBM Business Partner, Morpheus, who operates out of the UK. Bleddyn Williams is the contact at www.ignite400.org. Kelly Schmotzer introduced me to the Ignite folks when I needed some Java AD help and some big shoulders for commiseration. They are a first class shop and are most knowledgeable in the WebSphere world. A trip to their site is sure worth your time.

Start Telephone Directory Application at Ignite400

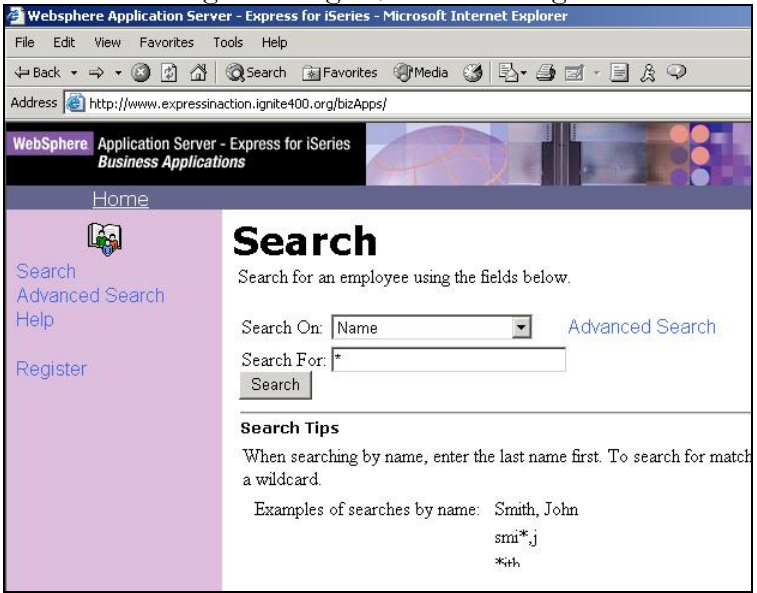
Now that we know how to do the basic Telephone Directory Application, let's go out to the ignite400 Web site and see how they implemented the Telephone Directory Application. As you will see, they are currently on WAS V5.0. Hold, on, there's learning ahead. The URL to get there is as follows:

`http://www.expressinaction.ignite400.org/bizApps/`

When you type in this URL, you will see the IBM Intro panel similar to that shown in Figure 19-6.

Figure 19-6 Telephone Directory Application at Ignite/400

This looks like the vanilla WAS V5.0 IBM Telephone Directory application. It does because it is. From here, select the IBM Telephone Directory application and you will see a panel similar to that shown in Figure 19-7.

Figure 19-7 Ignite/400 Search Page

Search Tips

Within the panel in Figure 9-2, you see that the IBM panel offers some tips for how to search for entries in the directory. The text says that when searching by name, you should enter the last name first. To search for matches that begin or end with the same characters, use an asterisk as a wildcard. They also supply a few examples that are repeated below:

Smith, John
smi*,j
***ith**

In this case, we have chosen to see the full directory listing. To do this, we type in an “*”, just as with the Nipper search from above, and

we submit the search. In no time at all, we receive a results panel very similar to that shown in Figure 19-8.

Figure 19-8 Search List for Ignite400 ITD Application

The screenshot shows a Microsoft Internet Explorer browser window displaying a search results page. The browser's address bar shows the URL: <http://www.expressinaction.ignite400.org/bizApps/>. The page header includes 'WebSphere Application Server - Express for iSeries Business Applications'. The main content area features a search results table with the following data:

Name	Phone (Primary)	Email Address
Andelin, Nathan	801-568-1669	nathanma@haaga.com
Cancilla, Robert J		bobc@ignite400.org
Cioffi, Jim	800-555-1212	jacioffi@yahoo.com
Cook, Drew	01506 508204	dcook@glenmorangieplc.c
DeCandio, George	919 254-1569	
Dolinar, Brian	313-966-2514	BDolinar@CTS-mail.com
Hoover, Tom M	905-316-8308	thoover@ca.ibm.com
Kelly, Brian W	570-829-5926	bkelly@kellyconsulting.com

On the left side of the page, there are navigation links: Search, Advanced Search, Help, and Register. The browser's status bar at the bottom shows 'Done'.

The List

Now, kindly notice the last person on the list. It is none other than yours truly, since I took up Bob Cancilla's offer to register, just as you can if you choose. To see my record, you would just click on the last name in the list as shown in Figure 19-8, and you will see a panel similar to that shown in Figure 19-9.

Figure 19-9 Record From Ignite/400 ITD Application

Search

Have Same Manager Report-to-Chain

Update Profile Change Password Delete Profile

Kelly, Brian W

Preferred First Name: **Brian**

Job Responsibility: **Information Technology Consultant**

Phone (Primary): **570-829-5926**

Fax: **570-208-1577**

Email: bkelly@kellyconsulting.com

Business Address: **11 Marjorie Avenue**

LET'S GO WebSphere
Getting Started
WebSphere Development
Studio Client runtime (OS/400)
By Example - Get Value and the Best
WebSphere Development and Deployment
Solutions

Search for Another User

As you can see, instead of using my picture, which admittedly is outdated in my books, I used a piece of the sample cover from a book I had just finished at the time. It serves where my picture otherwise would be.

Some of you may know Kelly Schmotzer from IBM who I mentioned earlier in this chapter. In the OV/400 days, she was known as the “Cha-Ching” lady because she would consistently make the cash register noise when she discussed costly items in her presentations. Kelly suggested that I test the Ignite/400 Telephone application. I did and I noticed that she is there too. She’s got her own little Directory Entry. Just for the Halibut, let’s see if we can find Kelly’s record and picture out there anywhere. Figures 19-10 and 19-11 tell the whole search story.

Figure 19-10 Search for Schmotzer

Search On:	<input type="text" value="Name"/>	<input type="button" value="A"/>
Search For:	<input type="text" value="SCHMOTZER"/>	
<input type="button" value="Search"/>		

Figure 19-11 Schmotzer Found

Schmotzer, Kelly		
Preferred First Name:	Kelly	
Job Responsibility:	SMB Marketing Manager	
Phone (Primary):	216-664-7226	
Email:	kschmoltz@us.ibm.com	
Business Address:	MobileEmployee Cleveland, OH	

It looks like we've captured Kelly's record on Ignite400. Cha-Ching!

OK, we've done enough with the IBM Telephone Directory for today. If you are a V5.0 user and you'd like a better perspective on how to set ITD and LDAP up for that version of WAS Express, continue reading the chapter appendix, which is coming up next.

Chapter 19 Appendix

WAS 5.0 Express ITD Installation

The first order of business in setting up WAS 5.0 ITD even before creating the WAS instance in which to run the ITD application, is to assure that the Lightweight Directory Access Protocol (LDAP) is up and running. To do this, you need iSeries Navigator.

Go to the TCP Server section with iSeries navigator and look for an application called Directory. If it is not running, start it. When you get this done, you can proceed with the running of the ITD setup script.

In V5.0 the LDAP sub Wizard in the Create WAS Server Wizard does not exist. Therefore, you must set up the IBM telephone directory with LDAP prior to creating the WAS instance. In my book, “[The iSeries WebFacing Pocket Guide](#), Lets Go Publish, 2003, this process is spelled out in detail. In this book, we are concentrating on the newer WAS V5.1, which does not need the Setup routine.

To run the script, get into the Qshell mode from the command entry screen by entering the following

QSH

From here, make a determination as to whether these options are OK for you and then run the following script:

```
itdsetup -w passworda -a passwordu  
-u -o
```

Script Parameters

There are a few more options than the four we chose to use, and you certainly may want to change the user names and passwords for your shop.. As we explain the meanings of all the parameters below, you can get an appreciation for why we chose only the `-u` and the `-o` options:

-w This is a required parameter. It is your LDAP administrative password.

-a This is a required parameter. It is the `iweAdministrator's` password. The `iweAdministrator` profile is created during the LDAP installation, so choose a password for this user name with this parameter. This super user is authorized to manage all IBM Telephone Directory data in the LDAP directory.

-u Update LDAP schema. You would set this option if you have not set up ITD (Telephone Directory) previously. The first time that you run `itdsetup` to a new LDAP server, you must specify this option to set up the schema for the IBM Telephone Directory data on that server.

-o Open Enrollment. This option determines whether users can register and delete themselves in the telephone directory. If the option is not specified, administrator access is required and you are required to log on before registration. When you are playing with this application, as we are to determine its applicability in our shops, the open enrollment methodology is recommended.

Other options are spelled out in an IBM Redpiece titled [WebSphere Application Server – Express V5.0 for iSeries](#). It is an excellent reference for anybody moving to WAS Express from any other platform or no platform. IBM's number for the Redpiece is SG24-6555-00. It is available on the Web at www.redbooks.ibm.com. Just search for “WebSphere Express” and it will come up. As of this writing, the date of the book was getting old, February 2003. Hopefully, IBM will update it with good stuff about WDSv5.1.2 and

better. When you download the book, you can then browse the manual or bring it down as a PDF file to your PC.

In addition to being IBM's only bible on WAS iSeries Express, this Redbook is a great reference for the WAS Express installation and the ITD application installation. We do not present all options here in detail since our objective is to provide a simple, non-confusing fast path to implementation. If this Redpiece were not available, our job would be much more difficult. However, the Redpiece is very good and it is free for the taking on the IBM Redbook Web site

Chapter 20

Using WebFacing: Starting a WebFacing Project

WebFacing Organizes Work in Projects

By definition, a project is a series of tasks necessary to complete a mission. When the mission is WebFacing, the project, quite simply, is a WebFacing project. All of the work that the WebFacing Wizard does on your behalf is placed in the WebFacing Project folder for each WebFacing application that you create.

Once again, this folder will contain all aspects of the WebFacing project and quite handily, it is given the same name as the WebFacing project itself. In this PC folder, the WDS Sc Eclipse WebFacing plug-in stores all of the converted resources for the WebFaced application. The items that arrive in the folder through the conversion process include servlets, JSP files, Java files, static HTML documents, XML, etc. Additionally, the folder also holds any associated J2EE metadata. Metadata is data about data, and data about what to do with data. This is necessary for a full description of the project.

The Connection Wizard

One of the first things that you should do when you get your first WDS_c functional panel in the Remote Systems Explorer Perspective is to bring down the Fixes. Since we have already done that in Chapter 9 (Figure 9-25), the second most important item is to establish a connection to your AS/400.

If you are on a LAN it should be no issue with token ring or Ethernet. Most of the time I connect remotely with DSL over the Internet over a VPN connection and that is OK also. If you have a Client Access or iSeries Access connection that is LAN based (not twinax or ASCII) you should have no physical networking issues. You just need the IP address of your host or the DNS name or an entry in your workstation *hosts* file pointing to your IP address.

Once you've got that, your next step is to create a connection from WDS_c to the AS/400.

Figure 20-1 Start the Create Connection Wizard



To start the Connection Wizard, click the plus sign next to the iSeries line in the RSE perspective as shown in Figure 20-1. You will be taken to a panel very similar to that shown in Figure 20-2 below

Figure 20-2 Launching the Connection Wizard

New

Remote iSeries System Connection

Define connection information

Parent profile: DP000BK

Connection name: 192.168.0.250

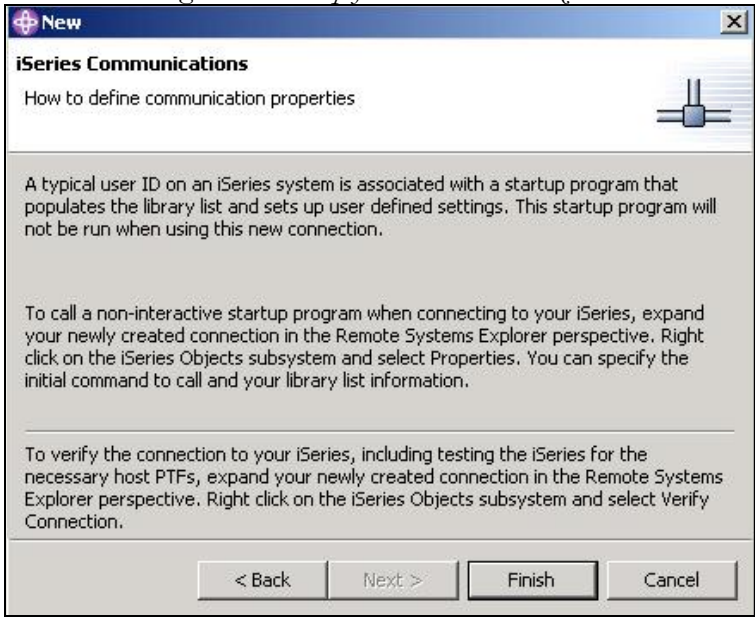
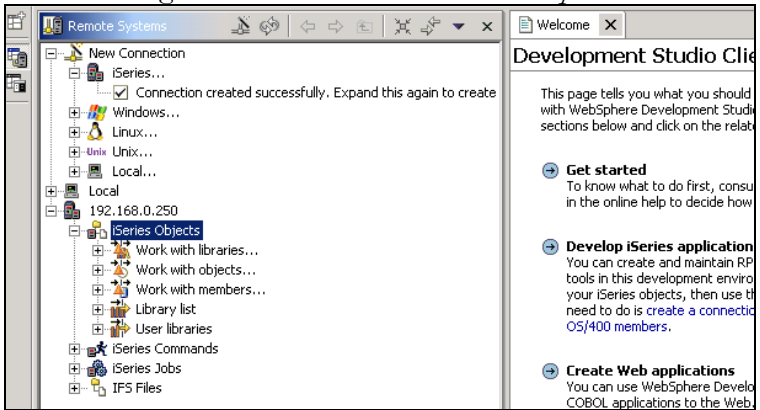
Host name: 192.168.0.250

Description: Connection to Host A5/400 (iSeries)

Verify host name

< Back Next > Finish Cancel

The Parent profile can be left as is. It defaults to the name of the device on which you are running WDS. There are no more entries for the connection Wizard. When you have completed your work on this panel, if you hit “NEXT,” you will be taken to a Help Panel for the connection Wizard as shown in Figure 20-3 below. If instead, you hit Finish, the connection will be created and you will be taken to a panel similar to that shown in Figure 20-4.

Figure 20-3 Help for Connection Wizard*Figure 20-4 Connection to iSeries Completed*

Since we used the IP address to label our connection, the WDS Client Eclipse Remote Systems Explorer (RSE) information for navigating

the AS/400 (iSeries) connection is shown under the item 192.168.0.250. If you hit the “+” sign next to Work with Libraries, for example, you will note quickly that this is functionally about the same in function as the Work with libraries option in PDM. When you get this far, even before starting your WebFacing project, it would help you to click on some of these options to understand the tremendous facility packaged with the WDS Sc Eclipse environment’s Remote Systems Explorer (RSE).

Library Lists

As you will see when you begin the WebFacing Wizard, you can select specific libraries to view or you can grab all the libraries in the library list for the job. In the panel shown in Figure 20-4 above, you see a section for Library list. If you need to add some libraries to the list for this iteration of WebFacing, do so before you get into the WebFacing Wizard so that the libraries will be available in your list while you are using the Wizard.

When we get into the Wizard and you choose to select from the library list, you will be able to choose the library first, then you can select the source file(s) and then you can pick the specific display file source members that you want it to convert (WebFace).

Creating Your First WebFacing Project

Now, that you have been introduced to the WDS Sc Eclipse Intelligent Development Environment (IDE), a.k.a. the WorkBench, and you now know about its notion of projects, perspectives, views, and tools, and you have your connection to the AS/400 created, let us start the WebFacing process by taking the following clicks from any WDS Sc V5.1.2 Eclipse perspective: If you are using a different version of WDS Sc, the work is similar but I would advise using the latest and greatest. Near the end of 2004, V5.1.2 was clearly the version to have.

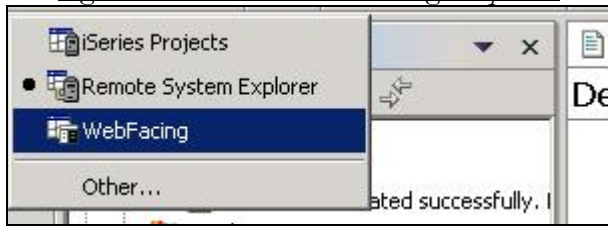
Start by clicking the icon to “Open a Perspective” on the left side of the RSE indented view as shown in Figure 20-5 below:

Figure 20-5 Select the Open Perspective Icon

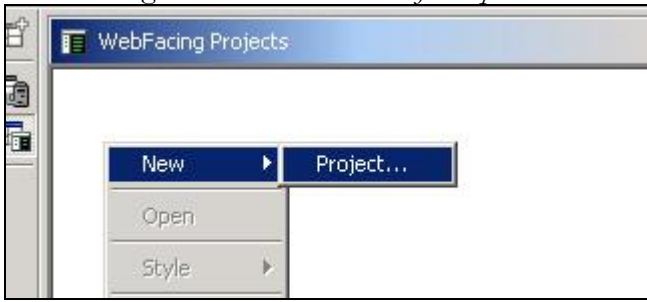


Our objective in performing this task is to change from the RSE perspective (collection of views) to the WebFacing perspective.

Figure 20-6 Choose the WebFacing Perspective



Select the WebFacing item from the panel in Figure 20-6 and release the mouse button. The WebFacing perspective will open with no projects shown. It will be empty the first time as shown in Figure 20-7 below.

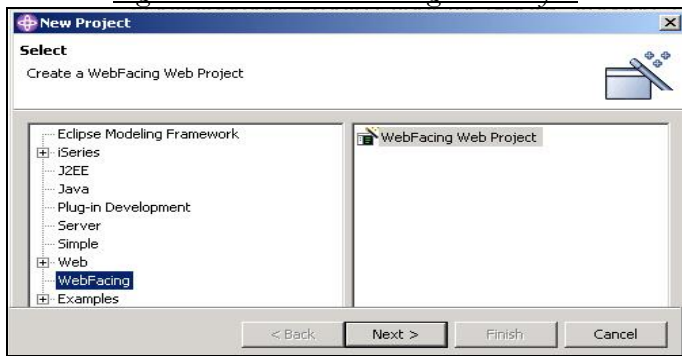
Figure 20-7 Select New Project Option

Once within the WebFACING perspective, right click in the white space as shown in Figure 20-7 above and select the New Project option

The clicks for this are shown below:

Right Click>> New>> Project

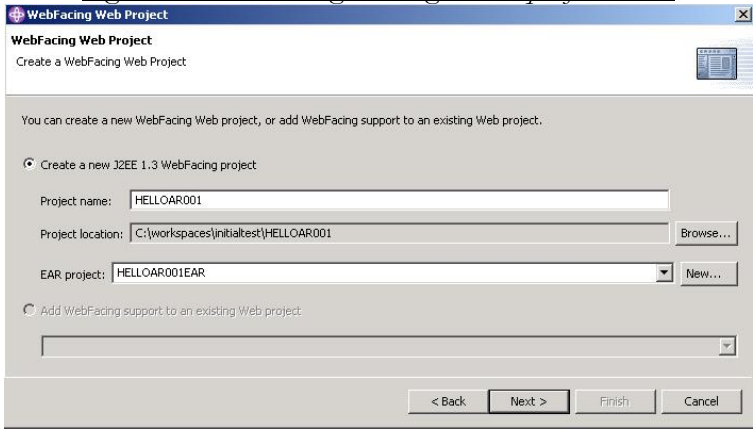
A panel similar to the one shown in Figure 20-8 will appear.

Figure 20-8 Select WebFACING Web Project

From here, (Figure 20-8) pick the WebFaced Web Project option on the right panel as shown.

When you select the WebFaced Web Project item, the WebFaced Wizard officially begins to do its thing. You can see this in the dialog in Figure 20-9 below.

Figure 20-9 WebFaced-- Filling in the Specifics Part I



The first part of the Wizard is to find out what you would like to call your project:

Fill in the Blanks

When you fill in a name for your project, by default it serves as the context root. You will append the context root to the domain name in your browser URL in order to access the WebFaced application. As you can see, in Figure 20-9, you also give your project an Enterprise Application project name. However, you do not have to type this. While you are typing the Project Name, the system types the EAR project name.

Project and Enterprise Archive (EAR) Project

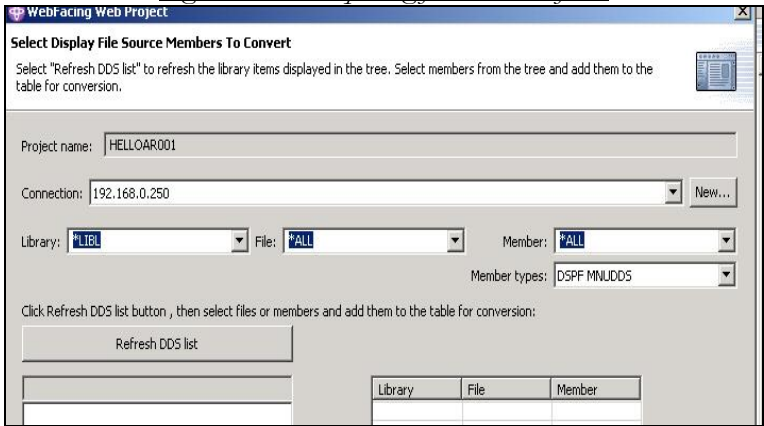
Enterprise Archives part of the J2EE specification and, just like Web archive files (WAR), they help make applications easier to export, import, and deploy on the AS/400 or iSeries servers. In addition to creating a project, WDS's WebFacing perspective also creates an EAR file for you. As you will see as we progress, the EAR file is a like a big container s that you whole project can be moved from system to system and from WebSphere Server to WebSphere server merely by referencing it by one name.

When you have completed the panel shown in Figure 20-9 above, click on *Next* to continue with the WebFacing Wizard, You will then be presented a panel similar to that in Figure 20-10.

Define Source for DDS

As you can see in Figure 20-10, you must specify to the Wizard the location of the DDS and other necessary information about the material that you wish to convert (WebFace).

Figure 20-10 Preparing for DDS Refresh



Bring Down Your DDS List

The next step in the process, as shown in Figure 20-10, is to provide the connection name, the library (or library list *LIBL), the source file, and the member. This is the information that the AS/400 needs to return a list of libraries or the library contents (depending on what you ask for). From this list, you can navigate to the DDS for the panel that you wish to convert. As you can see in the panel shown in Figure 20-10, we are asking to see the **library list** and **all** source files in the libraries and **all** members in the source files. Remember that you must have your library in the library list if you choose this option. Otherwise, you have to select one library at a time.

As you also see in Figure 20-10, we changed the *LIBL designation for Library to the HELLO library where we want to begin our conversion. If you have lots to convert, you may wish to use the *LIBL option instead.

To start navigating through the iSeries object list to bring down the display files to WebFace, click on the “Refresh DDS List” button. You will then get an AS/400 sign-on panel similar to that shown in Figure 20-11.

Figure 20-11 Sign-on Panel for iSeries or AS/400



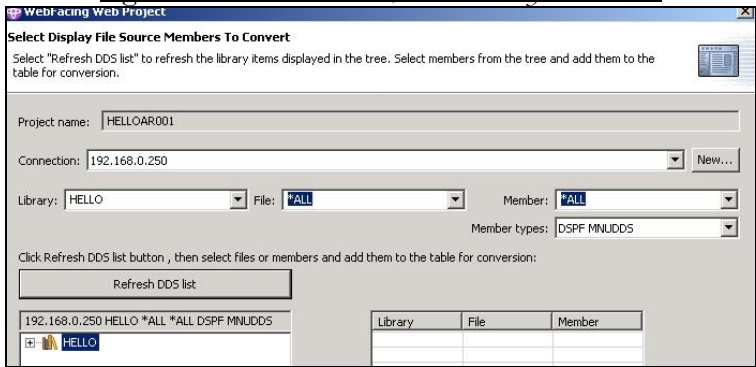
Sign On First

When you sign on, you will be taken to a panel such as that in Figure 20-12, in which you will be asked to select the source DDS to convert.

Note:

Figure 20-12 shows a crunched up version of the real selection panel. Since this is not an 8.5 X 11 book, it makes it easier for you to see when we enlarge the active portions of screen snapshots and remove wasted space. But, this does affect the overall look of the panel.

Figure 20-12 Select the AS/400 Library to Convert



Pick Your DDS Library

Find your library in the library list (HELLO). Then, keep hitting plus until you get your QDDSSRC file and the DDS member that you want to convert with WebFaced.

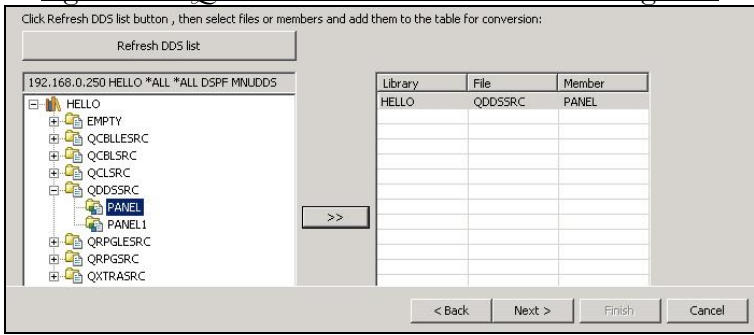
Tip:


What if your library is not in the library list? Unfortunately, at this point, you would have to specify the name of the library as we have done or you would have to go back to the RSE perspective and under the library list option, you can add your libraries to those that will be shown during WebFaced. Of course, you can also change the job description of your signon profile to include the libraries that you would like to convert.

Drilling Down for DDS

OK, enough for library list changes. Let's get back to WebFacing using the information in the panel in Figure 20-12.

Figure 20-13 QDDSSRC Member PANEL For Migration



As you can see in the panel it is already opened and the member named PANEL is selected and moved to the selection side. When you originally enter this library, however, it is all closed up in the view. So, you would start at the Hello library, and open the plusses by clicking and drilling down to the PANEL format for this case study. Select PANEL as we have done. Then click on the little double arrow button  to move it to the right side.

Your panel will then look almost exactly like the one shown in Figure 20-13. Note the right side shows the work that you have already completed in this WebFacing selection section of the Wizard. Our work is already done at this point since we have just one panel to convert.

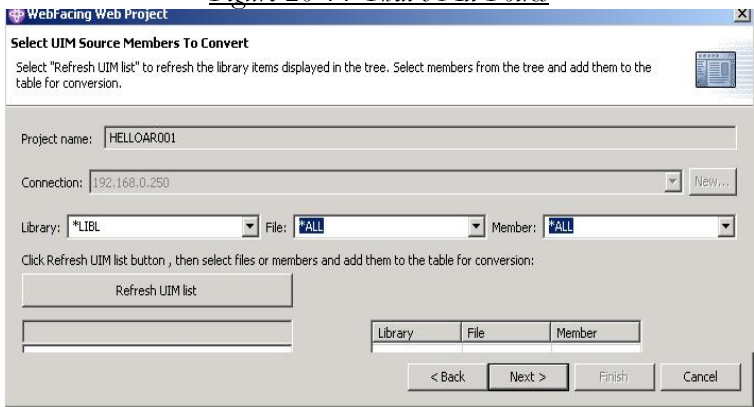
As you can see, the library that contains our DDS is aptly named HELLO. You can see it is the only library in the Window shown in Figure 20-13. The WebFacing navigation process brought us through

the Hello Library to the QDDSSRC source file and on to the member PANEL as shown on the left side of Figure 20-13.

Click on Conversion Items

Once you get to this panel, to repeat, use the double arrow button in the middle as discussed above (>>) to select the PANEL DDS source member for WebFacing. As you click the double arrows, the objects that you are selecting will appear on the right side of the screen as shown (highlighted) in the above panel. Notice the latest selection for PANEL is highlighted in blue (reverse image for you b/w viewers). You can continue navigating in your projects to select more DDS for conversion. In this Advanced Hello World case study, you have done enough. Feel free to click the Next button to continue. A panel similar to that shown in Figure 20-14 will appear in case you want to convert UIM (Menu) objects.

Figure 20-14 That's All Folks



Menu's Anyone?

At this point with the Wizard, you receive a blank panel such as the mini version in the panel shown in Figure 20-14 above. This is asking if you have any UIM type source to convert. Again there are no menus or Help in this case study project, just the DDS that we already

selected. To continue with the Wizard after your first DDS selection, click on the Next Button.

You have now completed the first phase of WebFacing, the selection of the items to convert. In Chapter 21, you will continue the WebFacing process by picking the CL program that drives the application and you will spiffy the panels up a bit with a Web style. Following these actions, you will be in a position to tell the Wizard to go ahead and “*Create a WebFaced application.*”

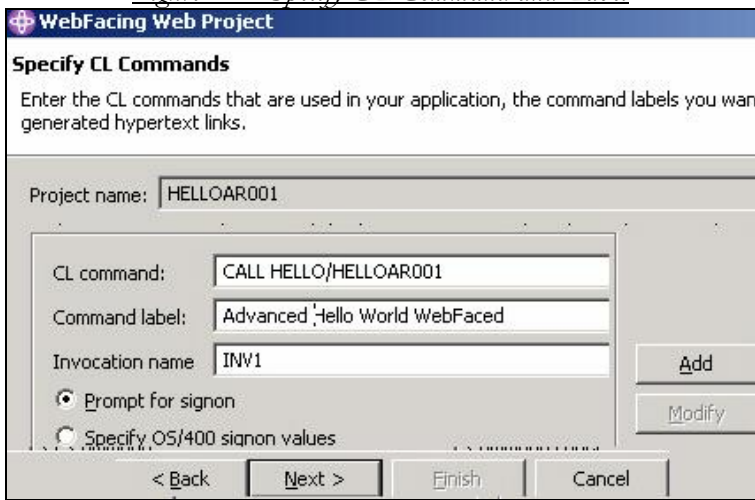
Chapter 21

Using WebFacing: Select CL Driver and Style; Convert DDS

CL Driver

At this point, the DDS has been selected along with any menus and help text that you want to be WebFaced. The Wizard knows that migrated DDS alone does not make a Web application. Some program someplace must drive the application. In this chapter, you get to pick the program that the WebFacing Wizard will use to link the migrated panels into a Web application.

So, when you hit the Finish button in the panel shown in Figure 20-14, you will be greeted with a panel similar to that shown in Figure 21-1.

Figure 21-1 Specify CL Command and Label

Specify the WebFacing CL Driver

In this representation of the panel as shown in Figure 21-1, type in the CL Command that you want to call, and provide a Command Label for it. The command label you provide appears on the index / welcome page (JSP in WDS_c 5.1.02) for the application when it is called from the browser, so you might want to give this some thought. As you can see in Figure 21-1, we filled in the parameters as follows:

CL Cmd: **CALL HELLOAR001**

Cmd Label: **Advanced Hello
World WebFaced**

In the Advanced Hello World example, there really is no CL program that drives the application, since this application is simple and is all accomplished in RPG. Thus, we specify the CALL CL command with

the program HELLOAR001. This is name of the green screen RPG program used for the application.

COBOL WebFacing

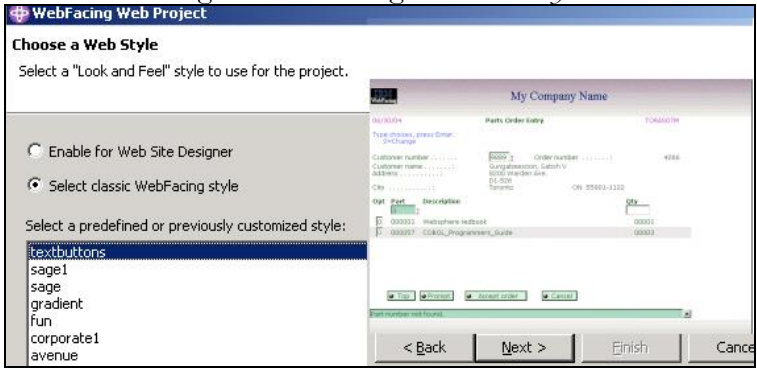
There is also a program called HELLOAC001 that we have in our code stable. It is a COBOL version and runs the same way. It uses the same display file, PANEL. At this point, either program can use the WebFacing objects that we are about to create. However, the name of the COBOL or RPG program must be specified here. We are going to choose the RPG program, though it would be perfectly OK for you to specify the COBOL program in this space if that were your preference.

After specifying the name and label, make sure you click the Add Button for this selection to take hold. It is easy to forget his step. When you click *Add*, the information you specified on the top of the page is taken to the bottom portion of the panel as shown in Figure 21-2.

Figure 21-2 CL Program Selection Panel Completed

CL Command	Command Label	Invocation name	User ID
CALL HELLO/HELLOAR001	Advanced Hello Wo...	INV1	*PROMPT

As a point of note, when you click the Add button in Figure 21-1, it is no longer highlighted. Instead, the Modify button would be highlighted. To proceed with WebFacing, from the panel in Figure 21-2, above, click the **Next** button (not shown in Figure) , and you will be taken to the panel shown in Figure 21-3.

Figure 21-3 Selecting Your Web Style

Making It Pretty With Style

The blurry image on the right side of Figure 21-3 gives you a clue about the look of the various styles that you may choose. Styles are like masks that guide the look and feel of applications. Just like a mask, when you put a style on your WebFaced application, your Web page looks lots different, based on the particular style that you choose.

By highlighting the name of the style, you get a general idea of how your Web pages will look. In this newer WDS version of WebFacing a few favorite styles of mine from the old WDT days, such as “spiffy blue,” have disappeared.

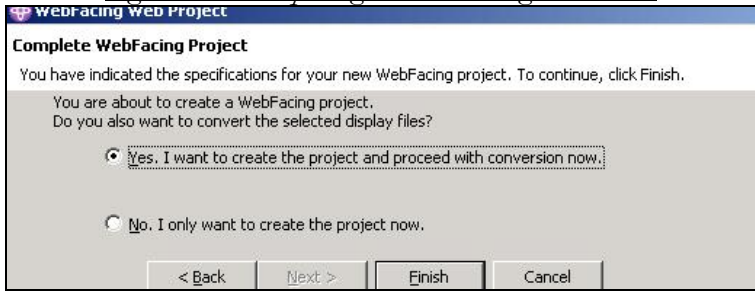
No More Spiffy Blue

I must admit that when I saw “spiffy blue;” as one of the original choices, I thought that Jim Sloan, the CL and QUSRTOOL great former IBMer had made a guest appearance in Toronto. That would have been nice. Spiffy was one of Jim’s key phrases in his fine presentations. Anyway, “spiffy blue” and others are “among the missing” in the WDS versions of WebFacing. I would hope that IBM increases the choices in the future, rather than decrease them.

In WDS V5.1.2, IBM added a new style called textbuttons. This is very handsome. However, unlike some of the prior styles, it does not

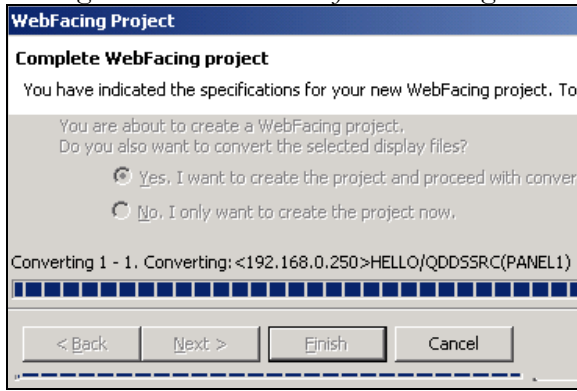
permit the PF key and Enter key to be mapped as buttons in the left pane. After we use this style in this example, I will show you a panel from a former version that has both the buttons and the PF keys enabled. For now, select your style and click on Next to continue with the panel in Figure 21-4.

Figure 21-4 Completing the WebFacing Conversion



Say Yes to Get it Done!

Select the option to create the project and proceed with the conversion now, and then click the Finish Button. On the way to building the project, the Wizard will show you some status messages, such as those in the panel in Figure 21-5.

Figure 21-5 WebFace Project Build Progress

When the project is built, you will see the completed WebFacing project name in the WebFacing perspective (Project View) as shown in Figure 21-6 on the left side of your WorkBench. You will also see the Conversion Log as in Figure 21-7. It will be on the right side of your WorkBench.

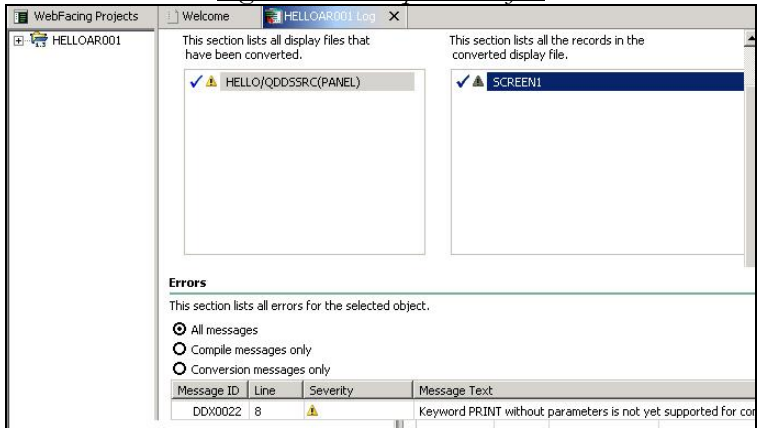
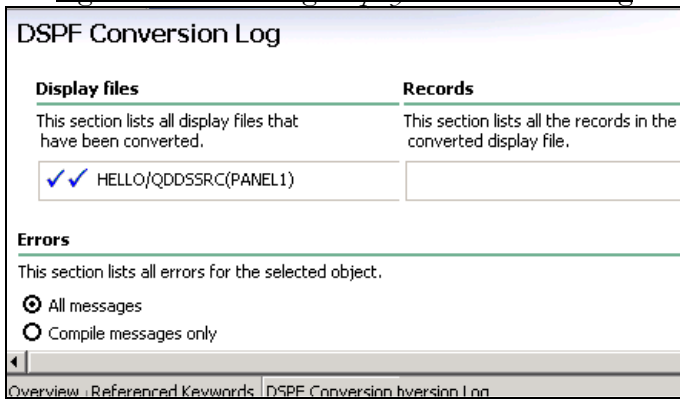
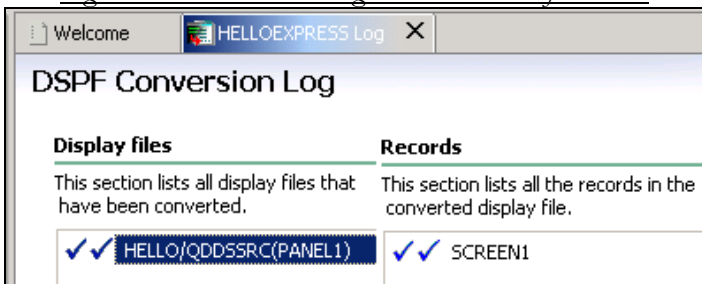
Figure 21-6 Completed Project

Figure 21-7 WebFacing Display File Conversion Log

Check Out the Conversion Log

If the names of the display file and the records do not appear in the log in Figure 21-7, for example, click on the check marks. You will then get information about how well the panel(s) converted (Figure 21-8). You will notice that there is just one error with this conversion.

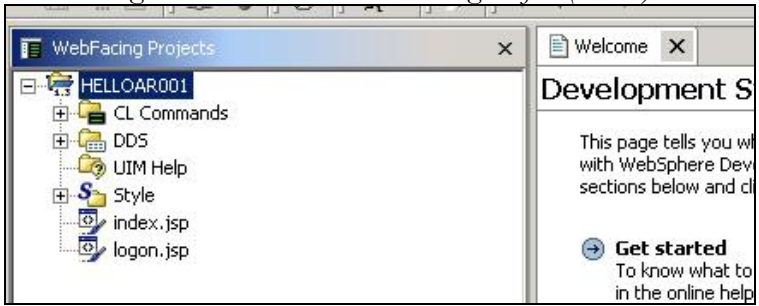
The PRINT keyword is still not supported so an error appears in the log. To see the error, however, you must scroll down from the panel shown in Figure 21-8

Figure 21-8 Conversion Log – Additional Information

Looking At Results

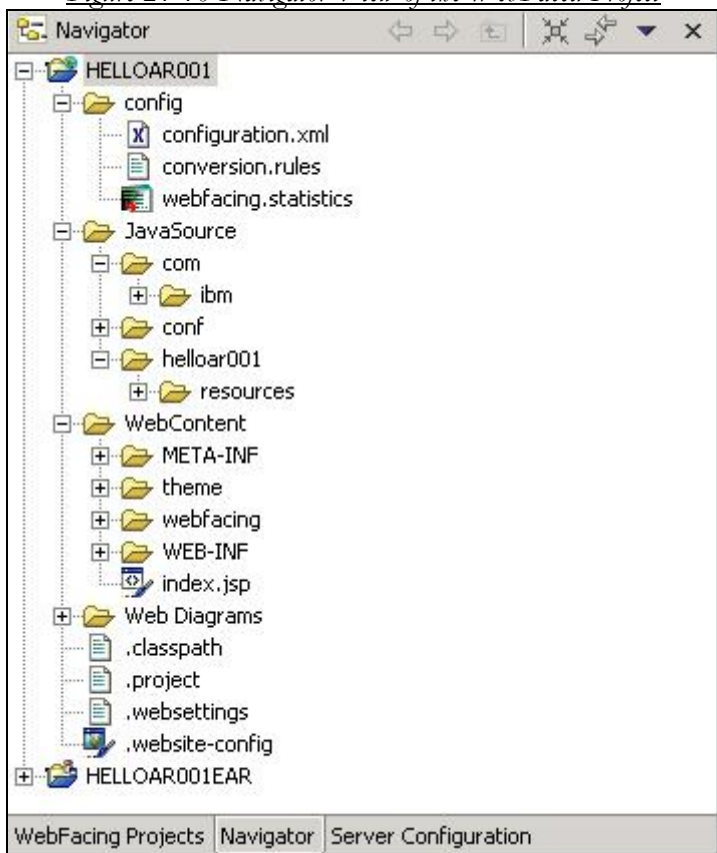
Now, take your mouse and expand all of the plus signs in the project view on the left pane of Figure 21-6, so that you can see what is really there. Start right next to HELLOAR001. Your converted WebFacing project view will expand to look like that shown in Figure 21-9.

Figure 21-9 Converted WebFacing Project (View)



C'est Fini

There it is, a completed WebFacing project shown in the WebFacing project view of the WDSv5.x Eclipse WorkBench. Click on the Navigator view to get a better perspective of all that is contained within a WebFacing project. You will see a panel similar to Figure 21-10.

Figure 21-10 Navigator View of the WebFaced Project

You may notice that on the bottom of Figure 21-10, there are three buttons:

WebFacing Projects
Navigator
Server Configuration

The view you are looking at in Figure 21-9 is the WebFacing Projects view. It is highlighted in that panel. Before you continue on to exporting the project to the internal WebSphere Express Server, click the Navigator button to get a different view as shown in Figure 21-10. Open all the items in the view and get an appreciation for all that WebFacing has done for you. Check this out in Figure 21-10.

Project Navigator View

Figure 21-10 shows a partially expanded Navigator view. Look at all the additional plusses that can be hit. One simple display panel can cause WebFacing to create many objects.

In Chapter 22, you will be asked to continue the conversion plan as we test the converted WebFaced project on the WDS*c* Eclipse Internal WAS Express server. For all intents and purposes, at this stage, you should know that WebFacing has left the building

WebFacing Has Left the Building

Along with Elvis, at this time there is no real reason for us to be concerned about WebFacing. The WebFacing Wizard has done its job and it has placed the results of its actions in the WebFacing projects folder on your PC. Now, depending on your use of the Eclipse WorkBench, it is up to you to do some more work with WDS*c* before you transfer the project to the iSeries server. This work will help assure that the project will run on the iSeries, without actually having to do the work to put it there.

Now, in a sense, we are going to take door number three from the page shown in Figure 21-10. When you click the Server Configuration button, the Navigator view changes to the server view with one entry titled *Servers*. We'll be working with servers in Chapter 22. However, if you have any prior WDS*c* or WDT experience, you can immediately see that the option to open door number three is far easier than with the older WDS*c* versions.

Congratulations on your first WebFaced project with WDS*c* Eclipse!

Chapter 22

Using WebFacing: Working with Internal Server to Test WebFaced Application

Testing with WDS Sc V5.1.2

Once you've got your WebFaced application converted on your PC, you can test it with your remote AS/400 without having to move the application to the AS/400 or iSeries WebSphere Express server. The V5.1.2 WDS Sc Eclipse WorkBench comes with several WebSphere Server versions at the V5.X level if you choose to install them. One of these servers is WAS Express Test Environment, which we installed in Chapter 9.

It helps to understand that any WebSphere server can communicate with any host system. The WAS Express test environment on your development PC is a bona fide WAS Express 5.1 server and thus, it has the hooks necessary to talk to a server program on an iSeries. In this case, the server program is the iSeries WebSphere server support for WebFacing. This support runs independently of whether the iSeries machine itself runs WebSphere Express. In fact, as we will say several times to make the point, to test WebFacing, you do not have to install WebSphere Express on your iSeries. I repeat--- **to test**

WebFacing, you do not have to install WebSphere Express on your iSeries.

Start TCP/IP WebFacing Server on iSeries

Before you try to run the project on your development PC, you better make sure that you have started the WebFacing Server code on the target AS/400 or iSeries. Again, your WebFaced application does not have to be resident on the iSeries and the WAS Express on the iSeries does not have to be started since it doesn't even have to be there. To make sure that your WebFacing server is active on the iSeries, you can issue a WRKACTJOB command and look for the following jobs in the QSYSWRK subsystem:

QQFVTSVR
QQFWFSVR

Both of these jobs must be executing on your iSeries server to run WebFacing in the internal testing environment as well as the production environment on the same system as your WAS express server. If you go to QSYSWRK and you find the servers missing from the list, you can set WebFacing attributes in iSeries Navigator to start the WebFacing server whenever TCP/IP is started so that, barring any problems, the servers will always be there when needed.

In iSeries Navigator, you can set this parameter as follows:

Network>> Servers>> TCP/IP>
WebFacing

Right click on WebFacing for the *properties* display and then place a checkmark to automatically start WebFacing. Just click on the box in

the properties display that triggers WebFacing to start when TCP/IP is started.

You can also right click on the WebFacing item in iSeries Navigator and select the Start option to immediately start the WebFacing server.

IBM has also supplied a green screen command so that you can start the WebFacing server on a 5250 green screen command line as follows:

STRTCPSVR SERVER (*WEBFACING)

Unfortunately, there is no green screen interface to automatically start the server when TCP/IP starts.

Pardon a short commentary but again I am compelled. IBM has decided that even though its loyal constituency likes the green screen approach, the company has a reluctance add even simple functions to the native interface. There is no question in my mind that this would be a simple change and it would make it simpler for you and I to administer. With a native command line interface, we would not have to bring out the big and bulky and often Windows-unstable Operations or iSeries Navigator to perform this and other simple functions. Come on IBM, there is nothing wrong with having a complete native interface for important functions! Enough said!

In addition to being able to start WebFacing from iSeries Navigator, you can also stop it from iSeries Navigator. IBM also created a green screen command for ending the WebFacing server from a command line on your iSeries or AS/400. You do so by entering the following command:

ENDTCPSVR SERVER (*WEBFACING)

Remotely Testing WebFaced Applications

With very little work, and no WebSphere server work, you can prove your WebFaced application works using the internal WDS_c WebSphere Express server. It bears repeating that you don't even have to have a WebSphere server (any version) installed on your iSeries. After you test with the internal WebSphere Express Test server on your own PC, if your WebSphere Express or other WebSphere server implementer cannot get your application running on your AS/400 or iSeries WebSphere Express Server, then you know for sure that it "ain't" an application problem."

So that there is no mistake about what I am saying, here it is again. You can unit test a Web or a WebFaced application that communicates with an ILE program on an iSeries host without deploying the JSP files and servlets to the host iSeries. Unlike WDS_c V4, with WDS_c V5, you don't even have to set up communications using the iSeries Host Information wizard. When we suggest that the WAS Express in WDS_c V5.x is well integrated, that's exactly what we mean.

No WebSphere Server Required

You no longer have to mumble the question to yourself: "Wouldn't it be nice if I could test my application on the PC without having to use WebSphere on the iSeries at all? Why? Because you can! Moreover, now that you have created your JSPs and other WebFacing objects with the Eclipse-based WDS_c WebFacing Perspective, the ease of testing will be another unexpected surprise. With no iSeries server work at all, you can test your application. I like saying that. Of course, you do need to have the right perspective. And, this time, unlike Version 4.x of WDS_c, you do not even have to change your WorkBench perspective.

No More Server Perspective

With the V4 Server perspective, for example, you had to tell the Eclipse IDE that you wanted to test locally. You had to use a special perspective called the Server Perspective to define information about your iSeries server. Following this, you would select your Input panel, and then you would select, “Run on Server.” It used to be reasonably easy with WDS V 4.x. With V5.x, and later, it is even easier.

All you’ve got to do is right click on the WebFacing Project name while in the WebFacing perspective, and select “Run On Server.” With Version 5.X, the rest just happens as you are about to see.

Run on WorkStation Server (PC WAS)

Before you actually run the application on the V5.x WDS Eclipse internal test WAS Express, let’s take a quick pause to see what this application once looked like when it was running in green screen mode on your AS/400 or iSeries. The sample application is very simple. As you can see, there is one panel defined with one input field and one output field and lots of constant “fluff” information.

Show Me How It Looks Regular!

By the way, past tense is inappropriate for the green screen version. You do not have to change a line of DDS or RPG code to make this application run under WebSphere Express from its green screen form. While you are running the application from your browser, which you are about to do, you can, at the very same time, call the same program with the green screen interface as show in Figure 8-1 and Figure 8-2 below. When you do, the only copy of the object program in memory handles both user interfaces at the same time.

Green Screen Advanced Hello World

The application starts with a request for a language. The idea is to take a sentence with Hello World and display it in another language. The requested language is the language in which the sentence is to appear. However, since I never actually translated the languages in my database, the application sends a little message instead helping you know that you've got it right. If you ever use this application as a demo in your own shops, feel free to prime the database with the foreign translations. The application will work exactly the same.

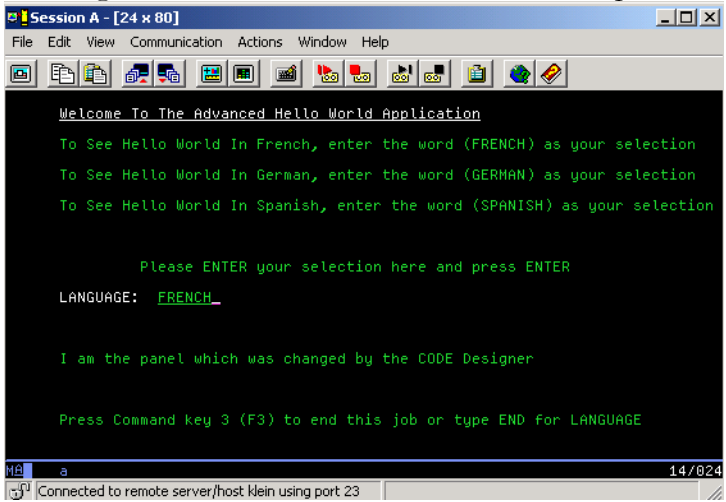
Calling Advanced Hello World

The data from the input field is used to look up a record in a database that matches the language key. The database is on the AS/400 or iSeries. The matched record response from the database lookup is written as an output field on the panel.

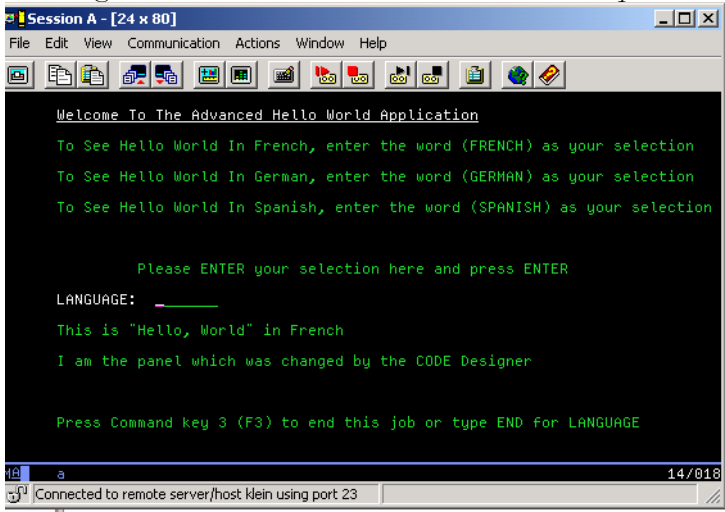
We're now ready to give it a whirl. The way you start the application after assuring that the HELLO library is in your library list, is by keying the following command on the command line of the iSeries or AS/400 server:

```
CALL HELLOAR001
```

The next panel that is displayed looks just like that shown in Figure 22-1

Figure 22-1 Hello World Green Screen Panel - Input

Start by typing the word *FRENCH* for your input, and press Enter. You will then see the panel as shown in Figure 22-2.

Figure 22-2 Hello World Green Screen Panel- Output

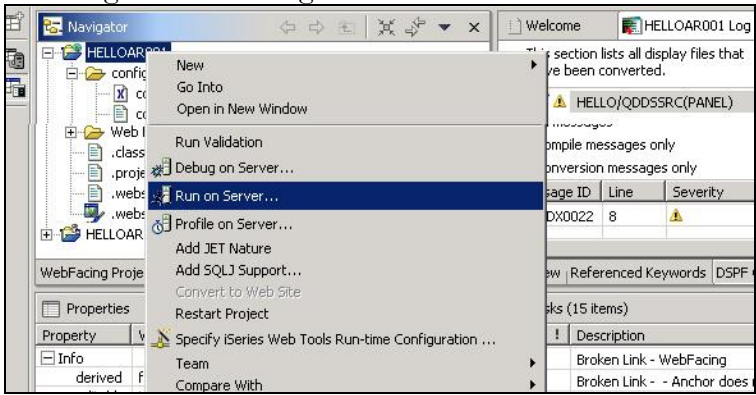
In this particular case, *FRENCH* is one of the records in the database that program HELLOAR001 fetches. It then writes the contents of the DB field (MESSAG) to the display panel as output.

Take a good look at this screen sequence, since it is the same sequence that you will see when the WebFaced application is run on the WebSphere Express server.

From Green Screen to PC

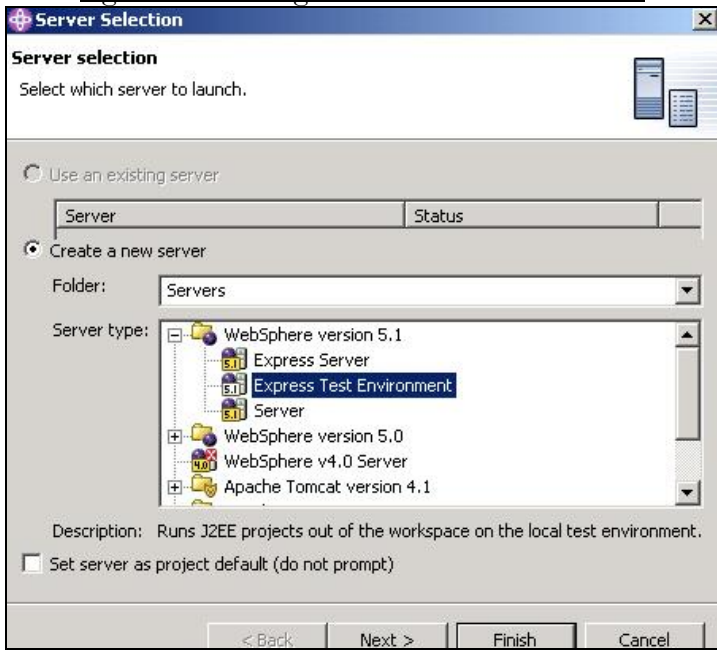
So, now that we have seen the application in raw green screen form, you know enough to run it on the internal WAS server. To do this, right-click the WebFaced project that you would like to test, and then select *Run on Server*. The V5.1.2 clicking and menu action for this is shown in Figure 22-3.

Figure 22-3 Running on Test Server in V5.1.2 WDS

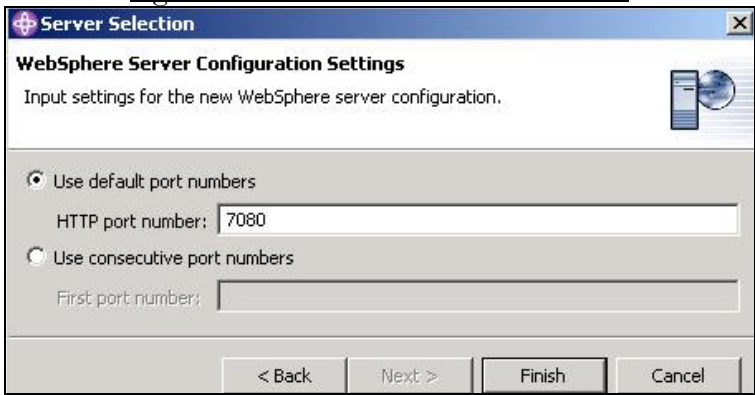
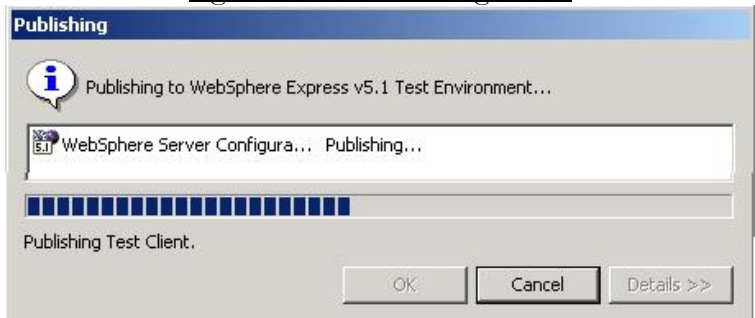


WDS Eclipse Tests Easy

All you have to do now is select the item directly from the WebFacing perspective's drop down menu as shown above. The first time you say *Run on Server* in V5, you select a test environment server configuration. Notice that there are many more action options in the drop down list than just *Run on Server*. For the purpose at hand, when you select *Run on Server*, you will be taken to a panel that looks very similar to the one shown in Figure 22-4.

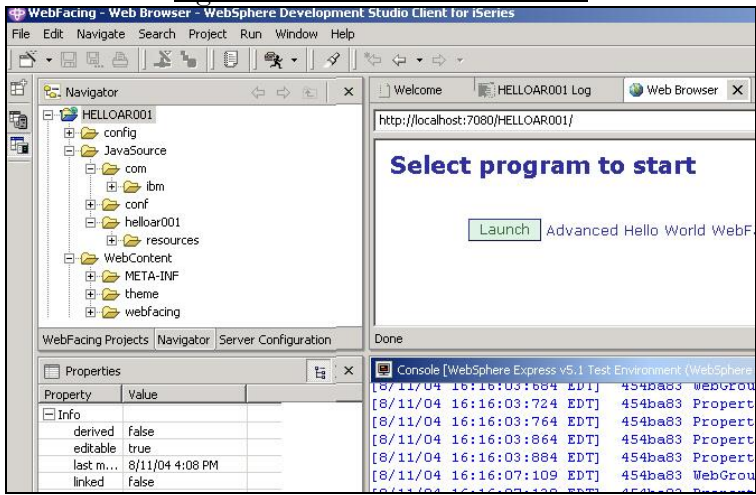
Figure 22-4 Selecting the WAS Server Environment

As you can see in Figure 22-4, you get to select the server type upon which you want to test the WebFaced project. In this case, select the WAS Express V5.1 Test Environment. When you click Next, the Wizard shown you the local ports it is to use. Clicking Finish from the panel shown in Figure 22-5 creates the Test server configuration and sets the process in motion to publish the WebFaced project objects to the internal server that you just configured. The publishing is shown in Figure 22-6.

Figure 22-5 Look at the Localhost Port 7080*Figure 22-6 The Publishing Process*

Start WebFaced Application on Internal Server

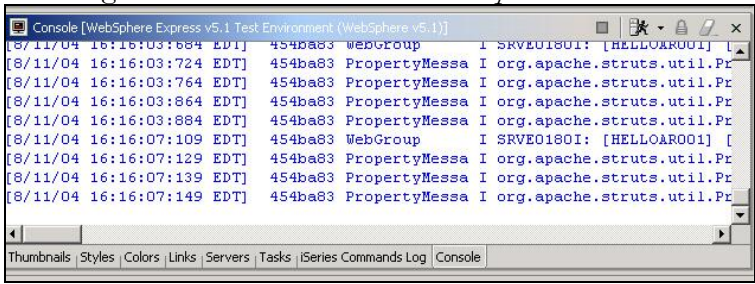
When the project is published, the WAS Express Test Environment server automatically starts up with the initial page of the WebFaced application as its Welcome Page. The Welcome Page comes up in a browser view within the upper right panel of the WDSclipse Workbench as shown in Figure 22-7.

Figure 22-7 Internal Browser Window

The Select Program to Start Panel is a generic launch JSP that you can change and incorporate in your static part of your Web Site. On the rightmost upper view of the WDS Eclipse WorkBench, as you can see in partial page capture in Figure 22-7, there is a Web Browser launched. The browser is part of the WebFACING perspective. The WebFACING Welcome Page is fairly stark and simple as you can see. It gives just the option of launching the application

Before we run the application, take a look at the Console view in the bottom right side of the WebFACING perspective in the WDS Eclipse Workbench. This is a bit larger in Figure 22-8. Notice the slew of log notations describing the process of opening the WAS Express test server. This is the real Console for the WAS Express testing version that is now running on your PC. Immediately after the server comes up, and before any application is launched, the last line of the Console Log always says that the “server is open for e-business.” That’s a nice warm thought.

In other words, your project has been published to the internal WAS Express on your PC and the server and the application are ready for testing. A few seconds after that, the internal browser contains the launch panel and you are set to go.

Figure 22-8 Console Internal WAS Open For Business

Run App with Internal WDS Sc Browser

From the panel shown in Figure 22-7, let's click on the launch button to run the application. When you click on the button, the first page of the application will be launched.

Well, not exactly. This is an AS/400 and we told WebFacing through the Wizard that we wanted to force the user through a sign-on verification sequence. To satisfy this requirement, the next panel that you see after launching this application is the sign on screen as shown in Figure 22-9.

Figure 22-9 Signon Screen for WebFacing Application

Sign on for WebFacing Test

When you sign on to the AS/400 or iSeries system, you will be taken to the first page of the application as shown in the page in Figure 22-10. The application is running on your PC WAS Express Test Environment, but the WAS is getting its data from the iSeries or AS/400 through the WebFacing Server that you started.

You may recall that the PANEL display file format SCREEN1 is the first panel in the application and it requests that a language be typed for translation. The panel using the textbutton style looks like that shown in Figure 22-10. If, however, we chose to use the avenue style, it would look as shown in Figure 22-11

Figure 22-10 Hello World Input Panel – from iSeries

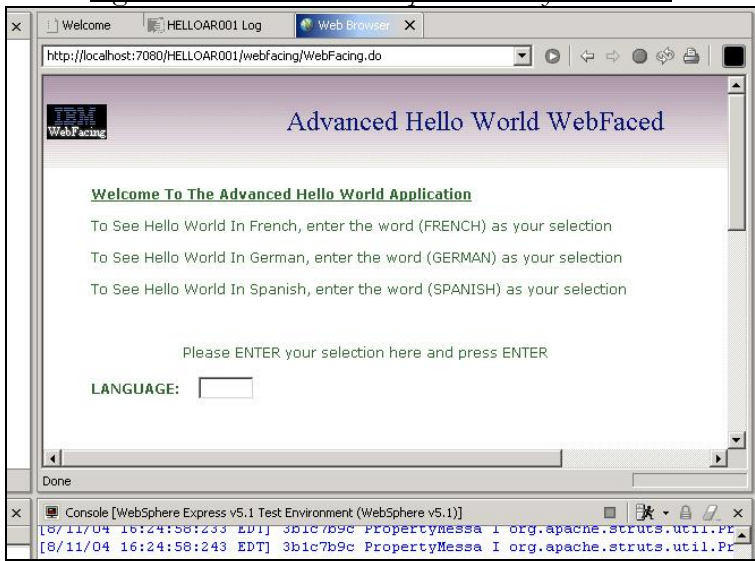
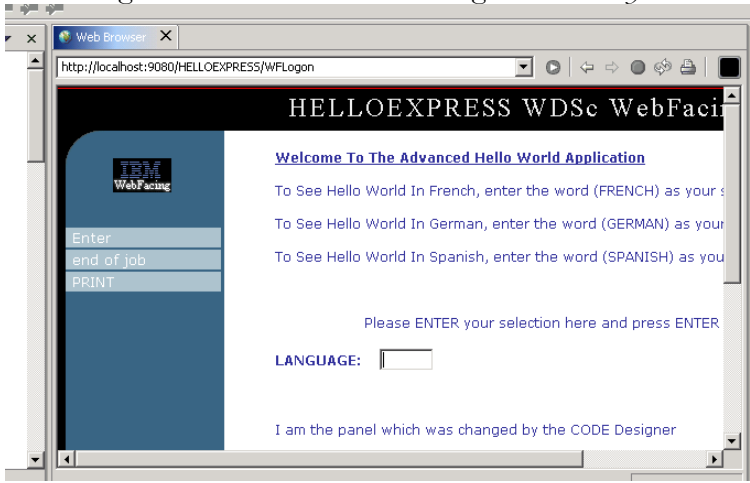


Figure 22-11 Initial WebFaced Page –Avenue Style

WebFaced Initial Input Process

As you can see both versions of the panel are quite a bit more attractive than their green screen counterparts. Moreover, the text for the Enter and Function keys are available in the Avenue style in the left frame of the panel in case you want to drive the application with mouse clicks instead of Enter and PF keys. Additionally, the new WDS also incorporates plug-ins for the keyboard, thereby enabling the Enter key and the PF keys to be used in your browser session. Unlike earlier versions, which were mouse driven only, this nice new version can be driven with the PF and Enter keys if you choose.

To move on with the application, type FRENCH in the input field and press or click Enter. You will be taken to the panel in Figure 22-12.

Figure 22-12 WebFacing Output Panel

WebFacing Output Process

As you can see, we got the right answer after the DB lookup – just the same as when done via green screen.

Ending WebFaced Application

By clicking on the end of job button in the left pane of the pages with buttons, or by hitting F3 on any Web style page in which F3 is enabled, the program launched by WebFacing is ended and you return to the original page as shown in Figure 22-6.

Nice Job

OK how's that for a job well done? Please note that this success does not mean that we are finished with the project. Though the project tested well, this converted WebFacing project is not yet running on your AS/400 or iSeries WebSphere Server. We're about to change that in Chapter 23.

Chapter 23

Using WebFacing: Exporting WDS Sc WebFacing Project

Destination: iSeries WAS Express V5.1

Other than the IBM-built Telephone Directory application and the Survey Creator applications that ship standard with WAS Express V5.1, I would expect that the next most common type of Web application to be brought into WebSphere Express would be those applications, such as Advanced Hello World that we WebFaced in this book using the WDS Sc V5.1.2 WebFacing Tool.

In this exercise, we are at the stage where the application has been created and tested. In other words, you have already converted your display panel DDS into JSPs and a full Java environment for iSeries execution. It is now stored in the WDS Sc Eclipse HELLOAR001 project on your PC. This project folder and all of its objects are headed for the WAS Express on your iSeries or AS/400.

Extract Application From WDSc Eclipse Project

Once WebFacing, a process that runs on your PC, creates its output, objects in its project environment must be “exported” to a file structure that can then be “imported” into WAS Express for iSeries. In other words, the completed, already converted application must be extracted from the WDSc project environment to a special JAVA / PC file type (EAR). The EAR file can be created directly on your AS/400 or iSeries IFS, and, in a subsequent step it can be installed (imported) into a server instance running under WAS Express or another WAS V5 server product. However, many developers like to create the EAR file on their PC and then copy it to the iSeries IFS.

The WAS Express for iSeries Application Installation Roadmap

The major steps to create and install your HELLOAR001 WebFaced application to WAS Express (V5 and above) are straightforward. However, the devil is in the details. The major steps in total for WebFacing are as follows:

1. Create a WebFacing project
2. Select the DDS to be converted from the source files on your AS/400 or iSeries
3. Convert the application display file DDS to java objects
4. Test application on internal WAS server
5. Export the completed WebFacing objects (project files) to a PC file
6. Map a network drive from your PC to a holding area in the IFS (unless you export to the IFS)
7. Copy the PC file to the AS/400 or iSeries IFS via the mapped drive (unless exported directly to IFS)
8. Use the Manage Applications function to install the application under WAS Express. (Import)
9. Restart your WAS Express Server to prepare the application.
10. Test the Application from any browser that can reach the server.

Some may find it handier to combine steps 5, 6 and 7 into two steps. If you map the network drive prior to the Export, then you certainly can export directly to an IFS directory (share). From my experience, the WebFacing process is intensive enough, without having its output requiring a networked drive. I have found that if I export to a PC file first, the Export happens very quickly, rather than going directly to the IFS directory. Since most of my work is remote and with DSL, I may have different considerations than you would have in your shop. Once exported to your PC, the cut and paste copy to the IFS is a very simple and quick process. However, the choice is yours.

Start With Step 5 Now!

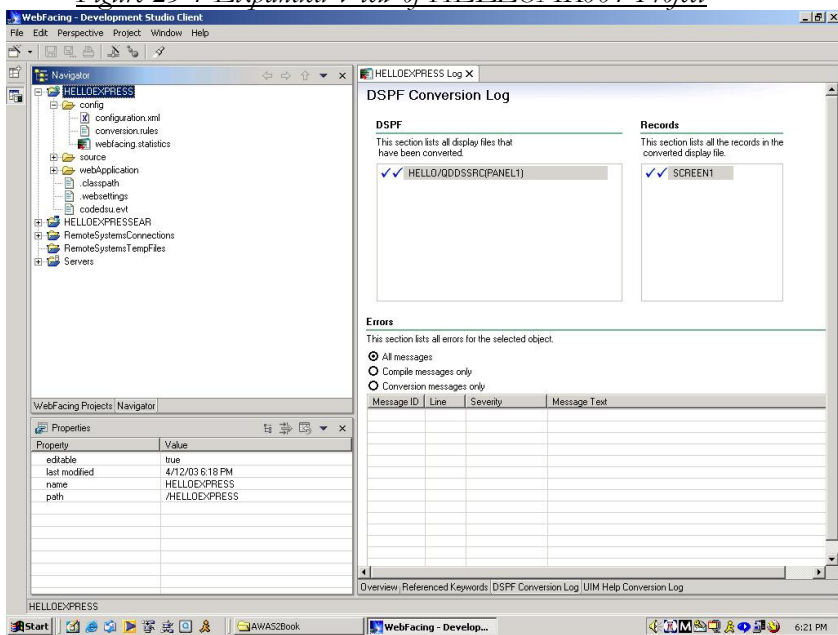
Since we have already presented how to WebFace, and we tested the results on the internal server, it is now time to deploy your application to the WAS Express platform. The case study in this Chapter starts by showing a close-up of your HELLOAR001 WDSc V5.1.2 WebFaced project after it is completed and all ready to be exported. In other words, the project has already been WebFaced and tested and we are getting ready to take it from the PC where it is now stored to

the WAS Express environment on your iSeries or AS/400. Thus, step 4 is already complete at this point of the game. This Chapter takes us through steps 5, 6, and 7. We then continue with Step 8 in Chapter 24.

Checking Out the Project

Figure 23-1 shows another “Navigator” view of the project. Figure 23-2 shows a close-up of the project in a collapsed form, waiting to be exported.

Figure 23-1 Expanded View of HELLOAR001 Project

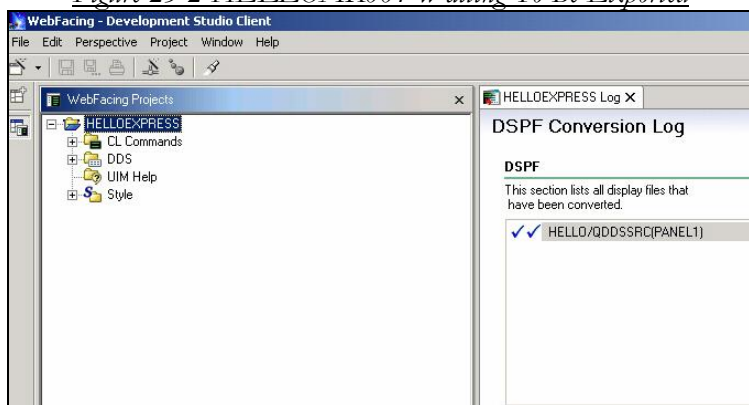


This is the partly expanded HELLOAR001 WebFaced project after it has been WebFaced. The WDSc V5 WebFacing Perspective is what you are looking at in Figure 23-1. Once the project is in this state, after internal testing, the next step is to bring it to the iSeries WAS Express.

Collapse the View

Since we have already tested the application in place and it works, it is ready to be exported. Take a look at the project panels. The first thing you can see above is the expanded project. To get a more concise view, and to give a better look at the project we collapsed it below. By hitting the minus signs in the project's Navigator view, it too collapses into something more docile looking. The collapsed project in the Project View is shown in Figure 23-2.

Figure 23-2 HELLOAR001 Waiting To Be Exported



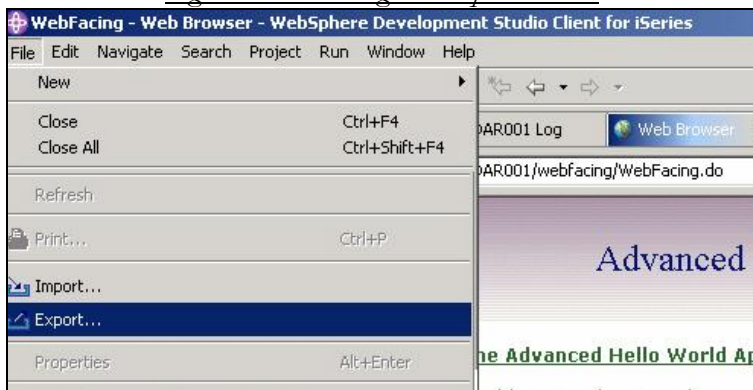
Step 5 WebFacing Deployment to WAS Express

Now, we're ready. To start the Export Process, select the project as shown in Figure 23-2, with a left mouse click. Release the mouse. Then, click the following:

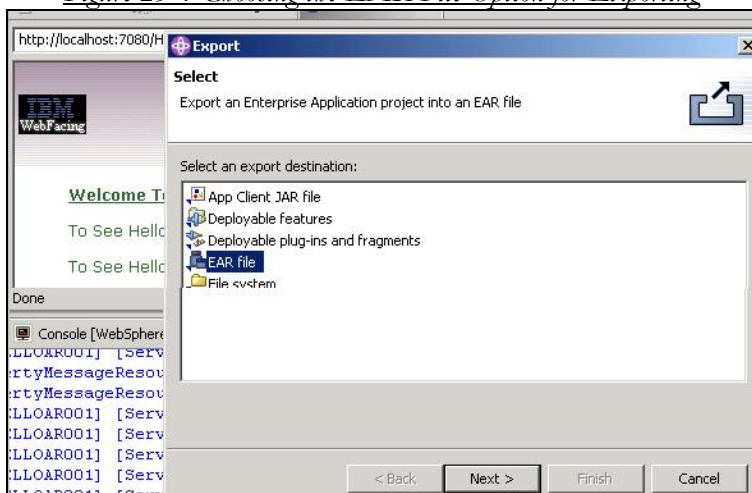
>> File >> Export

The start of the Export process is captured in the panel as shown in Figure 23-3.

Figure 23-3 Starting the Export Process



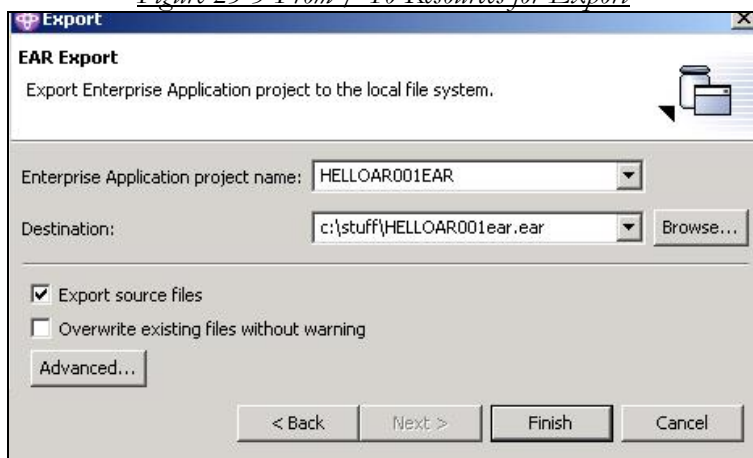
The next panel after choosing the Export option is shown in Figure 23-4.

Figure 23-4 Choosing the EAR File Option for Exporting

Pick Enterprise Archive File (EAR)

You should pick the option to export the WebFaced project to an EAR file. This is the best option for loading the project to an iSeries WAS Express server. An EAR file is a very inclusive file archiving structure (a super Zip file used in the Java environment). By selecting the EAR file format for Export, you enable the entire WebFaced project to be exported and deployed in one file with data about the project (metadata) included in the packaging.

In this way, the receiving WAS knows where to place everything. Thus, you are not required to deal with hundreds or thousands of individual conversion objects that would otherwise be exported separately. After you pick the EAR file mechanism in Figure 23-4, the next action for you is to pick the project name that you want to export as shown in Figure 23-5.

Figure 23-5 From / To Resources for Export

Specify the EAR Project Name

If the name of the Enterprise Application (specified during WebFacing) to export does not appear in the top window immediately, you can pull down the list box to select it. Of course, you can always type the name if you wish.

The other piece of information that you must provide to the WDS Sc WebFacing Export Wizard is the location to which you want to store the exported EAR file. In this case, as you can see in the panel shown in Figure 23-5, we have selected to store the file in a PC directory that we set up on the C:\drive of our PC. The exported file name is `HELLOAR001ear.ear`. This does not have to be the same name as the project.

Tip:

Suppose that you had mapped a PC drive, such as Drive K to an IFS share that pointed to a directory called `/INCOMING.` If you do not want the intermediate step of exporting to a PC file, then the bottom window can be filled out as noted below and the Export will take the project EAR file and place it on the IFS immediately:

K:\HELLOAR001ear.ear

In all WebFacing projects, when you eventually use the Manage Applications facilities of the iSeries WAS Express Admin GUI (port 2001) to build your application, you will not be using a shared drive. Instead, you will need to directly address the IFS directories involved. Thus the path would be as follows:

/INCOMING/HELLOAR001ear.ear

Note the differences in the slashes and also when referencing the IFS directly, you do not need to have mapped drive letters. You will also see that when you Export the application to a PC and map the drive for a copy, it will be to the same directory share, "INCOMING," as shown for the direct IFS export.

Don't Forget Your Source!

We recommend bringing your source files with you in the process (Export Source Files check mark in Figure 23-5). The last check mark option on the page asks if you want to overwrite another EAR file with the same name. In this case, we have no such file so it does not matter. However, if you export a project a second time after making modifications to your DDS, you may want to select the option to overwrite if you are using the same file names.

Finish it Up!

When you have completed filling in the necessary information in Figure 23-5, it is time to hit the Finish button. Depending on how many items you have within your project, the Export can be a matter of seconds or substantially longer. In this case, there is just one display file. Within several seconds after clicking the **Finish** button, the WebFacing Export Wizard returns you to the panel from which you started the process.

Step 6 Mapping a Network Drive

Create Directory On IFS

The first step is to create a directory on your IFS unless you plan to use an existing directory. In our case study, we recommend creating a directory called /INCOMING. Using an AS/400 command screen, you can do this by typing in the following command:

```
CRTDIR DIR (' /INCOMING ')
```

Create a NetServer Share

A directory per se cannot be shared. Instead you first must create what is called a share or shared directory. In Windows, this is done on your PC by right clicking on the folder that you want to share, then selecting properties, and then selecting share. You then get a panel that permits you to specify how you want to share the directory, and what you want to call the directory or subdirectory when it is shared. This name then gets broadcast across your network and you can map a drive to it if you like.

Same as Peer Windows Support

All this is done with Windows and you do not need an iSeries to use the peer networking share facility. IBM's NetServer merely complies with Microsoft's SMB peer network specification and therefore can and does serve as a peer in a Microsoft Network.

However, if you want to map your drive to the IFS and not to another PC, then you are talking about using the AS/400. In this case, you must involve the AS/400 GUI in the process. You must use the GUI (iSeries Navigator, formerly Operations Navigator) to name a shared directory on your iSeries IFS. When set up with iSeries Express, the share behaves exactly the same as any other share in the network. After all, it uses the same exact Microsoft SMB support that you use to share folders on your non-server oriented PCs.

Shares Need a GUI

Since the AS/400 is not a display-oriented machine, it has no real GUI of its own. Therefore it uses Operations Navigator or the newer iSeries Navigator. Use either of these to create your shares for IFS directories. Then the directories can be shared and are broadcast just as Windows peer (SMB) directories.

So, we might consider a “share” as a named external network point of reference. When it is created, it behaves as a first level directory on your IFS. However, just as with a Windows PC, a share name can point to a subdirectory. In our case, we would use the iSeries Navigator to give the name “INCOMING” to a share that references the newly created IFS directory /INCOMING. Once we do this, and it is broadcast, we can map a PC drive to the share, such as the H:\ drive.

I like to export to my own PC to have a copy of the EAR file, and then I like to copy it to the IFS for the import process. That is my preference. You may elect to go directly to the IFS. You would use the iSeries Navigator facility with NetServer to create a “share” in order to map a Network Drive. NetServer makes your AS/400 appear as a System Message Block (SMB) peer in a Microsoft architected peer network environment.

Right click on Network Neighborhood or Network places on your workstation as shown in Figure 23-6 below:

Figure 23-6 Map Network Drive

From here, select a drive letter and then place the path of your iSeries server in the format shown in Figure 23-7 below:

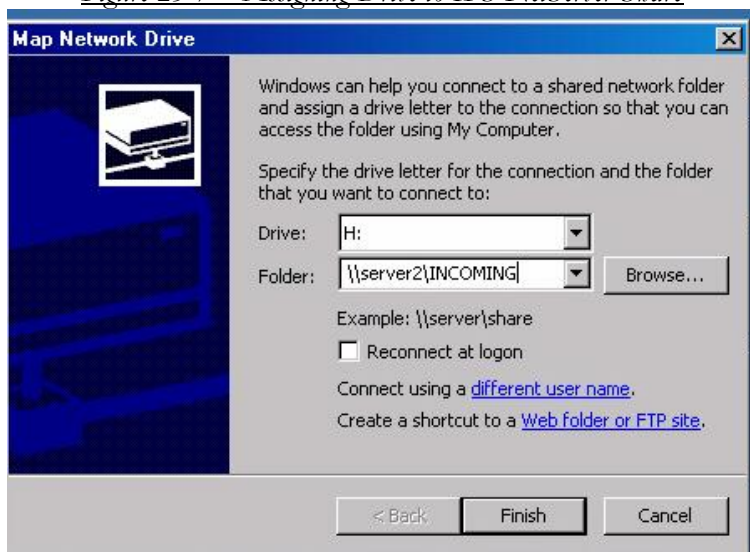
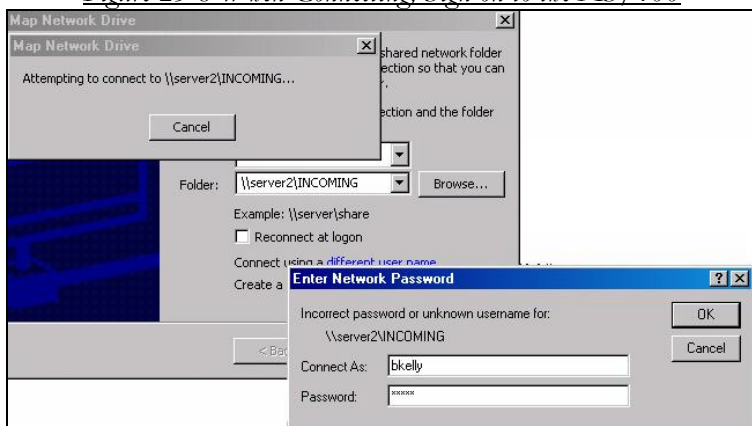
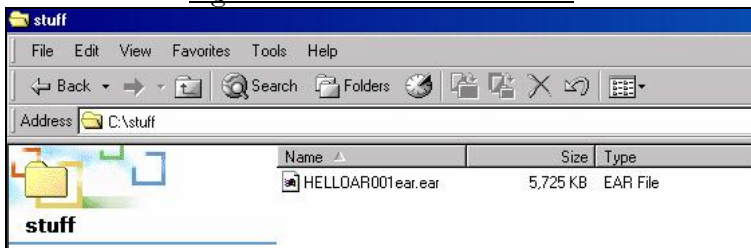
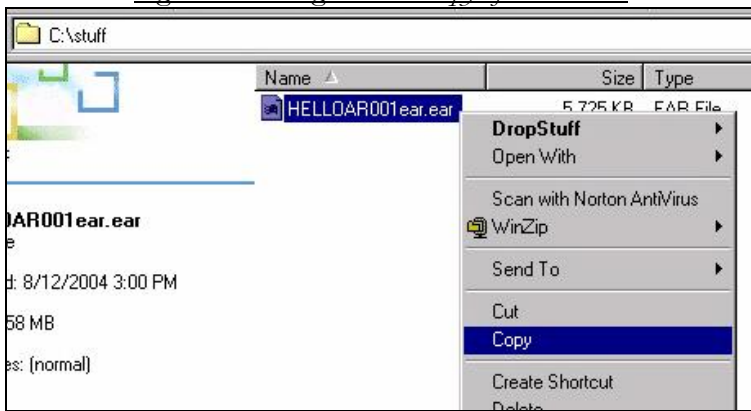
Figure 23-7 -- Assigning Drive to IFS NetServer Share

Figure 23-8 When Connecting, Sign on to the AS/400

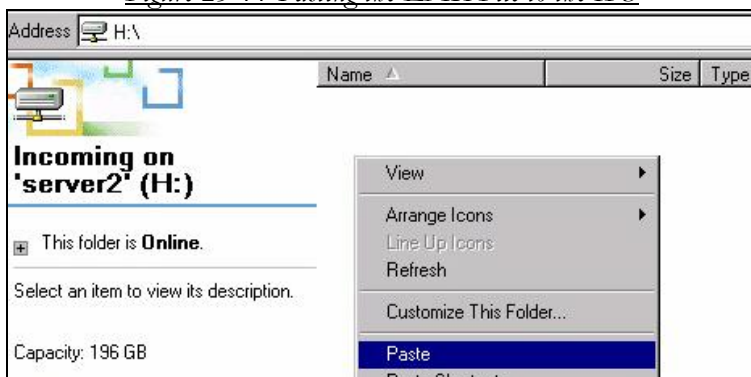
Step 7. Copy the PC file to the IFS via the Mapped Drive

When the H:\ drive is mapped to \INCOMING, you can then initiate a WINDOWS COPY from your PC by right clicking on the HELLOAR001ear.ear file in the Stuff directory of your C:\ drive. Select the *Copy* command. Then go into the shared drive and click on the *Paste* command. You will see the EAR file being copied to your IFS mapped drive.

Find the EAR file on your C:\drive (Figure 23-9) and prepare to copy it using the right click Copy function. (Figure 23-10)

Figure 23-9 EAR File on C Drive*Figure 23-10 Right Click Copy of EAR File*

Then open the folder (H: Drive) that is really the /INCOMING directory on your iSeries IFS as shown in Figure 23-11

Figure 23-11 Pasting the EAR File to the IFS

Right Click and select the paste option as shown in Figure 23-11 and release the right mouse button.

Figure 23-12 WebFaced EAR File Waiting to be Installed

As you can see in Figure 23-12, The WebFacing project file (EAR file) has been copied to the /INCOMING directory of your AS/400

In the next chapter, we will install this application to be driven by your AS/400's WebSphere Express Server.

Chapter 24

Using WebFacing: Installing (Importing) WebFaced Application to iSeries WAS Express

Step 8. Use the Manage Applications to Install the Application

At this point, as your Exported EAR file (HELLOAR001ear.ear), containing your WebFaced application is on the IFS in the /INCOMING directory, you are now ready to run the Manage Applications facility in the WAS Express Admin GUI in order to deploy (install) the WebFaced application to the WASEONE Express Server.

Though you can certainly import this EAR file to a WAS V5 Base server, we show the steps for WAS Express, since it is more widely used on iSeries than the base or the Network version. (It is the only free version.) In prior iterations of this dialogue, we would also show the steps to install a WAR file (predecessor to ear file) to a WAS 3.5 server. WAS 3.5 should not be used for WebFacing or Web activity anymore. It is too difficult and it is not supported.

First Need a WAS Express Server

This step assumes that you have a server built in iSeries WAS Express (WASEONE built in Chapter 12) to handle the application's installation process. If you do not, then to test this application, we recommend creating a WAS server and HTTP server called WASEONE as we have previously done in Chapter 12.

Your next step is to install the HELLOAR001 application to this new WAS Express server called WASEONE. The steps to install (import) your WebFacing project on the WAS Express are as follows:

A. Start the Admin GUI

From a Web Browser, type in the following URL to get the Admin GUI fired up:

```
http://your_hostname.your_domainname:2001
```

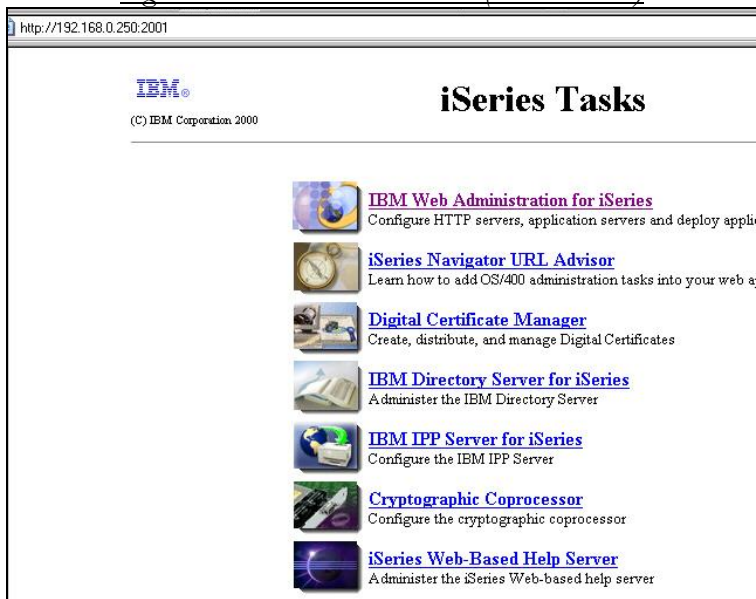
Or

```
http://your_IP_address:2001
```

B. Sign on to your WAS / HTTP Admin GUI.

After you sign on, you will be taken to the iSeries Tasks menu, (Figure 24-1) which runs from port 2001 of the browser-based HTTP Administration Graphical User Interface (Admin GUI).

Figure 24-1 ISeries Tasks Menu (Admin GUI)



C. Select Web Administration for iSeries.

Take the top option from the iSeries Tasks menu and you will find yourself in the Admin GUI for WebSphere Express and HTTP functions.

Figure 24-3 Admin GUI Manage Tab

If you do not immediately come to the Manage tab as shown in Figure 24-3, then select it regardless of which tab you see. The Admin GUI will then provide you with the options on the *Manage* tab. The most strighth forward approach is to click on the “Install New Application” menu item under the WAS Wizards heading on the left pane of Figure 24-3. However, from past tradition we take a route that today is not necessarily the best, but it is one with which you too may be familiar from past work with WebFacing. From the Manage tab, as shown in Figure 24-3, navigate down the left pane until you come to the *Applications* menu item. (shown in Figure 24-4)

Figure 24-4 Manage Installed Applications

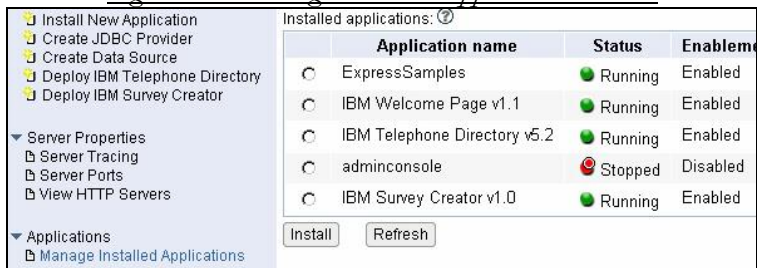
D. Click on Manage Installed Applications.

This is the function that is needed to get you to the panels that will enable you to invoke the Application Installation Wizard. As you

move the mouse around the item, you can see its descriptive text in yellow as shown in Figure 24-4.

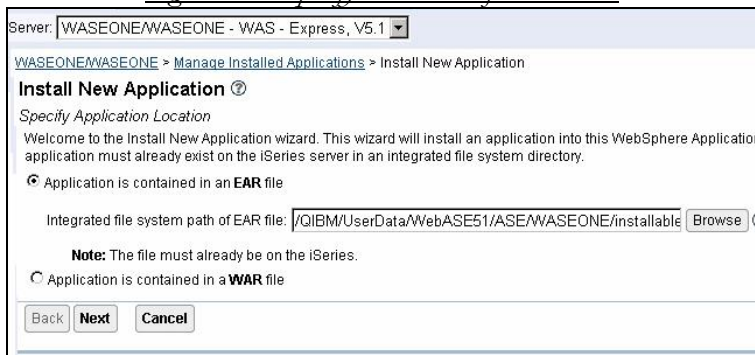
After clicking the *Manage Installed Applications* menu item, the right side of the page should change to look similar to the page shown in Figure 24-5.

Figure 24-5 Manage Installed Apps – Main Panel



E. Click on the Install Button

You can either click on the install button that you see at the bottom of the right pane or you can choose to click on the Install New Application menu option in the left pane of Figure 24-5, as noted above. Either way, your click starts the WAS Express New Application Install Wizard. The next step is that you will be taken to a panel similar to that shown in Figure 24-6.

Figure 24-6 Specify IFS Path of EAR File

F. Click on the Browse Button

If you know where the file is on the IFS, you can type it in directly. The Wizard primes you with the installable applications directory of the WAS instance structure in which you are loading the application. As noted previously, I recommend a different approach. In fact, we have loaded our ear file into the `/INCOMING` directory in the root file system of the IFS. It is not within the WAS instance directory structure for WASEONE. So, at this point, feel free to type the term `/INCOMING` in response and then press the Next button if you choose not to browse for the EAR file.

If you are unfamiliar with the IFS, the navigation with the browse button on the right side of the panel in Figure 24-6 is a much better option. It's your choice. If you choose the browse option, it may help you to more easily locate the WebFacing Project's EAR file on the AS/400 IFS.

If you choose to browse, you will see a pop-up panel in a different form than most you have seen. When I first saw it, I did not immediately know it had anything to do with me. It will be very similar to the abbreviated panel shown in Figure 24-7.

Figure 24-7 Browse Panel for IFS

G. Click on the IFS Root

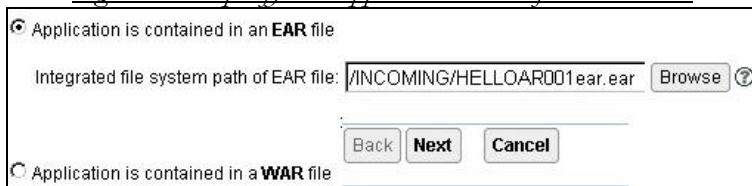
More than likely, you will see the full path to installable applications when you click browse. Scroll back to the root (/) directory and then you can go forward in your navigation to ultimately find the /INCOMING directory (folder) in the root of the IFS.

When you arrive there, in order to navigate to your EAR file, you must click on the “/” file system (root) to expand it to the next level of directories. Then, scroll until the panel looks similar to that shown in Figure 24-7. The end result of this in the WDSC V5.0 package for another project would look similar to the panel shown in Figure 24-8. IBM keeps changing the look and feel in this reasonably new product.

Figure 24-8 V5.0 Preparing to Select “INCOMING”

H. Select the INCOMING directory

From the WAS V5.1 panel in Figure 24-7, select the /INCOMING directory and then select the HELLOAR001ear file, and click the OK button. You will return to the selection panel as shown in Figure 24-9.

Figure 24-9 Specify the App Location – After the Browse

I. Click on the Next Button

This continues the import process and when you click on the Next button in Figure 24-9, you will be taken to a panel similar to that shown in Figure 24-10.

Figure 24-10 Providing Name and Install Options

Server: WASEONE/WASEONE - WAS - Express, V5.1

WASEONE/WASEONE > Manage Installed Applications > Install New Application

Install New Application ?

Provide Options to Perform the Install
Specify application deployment options

Application name: HELLOAR001EAR ?

Directory application installed to: /QIBM/UserData/WebASE51/ASE/WASEONE/installedApps

Pre-compile JSPs ?

Note: Pre-compiling JSPs can have significant performance impacts. When enabled, the JSPs in the application are compiled at installation time, causing the application install to take longer. When disabled, the JSPs will be compiled the first time they are accessed, causing the first use of the application to take longer.

Back Next Cancel

J. Assign Application Name

The name of the application is taken from the EAR file at this point. Take the default for the application as shown in the panel in Figure 24-10. It is the project name for the WebFaced application that you are importing from WDS.

Pre-compiling JSPs can have a significant performance impact. When this option is enabled, the JSPs in the application will be compiled at installation time. Thus, the application installation will take longer. When disabled, the JSPs will be compiled the first time they are accessed, causing the first use of the application to take longer. Though you wouldn't necessarily know it unless you looked deeply into the project folders, converted WebFaced projects do use JSPs.

So, it is a good idea to speed up operational execution by compiling the JSPs on the way in.

Take all the options as seen in Figure 24-10 and click on the Next button. You will then see the panel in Figure 24-11.

Figure 24-11 Map Virtual Hosts to Web Modules

WASEONE/WASEONE > Manage Installed Applications

Install New Application ?

Map Virtual Hosts for Web Modules

Map web modules to virtual hosts: ?

Web module	Virtual host
HELLOAR001	default host

Back Next Cancel

Proceed to the next step of this task.

K. Select Defaults

Unless you are a student of WebSphere and that is well beyond the intentions of this book, you will select the default option for the Virtual host parameter. Assure that `default_host` is in the list box and click the Next button to continue with the WebFaced HELLOAR001 application installation (import). You will see a panel similar to that shown in Figure 24-12.

Figure 24-12 Install Parameter Summary

WASEONEWASEONE > [Manage Installed Applications](#) > Install New Application

Install New Application ?

Summary

When you click **Finish** the installation will be started for the following application.

Integrated file system path of EAR file: /INCOMING/HELLOAR001ear.ear

Application name: HELLOAR001EAR

Directory application installed to: /QIBM/UserData/WebASE51/ASEWASEONE/installedApps

Pre-compile JSPs: Enabled

Map web modules to virtual hosts:

Web module	Virtual host
HELLOAR001	default_host

L. Click Finish to Create the Application under WAS Express

Figure 24-12 nicely summarizes all of the options that you have chosen for this particular WAS Express V5.1 application. When you are sure that these are the options that you want to run with, click the Finish button. The Wizard will then begin to install the WebFaced HELLOAR001 application. During the installation, you can watch the build process as it moves from *Installing and Disabled* to *Enabled and Stopped*, by using the Refresh buttons. This action is portrayed in the panels shown in Figure 24-13 and Figure 24-14.

Figure 24-13 Application is Installing

WASEONEWASEONE > Manage Installed Applications

Manage Installed Applications (?)






Data current as of 08:22:40 PM UTC on 08/12/2004

Installed applications: (?)

	Application name	Status	Enablement
<input type="radio"/>	HELLOAR001EAR	 Installing	Disabled
<input type="radio"/>	ExpressSamples	 Running	Enabled
<input type="radio"/>	IBM Welcome Page v1.1	 Running	Enabled
<input type="radio"/>	adminconsole	 Stopped	Disabled
<input type="radio"/>	IBM Survey Creator v1.0	 Running	Enabled

Figure 24-14 Application Is Installed and Stopped

Current Configuration for Server - WASEONE

Manage Virtual Hosts	Manage Installed Applications
<input checked="" type="radio"/> default_host	 HELLOAR001EAR
<input checked="" type="radio"/> admin_host	 ExpressSamples
	 IBM Welcome Page v1.1
	 adminconsole
	 IBM Survey Creator v1.0

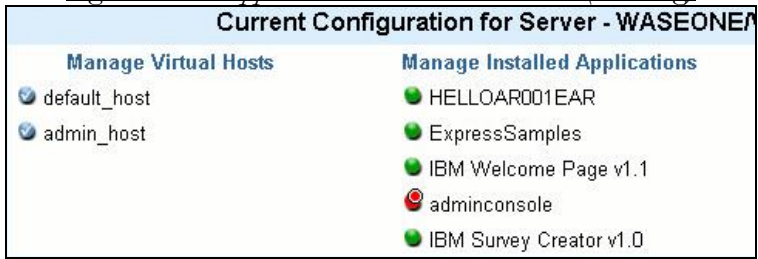
M. Check Application Build Status

Once the HELLOAR001 application is built and ready to be started, its status will appear as in Figure 24-14. The button next to it is colored red. At this point, if the WASEONE application server is up, the Start button in the left frame of the GUI will be highlighted. In the case shown in Figure 24-14, the application was built while the WASEONE server was up. The next step would be to stop and start the WASEONE server or to start the application. The safer bet is to do the former.

N. Recycle the Server

Either approach starts the application itself. However, sometimes, just as with WAS 3.5, you have to bring down the server and restart it before the application can become alive. So, you stop the server with the Admin GUI and then you restart it. When you restart WASEONE, after the application is loaded, the “Running” status of the server and the application changes and the application lights change from red to green to indicate that they are all started, including the new WebFaced HELLOAR001EAR. This is shown in Figure 24-15.

Figure 24-15 Application Installed and Started (Running)



It is always safest whenever you install a new application to a “started” or “unstarted” WAS server, that you should stop and restart the server. WebSphere is tricky about that, but a safe bet is to assure that you restart the server after adding a new application. Obviously, this can be inconvenient if you have a volatile and dynamic application environment. Sometimes you can take a chance and win. Sometimes you lose. Thus, it behooves the developer to assure that there is a test server available to thoroughly test the application, along with all of the necessary starts and restarts before deploying to the live server.

In Chapter 25 we will continue the WebFacing process by running the application from our iSeries using a Web Browser.

Chapter 25

Using WebFacing: Running the WebFaced Application on iSeries WAS Express

Time to Run

Once you have deployed your WDS_c V5 application to the WAS Express server, the next step is to run the application.

Launch Panel is Not Pretty

Most folks who look at the launch panel for WebFacing are at first thrilled because they got it to work. Quickly, however, the excitement wanes and there is disappointment because the launch panel presented is not at all handsome. In WDS_c 5.1.2 the main launch panel is lots nicer than it was and it is now built as a JSP instead of HTML with JavaScript. However, it still needs some work for application incorporation so it belnds with your Web site. In these next few pages we are going to stage the WAS Express and WDS_c V5 developers for the eventual modification of this panel ([index.html](#)). Between you and your Web designer, you should be able to prepare a much more

attractive page. The basic information to do this is provided in Chapters 26 and 27.

The index.html file et al.

The secret ingredient in a WebFaced application at the browser level is a little JSP. With WebFacing, in early versions it was named index.html. In later versions, such as WdSc V5.1.2, it is named index.jsp. This name is special because HTTP servers and Servlet servers working in tandem look for specific file names like index.html and index.jsp, and welcome.html, and if they are present when a URL arrives with no file, the servers present a “welcome page” to the browser.

The WebFacing Wizard builds the index.html / jsp file that you see in a WebFacing application, for you. It is deployed to your WAS instance in the proper instance / application “Web” IFS directory. This way, it gets called when you specify the web app name in your browser URL.

For this example, you may recall that the WAS Express web app (context root) that we specified for the WdSc project is as follows:

HELLOAR001/

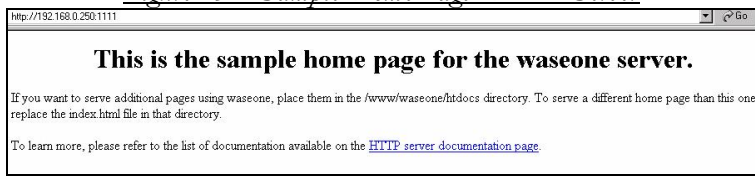
We specified this information when we named the project while the project was still on the PC. The context root is the same name by default in WdSc V5.1.2 as the project name. Of course this can be changed, but we had no need to do this in this application example.

Specifying the Correct URL in Your Browser

Whenever I create a WAS instance, I am unsettled until it works. I am not as comfortable as with a traditional COBOL or RPG program that it will perform as I expect. So, I take things incrementally to make sure I have the right perspective. Rather than test WebFacing on a new WAS server, I first test the home page to see if the HTTP server works. Then, I test the WAS server with the Example Apps (Snoop) to make sure it works. Then, since the Web environment works, after I import the WebFaced application into WebSphere, I feel much more confident in running the application.

If you look at Figure 25-1 below, you will see that I have included the sample home page for the WASEONE HTTP server. When I see this I know that HTTP is working fine. In other words, this page is the default page that the WASEONE HTTP server fires up when you come in from the Web to be welcomed. From a testing perspective, if this works fine as it has below in Figure 25-1, this helps me know that a good part of the URL – the host.domainname and the port number (1111 for this instance) are correct for this Web server instance.

Figure 25-1 Sample Home Page- HTTP Server



If you take the web app portion of the URL and you couple it with the HTTP instance port reference, you come up with the URL that should be used to call the WebFaced application from a WAS 3.5 or even an iSeries Express server. In the Advanced Hello World case study, this looks as follows:

**IP address; port#;context root---
web application**

**http://192.168.0.252:1111/HELLOAR0
01**

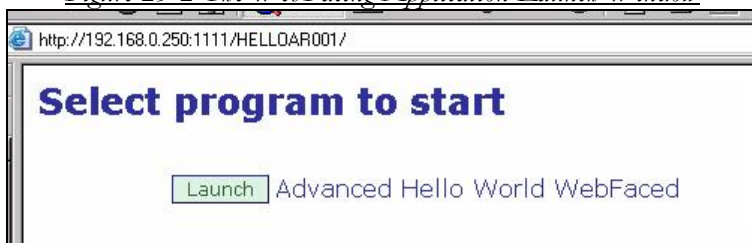
or

http://server2:1111/HELLOAR01

Launch the Application

So, to launch our WebFaced application from WAS 5.x Express for iSeries, you just type into the full URL into the browser address field as shown in Figure 25-2 below. This should be specified in the form shown above, and as shown again below in Figure 25-2.

Figure 25-2 The WebFacing Application Launch Window



Your First Look

When you finish with that URL if all things are OK, you will see a panel similar to that shown in Figure 25-2 above. Your Web Design

Team, more than likely will not particularly like this launch window as built for you in the WebFacing process. It is very stark. So, in Chapter 21, we give you some tips on how to modify it so that it can be much more handsome. The unmodified HTML starter page for the application built by WDS Sc V5.0 is basically the same as that built for the WDS Sc V4 project. However, IBM has switched the starter page to the JSP form in WDS Sc V5.1.2.

The HELLOAR001 default launch panel that we show in Figure 11-1 is launched from a file called `index.jsp`. As you can see, it is not very pretty and it would not fit into many application scenarios. Therefore, the page source needs to be altered once built to fit into your Web hosting environment.

Fully Functional Application Driver

The panel as shown, though not pretty, is fully functional. In our case study, there is just one RPG program involved called HELLOAR001. However, that is not a WebFacing limitation. You can WebFace many display files in the same run, as well as menus. Moreover, as shown in Chapter 26 or 27, you can change the starter page to deliver whatever look you like. It can be a menu of options such as a lead menu for many programs instead of a link for a simple RPG program as we've got in our little application. But, you've got to start someplace and that's why we are here.

Repeating for effect, in order to get the launch panel for the WebFaced application, as produced by WebFacing, you would type in a URL with the following format:

`http://yourhostname.yourdomainname:YourApacheHTTPServerPort#/YourWebFaceAppName`

Or

`http://yourIPAddress:YourApacheHTTPServerPort#/YourWebFaceAppName`

Names or IP Address?

For our example, we have a PC *hosts* file that is used to find the IP address of the iSeries. In the PC *hosts* file, we have an entry for “server2.” And an entry for “HELLO2.” Both entries point to the same non-routable (on the Internet) local IP address 192.168.0.250. To launch the WebFaced application from the WASEONE server then, you can use the IP address or the names in the host file, or the names in your DNS. The three URL forms that would launch this application therefore are shown below:

Two forms of names:

http://server2:1111/HELLOAR001

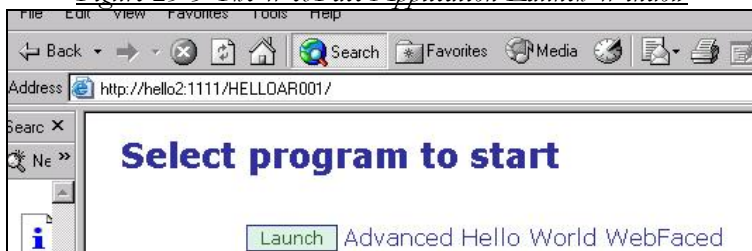
http://HELLO2:1111/HELLOAR001

Or

One form of IP Address:

http://192.168.0.250:1111/HELLOAR001

When you enter the URL as above, and press enter or otherwise launch the browser with that URL, you will get the “ugly” Application Launch menu repeated in Figure 25-3 for your convenience.

Figure 25-3 The WebFace Application Launch Window

Same Browser or New Browser

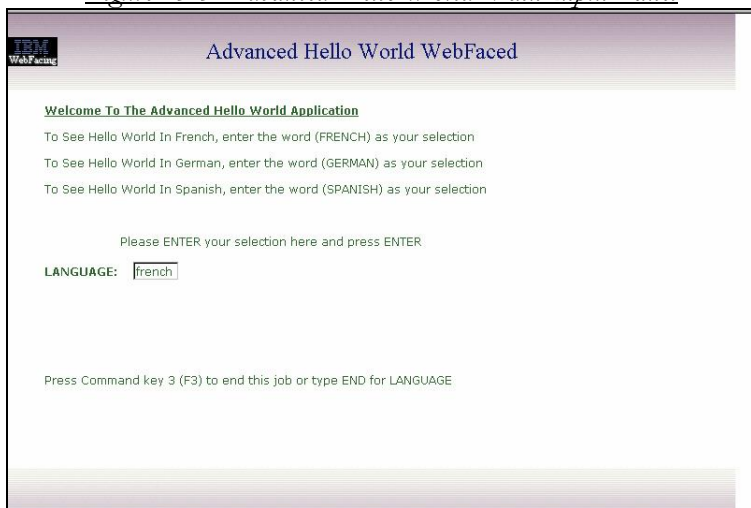
In Chapter 26 and 27, you'll see how you Web Designers can pretty it up quite a bit. From the application launch menu as shown in Figure 25-3, you can launch your newly created HELLOAR001 WebFaced application in a browser window. Because we built the application requiring a prompted sign on, the next panel you see will request a user id and password as shown in Figure 25-4.

Figure 25-4 Sign On to AS/400 or iSeries

Sign on to the AS/400

When you click the Logon button after filling in the panel in Figure 25-4, there will be a noticeable delay (at least the first time) as the panel loads and the code behind it is prepared for execution. After a lengthy number of seconds, the browser finally sneaks out the first Advanced Hello World panel driven by WebSphere Express on the iSeries as shown in Figure 25-5.

Figure 25-5 Advanced Hello World Main Input Panel



WebFacing Is Lots Better!

It sure looks just like the same panels shown in the WDS*c* Test Environment. And it should!

WebFacing has improved by orders of magnitude since its first release more than two years ago. Back then, there were no plug-ins to enable keyboard actions. Thus, to ENTER, you had to click the ENTER button. To F3 something, you had to click the F3 button. You could not use the keyboard for these functions. Now, the WebFaced application has plug-ins for the keyboard for ENTER and for the

function keys. You can use the mouse or the keyboard. This is ideal for machines like Macs that like everything to be mouse driven. But, it is now also appropriate for keying applications in which the ENTER key and the PF keys are productivity enhancers.

The code that was produced for WebFacing several years ago also had some nice features. For example, if you had text on a command key in your DDS, the conversion would place the text on the button as in Figure 25-7, rather than place the “F3” designation. Notice on the second button in Figure 25-7 – the “end of job” appears on the button in lower case. That’s because it was lower case in DDS. To make the conversion upper case, you would merely have to change the DDS, or you could make combination of U/l case that looks better than all lower case. You would not have to recompile the display file for the change to affect an application that you WebFace. The next time through WebFacing, the conversion of DDS would present an upper case End-Of-Job.

Can Get Fancy

To perfect their applications beyond that which WebFacing naturally provides, many companies such as APPCON, a premier apparel software house in Philadelphia, and a frequent IBM WebFacing reference account, have gone back to their DDS after seeing what happens during conversion. A few minor DDS changes can perfect the WebFaced application’s look and feel. However, as you can see, most applications, such as Advanced Hello World do not really have to be changed to achieve true Web function.

Running Advanced Hello World

This particular case study application has a database with three records. One record is for GERMAN, one for FRENCH, and one for SPANISH. Any other entry causes an application error message. That is how the program works, so it is OK. In the example in Figure 25-5, type in SPANISH as the Language and either click the ENTER button or hit the ENTER key.

Tip:

If you want to DFU the actual language translation into these records, that would be easy also. Thus, the record that SPANISH pulls from the DB could very well be the translation as it would actually be in SPANISH, rather than the Q & D database method that is deployed in the sample.

Figure 25-6 WebFaced Results Page Hello World

IBM
WebFacing

Advanced Hello World WebFaced

Welcome To The Advanced Hello World Application

To See Hello World In French, enter the word (FRENCH) as your selection
 To See Hello World In German, enter the word (GERMAN) as your selection
 To See Hello World In Spanish, enter the word (SPANISH) as your selection

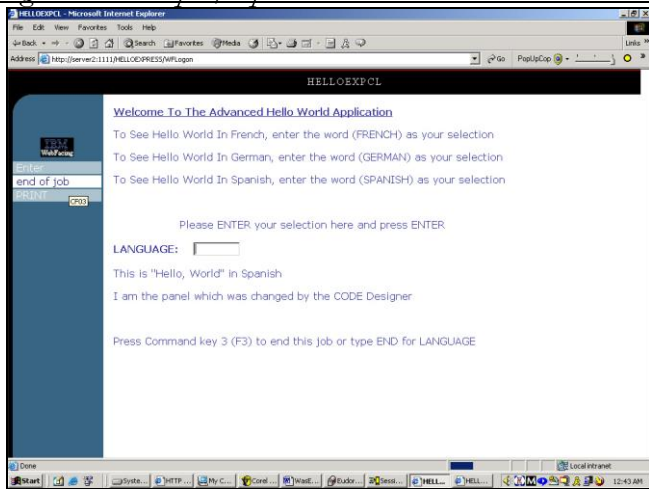
Please ENTER your selection here and press ENTER

LANGUAGE:

This is "Hello, World" in French

Press Command key 3 (F3) to end this job or type END for LANGUAGE

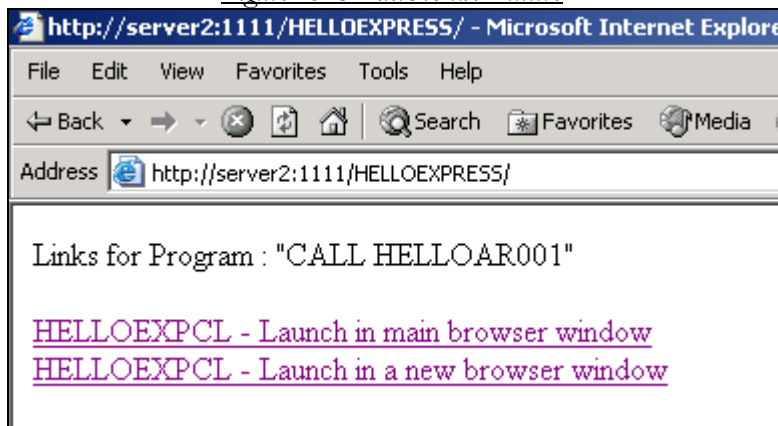
To demonstrate the power of the PF keys and the end of Job function, I have included the output page from another Advanced Hello World Project that I called HELLOEXPRESS. As you can see in Figure 25-7, the style sheet maps the function keys and the ENTER key to buttons on the left pane of the page.

Figure 25-7 Output/Input Panel For Advanced Hello World

End the WebFaced Application

When you are presented with the panel in Figure 25-7, the end of job button is not highlighted as shown in the page. Notice that the response has already come back properly from the DB lookup: In this case, we looked up SPANISH. The result is “This is “HELLO, World” in Spanish.” The application worked fine. It did its thing just as when we used the textbuttons style without the buttons! That means we were successful.

In Figure 25-6 or Figure 25-7, you can now type in another language here and press ENTER, or you can do as we did, hit F3 in Figure 25-6 and click the “end of job” button in Figure 25-7. When you run the mouse over this button shown in Figure 25-7, you see that it is highlighted and the command key behind it is also shown. That’s some slick function for a program that drives JSPs via RPG EXFMT operations. When you actually click on the end of job button, or you press F3, the WebFaced application is ended and you return back to the launching panel as shown in Figure 25-3 for WdSc V5.1.2 or Figure 25-8 for WdSc V5.0.

Figure 25-8 Back to the Future

Chapter 26

Using WebFacing: Re-Designing the WebFacing Launch Panel V5.0

The Sow's Ear

Figure 25-8 is the standard bearer WebFacing Launch Panel for HELLOEXPRESS, an application just like HELLOAR001, but WebFaced on V5.0 of WDSc (not V5.1.2). There is no beauty contest winner here. This chapter is intended to unmask the secrets of the index.html file that IBM uses to produce the launch panel so that your Web designer and you can come up with a better front page for this application. If you are using the older WDSc, this is your chapter. If you are using V5.1.x or greater, go to Chapter 27 and completely skip this work.

The HTML for the page, including the JavaScript is shown in Figure 26-1.

Figure 26-1 *index.html* File Produced by WebFacing

```

<!-- Version 1.1 -->
<html>
<head>
<META http-equiv="Content-Type" content="text/html;
charset=UTF-8">
  <script language="javascript">
    function launchApplication_W(app)
    { if(winActive == false)
      { winActive = true;
        window.open(app+'&turnCacheOff='+new
Date()).getTime(),'appa','scrollbars=yes,status=no,resizabl
e=yes,menubar=0');
      }
      else { window.event.returnValue = false ; }
    }
    function fullEscape(value)
    {
      var escapedValue = escape(value);
      var indexOfPlus = escapedValue.indexOf("+");
      while (indexOfPlus >= 0)
      {
        if (indexOfPlus < (escapedValue.length-1))
        {
          escapedValue = escapedValue.substring(0,
indexOfPlus) + "%2B" +
escapedValue.substring(indexOfPlus+1);
        }
        else
        {
          escapedValue = escapedValue.substring(0,
indexOfPlus) + "%2B";
        }
        indexOfPlus = escapedValue.indexOf("+");
      }
      return escapedValue;
    }
    function launchApplication_M(app)
    {
      app += '&turnCacheOff='+new Date()).getTime();
      if(winActive == false)
      { winActive = true;
        try {
          var t = document.getElementById("wfLink");
          t.href = app;
          t.click();
        } catch(any_exp){}
      }
      else { window.event.returnValue = false ; }
    }
    var winActive = false;
    window.closeWinListener = function() { winActive =
false; }
  </script>
</head>
<body><a id="wfLink" href=""></a>
  <!-- begin cl205999 -->
  <p>Links for Program : "CALL HELLOAR001"</p>
  <a href="#"
onclick="launchApplication M('WFLogon?inv=cl205999');return
false;">HELLO2AR001 - Launch in main browser window
</a><br>
  <a href="#"
onclick="launchApplication W('WFLogon?inv=cl205999');return
false;">HELLO2AR001 - Launch in a new browser
window</a><br>
  <!-- end cl205999 -->
</body>
</html>

```

Default index.html Page

As you can see the default index.html page is pretty plain and pretty boring. It displays just as it did when we tested it in the WDS Sc Test Environment and when we executed it live last Chapter with WAS Express. The HTML and JavaScript, shown in Figure 26--1, however, for the typical RPG / COBOL person certainly do not look as simple and easy as the page that they produce.

Same index.html as in Test

This time we are running the html and the WebFaced application from the AS/400 WAS Express, not the internal test WAS. It can be started and run from any browser, not just the Test session of the WDS Sc as shown previously.

Under JavaScript Control

Once it is called, you can see that there is a bunch of JavaScript that gets executed in order to ultimately launch your RPG application. You might enjoy trying to follow the logic in the html/JavaScript code if you have the patience. Of course, that is not a requirement for this book or for understanding what must be done to make it better.

When you enter a language as input to the test application, the panel looks similar to that shown in Figure 22-11, and when you receive back the output it has an extra line with the response. You can then click the end of job button or press F3 to end the job. With this simple set of instructions, you have accomplished the run requirements of Advanced Hello World. Now, let's try to perfect the application with a better welcome page.

OK, we're not designers here, so the most important thing that we can do is show you how you can make changes. By showing where you should make the changes, you can then instruct the real Web designer where to make the changes that are necessary to improve the look and

feel of the application, while still supporting the launch of the application with JavaScript.

Customizing Launch Page

Now, let's say that you think the panel shown in Figure 25-8 is ugly and otherwise unbecoming the look and feel of your fancy schmancy website. Well, it is ugly! So, it's okay to feel that way. How do you make it better?

You can optionally create your own pages to launch your applications. If you want to do this, it is a good idea the first time through to use the IBM produced `index.html` page as a starting point. If you happen to be an HTML guru, however, feel free to add the links that are shown in `index.html` to your own pretty pages.

COPY Parts of Index.html

Your links will vary depending on your approach, but will be very similar, if not the same as those shown for this project. It all depends on the command you are using to call the application. To use the links that work for this application case study, just copy and paste the link into the source for the your static html page that you choose to use as the launching point.

Look for links within `Index.HTML` that use the JavaScript function *launchApplication*. These links begin with the following:

```
<a href="javascript:launchApplication_W
```

You don't have to know JavaScript to use the JavaScript code that IBM has made for you. You just have to know how to copy and paste lines of text. To use this JavaScript function, you will need to copy the source for the function from the `<head>` section of `index.html` into the `<head>` section of your own web page. To identify the function, look for JavaScript source that looks like the sample shown below:


```

<script language="javascript">
function launchApplication_W(app)
{

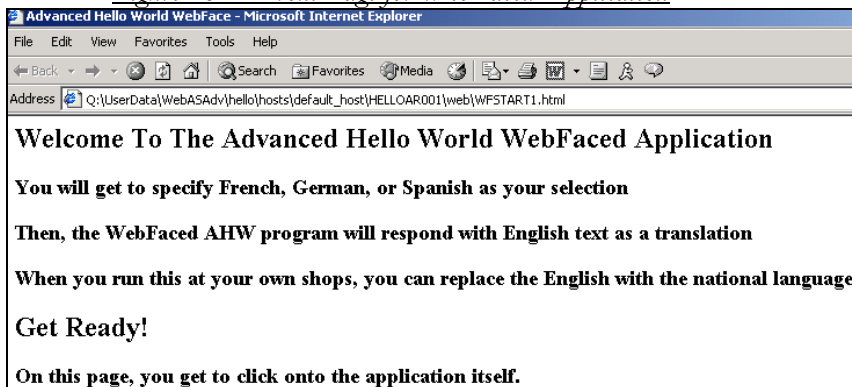
window.open (app, 'appa', 'scrollbars=no,s
tatusbar=0,resizable=yes,menubar=0');
}
</script>

```

Changing index.html

There is really a lot more function in the JavaScript that you will be copying, but the expanded text above gives you an idea of what to look for. So that we have a simple “pretend pretty” Web page to launch from, we created a static html page, as shown in Figure 26--2. We then modified the page representing your launch page in Figure 26--3, by adding in the components of the index.html file necessary to launch the WebFaced AHW application.

Figure 26--2 Front Page for WebFaced Application



The Starter Page

In Figure 26--2, you can see a reasonably plain Web page. As noted above, the HTML behind this is very simple and is shown in Figure 26--3. We modified this pretty pretender html code to include the Web information (java script) from the COPY. This is shown in Figure 26--4, as you can see.

As discussed above, the idea is to be able to launch the AHW WebFaced application from this modified page, rather than the plain index.html page that IBM provides. This page therefore, represents the type of page that you may already have on your Web site that already serves to launch your applications or your other links. Of course, we are sure that your page is better looking.

The simple HTML for the page in Figure 26--2 is shown in the panel in Figure 26--3.

Figure 26--3 Static HTML Unmodified “Pretend” Web Panel

```
<html>
<head>
<title> Advanced Hello World WebFace </title>
</head>

<body>
***** Insert the Launching Head information here: </p>

<h2>Welcome To The Advanced Hello World WebFaced
Application </h2>
<h3>You will get to specify French, German, or Spanish as
your selection </h3>
<h3>Then, the WebFaced AHW program will respond with
English text as a
translation </h3>
<h3>When you run this at your own shops, you can replace
the English with the national
language text </h3>
<h2>Get Ready! </h2>
<h3>On this page, you get to click onto the application
itself. </h3>
</body>
</html>
```

Merge JavaScript into the Starter Page

Your next step is to take the index.html code that was created for you, and as instructed above, copy in the JavaScript into the two places as shown in Figure 26--5. The objective is to have your panel look like the one shown in Figure 26--4.

Figure 26--4 Modified Static HTML for WebFaced Application

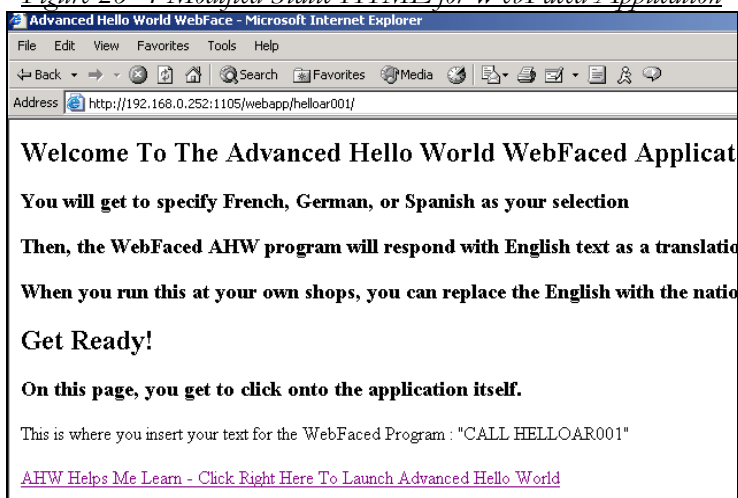


Figure 26-5 Merged HTML for Initial Panel

```

<HTML>
<HEAD>
<TITLE> Advanced Hello World WebFace </TITLE>

Beginning Of Index.HTML Insert

<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <script language="javascript">
    function launchApplication_W(app)
    { if(winActive == false)
      { winActive = true;
        window.open(app+'&turnCacheOff='+new
Date()).getTime(), 'appa', 'scrollbars=yes,status=no,resizable=yes,men
ubar=0');
      }
      else { window.event.returnValue = false ; }
    }
    function fullEscape(value)
    {
      var escapedValue = escape(value);
      var indexOfPlus = escapedValue.indexOf("+");
      while (indexOfPlus >= 0)
      {
        if (indexOfPlus < (escapedValue.length-1))
        {
          escapedValue = escapedValue.substring(0, indexOfPlus)
+ "%2B" + escapedValue.substring(indexOfPlus+1);
        }
        else
        {
          escapedValue = escapedValue.substring(0, indexOfPlus)
+ "%2B";
        }
        indexOfPlus = escapedValue.indexOf("+");
      }
      return escapedValue;
    }
    function launchApplication_M(app)
    {
      app += '&turnCacheOff='+new Date()).getTime();
      if(winActive == false)
      { winActive = true;
        try {
          var t = document.getElementById("wfLink");
          t.href = app;
          t.click();
        } catch(any_exp){}
      }
      else { window.event.returnValue = false ; }
    }
    var winActive = false;
    window.closeWinListner = function() { winActive = false; }
  </script>
</head>

*** END of Index.HTML
--- continued

*** BEGINNING of Static WEB Page

```

```

<BODY>
<H2> Welcome To The Advanced Hello World WebFaced Application </H2>
<H3> You will get to specify French, German, or Spanish as your
selection </H3>
<H3> Then, the WebFaced AHW program will respond with English text
as a translation </H3>
<H3> WHEN you run this at your own shops, you can replace the
English with thenational languageT text </H3>
<H2> Get Ready! </H2>
<H3> On this page, you get to click onto the application itself.
</H3>

*** End of Static Web page

*** Beginning of Index.HTML

<a id="wfLink" href=""></a>
  <!-- begin c1205999 -->
<p>This is where you insert your text for the WebFaced Program :
"CALL HELLOAR001"</p>
  <a href="#"
onclick="launchApplication_M('WFLogon?inv=c1205999');return
false;">AHW Helps Me Learn - Click Right Here To Launch Advanced
Hello World </a><br>
  <!-- end c1205999 -->
</body>
</html>

```

Two Blocks of HTML Code

For your HTML to look like that Figure 26--5, you have to identify the two blocks of code that are marked in Figure 26--5 without bold typeface. These came from the static web page that we produced. You either insert these two pieces of code into the html as it is deployed in your IFS file for your web instance, or you change the static web page by copying the two boldface sections of index.html into the merged page as shown in Figure 26--5.

Copy Original

In either case, it is good for you to make a clean copy of the index.html file in your web instance / HELLOAR001 IFS web directory before you modify it. You know why... just in case you mess something up. If you successfully modify the index.html file in place after copying it, you have finished your work. If you modify the static web page by bringing in the JavaScript pieces, then you must replace the index.html file in your web instance / HELLOAR001 with

the merged html file. It must be named `index.html` for your WebFaced application to call it as its launch panel.

Not Really a Programmer's Job

If you happen to have a slick Web designer in house or on contract, more than likely, they know how to deal with the JavaScript launching in a way that will maintain the look and feel of your web site. It is certainly not our intention to suggest that this is the only way to do this, but it is the most straightforward, and our objective is not to teach HTML, Java or JavaScript.

If you are deploying whole applications, then you will be converting many series of menus that already work in a green screen environment. In this case, just a small amount of touch up work in the `index.html` file may be all you need.

Well, That is That!

In the WebFacing chapters and in the Appendices, we sure completed a lot of things. We WebFaced and deployed the AHW application twice, once to V3.5 WAS and once to WAS Express. One time we did it in the main book with V5 and another time in the Appendix with V4. We even deployed the application to the test WAS Express server within WDS Sc Eclipse.

There are lots more things to learn about WebFacing, but we accomplished the mission for this primer, which was to get you started. The steps to WebFace big applications are the same as for this simple application. When you do big applications, of course, you have to make sure that you have all of the components isolated for conversion. Unless you are doing it all, that may be the more difficult up-front task.

We would recommend getting a test WAS Express server or WAS 3.5 instance set up, such as the hello instance we used, along with respective test HTTP instances, and then start running through various WebFacing scenarios. You'll start getting pretty good with the

WebFacing tool and you will gain comfort with the WDS V5 Eclipse WorkBench. You can expect to see these around for a long time to come as the IDE of the IBM workstation development tools.

Best wishes as you attack your green screen inventory. May all your WebFaces be happy faces!

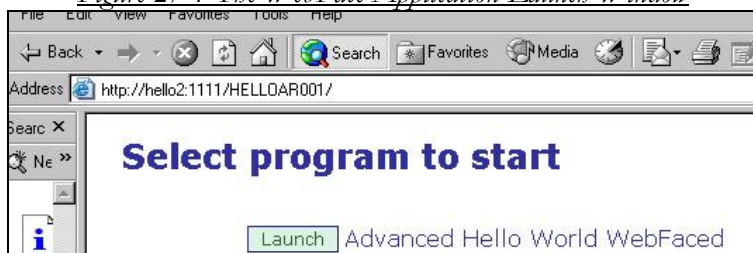
Chapter 27

Using WebFacing: Re-designing the Launch Panel for WDSVC V5.1.2 and Higher

New WebFacing Launch Panel

With the newer versions of WDSVC, which we recommend highly, the launch panel is actually more handsome than the panel look and feel that we worked with in Chapter 26. Though nicer, it is probably still not what you want to drive your application from your main Web site. Structurally it is much different than the HTML panel from Chapter 26. IBM's WebFacing process now creates a ".jsp" file instead of the more primitive html code. Because of these changes, the panel is more versatile and fits in better with the new J2EE WebSphere versions.

You may recall the look of this panel shown in Figure 27-1 in our exercises in Chapter 25.

Figure 27-1 The WebFace Application Launch Window

The Code Behind the JSP

The code that actually starts our Advanced Hello World application is rather simple. The real work that goes on behind the scenes is handled by a relatively new facility called Struts. Using Struts, a group of special servlets that IBM has created for WebFacing run the application.

We are not interested in this chapter in the whole execution environment. In fact, the most important code that we need to work with in this chapter is contained in the `index.jsp` file that is produced during the WebFacing process. This is the first page that you would see when you run the HELLOAR001 application. We have referred to this panel and the WebFacing Launch panel. The objective of this Chapter is to show you how you can extract the code from `index.jsp` and use it in your own Web pages to make the WebFacing launch more natural.

TIP:

What is Struts? Though the actual topic of Struts is out of the scope of this book, a simple explanation is still necessary. The Struts framework was created by Craig R. McClanahan and was donated to the Apache Software Foundation (ASF) in 2000. The Struts framework was developed to make it easier

for web developers to create powerful and robust web applications based on the JavaServer Pages (JSP) and Java Servlet technologies. The Struts framework provides a solid foundation upon which Web applications can be built.

You can visit the Apache Software Foundation to read more about the topic of Struts.

Apache Software Foundation
<http://struts.apache.org/>

Calling the WebFaced Application

As you may recall, this is the URL that we used to start the HELLOAR001 WebFaced application in Chapter 25. The code shown in Figure 27-2 is what drives the application:

`http://server2:1111/HELLOAR001`

As you would see if you opened the `index.jsp` file, the full set of code is rather long. However, we can sum up its function by noting that `index.jsp` uses a cascading style sheet to tell our application how the page will aesthetically be structured, as well as how the controls, the launch button, and the text will be displayed (Color, font, graphics). To net it out, the only line of code that is of importance to this effort is shown in Figure 27-2.

Figure 27-2 Part of index.jsp Code

```
<jsp:include  
page="webfacing/services/invocation/html/_  
invocations.jsp" flush="true"/>
```

This code is very simple. As shown in Figure 27-2 this code uses a Struts tag to access a jsp *include* directive to tell the application server to “include” another page from the point where this line of code is being executed. In other words, as in this example, it grabs another jsp page called:

invocations.jsp

This jsp is located in the following folder: within the WebFacing IFS structure.

webfacing/services/invocation/html

The “include” facility copies the code into index.jsp from where it is being called. A simple way for you to be able to work with this “include directive” to make the function part of your own Web page is to copy and paste the contents of invocation.jsp directly over this line of code. This would be done to separate the actual initialization code from the welcome page, index.jsp.

That’s about as complicated as I want to get at this point. However, there are still a few other things that must be done to be able to lift this code into our own Web pages. To really get down to the “good stuff,” you should take a hard look at the invocation.jsp file itself as shown in Figure 27-3.

Figure 27-3 *Advanced Hello World invocation.jsp*

```

<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<table border="0" width="100%">
<!-- begin INv1 -->
  <% /* sample programatic invocation
    <a href="WFInvocation.do?inv=INv1">Advanced Hello World WebFaced</a>
  */ %>
  <html:form action="WFInvocation.do" styleId="INv1" onsubmit="var
rv=false;if(false!=document.wfflag){rv=true;}document.wfflag=false;return
rv;">
    <html:hidden name="inv" value="INv1" property="inv"/>
    <tr>
      <td width="50%" align="right"><html:submit styleClass="launchbutton
buttonup wpsButtonText nav-h-normal"><bean:message
key="invocations.launch"/></html:submit></td>
      <td width="50%" align="left">Advanced Hello World WebFaced</td>
    </tr>
  </html:form>
  <tr><td>&nbsp;</td></tr>
<!-- end INv1 -->
</table>

```

The Real Magic

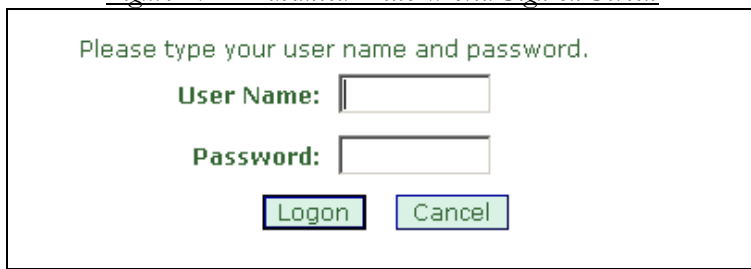
The invocation.jsp file contains Struts related elements as you may be able to see with a trained eye. Recognizing or understanding this is certainly not a requirement to understand this book. It is more geared to the folks in your shop who do your Web design and who may use Struts in their normal work would . Though this is activity is certainly not beyond the capabilities of a technician in an iSeries shop, it is atypical.

On the other hand, Web designers with an affinity for Struts would have an appreciation for this code section. The major elements within the invocation.jsp file shown in Figure 27-3 are provided by two very important snippets that relate to how the actual launch panel really works.

Option One of Two

There are only two ways that you can go to launch the application. In method 1, in essence, you can call the sign on page directly. In this case, the first page you would see is shown in Figure 27-4.

Figure 27-4 Advanced Hello World Sign on Screen



Please type your user name and password.

User Name:

Password:

You could create a launch panel with a look and feel that blends well with the rest of the pages in your site.

Just Sign Me On!

As noted above, you can call the sign on screen directly, without having to click on a launch button merely by using the URL below:

```
http://<your ip or computer
name>/<Hello world
Application>/WFInvocation.do?
inv=INV1
```

OR

```
Http://server2:1111/HELLOAR011/WFInvoca
tion.do?inv=INV1
```

Of course the URL would look different depending on how you call your server and how you call your application. The main point here is that you can call the application in this manner and it then calls the servlet that handles the invocation directly. Just pass it the parameter *inv* and the value *INV1* as shown above:

Tip:

WFInvocation.do?inv=INV1. Anything after the actual *WFInvocation.do* is the parameter string, the *inv* is the name of the parameter and *INV1* is the value.

There is one slight drawback calling your application this way. The end user will see the parameters for invocation in their URL location bar. So, they will see more than just the following:

```
http://server2:1111/HELLOAR001/WFI
nvocation.do
```

They will also see *inv=INV1* appended to the URL.

As you may be able to discern, by calling your application directly in this fashion, as in this example, you could create hyperlinks to the sign on screen wherever you choose at any location on any Web page. See code in Figure 27-5.

Figure 27-5 Snippet from invocation.jsp

```
<% /* sample programatic invocation
   <a href="WFInvocation.do?inv=INV1">Advanced
Hello World WebFaced</a>
*/ %>
```

Option Two of Two

There is another option, and I would expect that most would probably agree that this option is the better way to go. In essence, this option is the same as the first one shown above, with one exception, the end user does not see the parameters for invocation appended to the URL. This option is the same one our current Advanced Hello World application uses. Take a look at the code in Figure 27-6 now from the invocation.jsp file:

Figure 27-6 *Invocation.jsp Second Option*

```

<html:form action="WFInvocation.do"
styleId="INV1" onsubmit="var
rv=false;if(false!=document.wfflag){rv=true;}
document.wfflag=false;return rv;">

<html:hidden name="inv" value="INV1"
property="inv"/>
  <tr>
    <td width="50%" align="right"><html:submit
styleClass="launchbutton buttonup wpsButtonText
nav-h-normal"><bean:message
key="invocations.launch"/></html:submit></td>
    <td width="50%" align="left">Advanced Hello
World WebFaced</td>
  </tr>
</html:form>

```

If you are not a Web designer or a Web programmer, the above code more than likely looks “difficult and unintelligible” to you, but it’s actually very simple. This block of code does just a few things.

First of all, it creates a form and gives it the action WFInvocation.do as noted in Figure 27-7

Figure 27-7 *WFInvocation.do*

```

<html:form action="WFInvocation.do"

```

The action calls the page when the form is submitted, or in our case, when the *Launch* button is pressed. The next block of code accomplishes something very important to us. It is shown in Figure 27-8

Figure 27-8 *Make It Hidden*

```

<html:hidden name="inv" value="INV1" property="inv"/>

```

This code basically says to create an element in our form and make it hidden. In other words, it exists only in our code and cannot be seen on the screen / URL. You would give this hidden element the name

inv and assign it the value INV1. The property attribute is not needed for its use with WebFaced code.

If you recall from the first option, it required that we pass a parameter and a value for that parameter to WFInvocation.do in order to see the sign-on page. Any alteration of that parameter string would produce a blank page. The line of code that produces the hidden form element is shown in Figure 27-9.

Figure 27-9 Launch Block

```
<html:submit styleClass="launchbutton buttonup
wpsButtonText nav-h-normal"><bean:message
key="invocations.launch"/></html:submit>
```

A button is produced from this code, our launch button. When you click the button, it submits the form to the WFInvocation.do servlet. The only information from this form that is being sent is the hidden element with the parameter inv and the value of INV1.

Again, this example is almost identical to the first option but with the noted exception. The end user does not actually see the parameters that we must pass in order to actually get the sign on screen for the application.

This WebFaced Advanced Hello World application (HELLOAR001) relies heavily on Struts to do its grunt work. Most of the JavaServer pages that we have already shown (index.jsp and invocation.jsp) use Struts methodology.

Getting the Job Done

You can check out the code in Figure 27-10 in pure html to get this job done. You can take this code, for example, and place it over the code in index.jsp. This would then produce the same results as the launch screen. The code in Figure 27-10 below utilizes everything we've discussed in this Chapter so far.

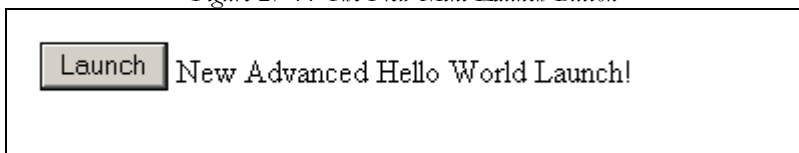
Look How Easy!

Figure 27-10 Advanced Hello World Code Snippet

```
<form name="InvocationForm" method="post"
action="/HELLOAR001/WFInvocation.do">
  <input type="hidden" name="inv"
value="INV1">
  <input type="submit" value="Launch">
  New Advanced Hello World Launch!
</form>
```

The code in Figure 27-10 produces the line below in Figure 27-11, and of course, that line (via the code) can be inserted into any Web page from which you would like to launch the Advanced Hello World application or your next big live Web application:

Figure 27-11 The New Mini Launch Button



As you can see, this code renders a simple gray button and some text next to the button. When the button is clicked, it launches the sign-on screen directly and links to the WebFaced application. Yes, this may look less handsome than our initial launch screen, but you would not use this as is. You would place this line in an existing Web page in which it would be a cooperative element and would blend in.

This block of code shows how easy it is to launch the HELLOAR001 WebFaced application or any WebFaced application without the use of Struts directly as our initial example does. By using this simple notion, this button provides you a simple method of launching your applications from the page of your choice.

By using the above method we could totally skip the launch screen by simply creating a jsp that redirects the user using our form or simply

using the above code to house a graphic that when clicked calls the submit of our form. The possibilities are endless and very easy to accomplish for any Web developer, or any person so inclined. All you need is what we have shown; either simple html or JavaServer pages.

Introducing WebFacing to Your Web Site

Suppose you already have a Web site, a personal Web site or an organizational based Web site, from which you would like to fire off your WebFaced application. How would you do it?

If you chose to follow the previous examples described in this chapter, the process would be reasonably simple and straight forward. We'll show you how to get that done right now.

If you look in Figure 27-12, you will see a sample Web page for a fictitious company, the infamous XYZ Company. Obviously, this can be a page from a full corporate Web site or a personal one page Web site. Either way, this example applies.

Figure 27-12 XYZ Corporation Before WebFacing



This page can be thought of as a template for any company or personal web site. As you can see, the site is rather bare, but it will suit our needs for this example. To keep it clear and simple, this Web site is done in just plain html using graphics and tables. Utilizing what we learned in this chapter, we will add a launch button to the web page to allow visitors to launch our WebFaced application directly without having to see the IBM launching screen.

Figure 27-12 shows the Web Page as it could look after we added the launch button (online system in blue) and code to the page.

Figure 27-13 XYZ Corp with a graphical launch button



Now our company web site allows users to launch our web faced application directly from the main page. Below is the clip of code in which was mentioned earlier in the chapter.

Figure 27-14 XYZ Corp WebFaced Online System Graphic Link

```
<center>&nbsp;  <B>XYZ Corporation</B><BR>123 WebFacing Ave<BR>IBM, New Y
<br>
<br>
<br>
<br>
<a href="WFInvocation.do?inv=INV1"src="images/Online_System.gif"></a>
<br>
```

Simple!

That's right! Only one line of html accomplishes this task. All you need do here is add a hyperlink to your WebFaced application. The link points to `WFInvocation.do?inv=INV1` and instead of seeing ugly text, you see a nice graphical button or link which launches the application.

There's more good news. This line of code can be placed anywhere on your main web site. It is important to note that where your Web site sits on your server is very important. The link to the WebFaced application provided here must be absolute. For example, if the Web site that will house this button is at the root of your server under '/', you must provide the full path to the WebFaced application. In other words, if the WebFaced application were called HELLOAR001, as in our Advanced Hello World WebFaced application, the hyperlink would be

```
<a  
href="/HELLOAR001/WFInvocation.do?inv=I  
NV1>
```

Another Example!

Suppose you would like to use the last example I provided, in which we use an invisible form and a launch button to fire up the WebFaced application. Just take a peek at the picture in Figure 27-15 to see the difference:

Figure 27-15 XYZ Corp Launch Button Using a Form

This example utilizes the example using a form and a submit button. The code is provided below:

Figure 27-16 XYZ Corporation Web Faced App Form

```

<form name="InvocationForm" method="post" action="/HELLOAR001/WFInvocation.do">
  <input type="hidden" name="inv" value="INV1">
  <B>XYZ's Online WebFaced Application</B><BR>
  <input type="submit" value="Launch">
</form>

```

All you do in this example is create a form and set the form's action attribute to **/HELLOAR001/WFInvocation.do**. Since Advanced Hello World is installed in a different directory, you must use an absolute path. Again the action attribute will be different depending on what you named your WebFaced application.

Although these examples are very simple, they leave you with a wide range of possibilities. These are just two simple ways of launching the recently WebFaced Advanced Hello World application, HELLOAR001.

Mission Accomplished!

It is definitely true to say that we delved into some pretty deep territory in this chapter. We began by analyzing the various aspects of our Advanced Hello World WebFaced application and the two pages that drove it, *index.jsp* and *invocation.jsp*. We then discussed how Struts plays into the big picture.

Working through the options, we accomplished the task at hand. We simplified our launch panel and showed how a simple block of code could be utilized in many different ways. With the knowledge gained in this chapter, you can now work with the launch panel and produce the results you desire for your shop..

A Hearty Thank You:

I would like to acknowledge and thank Matt Beutel, a student at Marywood University for his work in analyzing the functions used to launch WebFaced applications and his work in showing how this can be made to blend well with existing pages. This Chapter is a reflection on the fine work that this young man performs day in and day out as a student, and as an intern in the AS/400 shop in which I consult.

Thanks again Matt

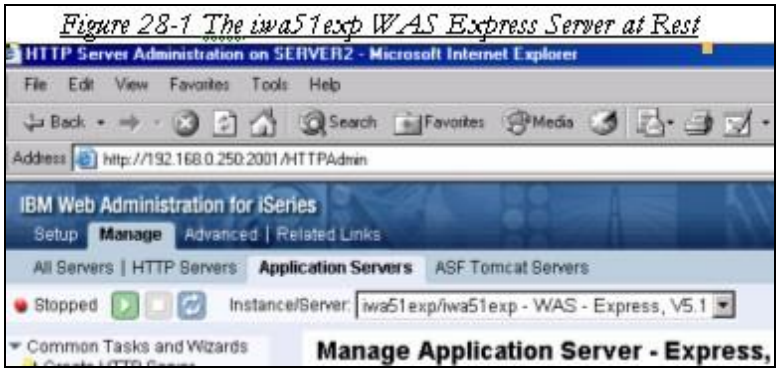
Chapter 28

Using iSeries Access for Web

Starting WAS Express & HTTP Servers

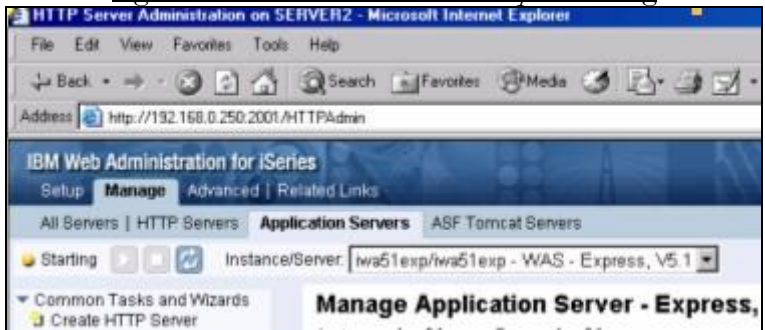
In Chapter 15, we set up iSeries Access for the Web so that we can now use it. You may recall seeing a panel similar to the one shown in Figure 28-1.

As noted previously, iSeries Access for Web depends on there being a fully functional WAS Server available as well as an HTTP server. In Chapter 15, we installed the iAWeb product to the iwa51exp WAS Express server that we had built specifically to run iSeries Access for Web. This server is linked to the iwa51exp Apache HTTP server.



Before you can use iSeries Access for Web (iAWeb), you must start the WAS Express Server *iwa51exp* in which it runs and you must also start the companion HTTP server of the same name. To get the WAS Express Server going, you would click on the green and white start button in the left frame of the panel shown in Figure 28-1. In a short while, the panel will change from a stopped mode to “starting.” As shown in the panel in Figure 28-2

Figure 28-2 WAS Instance iwa51exp Is Starting



While the WAS Express instance is starting, if you were to perform a WRKACTJOB command from an AS/400 command line, you would see a panel similar to that shown in Figure 28-3.

Figure 28-3 WAS Express Server Is Starting

```

Work with Active Jobs                                     S105PV4M
                                                         07/31/04 15:53:54
CPU %:    92.8      Elapsed time:  00:00:02      Active jobs:   170

Type options, press Enter.
 2=Change  3=Hold  4=End   5=Work with  6=Release  7=Display message
 8=Work with spooled files 13=Disconnect ...

Opt  Subsystem/Job  User           Type  CPU %  Function           Status
----  -
QASE51  QSYS            QSYS          SBS   .0     PGM-QASESTRSVR    DEQW
      IWAS1EXP     QEJBSVR       BCH   74.1   PGM-QASESTRSVR    RUN
QBATCH  QSYS            QSYS          SBS   .0     PGM-QASESTRSVR    DEQW
QCMN    QSYS            QSYS          SBS   .0     PGM-QASESTRSVR    DEQW
QCTL    QSYS            QSYS          SBS   .0     PGM-QASESTRSVR    DEQW
QSYSSCD QPGMR          QPGMR         BCH   .0     PGM-QEZSCNEP      EVTW
QHTTPSVR QSYS           QSYS          SBS   .0     PGM-QZHBMAIN      SIGW
      ADMIN       QTMHHTTP     BCH   .0     PGM-QZHBMAIN      SIGW
      ADMIN       QTMHHTTP     BCI   .0     PGM-QZSRLOG        SIGW
More...

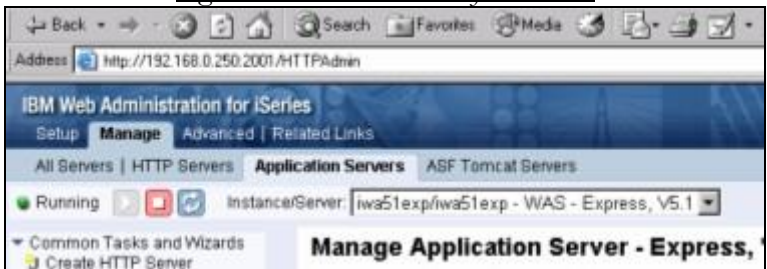
Parameters or command
====>
F3=Exit  F5=Refresh  F7=Find  F10=Restart statistics
F11=Display elapsed data  F12=Cancel  F23=More options  F24=More keys

```

You may grow impatient waiting for the WAS instance to start. It takes a while. So, from the WRKACTJOB panel you can tell whether it is doing a lot of work (74% of the system in Figure 28-3) or you can use the refresh option of the Admin GUI as shown in the panel in Figure 28-4.

Figure 28-4 Refresh the Starting Panel

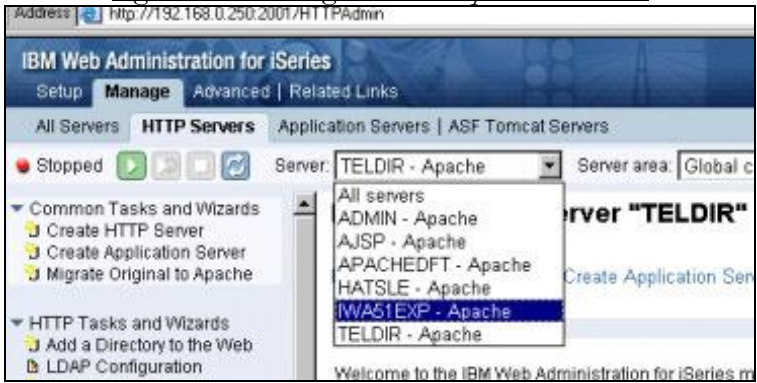
So, to refresh the panel to see if the WAS and the iAWeb application has started, use the blue refresh button as shown in Figure 28-4. Eventually, the WAS Express instance will start and you will see that it is started as shown in Figure 28-5.

Figure 28-5 WAS Started for iAWeb

Notice the word, running at the top of the left frame in the panel shown in Figure 28-5. When you scroll up on the right side of this panel, you will see that the iAWeb application has already started and is ready to roll. The

next step in getting iAWeb up and running is to get the companion HTTP server up and running. To do this, click on the Manage Tab and the HTTP Servers sub tab as shown in Figure 28-5. You will be taken to a panel similar to that shown in Figure 28-6.

Figure 28-6 Starting the iwa51exp HTTP Server



When you get to the panel as shown in Figure 28-6, you will select the iwa51exp server in the window as shown and then you will click the Start Button just as you did to start the WAS Express server named iwa51exp. This action starts the HTTP server. As you will see, the Apache HTTP server, unlike WAS Express starts almost immediately.

Figure 28-7 Apache HTTP Server iwa51exp Starts

As noted in Figure 28-7, the Apache server is started. The WAS server is also started. Thus, the iAWeb application is ready to be taken for a ride.

Starting the iAWeb Application from a Browser

The HTTP port that we selected for iwa51exp is 2042. Thus, that is how we tell our iAWeb application to talk to the HTTP server named iwa51exp. So, the very first steps after the HTTP server is up and ready to talk to the iwa51exp WAS Express server is to place the iAWeb launch URL into your browser and fire it off. The format of the URL is as follows:

`http://<server_name>:2042/
webaccess/iWAHome`

or for our environment:

`http://192.168.0.250:2042/
webaccess/iWAHome`

Tip:

If you substituted a different port than 2042 when the Web environment was created, you should use that port number in the above browser address. If all is well, you will get a sign on screen as shown in Figure 28-8.

Figure 28-8 Sign On to AS/400



Unless your system is a real ripper, the first iAWeb panel will cause you to Wait and Wait and Wait. It takes forever to appear so be patient. Finally, you will be filled with excitement as you see the very handsome panel as shown in Figure 28-9

Figure 28-9 iAWeb is Running in iwa51exp.



The iAWeb main panel is really well done and quite handsome. Moreover, there are lots that you can do with iAWeb. This book is not designed to tell you all about those things but the help text and IBM's documentation are poised to take you on a nice ride. It was my own personal experience that getting to the point that I could see this panel was the most difficult. It was fraught with many miscues and wrong turns and was not very easy. That's why we walked you through all of the ordering, prerequisites, installation, and set up acts that were needed to get you here.

We're just about done doing our thing. However, before we finish with this Chapter, we want to show you some of the neat things that you can do with IBM's iSeries Access for Web product. The first item on my list and probably

yours is 5250 emulation through your browser (without having to run iSeries Access or a Telnet client). The good news in all this is that running iAWeb is almost as easy as using a terminal.

Figure 28-10 Start a 5250 Session



Start 5250 Session

To start your 5250 session, on the left pane click on the Start 5250 session and the first panel you see will be the one shown in Figure 28-11. Remember, you have already signed on.

Figure 28-11 Start Panel for iAWeb 5250 Emulation

IBM  **iSeries Access for Web**

bkelly

Start Session

Server

Server:

Port:

Code page: ▼

Workstation ID

Use user ID

Specify workstation ID

Avoid duplicates for this user

Avoid duplicates with other users

General

Initial macro:

My Folder

My Home Page

Print

Messages

Jobs

5250

[Active sessions](#)

[Configured sessions](#)

[Start session](#)

Database

Files

Command

Download

Customize

Other

Related Links:

[iSeries Access for Web](#)

[iSeries Access](#)

[iSeries Navigator](#)

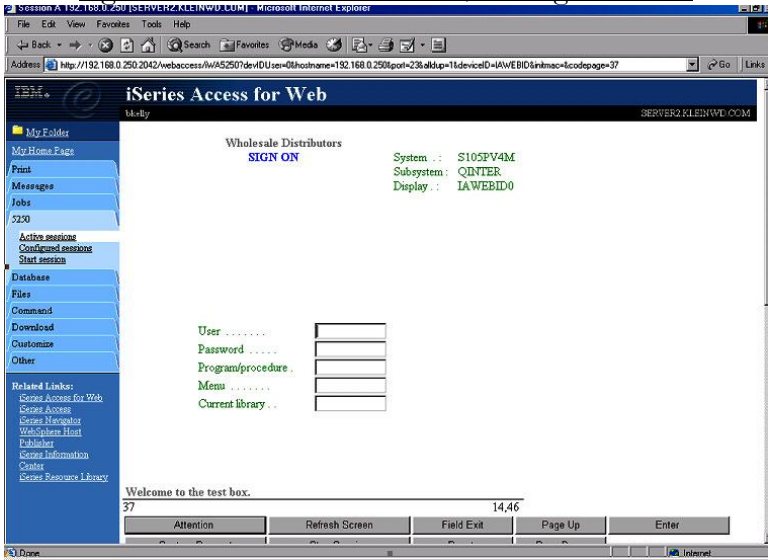
[WebSphere Host](#)

[Publisher](#)

The first functional 5250 panel that you get is very much like your session creation panel running the big iSeries Access product. You tell it your server and port (take the default) and you tell it the workstation ID that you would like to use when you come in. This is nice because your session can then be tracked and managed by WSID.

When you hit ENTER on that panel after completing it, you will be taken to a panel very similar to the one shown in Figure 28-12

Figure 28-12 5250 Emulation AS/400 Sign On Panel

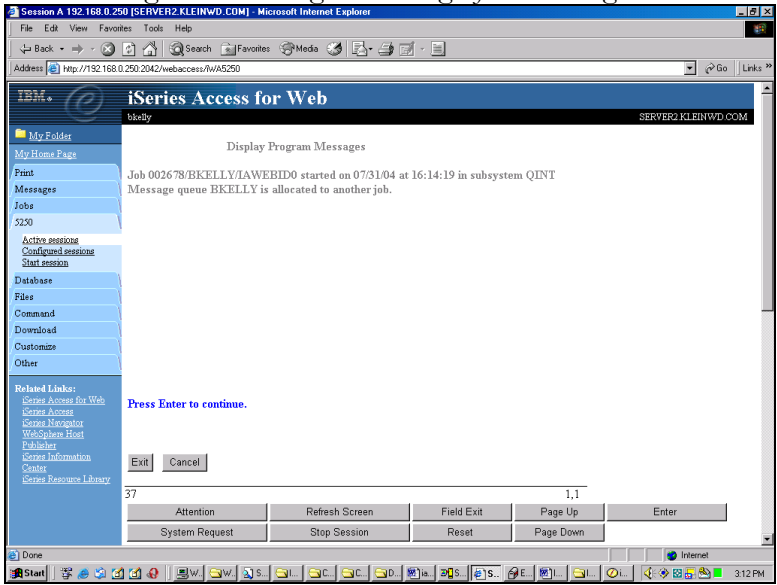


Sign on to the panel shown in Figure 28-12, and you will see one or two sign-on startup messages such as the ones shown in Figure 28-13 and 28-14.

Figure 28-13 Previous Signon Message

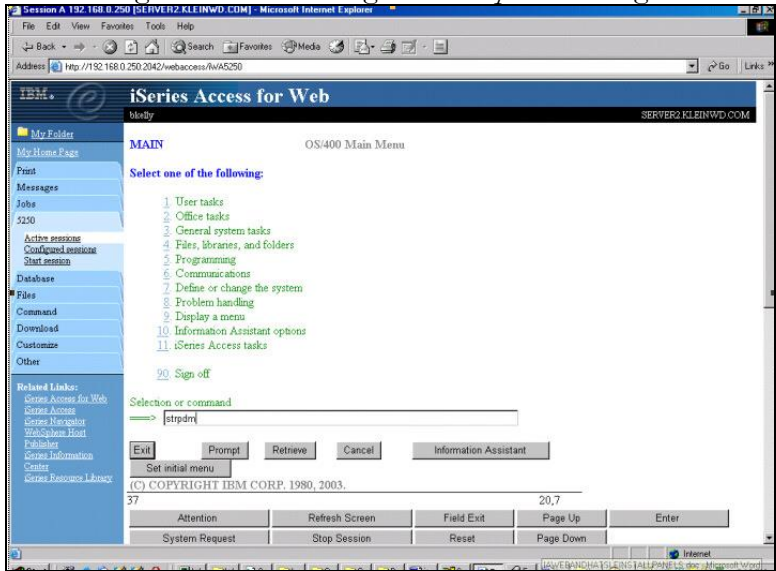


Figure 28-14 Program Messages for 5250 Signon



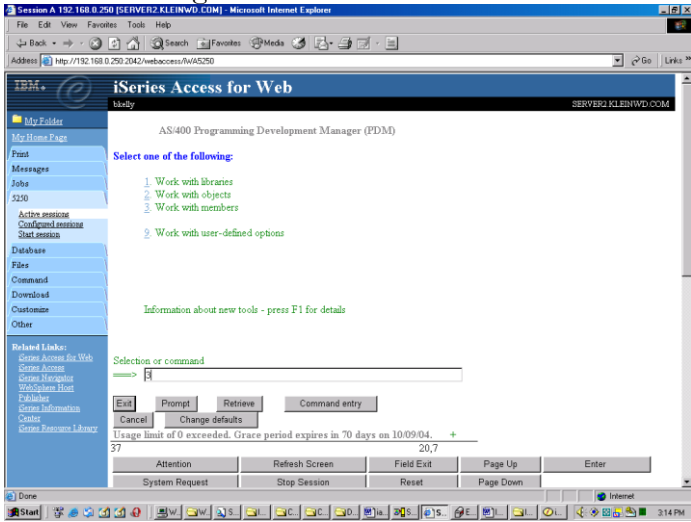
After you hit ENTER on those two panels, you will get the AS/400 Main Menu, shown in Figure 28-15, just as you would with a terminal session.

Figure 28-15 Start Program Development Manager



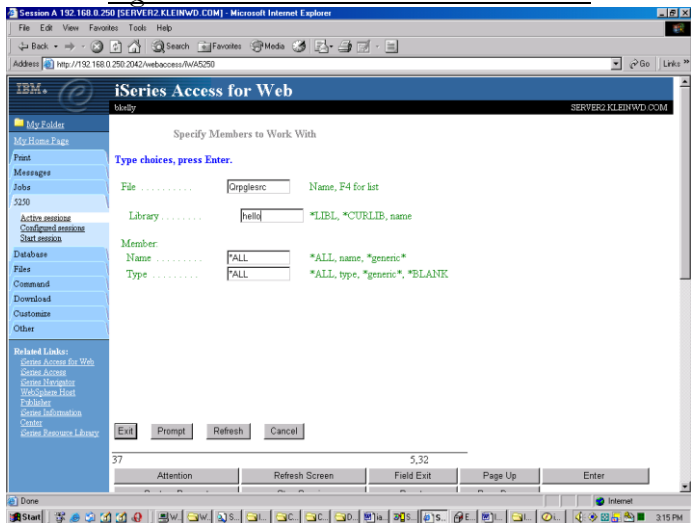
Since we are all programmers of sort, let's invoke the green screen Program Development Manager (PDM) from the Web by typing in the famous command STRPDM and then hit ENTER. You will see the Main PDM panel as shown in the panel in Figure 28-16.

Figure 28-16 Main PDM Panel



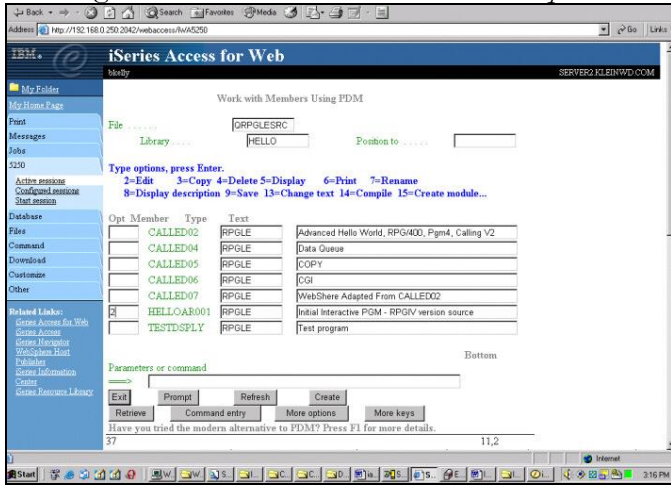
From the panel in Figure 28-16, select the 1 option for Work with members and press ENTER,

Figure 28-17 Pick Members to Work With

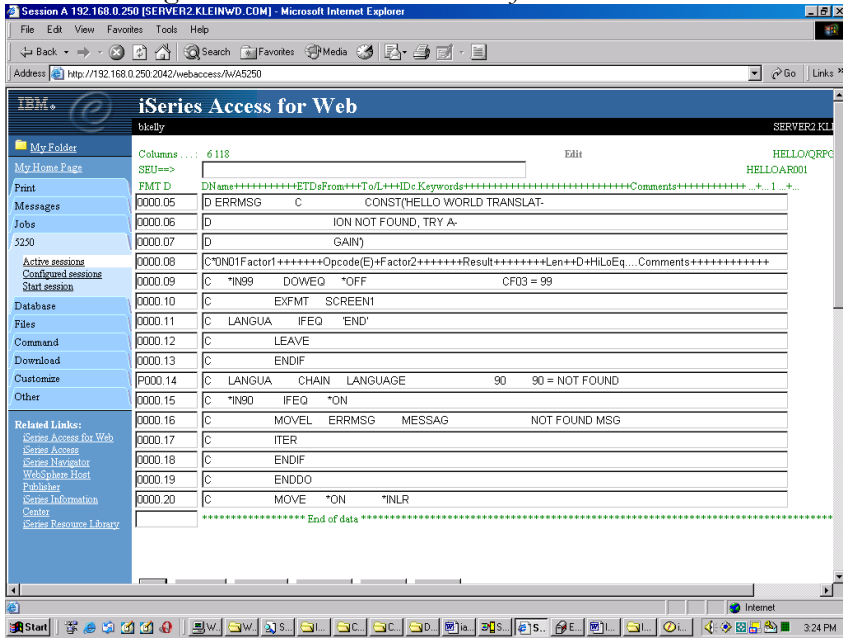


Fill in a Source File, such as QRPGLSRC for RPGIV, from Figure 28-17 and press ENTER.

Figure 28-18 PDM Work With Members panel



Place a 2 (EDIT Using SEU) next to HELLOAR001, the RPG program for our WebFaced example, and press ENTER.

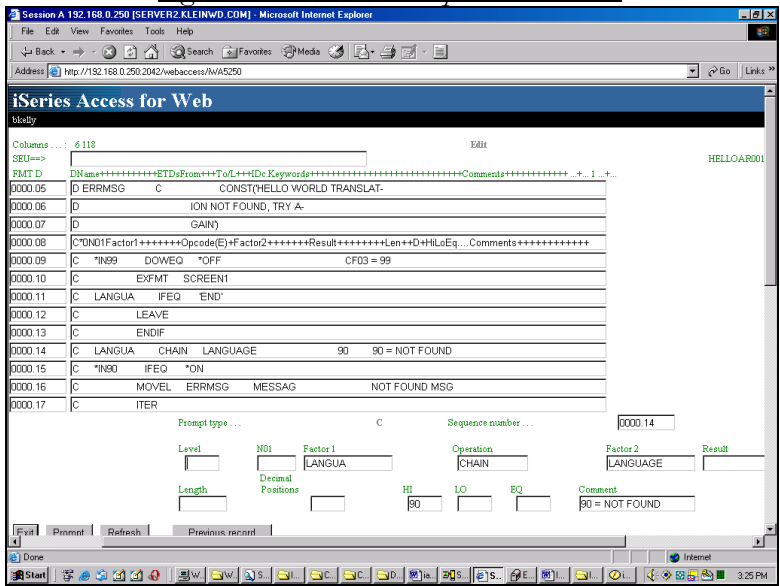
Figure 18-19 SEU Edit Session from PDM

One of the little glitches that I found using iAWeb to edit source RPG code was that the display did not work exactly as a terminal for SEU. For example, the lines are full of text as displayed, and though text can be inserted or deleted, you must delete a blank character on the line before you add a character to a line so there is room for the new character.

On the panel in Figure 28-19, you can see the top of the bottom of the SEU panel. If we prompted a statement and scrolled down, you would see how handsome the SEU prompter looks from the Web. This is shown in Figure 28-20.

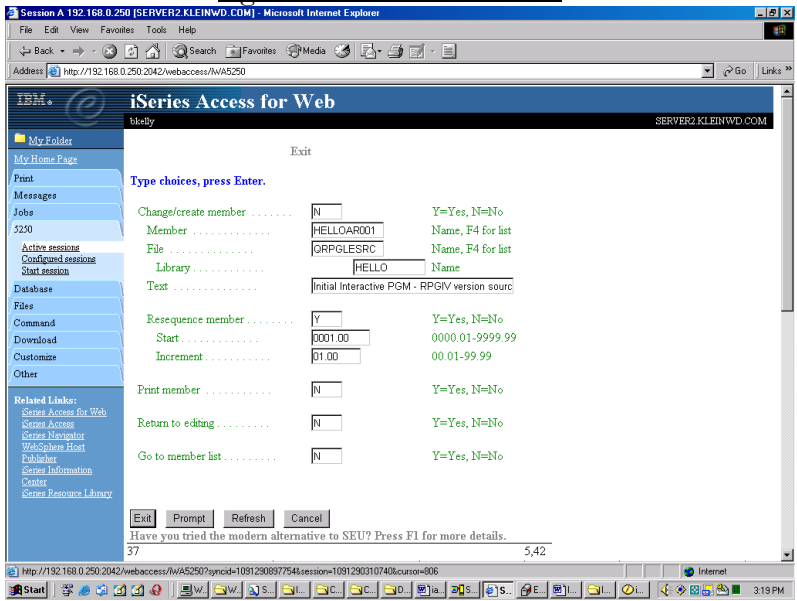
You also have to slide the scroll bar because the panel is not lined up w/o the prompter as you can see in the panel in Figure 28-20.

Figure 28-20 SEU Prompter Looks Nice



Notice how the prompter works in Figure 28-20, then hit F3 to end the SEU session and you'll get the SEU Exit panel as shown in Figure 28-21.

Figure 18-21 SEU Exit Panel



If you want to do more with iSeries Access for the Web, feel free to take IBM's links as follows:

- ✓ www.as400.ibm.com
- ✓ Library
- ✓ iSeries Information Center
- ✓ North America
- ✓ V5R3
- ✓ Connecting to iSeries
- ✓ iSeries Access
- ✓ iSeries Access for Web

For now, you have succeeded in ordering, installing, fixing, configuring, and using iSeries Access for Web.

Chapter 29

Using HATS/LE

Transformation to the Web

This Chapter completes our work on the WebSphere Host Access Transformation Services Limited Edition (HATS/LE) Version 5 (V5).

Host Application Transform Server Limited Edition (HATS/LE) is the newest member of the iSeries Access family entitlements. As noted in Chapter 10, it has no real relationship with iAWeb and I speculate that it is shipped with the iAWeb product more as a replacement for the Host Publisher entitlement than because iAWeb needs HATS/LE.

Conversion on the Fly

The major claim to fame of HATS Limited Edition is that it dynamically transforms 5250 screens with a point-and-click Web interface that is fairly attractive. Screens are converted on the fly and delivered as HTML to the end user's Web browser (Internet Explorer or Netscape). The screens do not particularly look like they are 5250 terminal oriented and thus, that is the major differentiator between HATS/LE and 5250 component of the iAWeb product.

The pages look more like WebFaced pages than terminal panels.

Introduction of HATS/LE to iSeries

On June 30, 2003, IBM began to include HATS/LE with new shipments of V5R2 iSeries Access Family. For those customers who had acquired V5R2 iSeries Access Family prior to this time, IBM continues to make HATS/LE available by ordering an iSeries Access Family Refresh. This information is covered in detail in Section II Prerequisites, Ordering, Installing, Fixing.

HATS/LE Vs. WebFacing

A major difference between HATS/LE and WebFaced applications is that no front-end work at the application level is required with HATS/LE. In fact, HATS/LE does not need access to the display file DDS source in order to do its "on the fly" conversion magic. It works directly with the applications pre-compiled display files in much the same fashion as Client Access for Windows.

A major differentiator is that HATS/LE driven 5250 applications can be spruced up in much the same fashion as WebFaced applications without all the work. Just like WebFacing, however, there is no need for programming skills while the product rejuvenates 5250 applications with a nice Web-like look and feel. It does this with a number of stock HTML templates that shape the data stream into a non-Telnet look. These templates can be tailored if you choose giving far greater capabilities than the iAWeb Telnet like capability. For example, you can change the

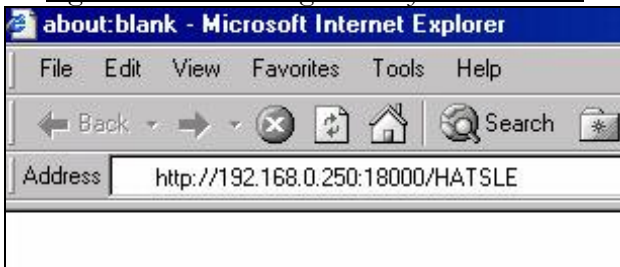
company name and logo in the template to spruce up the look of the applications you use on the Web.

Of course, you can't do any HATS/LE activity without WebSphere. Though IBM says it integrates with WebSphere Application Server, as we showed in Section III, in actuality HATS/LE is an application that runs under WebSphere. Just as iAWeb it runs on many different iSeries WebSphere types besides Express.

Using HATS/LE as your AS/400 Web Interface

When you have finished creating the various configurations that you want to use with HATS (Chapter 16) it's time to see what it looks like in operation. Actually, it is better to see what each new environment looks like as you create it. So, in our little example, we have just one session defined. In this way, it is easy for us to connect and become immediately functional. To get going with HATS/LE, type the URL from Figure 16-23 (The HATS LE Configuration Completion Panel), into a browser window as shown in Figure 29-14.

Figure 29-14 Launching HATS for 5250 Work



From this point on, you are working in an emulated terminal environment from the Web. As you would expect, the next thing you see is exactly the same thing you would see when you launch a terminal session. You get an AS/400 sign-on panel that looks very

much like a green screen panel except for a number of frills that you picked out in the configuration process as you can see in the sign-on panel shown in Figure 29-15.

Figure 29-15 HATS/LE 5250 Signon to Your AS/400

Wholesale Distributors
SIGN ON

System . . : Q
Subsystem : Q
Display . . : Q

User
Password
Program/procedure .
Menu
Current library . .

HATS/LE is smart enough to permit the ENTER key to work with this panel. So, after you type your sign on information, hit the ENTER key on your PC and you will see the typical next two panels in the green screen sign on sequence as shown in the panels in Figure 29-16 and 29-17.

Figure 29-16 Hats 5250 Signon / Change Password Panel

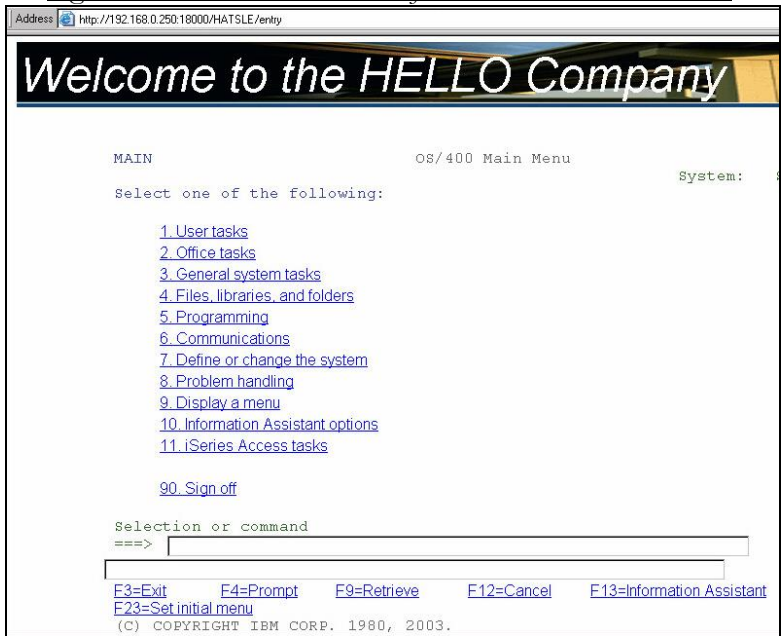
The options in Figure 29-16 are the same as any second panel of the signon sequence, though I have crunched the panel a bit to save some space. When you hit ENTER again, if you are already signed on at another terminal session of any kind, you will get the typical third panel of the signon sequence as shown in Figure 29-17

Figure 29-17 HATS/LE Message Q Allocation Message

The panel in Figure 29-17 shows that the user of this HATS/LE session is also signed on in another session via a terminal, iSeries Access, iSeries Access for Web or another HATS/LE session.

Press ENTER to continue the signon and you will come to the panel as shown in Figure 29-18.

Figure 29-18 HATS/LE Transformed OS/400 Main Menu



Now, there's a familiar sight isn't it? Of course, it uses the simplistic selections that I made in the configuration session. With other options, you can really gussy up this simple browser based 5250 emulation session. Notice that the link option makes all of the menu options and PF keys hot for the clicking. Other transformation options would produce buttons and other spiffy little things. You make the call.

Now, that you are on a main menu from green screen, I'm sure you can handle the rest from here.

To configure a different look and feel for HATS/LE, feel free to revisit the second half of Chapter 16 along with IBM's Web documentation. Once you have HATS/LE working as shown in Figure 19-18, tailoring it is merely a process of changing setup options and test driving HATS/LE to see whether you like what you would get.

Chapter 30

Summary and Conclusions

Easier Than You'd Think!

Once you make a WebSphere application work on any version and platform, you want to whoop and holler. For an AS/400 or iSeries professional, the difficulties of WebSphere are unlike any that have ever come your way. When you conquer this alien that lives in the AS/400 or iSeries land, you just have to feel good about it. Of course, WAS Express is not like that at all. Really!

WAS Has Been with You

Back in the V4.3 and V4.4 days of OS/400, IBM increased the disk usage of all AS/400s by automatically shipping the WAS V2.X server with the base operating system. Nobody seemed to care since few AS/400 aficionados had taken up

the WebSphere calling, and what's a little disk space anyway!

Then the intrigue of using servlet servers for applications caused some more iSeries implementers to move and the promise of the WebSphere Development Tools caused others to take the WAS plunge. Unfortunately for most, by the time they chose to come on board, they had to order the then current and free version (V3.5 standard) of the WAS server. The preinstalled version 2.X was well out of date.

Tough WAS Times Are Gone

Installing WAS 3.5 Standard Edition was at best difficult. But, IBM's trained Support Line people helped many through it successfully. If Rochester had designed WebSphere, it would not seem like it was built for Unix Nerds and Windows Weenies. Unfortunately, Unix Nerds designed and implemented WebSphere on big IBM platforms without considering those in iSeries land who aren't used to inventing a new tool for each sized nut that needs to be fastened.

Though it was tougher than anything I can recall in years, because it was unassuming and would often not work without giving obvious clues why, I got through it with IBM's help. And it worked very well and served Web applications flawlessly over the last few years in my case.

Enter WAS Express. What a breath of fresh air. You can install WAS Express in days and have something going. It is very AS/400-like. Compared to its predecessors, I think

you will like it. And, now with V5R3, it is free in this Express version along with the operating system.

The Web Tools

The IBM Web Tools are really not that bad. They're not how I would have done it and they're not exactly how you would have wanted them, but they are getting better and better. You can really use these tools, all of them to get yourself to dynamic data and the Web. IBM no longer tells its AS/400 constituents that they must use Java for the Web. WebFacing, iSeries Access for the Web, and HATS/LE have the bases covered for leveraging green screen development skills on the Web.

Ideally

We may never see ideal and that is a fact. IBM has four server lines and to assure compatibility between the lines, it has prepared major software components to port to all platforms. Unfortunately, for AS/400 purists, that's not part of the AS/400 tradition. The best thing that IBM could do for its AS/400 constituency is play to them, not a foreign audience. Though WebSphere is functional, it is not integrated. It may be free at some levels but it is not integrated.

The AS/400 database was integrated first and then given a name. It's still different from all other databases and in many ways it is better because it enhances programmer productivity in customer shops, not in IBM's Laboratories.

Since more than likely, my brand of ideal is probably not coming to the AS/400 any time soon, it is heartening that with just a little bit of knowledge of the tools discussed in this book, you can get your company on the Web in short order. Yes, it would be nice if it was

AS/400 style, but it isn't. So, the next best thing is to have tools that are capable of leveraging your skills and mine to get applications to the Web. The tools have finally arrived.

Enjoy!

Appendix A

Code for the Advanced Hello World Application

RPG/ COBOL DDS for the Advanced Hello World Application

The code for the HELLOAR001 application is on the following pages:

Figure A-1 PANEL Display File for Advanced Hello World

```

A**%FD Display File Panel For Advanced Hello World
A                                DSPSIZ(24 80 *DS3)
A                                CF03(99 'end of job')
A                                R SCREEN1
A                                PRINT(03)
A                                12 16'Please ENTER your selection here a-
A                                nd press ENTER'
A                                14 7'LANGUAGE:'
A                                DSPATR(HI)
A                                LANGUA R I 14 18REFFLD (REFFMT/LANGUA HELLO/LANGUAGE)
A                                MESSAG R O 16 7REFFLD (REFFMT/MESSAG HELLO/LANGUAGE)
A                                2 7'Welcome To The Advanced Hello Worl-
A                                d Application'
A                                DSPATR(HI)
A                                DSPATR(UL)
A                                4 7'To See Hello World In French, ente-
A                                r the word (FRENCH) as your selecti-
A                                on'
A                                6 7'To See Hello World In German, ente-
A                                r the word (GERMAN) as your selecti-
A                                on'
A                                8 7'To See Hello World In Spanish, ente-
A                                r the word (SPANISH) as your selec-
A                                tion'
A                                22 7'Press Command key 3 (F3) to end th-
A                                is job or type END for LANGUAGE'
***** End of data *****

```

Figure A-2 Language Database File for Advanced Hello World

```

***** Beginning of data *****
A*                                ('LANGUAGE DB FILE FOR HW')
A                                R REFFMT TEXT('NAME OF DB RECORD FORMAT')
A                                LANGUA 7 COLHDG ('LANGUAGE')
A                                ALIAS (LANGUAGE)
A                                TEXT ('KEY FIELD FOR LOOKUP')
A                                MESSAG 73 COLHDG ('HELLO LANGUAGE TEXT')
A                                TEXT ('CONTAINS LANG. TRANSLATION')
A* BELOW REPEAT OF LANGUA FIELD IS TO DEFINE IT AS THE KEY TO FILE
K LANGUA
***** End of data *****

```

Figure D-3 RPG/400 version of HELLOAR001 AHW Program

```

***** Beginning of data *****
FPANEL CF E WORKSTN
FLANGUAGEIF E K DISK
I 'HELLO WORLD TRANSLAT-C ERRMSG
I 'ION NOT FOUND, TRY A-
I 'GAIN'
C *IN99 DOWEQ*OFF CF03 = 99
C EXFMTSCREEN1
C LANGUA IFEQ 'END'
C LEAVE
C ENDIF
C LANGUA CHAINLANGUAGE 90 90 = NOT FOUND
C *IN90 IFEQ *ON
C MOVELERRMSG MESSAG NOT FOUND MSG
C ITER
C ENDIF
C ENDDO
C MOVE *ON *INLR
***** End of data *****

```


Figure D-4 RPGIV version of HELLOAR001 AHW Program

```

F*filename++IPEASFRlen+LKlen+AIDevice+.Keywords+++++
FPANEL      CF      E          WORKSTN
FLANGUAGE   IF      E          K DISK
D*ame+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D ERRMSG      C          CONST ('HELLO WORLD TRANSLAT-
D                                     ION NOT FOUND, TRY A-
D                                     GAIN')
C*0N01Factor1+++++Opcode (E)+Factor2+++++Result+++++Len++D+HiLoEq.
C      *IN99          DOWEQ      *OFF
C                                     EXFMT      SCREEN1
C      LANGUA        IFEQ        'END'
C                                     LEAVE
C                                     ENDIF
C      LANGUA        CHAIN        LANGUAGE      90
C      *IN90          IFEQ        *ON
C                                     MOVEL      ERRMSG      MESSAGE
C                                     ITER
C                                     ENDIF
C      ENDDO

```

Figure D-5 COBOL Version of HELLOAC001 AHW Program

```

***** Beginning of data *****
.....-A+++B+++++
PROCESS
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLOAC001.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT DB-LANGUAGE
        ASSIGN TO DATABASE-LANGUAGE
        ORGANIZATION IS INDEXED
        ACCESS MODE IS RANDOM
        RECORD KEY EXTERNALLY-DESCRIBED-KEY
        FILE STATUS IS MF-STATUS.
    SELECT DISPLAYPANEL
        ASSIGN TO WORKSTATION-PANEL
        ORGANIZATION IS TRANSACTION
        ACCESS MODE IS SEQUENTIAL
        FILE STATUS IS WS-STATUS.
DATA DIVISION.
FILE SECTION.
FD DB-LANGUAGE
    LABEL RECORDS ARE STANDARD.
01 LANGUA-RECORD.
    COPY DDS-REFFMT OF LANGUAGE.
FD DISPLAYPANEL
    LABEL RECORDS ARE STANDARD.
01 PANEL-RECORD PIC X(150).
WORKING-STORAGE SECTION.
01 PNL-INPUT.
    COPY DDS-SCREEN1-I OF PANEL.
01 PNL-OUTPUT.
    COPY DDS-SCREEN1-O OF PANEL.
01 WS-STATUS PIC XX.
01 MF-STATUS PIC XX.
01 INDON PIC 1 VALUE B'1'.
01 INDOFF PIC 1 VALUE B'0'.
PROCEDURE DIVISION.
BEGIN.
    OPEN I-O DISPLAYPANEL.
    OPEN INPUT DB-LANGUAGE.

```

```

PERFORM SCREEN-IO THRU EXIT-SCREEN-IO
  UNTIL IN99 OF PNL-INPUT = B'1'.
CLOSE-ALL.
  CLOSE DB-LANGUAGE DISPLAYPANEL.
  STOP RUN.
SCREEN-IO.
  WRITE PANEL-RECORD FROM PNL-OUTPUT
  FORMAT IS 'SCREEN1'.
  READ DISPLAYPANEL INTO PNL-INPUT
  FORMAT IS 'SCREEN1'.
  IF IN99 OF PNL-INPUT IS EQUAL TO B'1'
    GO TO EXIT-SCREEN-IO.
  MOVE LANGUA OF PNL-INPUT TO
    LANGUA OF LANGUA-RECORD
  READ DB-LANGUAGE
    INVALID KEY PERFORM LANGUA-NOT-FOUND
    NOT INVALID KEY PERFORM LANGUA-FOUND.
EXIT-SCREEN-IO.
  EXIT.
LANGUA-FOUND.
  MOVE CORRESPONDING REFFMT TO SCREEN1-O OF
    PNL-OUTPUT.
LANGUA-NOT-FOUND.
  MOVE 'HELLO WORLD TRANSLATION NOT FOUND, TRY AGAIN'
    TO MESSAG OF PNL-OUTPUT.
***** End of data *****

```


LET'S GO PUBLISH! Books:

(sold at www.itjungle.com , www.mcpressonline.com, www.iseriesnetwork.com)
 LETS GO PUBLISH! is proud to announce that more AS/400 and iSeries books are becoming available to help you inexpensively address your AS/400 and iSeries education and training needs: Our titles include the following: email letsGOPublish@kellyconsulting.com for ordering information

Getting Started With The WebSphere Development Studio for iSeries .

Your introduction to the new IBM strategy for Application Development. Includes a case study and examples of UI / Logic separation and CPW savings techniques.

The iSeries Pocket Developers' Guide.

Comprehensive Pocket Guide to all of the AS/400 and iSeries development tools - DFU, SDA, etc. You'll also get a big bonus with chapters on Architecture, Work Management, and Subfile Coding.

The iSeries Pocket Database Guide.

Complete Pocket Guide to iSeries integrated relational database (DB2/400) – physical and logical files and DB operations - Union, Projection, Join, etc. Written in a part tutorial and part reference style, this book has tons of DDS coding samples, from the simple to the sublime.

The iSeries Pocket Query Guide.

If you have been spending money for years educating your Query users, and you find you are still spending, or you've given up, this book is right for you. This one QuikCourse covers all Query options.

Getting Started With The WebSphere Development Studio Client for iSeries

(WDSC) Focus on client server and the Web. Your introduction to the client server and web development tools. Includes CODE/400, VisualAge RPG, CGI, WebFacing, and WebSphere Studio. Case study continues from the Interactive Book.

The iSeries Pocket WebFacing Primer.

This book gets you started immediately with WebFacing. A sample case study is used as the basis for a conversion to WebFacing. This interactive 5250 application is WebFaced in a case study format before your eyes. Either learn by reading the book or read while working along on your own system.

Migrating to WebSphere Express for iSeries: Your Roadmap for Migrating Applications to WebSphere Express

A Comprehensive guide designed to be your roadmap for moving to WAS Express for iSeries. It is loaded with examples and structured for easy learning. Through an easy to understand sample case study, you experience a real migration, and you learn the gotchas before they getcha! This book is designed to be a companion to all of your WAS Express migration efforts in the iSeries environment

Getting Started with WebSphere Express Server for iSeries: Your Step-by-Step Guide for Setting Up WAS Express Servers

A Comprehensive guide to setting up and using WebSphere Express. It is filled with examples, and structured in a tutorial fashion for easy learning. The book is designed to take you to a point at which you understand the notion of a servlet server, what WebSphere Express is, where it came from, how to order it, how to set it up, and how to make it work in your shop.

The WebFacing Application Design & Development Guide:

The Step by Step Guide to designing green screen iSeries applications for the Web. This is both a systems design guide and a developers guide. Using this guide, you will understand how to design and develop Web applications using regular workstation interactive RPG or COBOL programs. When you learn the tricks, and observe the sample code in action, you might choose to develop all your applications using this approach.

Can the AS/400 Survive IBM

Exciting book about the AS/400 in an iSeries World.

