

Chapter 1 **Extension**

What is New in WebSphere Application Server V6.1: Packaging, Architecture and Features

In this document you will see the important new features that have been added in WAS V6.1. Refer to the printed version of the book (get more details from www.WebSphereMentor.com, if you want to purchase one) to get a list of the features that were added to V6 as compared to V5.1. Here we discuss only the new features between V6 and V6.1. We are assuming that you read chapter 1 in the printed version of the book, which is a prerequisite to understanding the topics we cover in this extension document. The list below summarizes the important extensions to WebSphere Application Server V6.1 at a high level.

- Portlet application support (JSR 168)
- SIP servlet application support (JSR 116)
- Java 5 Software Development Kit (SDK)
- Java Server Faces (JSF) 1.1
- Completely re-designed Eclipse based Application Server Toolkit (AST) that supports a Jython editor for automation apart from J2EE development and testing environment
- Cloudscape 10.1.x support which is a pure Java based database based on Derby (an Open source RDB)
- Many new features and extensions as discussed in **New Features** section below (Refer to V6.1 information center at <http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp> for complete list).

WAS V6.1 is a full installation. This means you don't install V6.1 on top of an existing V6 installation (V6.0.2.9 for example), the way you apply a refresh pack, fixpack or interim fix. This also means you don't need a WAS V6 installation as a pre-requisite to install WAS V6.1. Instead you install it right from the product binaries and then create the profiles etc. If you are familiar with WAS V6 and read the printed version of this book then you will not have any difficulty using the WAS V6.1 product. The WAS V6.1 core architecture remains the same as V6, but comes with many useful new features and extensions. In WAS V6.1 many tasks have been automated and merged to make it simple and straight forward as compared to WAS V6. For example, you can 1). Create a profile at the time of product binary installation itself and 2). Create a Cell configuration (Deployment Manager + Federated node within an application server profile) in one step. Refer to **New Features** section later in this chapter for more information.

Refer to Chapter 1 in the printed version of the book for a basic understanding of the relationship between product binaries, templates, profiles, servers, and various packages

and architectures possible with each package and, limitations if any. The book also discusses the different services and containers in the application server, deployment manager and products (HTTP server, Plug-in, Load Balancer, Caching proxy etc.) and their relationship with the WebSphere Application Server. All that information is in the printed version of the book and still is valid for V6.1.

WebSphere Application Server V6.1: New Features

You'll find a host of improvements and enhancements in V6.1 from V6. We will delve more into these new features on later chapter extensions, but following are the highlights:

- Profiles can be created during the installation of product binaries in WAS V6.1 ND environment unlike in V6. In WAS V6, the creation of profiles was a separate step after installing the product binaries.
 - This means you will have a single response file (responsefile.nd.txt) for both product binaries and profile creation during the silent installation. Refer to Silent Installation section of chapter 3 in the printed version of the book for more information about silent the installation process in WAS V6.
 - The base package in WAS V6.1 creates an application server profile by default during the product binary installation just like it used to do in V6. But name of the default application server profile in V6.1 is AppSrv01 instead of "default" as is in case of V6 making it uniform with the ND package.

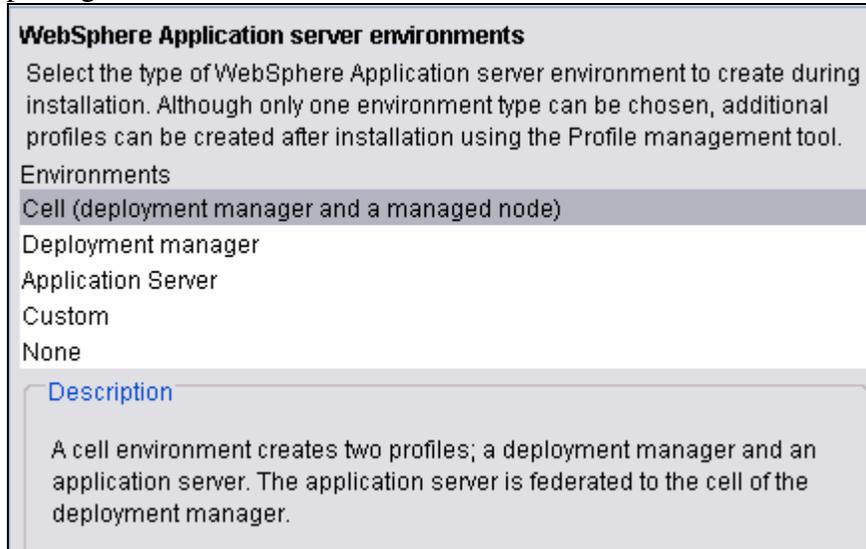


Fig 1a-1: Creation of profiles during the product binaries installation

- Create **Cell** profile (deployment manger profile + application server profile (without default applications) + federate node within the application server profile to the deployment manager's cell) all in one step in addition to the creation of deployment manager, application server and custom profiles individually just like you do in V6.

- New **profile template for Cell** has been introduced in product binaries to create a Cell based profile. Look for the **cell** template under **<WASV61-ROOT>\profileTemplates** directory as shown in the screenshot below. Refer to chapter 1 in the printed version of the book to know the difference between a profile template and actual profile.

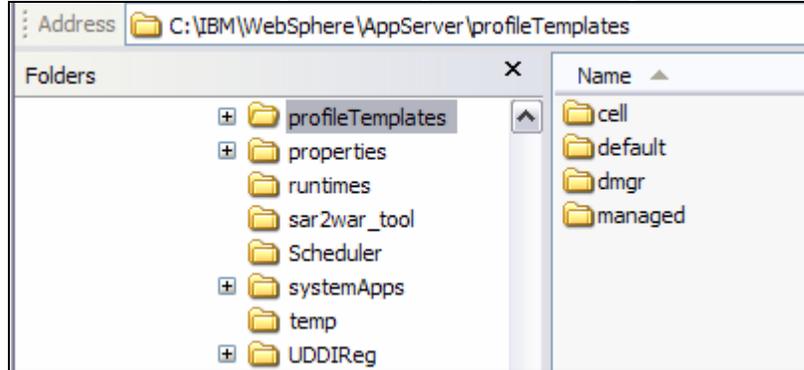
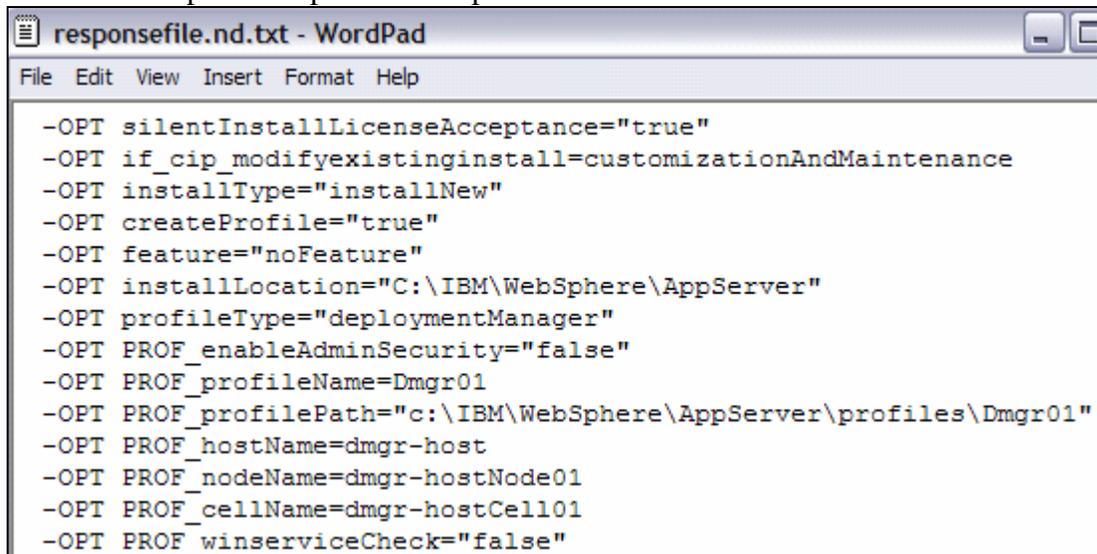


Fig 1a-2: Cell Template introduced in WAS V6.1

- WAS V6.1 silent installation response file (for example responsefile.nd.txt) uses user-friendly option names as opposed to difficult naming conventions used in earlier versions which were based on internal bean and property names that are vulnerable to changes between releases. Refer to the screenshot below for an example of the silent installation script and compare to the silent installation script of V6 sample in chapter 3 in the printed version the book.



```
-OPT silentInstallLicenseAcceptance="true"  
-OPT if_cip_modifyexistinginstall=customizationAndMaintenance  
-OPT installType="installNew"  
-OPT createProfile="true"  
-OPT feature="noFeature"  
-OPT installLocation="C:\IBM\WebSphere\AppServer"  
-OPT profileType="deploymentManager"  
-OPT PROF_enableAdminSecurity="false"  
-OPT PROF_profileName=Dmgr01  
-OPT PROF_profilePath="c:\IBM\WebSphere\AppServer\profiles\Dmgr01"  
-OPT PROF_hostName=dmgr-host  
-OPT PROF_nodeName=dmgr-hostNode01  
-OPT PROF_cellName=dmgr-hostCell01  
-OPT PROF_winserviceCheck="false"
```

Fig 1a-3: User-friendly options introduced in response file in WAS V6.1

- Profile creation is made easier, simpler and with fewer steps.
 - WAS V6.1 introduced two options during the profile creation (**Typical** and **Advanced**). If you quickly want to create a profile for development, education or test purposes select the Typical option that uses default values (instead of prompting to enter values for profile name, ports etc.) during the profile creation.

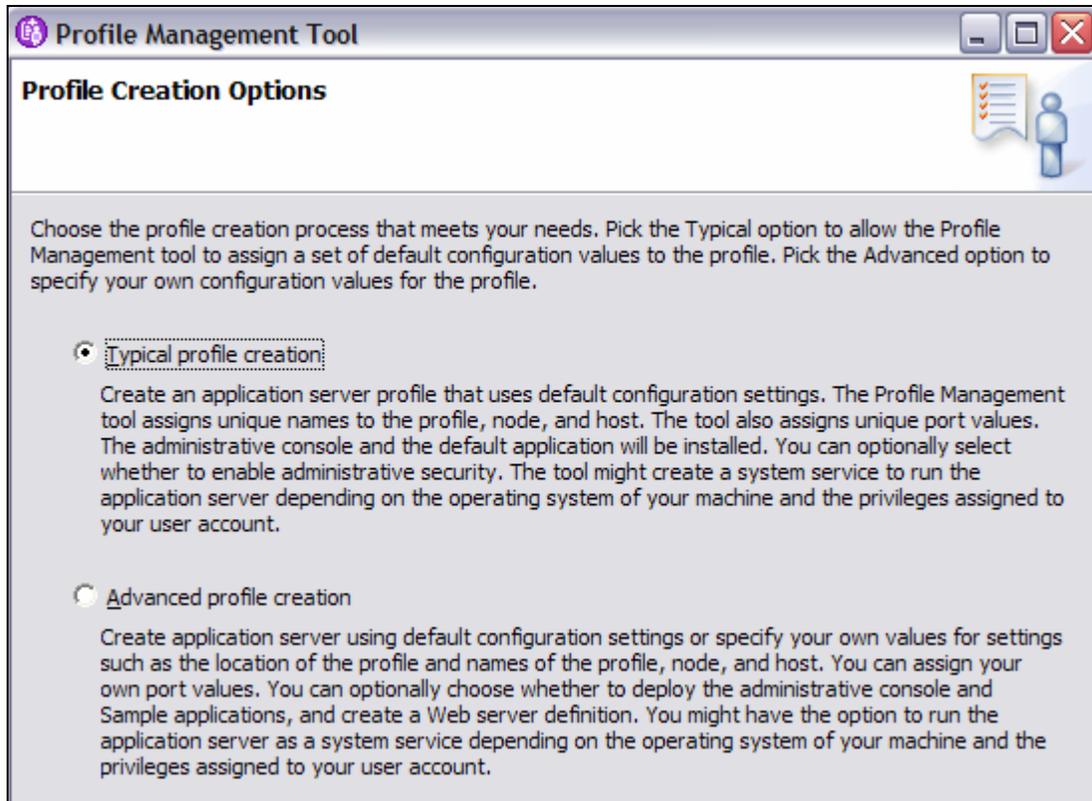


Fig 1a-4: Profile Options to select during the profile creation

- Support of Non-Root (or Non-admin group member in windows) installation in both silent and GUI installation modes. But some of the activities that requires root (admin) privileges won't be performed in case of a non-root installation. For example 1). Creation of windows service, 2). Detection of port conflicts against other user installations and 3). Setting ports below 1024 for WebSphere process are not possible that requires root (or admin) privileges. You will learn more on this in the chapter 3 extension.
- Able to select the **Default ports** (9060, 9080 for example) or **Recommended ports** (automatically setting the unoccupied ports during the installation in multi-profile environment. 9061, 9081 for example).
 - You can set the default ports in multi-profile environment with click of a button that was not possible in V6 (you had to manually change the ports). This feature is very useful if you know that you don't start the profiles with conflicting ports at the same time.

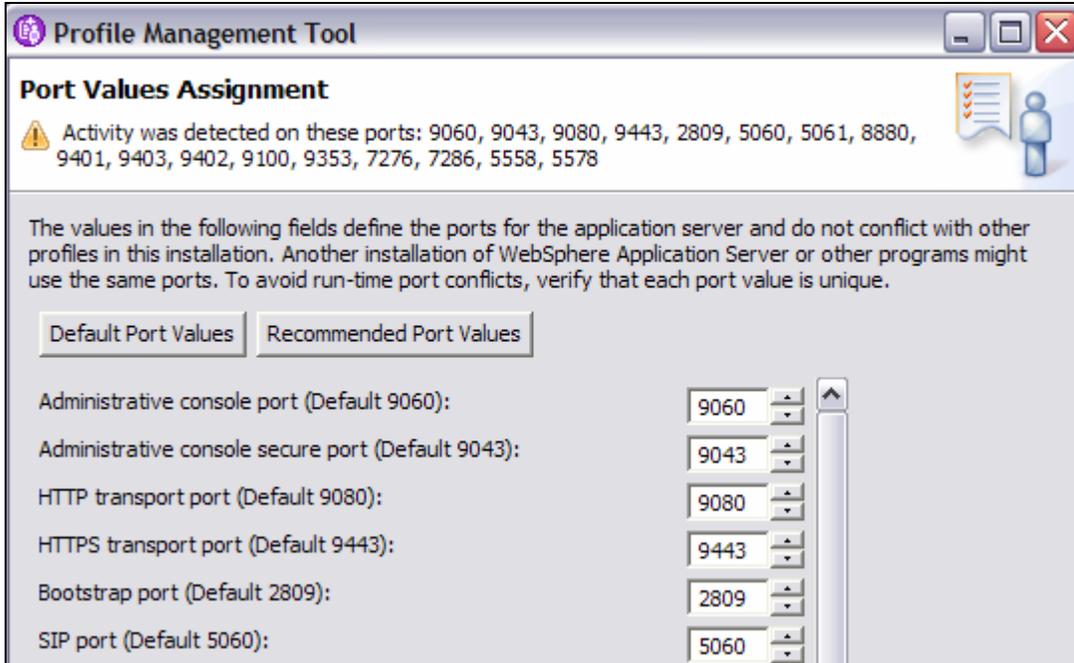


Fig 1a-5: Selection of Default (or Recommended) Ports with click of a button

- Create **Out of Box** secure environment by providing option to enable administrative security during the profile creation.
 - WebSphere uses the file-based custom registry to store the administrative user(s) information.
 - In WAS V6.1 Administrative security has been separated from application security for better control unlike in previous versions.

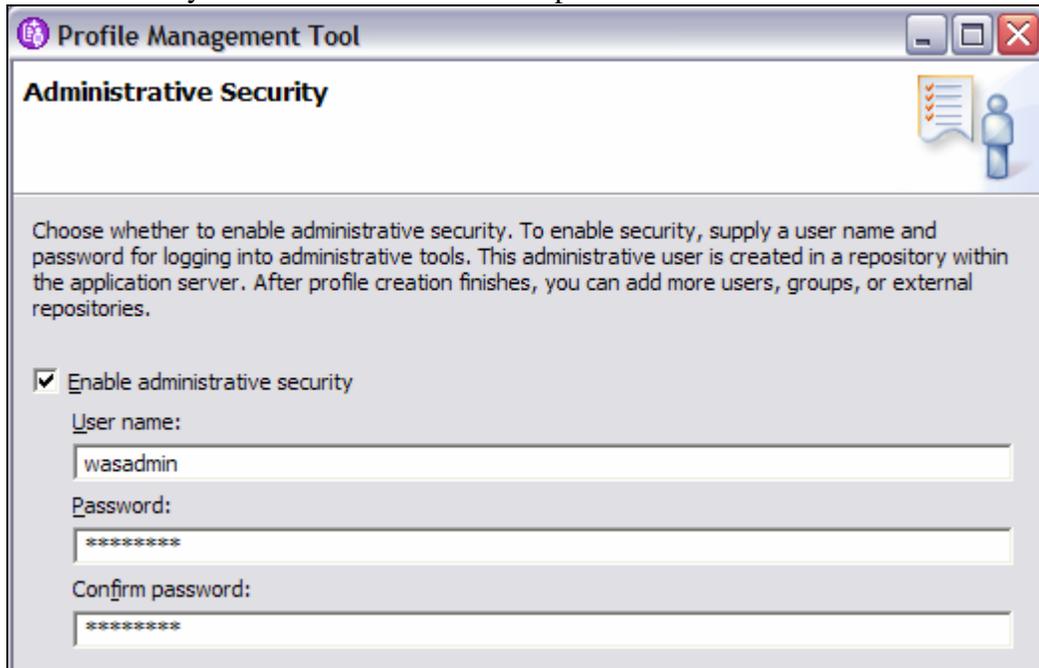


Fig 1a-6: Enabling Administrative security during the profile creation

- Support for federated repositories meaning that you can federate multiple disparate user registries (combination of LDAP, DB and file based user registries) into a WebSphere Cell. The federated repository feature is supported through the integration of the Virtual Member Manager (VMM) into the WebSphere Application Server. VMM provides a single view of multiple user repositories.
 - You can configure federated repository option through the admin console (for LDAP and file based registries) or command line utilities. Navigate to **Security|Secure administration, application.....**, select **Federated Repositories** from the list of Available realm definitions and click on the **configure** button to configure the federated repositories feature as shown in the screenshot below.

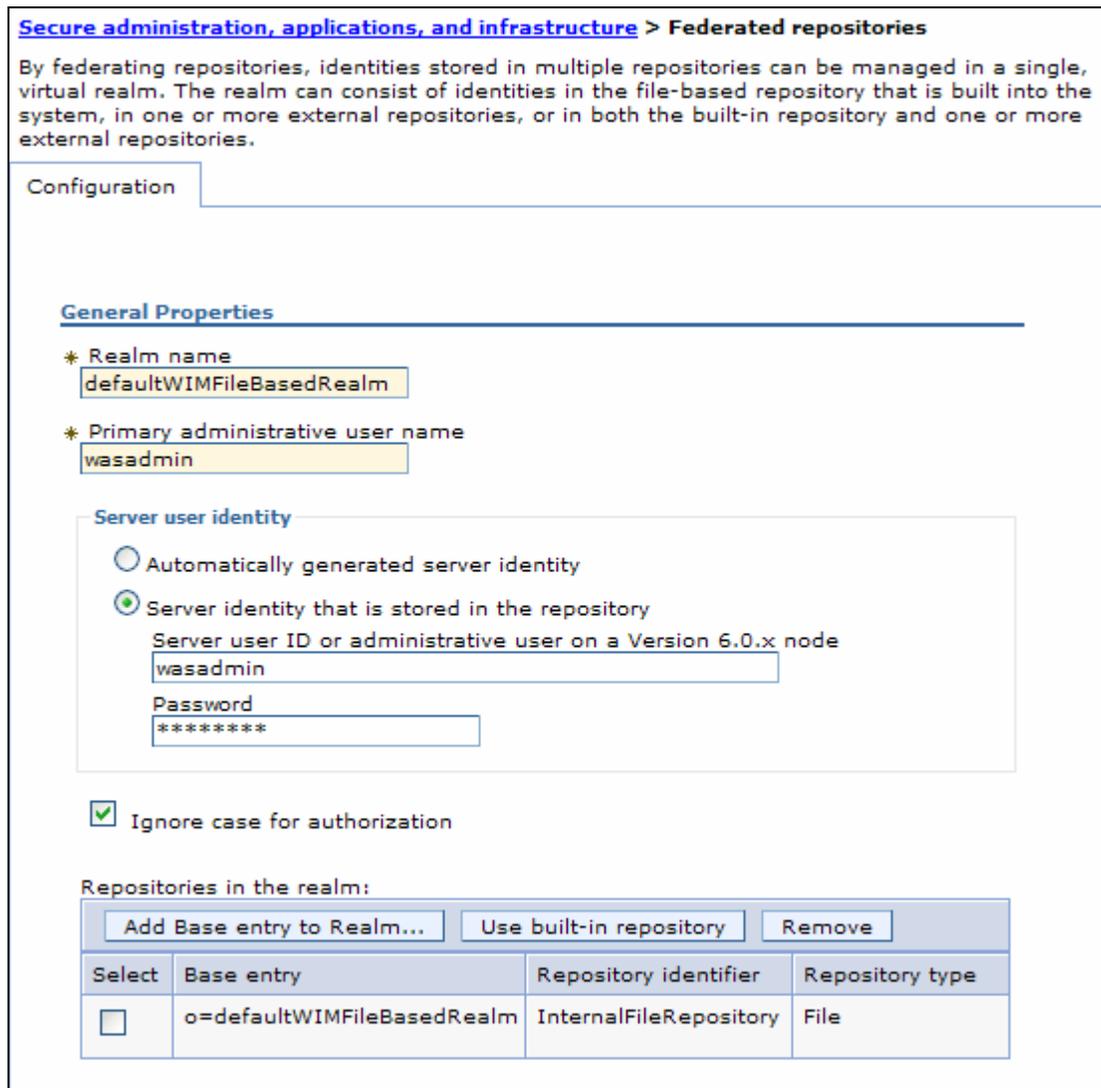


Fig 1a-6a: Configuring federated repositories from the admin console

- Option to deploy admin console application (adminconsole.ear) and the default application (snoop, hitcount and Hello servlets) or not during the application server profile creation.

- You need to deploy administrative console application if you want to administer using the GUI based console. If you want to administer only using the wsadmin tool then admin console application is optional but recommended.
- It is always a good practice to install the default applications to verify the server configuration immediately after the installation and test the configuration after reconfiguring the environment (for example after configuring the application server with Web server or configuring for session persistence).



Fig 1a-7: Selecting to deploy adminconsole and default applications during profile creation

- ISMP based Profile Creation Tool (PCT) has been replaced by eclipse based Profile Management Tool (PMT) if you want to create profiles using the GUI console. You can invoke PMT from <WASV61-ROOT>\bin\pmt.bat (sh) directory as shown in the screenshot below.

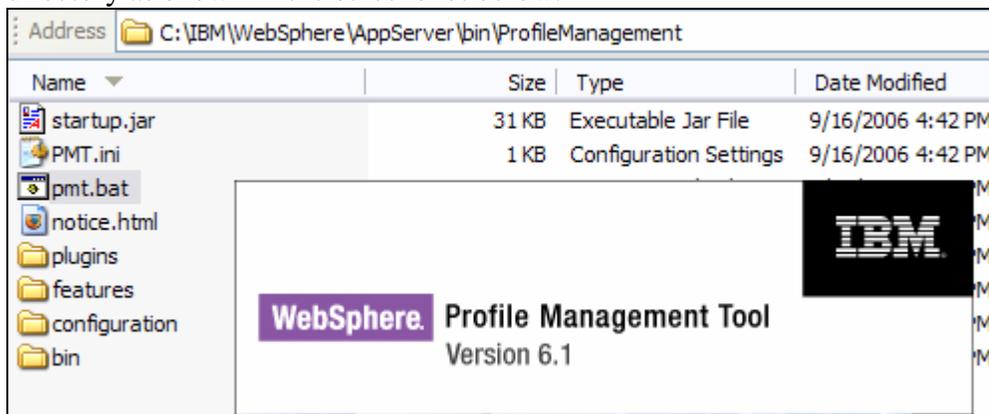


Fig 1a-8: Invoking Profile Management Tool introduced in WAS V6.1

- Command line based wasprofile.bat(sh) tool has been replaced by manageprofiles.bat(sh) tool if you want to create, delete, list profiles and many new functions from a command line. You can invoke the

manageprofiles.bat command line script from <WASV61-ROOT>\bin directory as shown in the screenshot below.



```
C:\IBM\WebSphere\AppServer\bin>manageprofiles.bat -listProfiles
[AppSrv01, Dmgr01, AppSrv02]
C:\IBM\WebSphere\AppServer\bin>
```

Fig 1a-9: Using manageprofiles command line tool introduced in WAS V6.1

- Option to create Web server definition during the profile creation itself by providing the HTTP server and plug-in information as shown in the screenshot below. Refer to printed version of the book to know more about managing the Web server from the administrative console in Chapter 5.
 - In V6 it was only possible by manually configuring from the admin console (in ND package) or running the configurewebserver.bat(sh) after the plug-in installation.

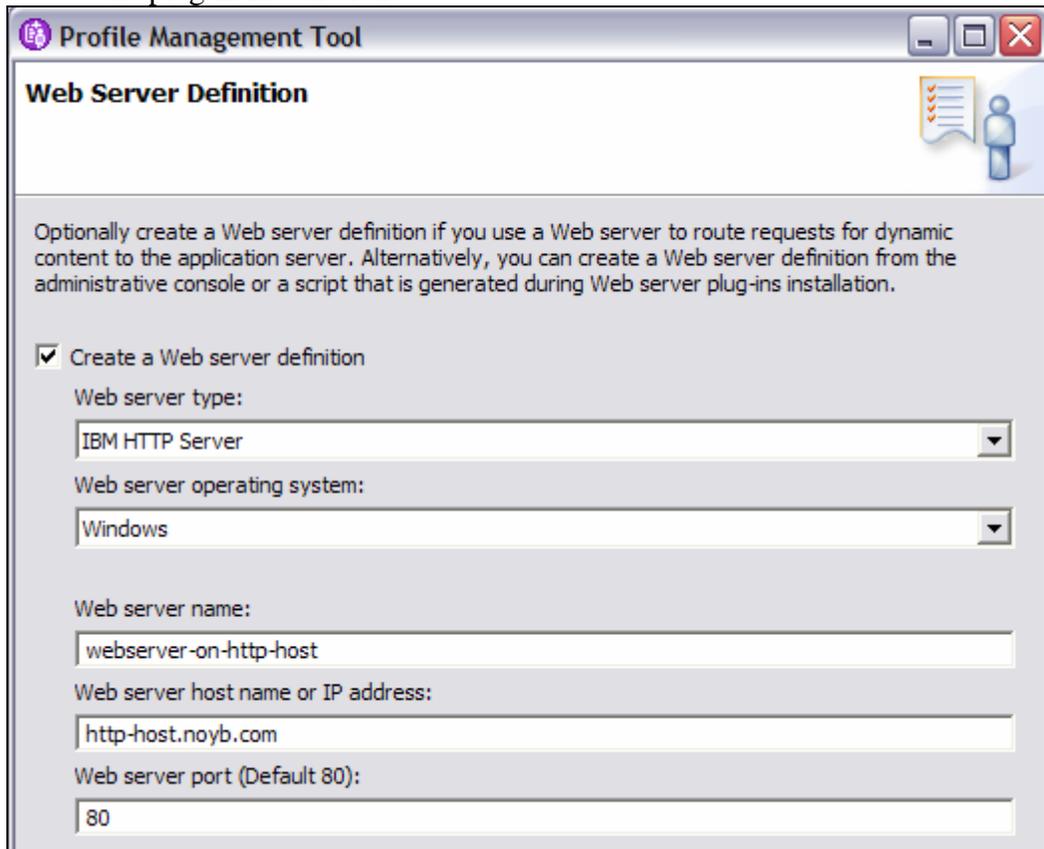


Fig 1a-10: Creation of Web server definition during the profile creation

- Ability to create Web server definition from the admin console, if it was not created during the profile creation even in Base/Express packages. This feature is very handy if you want to quickly configure the Web server without copying and executing the configurewebserver.bat (sh) file from the web server machine after installing the plug-in module.

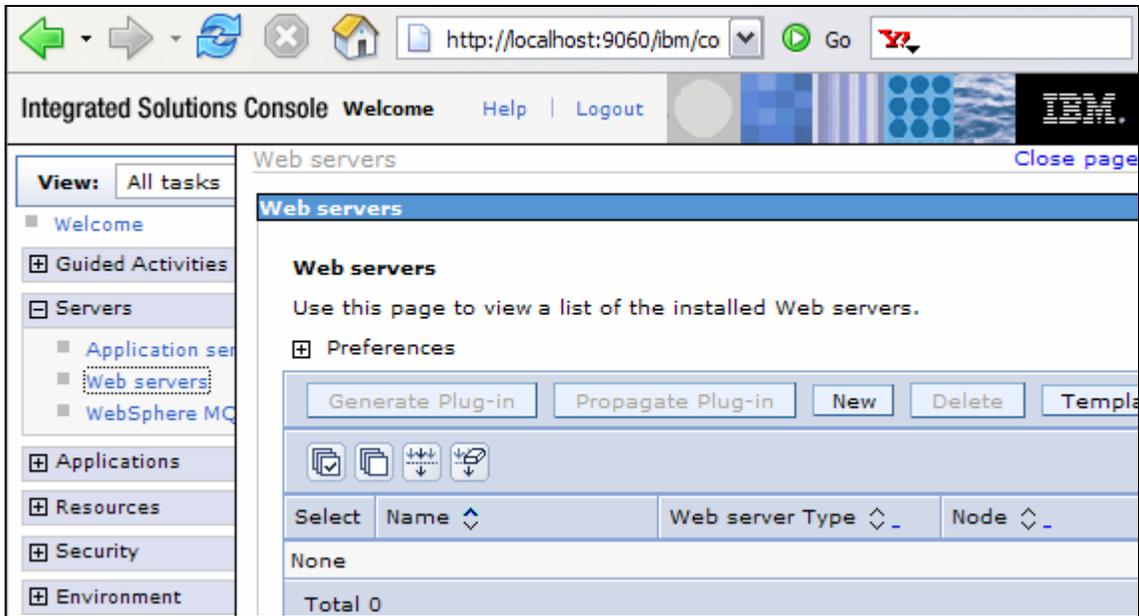


Fig 1a-11: Creation of Web server definition from the admin console

- New Command Assistance function helps administrators to view the Jython based wsadmin script for the last operation performed on the admin console and optionally record that script in a log file. Administrators can cut and paste these scripts to develop wsadmin scripts without writing them by performing them on the admin console for the first time and using the generated script to automate it.
 - The screenshot below depicts how the Command Assistance feature helps you to capture the wsadmin Jython scripts for the last operation you performed on the admin console. The operation we performed as seen in the screenshot can be replicated by: navigating to **Applications|Enterprise Applications** to list all the applications deployed in the cell. The equivalent Jython script is `AdminApp.list()`. You need to click on **View administrative scripting command for last action** that appears on the right side of the screen if you want to view the script.

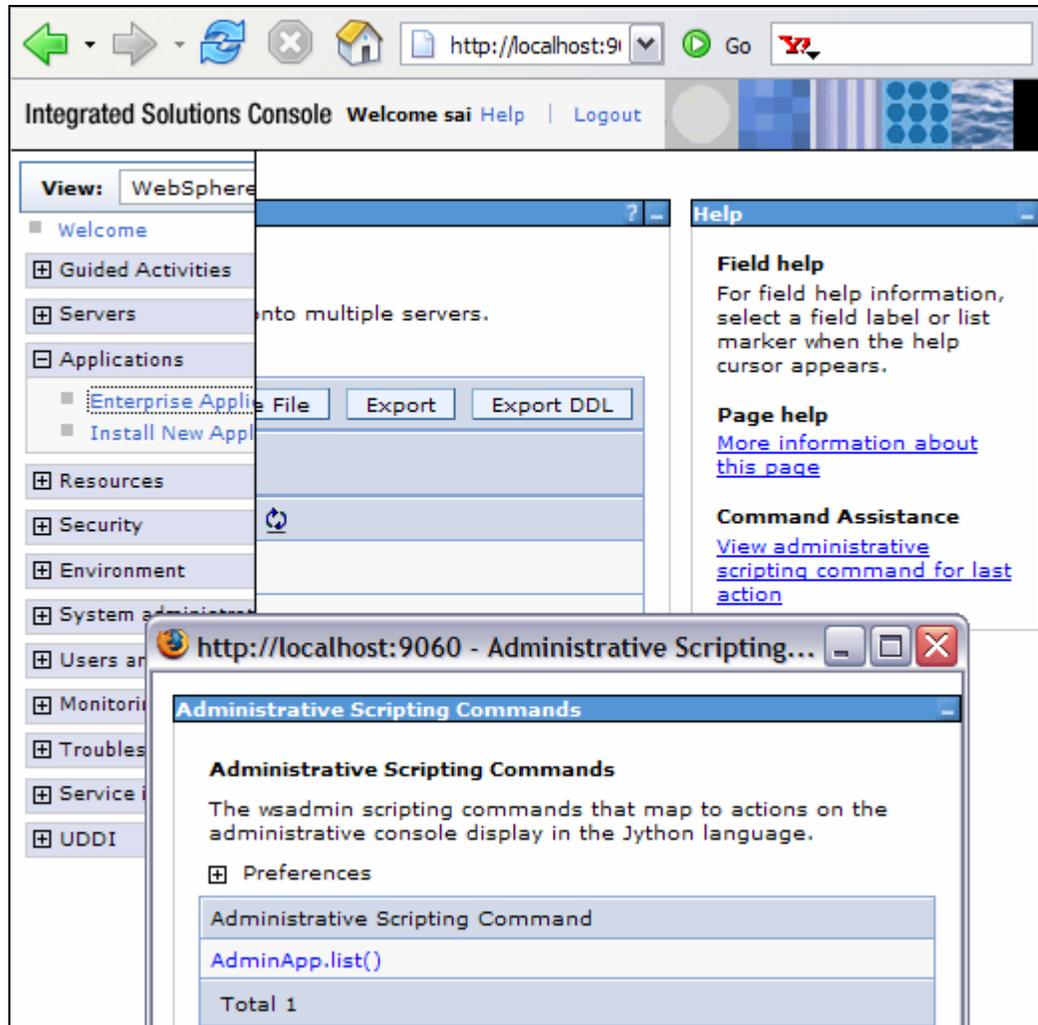


Fig 1a-12: Able to see Jython scripts executed for the last operation from the admin console

- **Customize** Integrated Solution Console (ISC) based administrative console to display only the frequently used tasks in the customized view.
 - To use this feature, choose the **My tasks view** on the top left corner of the admin console and select the check box against the tasks you want to be visible on the screen. Since then use the My tasks view after logging into the admin console to see only the tasks you operate frequently. You can select All tasks or WebSphere Application Server views at any time if you want to see the default options.

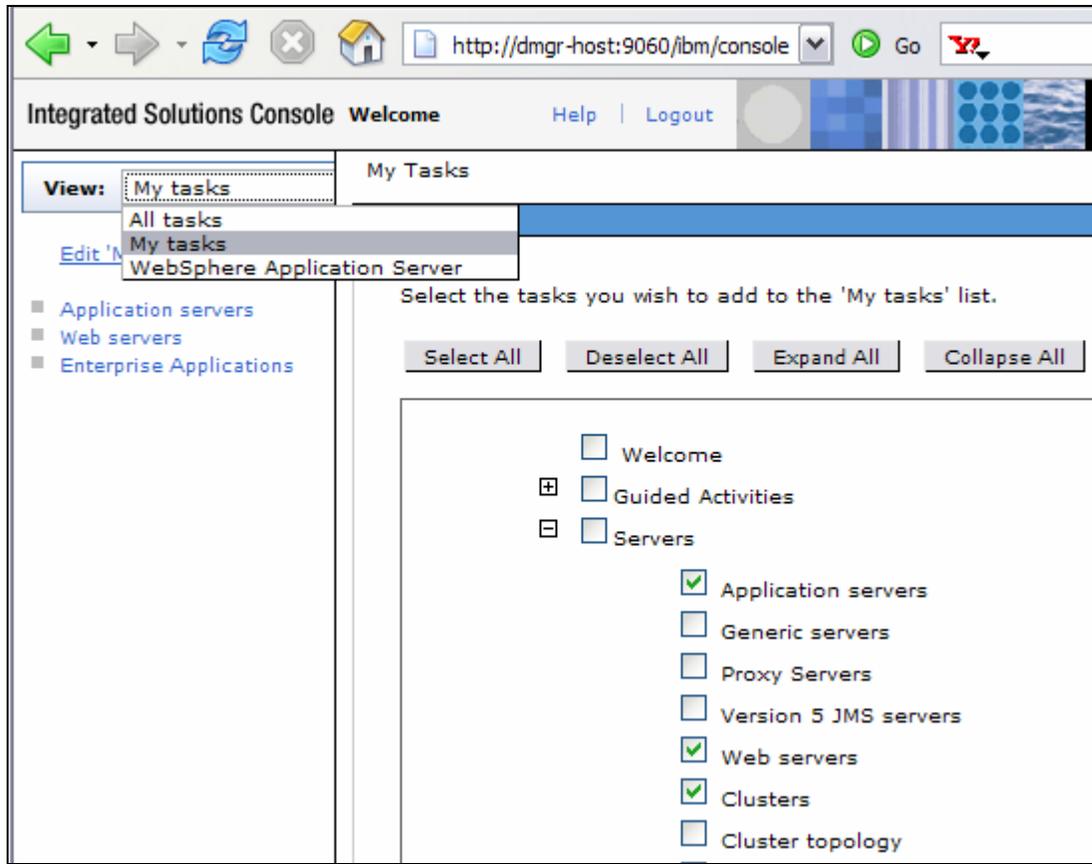


Fig 1a-13: Customizing admin console

- Application installation through the administrative console has been simplified by showing only the wizards required for that application (instead of showing all the pages whether relevant to the application or not) to expedite the application installation process.

Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

Local file system

Full path
tableApps\PerfServletApp.ear

Remote file system

Full path

Context root
 Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

Prompt me only when additional information is required.

Show me all installation options and parameters.

Fig 1a-14: Selecting option to show only required pages during the application deployment

- Generation of certificates and key management has been simplified by providing SSL configuration through the administrative console. No need to use ikeyman utility in WAS V6.1 anymore (even though it is still supported). Refer to updates to the security chapters for many security enhancements that have taken place in V6.1, when those chapter extensions become available.

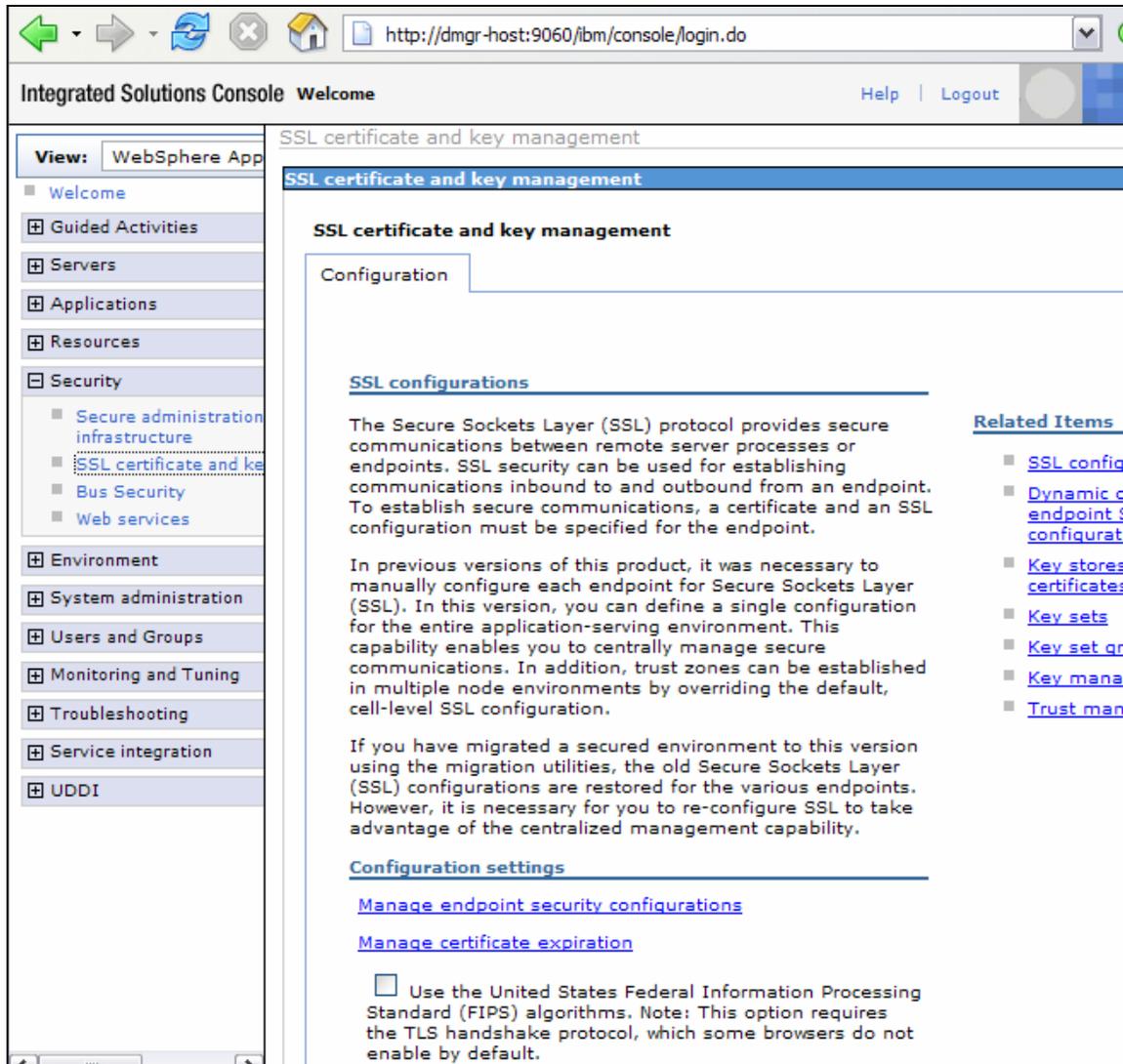


Fig 1a-15: Configuring and managing SSL and keys from the admin console

- Service Integration Bus (SIBus) which is foundation for Service Oriented Architecture (SOA), Enterprise Service Bus (ESB), JMS based messaging and web services integration now supports a file-based datastore to persist the messages (without having to have a database) simplifying the configuration. JDBC based datastores are still supported as in case of V6.
 - File datastore has to be kept in highly available (HA) shared drive if HA is the consideration.

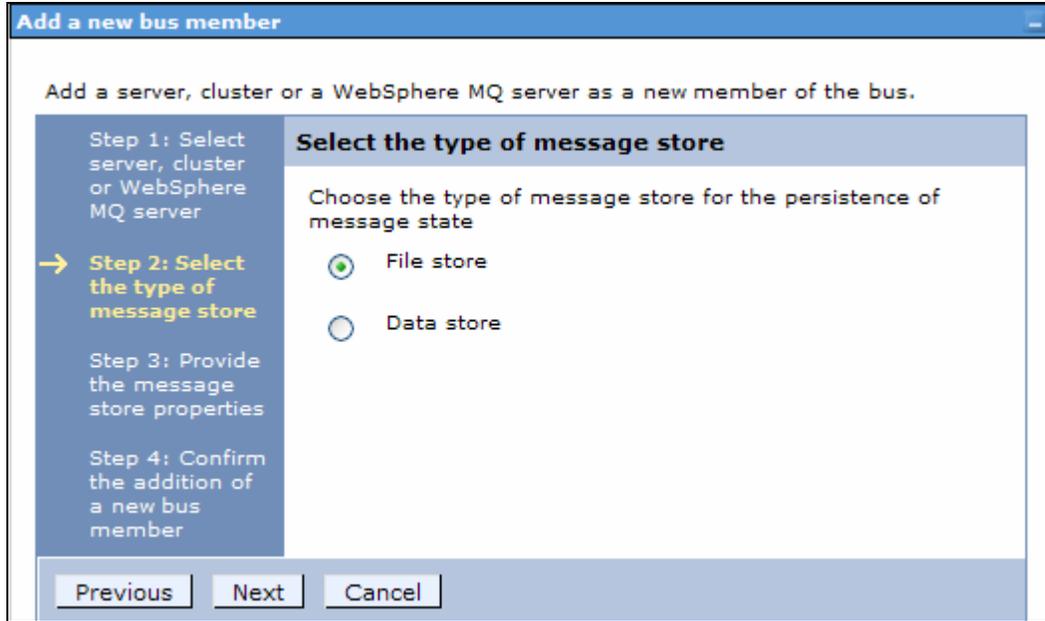


Fig 1a-16: Creation of Web server definition from the admin console

- Enhanced and redesigned Application Server Toolkit (AST) based on eclipse framework which is a subset of Rational Web Developer (RWD).
 - You can use AST not only to create and test wsadmin scripts or to assemble and deploy applications by administrators but also develop J2EE applications by the developers if needed. The sample portlet and SIP applications you are going to use in chapter-3 and 4 extensions later, to test the portlet and SIP containers are developed using the AST tool.
 - AST does not contain a built-in test server (WAS V6.1 runtime unlike Rational Application Developer (RAD) or Web Developer (RWD). You need to connect to the external application server from AST if you want to test the wsadmin scripts or deploy and debug the applications if needed.

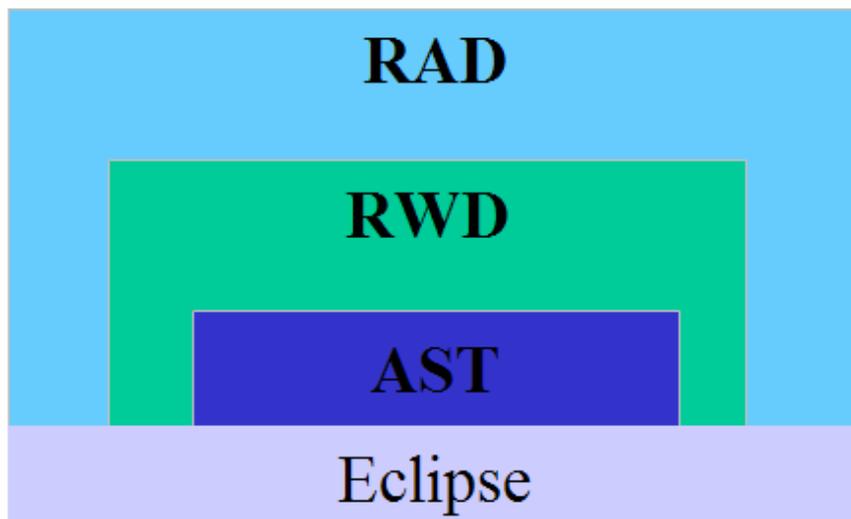


Fig 1a-17: Relationship of AST with other development tools

- AST Jython editor that comes with color syntax highlighting, code completion, keyword help, useful for administrators to build wsadmin scripts to expedite configuration, deployment, automation and testing phases.

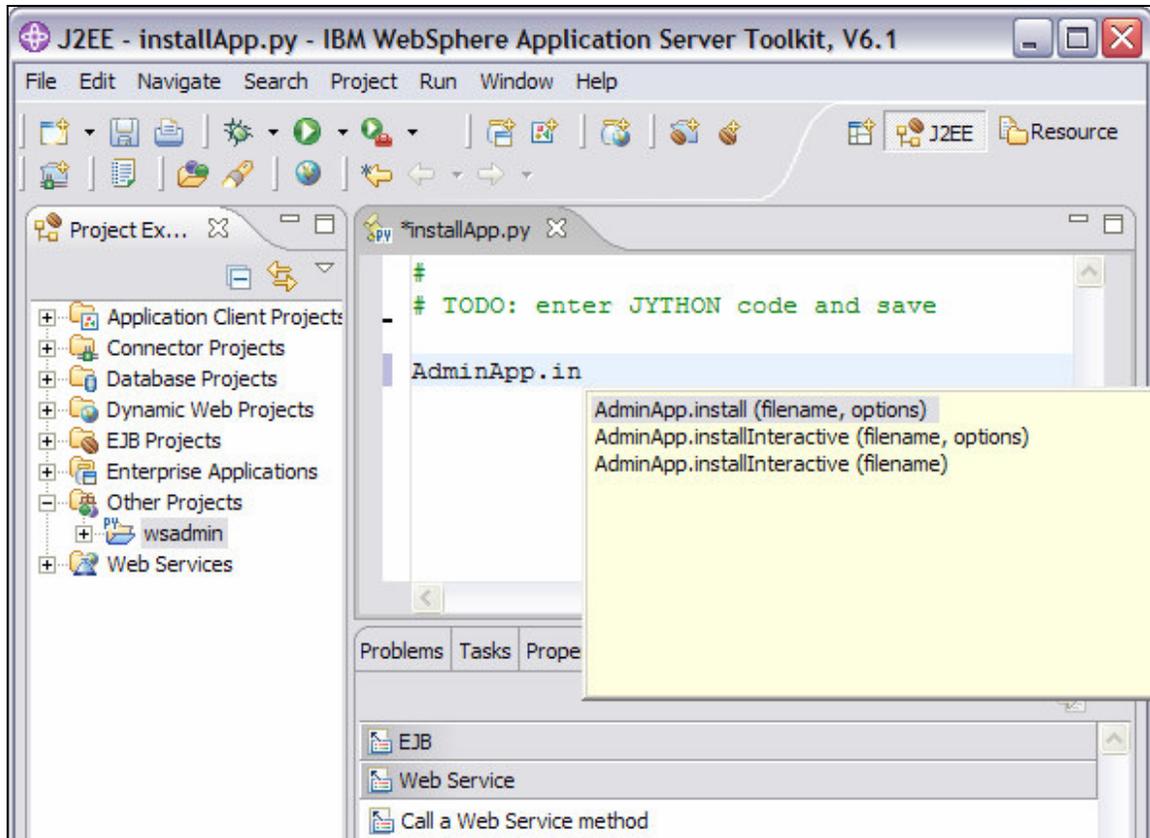
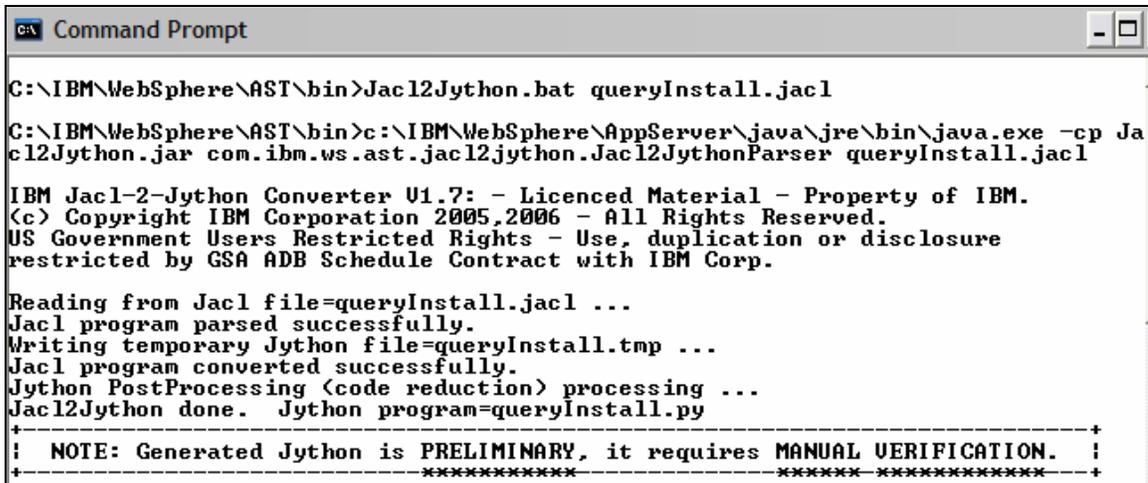


Fig 1a-18: Jython editor in AST to quickly develop, test and automate deployment

- AST comes with **Jacl to Jython Conversion Assistant** command line tool to convert existing JACL scripts to be converted to Jython scripts. Jython is the new strategic direction for wsadmin scripting in WAS V6.1 and forward. This tool (Jacl2Jython.bat (sh)) is available under <AST-ROOT>\bin directory. Refer to the new chapter on AST when it is available online for more information on this tool and its limitations.
 - To convert JACL script (script.jacl for example) to Jython script run **Jacl2Jython.bat (sh) queryInstall.jacl** for example from the command line from <AST-ROOT>\bin directory. The tool will generate Jython based **queryInstall.py** after its process completion.



```
C:\IBM\WebSphere\AST\bin>Jacl2Jython.bat queryInstall.jacl
C:\IBM\WebSphere\AST\bin>c:\IBM\WebSphere\AppServer\java\jre\bin\java.exe -cp Jacl2Jython.jar com.ibm.ws.ast.jacl2jython.Jacl2JythonParser queryInstall.jacl

IBM Jacl-2-Jython Converter U1.7: - Licenced Material - Property of IBM.
(c) Copyright IBM Corporation 2005,2006 - All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADB Schedule Contract with IBM Corp.

Reading from Jacl file=queryInstall.jacl ...
Jacl program parsed successfully.
Writing temporary Jython file=queryInstall.tmp ...
Jacl program converted successfully.
Jython PostProcessing (code reduction) processing ...
Jacl2Jython done.  Jython program=queryInstall.py
+-----+
+ NOTE: Generated Jython is PRELIMINARY, it requires MANUAL VERIFICATION. +
+-----+
+*****+
+-----+
```

Fig 1a-19: Converting JACL to Jython script using the Jacl2Jython tool

- The Jacl2Jython conversion tool creates some temporary files along with the Jython script. You have to verify and manually edit the Jython script if required after its generation. Refer to **Jacl2Jython.doc** file under <AST-ROOT>\bin directory for information about the tool and its limitations.

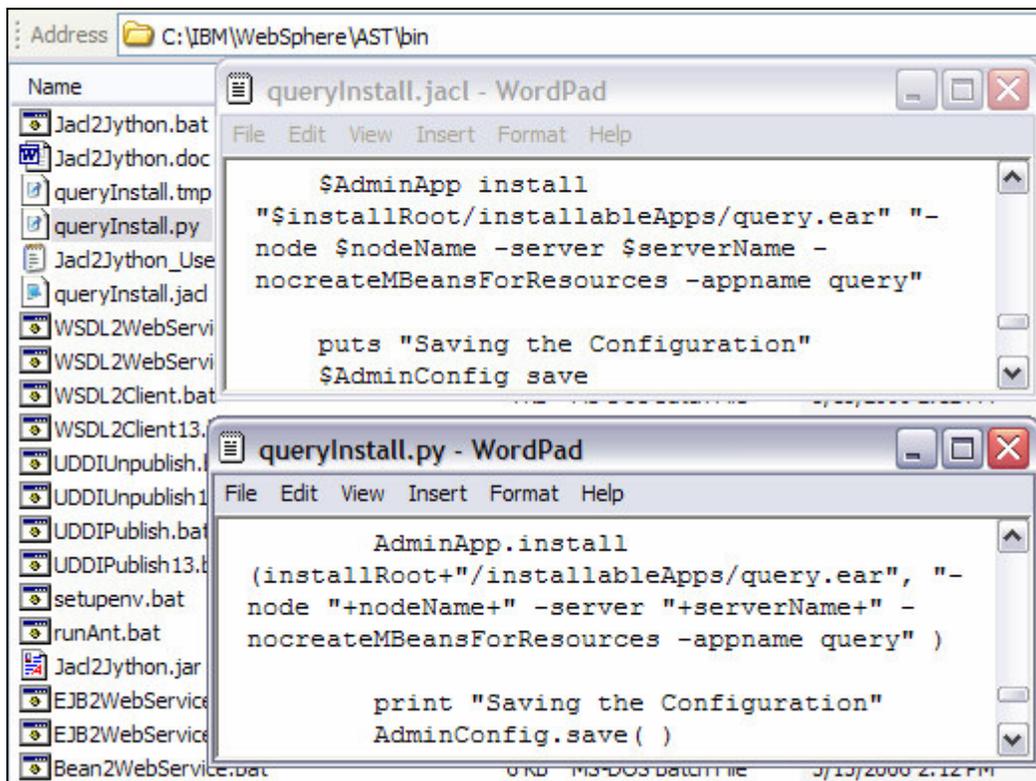


Fig 1a-20: Sample JACL converted as Jython using the tool

- Lightweight runtime package to support administrative functions, instead of installing express or application client packages to do administrative actions from a remote machine.

- You can elect to install just the Administrative Client during the Application Client installation or copying the admin client JAR file with IBM JRE on a remote machine. Refer to the new chapter on AST when it is made available for more information on this topic.

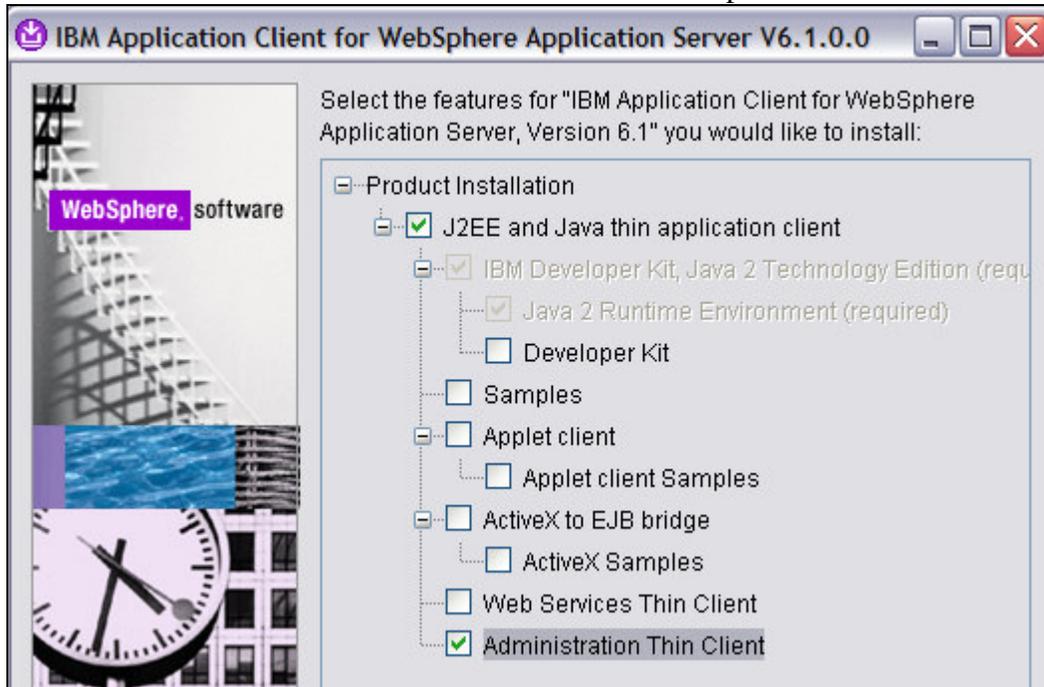


Fig 1a-21: Installing Administrative Thin Client on remote machine

- New J2ME based JVM, designed and developed from the ground up by IBM which is fast with a smaller foot print and contains no Sun intellectual property.
 - Administrator can look forward to more best practices on tuning the JVM in V6.1 that are specific to this new JVM.
- Provides runtime environment and life cycle management for JSR 168 compliant portlets
 - **Advanced features like portlet aggregation, personalization and collaboration are not supported** in WAS V6.1. You need to use **WebSphere Portal server V6** if such features are needed in your application.
 - IBM provides JSP Tag library using which developer can write applications aggregating multiple JSR compliant portlets on a single page.

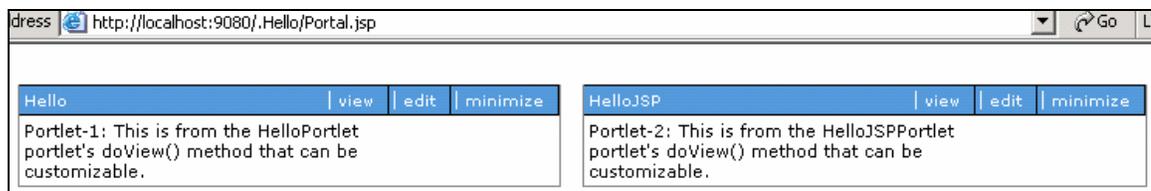


Fig 1a-22: JSR 168 portlets that have been aggregated using the IBM Tag libraries

- Provides runtime environment for JSR 116 compliant Session Initiation Protocol based applications.
 - Session Initiation Protocol (SIP) is a light weight and extensible protocol, used to negotiate call between two end points.
 - Some of the applications that are supported by SIP are instant messaging, presence, Registration and network games.

There are also host of new features and functionality introduced in WAS V6.0.2 (and earlier versions) that have been carried forwarded to V6.1 with more functionality. Some of the notable ones are:

- WebSphere Proxy Server is a new type of server that receives requests from clients initially on behalf of content servers and work load manages and routes the requests across On Demand Configuration enabled (ODC) content servers and and optionally Generic servers (for example HTTP servers) depending on the policies and filter classification definitions. WebSphere Proxy servers can secure the transport (using SSL), content and protect the identity of application servers using the response transformation feature (URL re-writing). ODR (On Demand Router) which is an extension to this Proxy server with request flow management to enforce SLA guarantees over and above what a Proxy server supports. ODR is supported in the WebSphere Application Server V6 - Extended Deployment Package. Refer to Chapter-27 (free download available at www.WebSphereMentor.com) for more information about this server. Proxy Server in WAS V6.1 can now route requests for Session Initiation Protocol (SIP), web services apart from the HTTP requests.

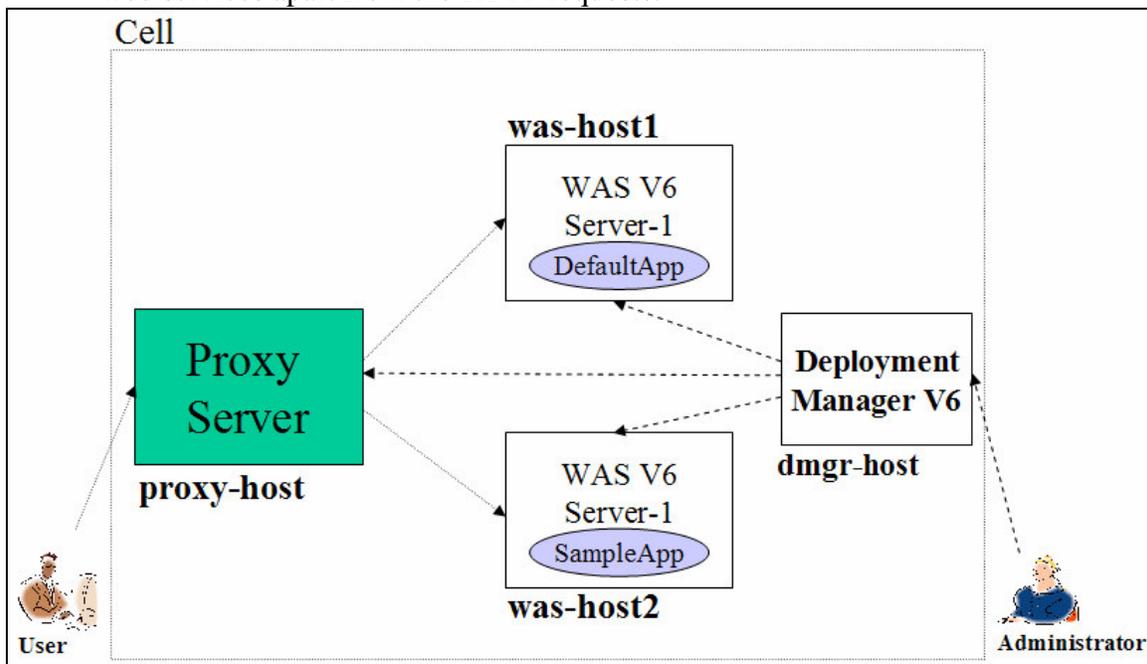
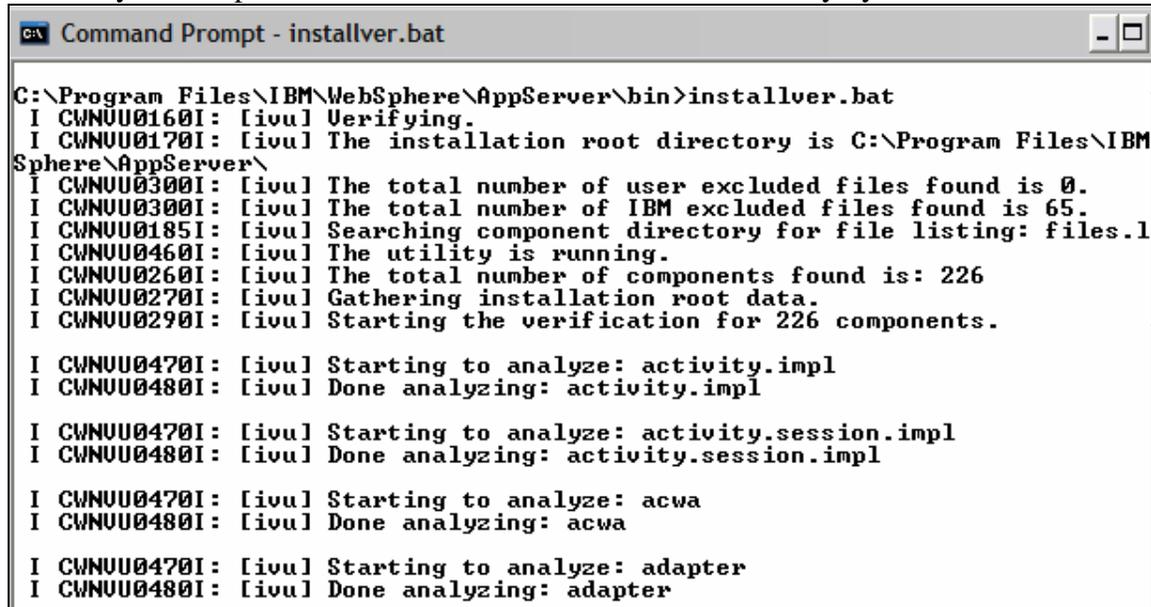


Figure 1a-23: Proxy server routing requests across application servers on federated nodes

- Installation Verification Utility (installver): Helps administrators to check the consistency and corruption in the installed image to make sure that the images are not altered after the installation. You can use this utility immediately after installation of product binaries or a fixpack, refresh packs, interim fixes or anytime later when you are doubtful of the integrity of the product binaries. This utility will be placed under <WASV61-ROOT>\bin directory by default.



```
C:\Program Files\IBM\WebSphere\AppServer\bin>installver.bat
I CWN000160I: [ivul] Verifying.
I CWN000170I: [ivul] The installation root directory is C:\Program Files\IBM
Sphere\AppServer\
I CWN000300I: [ivul] The total number of user excluded files found is 0.
I CWN000300I: [ivul] The total number of IBM excluded files found is 65.
I CWN000185I: [ivul] Searching component directory for file listing: files.l
I CWN000460I: [ivul] The utility is running.
I CWN000260I: [ivul] The total number of components found is: 226
I CWN000270I: [ivul] Gathering installation root data.
I CWN000290I: [ivul] Starting the verification for 226 components.

I CWN000470I: [ivul] Starting to analyze: activity.impl
I CWN000480I: [ivul] Done analyzing: activity.impl

I CWN000470I: [ivul] Starting to analyze: activity.session.impl
I CWN000480I: [ivul] Done analyzing: activity.session.impl

I CWN000470I: [ivul] Starting to analyze: acwa
I CWN000480I: [ivul] Done analyzing: acwa

I CWN000470I: [ivul] Starting to analyze: adapter
I CWN000480I: [ivul] Done analyzing: adapter
```

Fig 1a-24: Verifying product binaries using the installver program

- Installation Factory (IF): IF tool greatly reduces the time and effort during the installation, configuration, deployment and maintenance activities of the administrator by combining and automating various administrative steps into a single step process. Refer to Chapter-29 (free download available at www.WebSphereMentor.com) for more information about this tool.
- **Update Installer** and **IBM Support Assistant** are included and shipped along with the product images CD in WAS V6.1. Only one copy of update installer is installed on any single machines for easy maintenance. All the product updates (applying refresh packs, fixpacks and ifixes) are done through this update installer.

Welcome to WebSphere Application Server Network Deployment

IBM WebSphere Application Server Network Deployment, Version 6.1 is an integrated platform that contains an Application Server, Web development tools, a Web server, and additional supporting software and documentation. This launchpad may serve as a single point of reference for installing your Application Server environment.

We recommend viewing the [installation diagrams](#) for illustrations of common application server environments. For full documentation visit the on-line [WebSphere Information Center](#).

To begin, select an entry from the list below to initialize a product installation wizard. Alternatively, select a product on the navigation list to left to read descriptions of the products, and browse help documentation and support links before starting an installation wizard.

- **WebSphere Application Server Network Deployment**
 [Launch the installation wizard for WebSphere Application Server Network Deployment.](#)
- **IBM HTTP Server**
 [Launch the installation wizard for IBM HTTP Server.](#)
- **Web Server plug-ins**
 [Launch the installation wizard for Web Server plug-ins.](#)
- **Application Clients**
 [Launch the installation wizard for Application Clients.](#)
- **Application Server Toolkit**
 [Launch the installation wizard for the Application Server Toolkit.](#)
- **IBM Update Installer for WebSphere Software**
 [Launch the installation wizard for IBM Update Installer for WebSphere Software.](#)
- **IBM Support Assistant**
 [Launch the installation wizard for IBM Support Assistant.](#)

Fig 1a-25: Inclusion of Update Installer and ISA along with product images

- The IBM Support Assistant (ISA) tool helps administrators to find answers to questions and investigate problems. ISA can accelerate problem resolution by providing quick access to supported information and tools for IBM products. ISA also helps you automatically collect troubleshooting data from WebSphere Application Server (WAS) machines depending on the symptoms you specify. ISA can electronically open PMR and send collected captured data to the IBM WebSphere Support FTP site. Refer to Chapter 30 (free download available at www.WebSphereMentor.com) for more information about this tool.

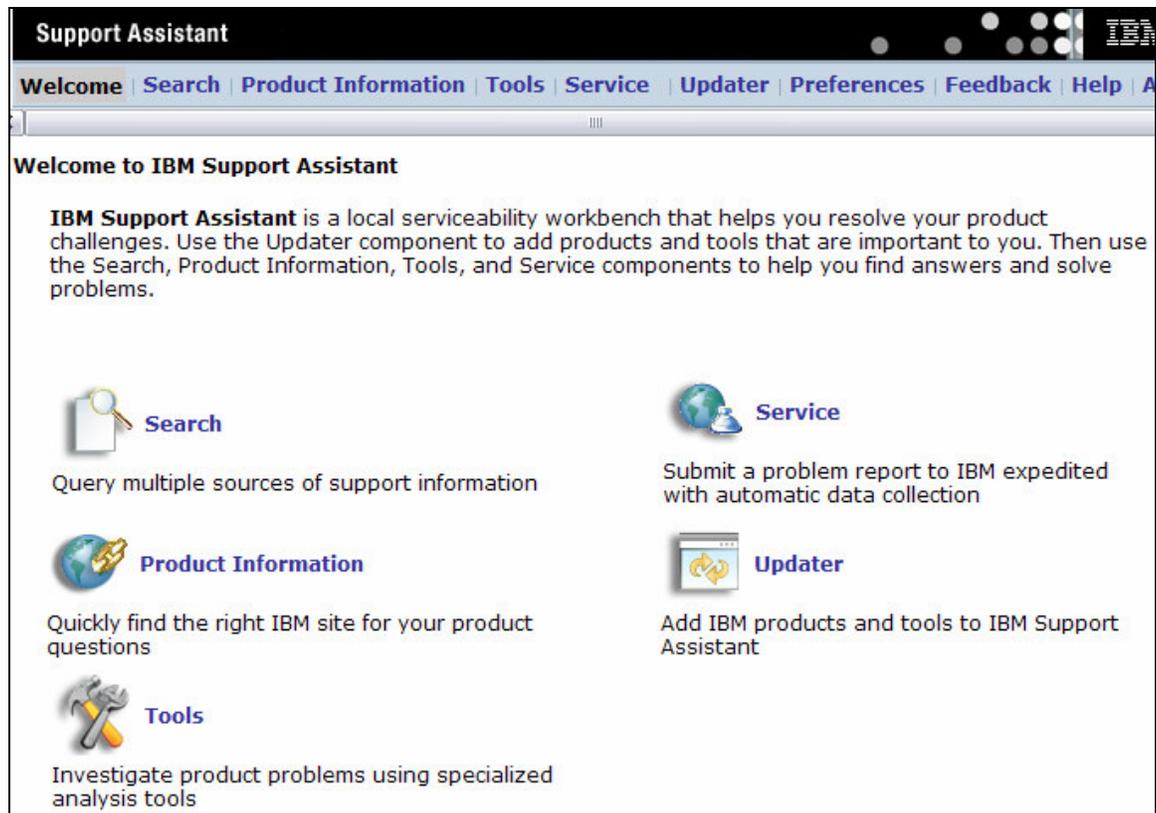


Fig 1a-26: ISA Welcome Screen

- WAS V6.0.2 and higher can automatically perform a heap-dump as soon as it detects that a memory leak condition has occurred providing early detection of the failing server. Memory leaks will occur when the garbage collection (GC) process can not free objects whose references are still held unintentionally in the JVM even though they are no longer needed. This light-weight process is designed to provide early detection of memory problems in test and production environments, with minimal performance overhead. You can use the automatically generated heap dump file as an input to the Memory Dump Diagnostic for Java (MDD4J) Tool (available under IBM Support Assistant Tool (ISA)) to detect the exact cause of the memory leak. Refer to Chapter 30 (free download available at www.WebSphereMentor.com) for more information about the MDD4J tool. This feature is not available if you are using the Sun JVM.
 - Automatic heap dump generation is disabled by default. To enable it navigate to **Servers|Application Servers|serverx|Performance and Diagnostic Advisor Configuration|Runtime**. Select the **check box** for **Enable automatic heap dump collection** and click **OK** and, save the configuration. You need to perform this step on each cluster member if you are in a cluster environment.

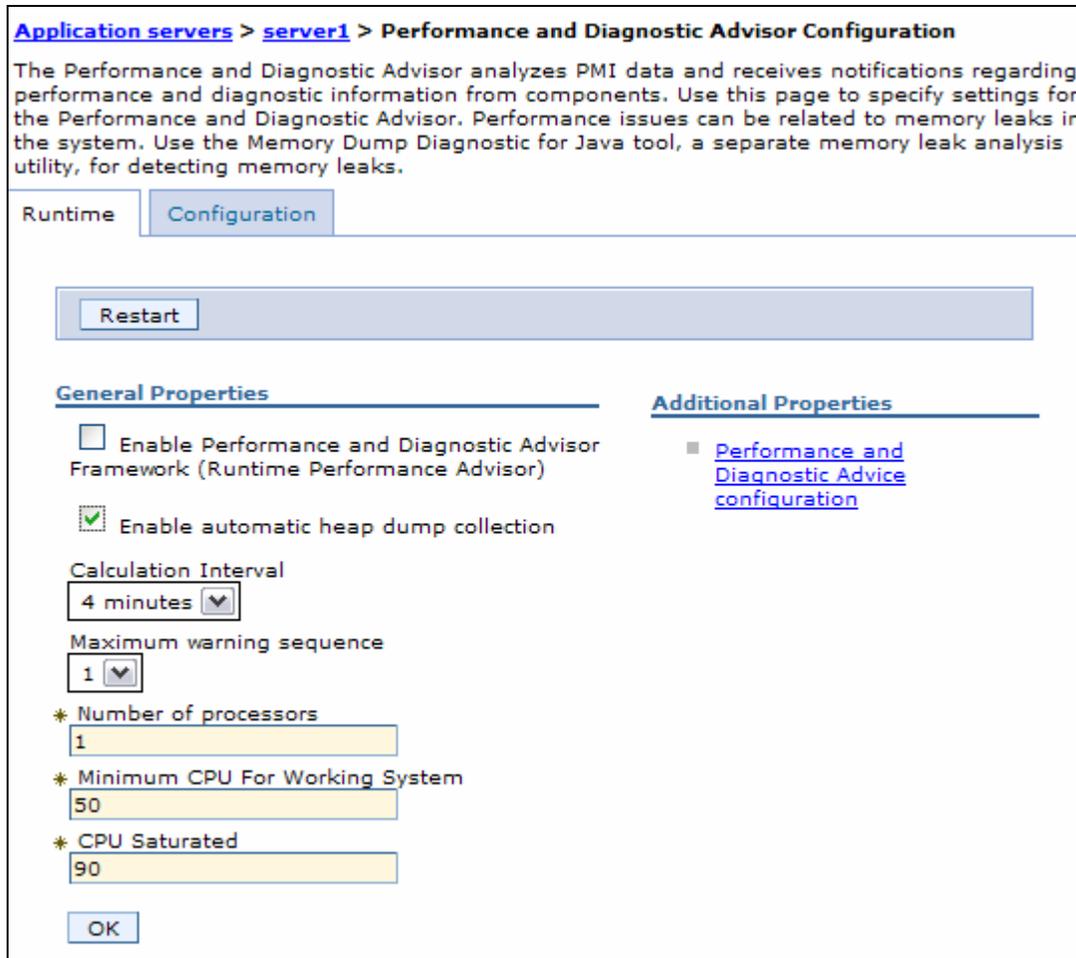


Fig 1a-27: Enabling automatic heap dump collection

- WAS V6.0.2 and higher can automatically detect hung threads and writes the required information (hung thread name, hang time etc.) in the server's SystemOut.log file providing early detection of failing server. A hung thread (for example thread dead lock) in a J2EE application may occur due to a defect in the J2EE application. You can take the thread dump (javacore file) once you find the hung thread situation in the log file and use that thread dump file as an input to the ThreadAnalyzer tool (available under IBM Support Assistant Tool) to detect the exact cause of the hung thread. Refer to Chapter 30 (free download available at www.WebSphereMentor.com) for more information about the ThreadAnalyzer tool.
 - Hung thread detection is enabled by default. You can fine tune the detection policy by customizing the thread monitor interval and threshold values by providing the following custom properties to reduce false alarms. Navigate to **Servers!Application servers!serverx!Administration!Custom properties** to enter the property information as shown in the screenshot below. You need to perform this step on each cluster member if you are in a cluster environment.
 - Hung thread detection can be disabled by setting the **com.ibm.websphere.threadmonitor.interval** property value to **zero** or less than zero.

The screenshot shows the 'Custom Properties' configuration page for 'server1'. It includes a description of custom properties, a 'Preferences' section with 'New' and 'Delete' buttons, and a table of properties. The table has columns for 'Select', 'Name', 'Value', and 'Description'. Two properties are listed: 'com.ibm.websphere.threadmonitor.interval' with a value of 300 and a description 'Interrogate threads every 300 secs', and 'com.ibm.websphere.threadmonitor.threshold' with a value of 900 and a description 'Consider it hung if thread stays more than 900 secs'. A 'Total 2' summary is shown at the bottom of the table.

Select	Name	Value	Description
<input type="checkbox"/>	com.ibm.websphere.threadmonitor.interval	300	Interrogate threads every 300 secs
<input type="checkbox"/>	com.ibm.websphere.threadmonitor.threshold	900	Consider it hung if thread stays more than 900 secs
Total 2			

Fig 1a-28: Customized hung thread properties

<begin note>

If you have worked on earlier versions of WAS then you can start using the trial version of the product. Use the chapter 3 and 4 of the printed version of the book along with this document to jump start on WAS V6.1. Download WAS V6.1 Base package trial version from http://www-128.ibm.com/developerworks/downloads/ws/was/?S_TACT=105AGX10.

<end note>

-30-

-30-