



**Carla Sadtler
Susan Hanson**

WebSphere Application Server: New Features in V8.5.5

IBM® WebSphere® Application Server helps drive business agility with an innovative, performance-based foundation to build, reuse, run, integrate, and manage service-oriented architecture (SOA) applications and services. From business critical enterprise-wide applications to the smallest departmental level applications, WebSphere Application Server offers reliability, availability, security, and scalability.

WebSphere Application Server V8.5 addresses the needs of today's agile enterprises and developers. It provides increased scalability, resiliency, and security for critical applications, and the flexibility to deploy new offerings quickly and efficiently. It includes a lightweight and powerful, yet simple, application server to satisfy multiple requirements around a simplified "low-end" application environment. For the developer, it provides an improved developer experience and a simplified server configuration that can have multiple versions and be maintained in source control along with the applications.

This IBM Redpaper™ publication presents a high-level view of some of the features and enhancements in WebSphere Application Server V8.5. and WebSphere Application Server V8.5.5.

WebSphere Application Server overview

Application infrastructure trends show a push towards rapid application development and delivery. This trend is driving simplified, integrated, and automated development and operation lifecycles. The explosion of mobile, social, and cloud applications is driving unprecedented demands on middleware infrastructures. The combination of huge transaction volumes against massive amounts of data with little tolerance for delays is also driving the need for elastic caching technologies. An increase in the use of cloud delivery models provides elasticity, scale, multi-tenancy, and context for different form factors and access methods. WebSphere Application Server provides a standards-based, high performance application foundation that addresses these trends.

The latest version of WebSphere Application Server continues to address the ever-changing trends driving the industry. This new release is a major step forward in several areas, specifically, resiliency and developer productivity.

WebSphere Application Server builds on over a decade of leadership for Java application servers by expanding its programming model portfolio to allow more flexibility in application development and by improving administrator productivity, increasing security robustness, and delivering new functions for added resiliency.

WebSphere Application Server provides the following enhancements:

- ▶ Rapid delivery of new applications and services
- ▶ Improved operational efficiency and reliability
- ▶ Increased security and control
- ▶ Packaging enhancements

Rapid delivery of new applications and services

WebSphere Application Server can help businesses offer richer user experiences through the rapid delivery of innovative applications. Developers can jumpstart development efforts and use existing skills by selecting from a comprehensive set of open standards-based programming models. Developers can use this function to better align project needs with programming model capabilities and developer skills. WebSphere Application Server also speeds application delivery by encouraging reuse and extending the life of existing application assets.

The following features of WebSphere Application Server can help with rapid delivery of new applications:

- ▶ The Liberty profile
- ▶ Expanded tools and tool bundles
- ▶ OSGi programming model enhancements
- ▶ Support for Java 7 features
- ▶ Migration toolkit enhancements
- ▶ Web 2.0 and Mobile Toolkit
- ▶ Service Component Architecture OASIS programming model implementation
- ▶ **(New in V8.5.5)** Multi-threaded programming model support

The Liberty profile

WebSphere Application Server V8.5 includes a Liberty profile, which is a highly composable and dynamic application server profile. The Liberty profile is included with each WebSphere Application Server package except the Community Edition, and a new Liberty Core package is available with V8.5.5.

The Liberty profile is designed for both development and production. For a developer, the Liberty profile focuses on those tasks that a developer does most frequently and makes it possible for the developer to complete those tasks as quickly and as simply as possible. For production environments that have the characteristics that are supported, it provides a dynamic, small footprint run time to maximize system resources. Fidelity to a WebSphere Application Server full profile environment, with the same reliable containers and quality of service (QoS), provides an easy deployment from development to your quality assurance (QA) and production environments.

Programming model

The Liberty profile provides support for the following features defined by the Java EE 6 Web Profile:

- ▶ Servlet 3.0
- ▶ JavaServer Pages (JSP) 2.2
- ▶ Expression Language (EL) 2.2
- ▶ Debugging Support for Other Languages (JSR-45) 1.0
- ▶ Standard Tag Library for JavaServer Pages (JSTL) 1.2
- ▶ JavaServer Faces (JSF) 2.0
- ▶ Common Annotations for Java Platform (JSR-250) 1.1
- ▶ Java Transaction API (JTA) 1.1
- ▶ Java Persistence API (JPA) 2.0
- ▶ Bean Validation 1.0
- ▶ **(New in V8.5.5)** Enterprise JavaBeans (EJB) 3.1 Lite and Interceptors 1.1 (a subset of the full EJB 3.1 specification, focused on session beans and local interfaces)
- ▶ **(New in V8.5.5)** Managed beans 1.0
- ▶ **(New in V8.5.5)** Contexts and Dependency Injection (JSR-299 1.0 and JSR-330 1.0)

(New in V8.5.5) The Liberty profile supports Java messaging service (JMS) and message-driven beans. A new lightweight single server message provider is included with Liberty for development and testing of messaging applications. A WebSphere MQ client is included for access to WebSphere MQ for production messaging. The JMS support in Liberty is also fully interoperable with the service integration bus in the full profile. The JMS client in Liberty can access the service integration bus and the full profile JMS client can access the messaging provider running in Liberty.

(New in V8.5.5) Liberty provides both client and server functionality for SOAP-based web services. The supported APIs include:

- ▶ JAX-WS 2.2
- ▶ JAXB 2.2
- ▶ Web Services for EE 1.3
- ▶ POJO and EJB-based web services

The following JDK bundled APIs are also available:

- ▶ SAAJ
- ▶ JAXP
- ▶ StAX

The SOAP/HTTP and WS-Security protocols are supported.

Web services can be secured without WS-Security through the use of basic authentication along with web application transport constraints. With WS-Security, a default keystore and truststore contains the keys for authentication and encryption. The following WS-Security policies are available:

- ▶ Username Token Profile 1.1
- ▶ X.509 Token Profile 1.1
- ▶ WS-I Basic Security Profile 1.1

Additional support for developing web services has been added to the WebSphere Application Server Developer Tools for Eclipse, including wizards for top-down WSDL-to-Java generation, for bottom-up POJO web service development, and to simplify adding policies to WSDL.

(New in V8.5.5) Support for MongoDB has been added to provide access to a scalable, document-oriented NoSQL database. Applications can get a reference to the database using injection, JNDI lookup, or J2SE style. After the reference is obtained, access to the database is through the MongoDB Java API. MongoDB support is not available in Liberty Core.

Runtime environment

The Liberty profile provides a lightweight server with a small memory footprint. Using WebSphere Application Server Developer Tools for Eclipse, you can create a server configuration quickly with just a couple of mouse clicks. The simple and flexible configuration is stored in the `server.xml` file and contains the configuration for the runtime instance. You can edit the `server.xml` file directly by using an XML editor or an Eclipse-based editor. The configuration lists the features (capabilities or bundles) that are installed in the server. By defining just the features that you need, the Liberty profile provides the smallest runtime footprint for applications. This dynamic run time allows the adding of features and updating of configuration parameters without requiring you to restart the server.

For team development, the Liberty profile provides support for shareable configuration snippets that you can use to keep the application and configuration components together and to share a single copy for the development team. This shareable configuration reduces the impact of making a single change and replicating that change to the entire development team. For example, if you have 50 developers and need to change a data source definition, a single change to the shared snippet file is picked up automatically by all the developers. This situation is preferable to all 50 developers making the change manually to their own configuration.

You can also package a compressed archive of a configured Liberty server type along with its applications and configuration and then distribute that compressed archive to developers on the team or directly into a QA or production environment.

To maintain HTTP session failover and high availability, the Liberty profile can persist session data to a database or can interoperate with IBM WebSphere eXtreme Scale and the IBM WebSphere DataPower® Appliance XC10 V2 caching appliance.

(New in V8.5.5) Liberty supports WebSphere Web Cache (sometimes referred to as DynaCache), providing a local caching service for dynamic web content. For scalability or to cache other data, Liberty can interoperate with WebSphere eXtreme Scale or the WebSphere DataPower XC10 V2 Appliance.

Serviceability and troubleshooting

The Liberty profile has a unified logging component that provides text log entries. Liberty also provides base implementations of trace and First Failure Data Capture (FFDC) services for

runtime and application code to gather debug information. Messages are written to stdout as well as to the defined trace destination. OSGi logging output is intercepted and output through the trace support. There is also interception of java.util.logging output.

(New in V8.5.5) The binary logging capabilities from the full profile are also available in the Liberty profile. Binary logging is an alternative to the default log and trace facility. Binary logging is a high performance log and trace facility based on the full profile High Performance Extensible Logging (HPEL) technology. Logging in binary format is faster than text logging.

(New in V8.5.5) Timed operations generate a logged warning when JDBC calls in the application server are operating more slowly or quickly than expected. Periodically, the timed operation feature will create a report in the application server log detailing which operations took longest to execute. If you run the server dump command, the timed operation feature will generate a report containing information about all operations it has tracked.

Monitoring

The Liberty profile provides monitoring support for the following runtime components:

- ▶ JVM
- ▶ Web applications
- ▶ Thread pools
- ▶ **(New in V8.5.5)** Database connection pools
- ▶ **(New in V8.5.5)** Messaging
- ▶ **(New in V8.5.5)** Web services

(New in V8.5.5) Liberty extensions

The Liberty extensions System Programming Interface (SPI) provides the ability to extend the Liberty profile with custom features by using OSGi bundles, including web application bundles. For example, you could extend the OSGi applications programming model by adding new annotations or custom configurations. Or, you could provide a custom user registry. This support can be used to integrate third-party function into the Liberty runtime and development tools.

The product extensions consist of a set of directories and files that are made known to Liberty by a properties file. Independent products can enhance a Liberty installation by registering new features using the properties file. Users that create features can install them in a built-in extension for convenience. This built-in extension can also be used by products that embed a Liberty install and service.

The SPI supports the full lifecycle of packaging, installation, and un-installation. The SPI includes plug points that you can implement as features, services for run time, and instrumentation for monitoring and problem determination.

Deploying applications

Using the Liberty profile provides multiple deployment options. You can add the application definition to the server.xml file directly or drop the application into the “monitored” directory. The server picks up these changes automatically and deploys the newly added application into the runtime environment. To uninstall an application, remove the application from the server.xml file and save or remove the application from the monitored directory.

Administration

You can manage the lifecycle of Liberty server instances through an integrated development environment (IDE) or through command-line programs that you can use to create, start, and stop the server or get the status of the server quickly and easily. New features have been added to the administration options for Liberty in V8.5.5 for operational efficiency and convenience, and to introduce high availability features.

(New in V8.5.5) Liberty servers can now be seen as a part of a common management domain, called a Liberty collective. Liberty servers join the collective by registering with an operational registry and adding the collectiveMember feature to their configuration. The collective controller provides full lifecycle management to all members in the collective, including product installation and maintenance, and operational access to all servers in the collective, without requiring an agent. Operational management includes commands to start and stop servers, invoke JMX MBeans, and perform file transfer in support of configuration changes and application installation. All WebSphere Application Server editions can be members of a collective, but only Network Deployment or WebSphere Application Server for z/OS® can be the collective controller.

(New in V8.5.5) Liberty servers that are members of a collective can be configured into a server cluster for high availability and scalability. The cluster can be treated as a single object in the collective, simplifying the operational management of the servers in the cluster. The members of the cluster can be configured individually, or can share a configuration. A web server plug-in is used to distribute work across the servers in the cluster.

In Network Deployment environments that use the Job manager, you can also use the Job manager to centralize management of the server lifecycle for Liberty profile servers. The job manager acts as a single point of management for agentless installation by using an operating system user ID and for starting and stopping the server instance.

Security

The Liberty profile is secure without modification. All opened ports are local host only and expose no remote management by default. The Liberty profile supports the following key security capabilities:

- ▶ A user registry holds information used for the authentication of users. The Liberty profile supports the following user registries:
 - The basic XML-based registry, which supports users, groups, and roles for role mapping. **(New in V8.5.5)** Passwords in the basic registry can be stored as a one-way hash for increased security, preventing the original password from being revealed, even if the server configuration becomes readable to unintended parties.
 - An LDAP user registry.
 - SAF registry (IBM z/OS).
 - **(New in V8.5.5)** Federated LDAP registries, where two or more LDAP registries are defined so that the operations, such as a search for a user, are executed on all the registries.
 - **(New in V8.5.5)** Custom user registries installed as an extension to Liberty.
- ▶ Role-based authorization verifies whether a user or group belongs to a specified role, and whether this role has the privilege to access a resource.
- ▶ **(New in V8.5.5)** The Liberty profile provides a utility to support Advanced Encryption Standard (AES) encryption for passwords that are stored in the server.xml file.
- ▶ Custom Java Authentication and Authorization Service (JAAS) login modules can be configured to make additional authentication decisions or add information to the subject to make finer-grained authorization decisions inside your application. A JAAS custom login module uses a hashtable, callbacks, or shared state variables provided by the Liberty profile server to pass authentication data to the system login module.
- ▶ Single sign-on is supported, allowing web users to authenticate once when accessing Liberty resources that share the same Lightweight Third Party Authentication (LTPA) keys.
- ▶ Communications are secured with SSL connections using HTTPS.

- ▶ **(New in V8.5.5)** Security support is available for web applications using Servlet 3.0 and for EJBs when the `ejbLite-3.1` feature is present.
- ▶ Secure remote access by JMX client through a REST-based connector is supported.
- ▶ **(New in V8.5.0.2)** Authorization to resources by using the OAuth 2.0 protocol is supported. OAuth is an open standard for delegated authorization. With the OAuth authorization framework, a user can grant a third-party application access to their information stored with another HTTP service without sharing their access permissions or the full extent of their data.
- ▶ **(New in V8.5.5)** Integration with a third-party security service using Trust Association Interceptors (TAI). A TAI is used to validate HTTP requests between a third-party security server and a Liberty profile server. The TAI can be called before or after single sign-on (SSO).
- ▶ **(New in V8.5.5)** Web Services security is supported at the transport layer and at the message level. Transport-level security is based on a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) and is used to protect HTTP message contents point to point. Message level security is based on WS-Security.
- ▶ **(New in V8.5.0.2)** The `sync-to-OS-thread` feature for z/OS allows the synchronization of a Java thread identity (or JAAS subject) with the OS thread identity for the duration of the current Java EE application request. If you do not choose this option, the OS thread identity value is the same as the servant identity value.

(New in V8.5.5) There are several security configuration examples on the `wasdev.net` website for reference when configuring security for your applications on the Liberty profile:

<https://www.ibm.com/developerworks/mydeveloperworks/blogs/wasdev/entry/snippets?lang=en>

Installation

Installation is done using IBM Installation Manager or a simplified installation can be done by extracting an archive file.

(New in V8.5.5) For those who use the archive installation on distributed platforms, the following separate archives are available to allow you to install archives as you need them:

- ▶ Runtime archive: The same Liberty profile available in V8.5.0 plus the features added to complete the Java EE 6 Web Profile support.
- ▶ Extended archive: Adds web services support, JMS and message-driven bean support, and MongoDB support. (Not available with Liberty Core.)
- ▶ Extras archive: Adds embeddable EJB container and JPA client.

(New in V8.5.5) For those who install the Liberty profile with the IBM Installation Manager, the following enhancements have been made:

- ▶ In V8.5 the Liberty profile is in the same Installation Manager package as the full profile. In V8.5.5 the Liberty profile is a separate Installation Manager package.
- ▶ Making the Liberty Profile available as a separate installation package creates a smaller repository for installation.
- ▶ The embeddable EJB container and JPA client are available for installation as a feature of the Liberty package.
- ▶ The extended programming models (web services support, JMS and message-driven bean support, and MongoDB support) are available for installation as a feature of the Liberty package for all Liberty installations except Liberty Core.
- ▶ The IBM WebSphere SDK Technology Edition V6 installation package is optionally available for installation with the Liberty packages with the exception of Liberty Core.

- ▶ The IBM WebSphere SDK Technology Edition V7 installation package is optionally available for installation with Liberty Core.

For z/OS, archive installation can be achieved by first installing Liberty using the IBM Installation Manager, then using the server packaging tool to produce a pax file.

Expanded tools and tool bundles

IBM Rational® Application Developer continues to be a popular and feature-rich development solution for teams that develop software for WebSphere Application Server.

Rational Application Developer Standard Edition provides developers with the ability to have a single environment to help simplify and accelerate the core tasks of designing and writing code, testing WebSphere applications, and then maintaining those applications. Rational Application Developer stays synchronized with WebSphere Application Server in terms of strategic focus on technologies and standards, including Java Platform, Enterprise Edition (Java EE) levels, SOA, Web and Web 2.0, Portal, and the OSGi programming model.

As an alternative to Rational Application Developer for developers who do not need the full range of features it offers, WebSphere Application Server Developer Tools for Eclipse V8.5.5 is available for developing applications targeted to WebSphere Application Server full profile and Liberty profile as a run time. WebSphere Application Server Developer Tools for Eclipse contains the tools required for the development of Java EE applications (beyond what's in Eclipse), OSGi applications, and applications using the web and mobile programming models. It includes support for deploying applications to existing WebSphere Application Server V7, V8, V8.5 installations and the V8.5 Liberty profile.

WebSphere Application Server Developer Tools for Eclipse is available at no charge for developer desktops. When you install from the Marketplace you typically pick WebSphere Application Server Developer Tools for Eclipse for the WebSphere Application Server version that you are using, which filters out support for the other versions. After you have installed the tools, you can see and install additional features, for example, support for a different version of WebSphere Application Server.

For more information about downloading WebSphere Application Server Developer Tools for Eclipse, see:

https://www.ibm.com/developerworks/community/blogs/wasdev/entry/downloads_final_releases?lang=en

(New in V8.5.5) Enhancements have been made to Rational Application Developer 9.0 and WebSphere Application Server Developer Tools for Eclipse in support of the new V8.5.5 capabilities:

- ▶ Support for developing JAX-WS web services for the Liberty server, new EJB support, and additional templates to support WS-Security X.509 token profile
- ▶ Support for targeting and installing user-defined Liberty features
- ▶ Enhancements to Maven integration, most notably OSGi with EJB and JPA project conversion
- ▶ Enhancements to the web and mobile web development tools, including jQuery and Dojo support
- ▶ Improvements to EJB and CDI development tools, including the new beans.xml deployment descriptor editor
- ▶ New editor for configuring the server's DynaCache mechanism for caching servlets, JSPs, web services, and commands

- ▶ Support of multiple versions of WebSphere Application Server
- ▶ SCA tools enhancements, including improved integration of web services bindings
- ▶ Code quality and productivity enhancements with sample-based profiling capabilities from integration with the IBM Monitoring and Diagnostic Tools for Java - Health Center
- ▶ Updates to Java EE Connector Architecture tools, including new adapters for IBM CICS®, IBM IMS™ and WebSphere

OSGi programming model enhancements

The OSGi application console in WebSphere Application Server V8.5 complements the command-line console. With this console, you can view applications, packages, services, and dependencies. In addition, you can view the wiring in the shared bundle space and drill down on content bundles.

WebSphere Application Server V8.5 adds the ability to use the OSGi programming model with Enterprise JavaBeans (EJB). The EJB is packaged and deployed in an enterprise archive (EAR) file or web archive (WAR) module and then can be used as part of an OSGi application. Each module has its own classloader and, by default, uses the parent-first delegation model. This model provides for modularity and is an extension of the existing technology, using the EJB container in collaboration with OSGi for classloading and lifecycle management. The OSGi lifecycle management allows for in-place update and extension. The EJB can be published as a service in the OSGi Service Registry, and the client does not need to know that the service is implemented as an EJB.

You can also now configure bean security in the Blueprint XML file of OSGi applications so that the methods of the bean can be accessed only by users who are assigned a specified role. You can configure bean-level security so that a single role is associated with all the methods of the bean, or you can configure method-level security, where different roles are associated with specific methods.

Support for Java 7 features

Application developers now have the flexibility to use features in Java 7 in applications that are deployed to WebSphere Application Server V8.5. This flexibility allows development and production environments to select the most appropriate level of Java support for the situation.

By default, the underlying Java virtual machine (JVM) that is used in WebSphere Application Server V8.5 is the same as in WebSphere Application Server V8.0, which is Java 6. This support provides stability for enterprise applications between versions of the application server. You can, however, choose to take advantage of new features in Java 7 by using the Selectable JDK option of WebSphere Application Server V8.5.

You can install Java 7 as a feature extension to a new or existing WebSphere installation. You can then switch between using Java 6 and Java 7 for application servers. In addition, on IBM System i® and IBM z/OS platforms, you can switch between 32-bit and 64-bit Java SDK.

You can configure the software development kit (SDK) for your topology by using one of the following methods:

- ▶ The administrative console
- ▶ The `wsadmin AdminTask` commands
- ▶ The `manageSDK` command-line utility

You can have a default SDK for a node and isolate Java 7 usage to specific nodes in the topology. You can specify the level of Java that is used for specific servers and clusters and for command-line tools.

You can build and test Java 7 applications using Apache Ant or Maven, in addition to using Rational Application Developer to develop, deploy, and test Java 7 applications.

Migration toolkit enhancements

WebSphere Application Server V8.5 makes it even easier to migrate applications from WebSphere and other Java EE application servers. Using the Application Migration Toolkit, you can migrate applications from older releases of WebSphere Application Server and migrate applications from Tomcat, Oracle, and JBoss.

The Application Migration Toolkit analyzes application source code to find potential migration problems, including removed or deprecated features, behavior changes, differences between JRE5 and JRE6, and Java EE specification changes or enforcements. You can review these potential migration problems and, where available, allow the Application Migration Toolkit to make a *quick fix*. The Application Migration Toolkit also provides guidance about how to change those items where no quick fix is available.

The Application Migration Toolkit works with Eclipse or Rational Application Developer and is available at no additional cost by downloading it from the IBM website at:

<http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/>

Web 2.0 and Mobile Toolkit

With Web 2.0 and Mobile Toolkit, WebSphere Application Server developers can build and deploy reliable mobile web applications using standard web technologies, such as HTML 5, CSS3, and JavaScript. The resulting applications are usable on various mobile platforms, including iOS, Android, and BlackBerry, using the device's web browser. The user experience is close to the experience of each mobile operating system and supports touch interactions.

WebSphere Application Server Web 2.0 and Mobile Toolkit simplifies the addition of Asynchronous JavaScript and XML (Ajax) rich desktop and mobile user interfaces and Representational State Transfer (REST) web services to Java web applications. Web 2.0 capabilities, such as Ajax and REST, help application developers to create more connected, interactive applications that result in higher customer satisfaction, user productivity, and enhanced decision making. New mobile Ajax components enable developers to create mobile web applications that run on mobile devices, such as smartphones and tablets.

Service Component Architecture OASIS programming model implementation

WebSphere Application Server V8.5 adds support for the Service Component Architecture (SCA) OASIS programming model implementation. The product provides partial support for the following OASIS specifications:

- ▶ OASIS SCA Assembly Model Specification 1.1
- ▶ OASIS SCA Policy Framework Specification 1.1 (OASIS policy attachment is supported, but not OASIS policy set definitions)
- ▶ OASIS SCA JMS Binding Specification 1.1

- ▶ OASIS SCA Web Service Binding Specification 1.1
- ▶ OASIS SCA-J Common Annotations and APIs Specification 1.1

EJB binding, plain old Java object (POJO), Java Architecture for XML Binding (JAXB), and Service Data Objects (SDO) are all supported as data types.

Multi-threaded programming model

Support for asynchronous work management (`java.util.concurrent.ExecutorService`) has been added to allow applications to perform work asynchronously, while retaining the context of the calling thread. Applications can submit tasks to run concurrently, with thread context that is managed by the application server. Tasks can be run in a fire and forget manner or the application can wait for the task to complete. The `ExecutorService` can be accessed through JNDI or resource injection.

Improved operational efficiency and reliability

WebSphere Application Server provides industry-leading performance, operational efficiency, and reliability. Companies can take advantage of WebSphere Application Server high performance to consolidate workloads and administrative work, which can then reduce total cost of ownership (TCO) without sacrificing system reliability. WebSphere Application Server transactional support helps companies maintain transaction integrity and overall reliability to minimize the likelihood of lost business opportunities because of failed transactions or system downtime.

The following features of WebSphere Application Server V8.5 can improve operational efficiency and reliability:

- ▶ Intelligent Management
- ▶ Highly available deployment manager
- ▶ Messaging infrastructure resiliency

Intelligent Management

Intelligent Management capabilities are available with WebSphere Application Server Network Deployment V8.5. It provides a virtualized infrastructure that redefines the traditional concepts of Java EE resources and applications and their relationships with one another. This application infrastructure virtualization facilitates the product's ability to automate operations in an optimal manner, increasing the quality of service. By introducing an automated operating environment with workload management, you can reduce TCO by performing more work using less hardware.

The Intelligent Management features extend the quality of service provided by your middleware environment. Configurable operational policies govern the performance and health of your applications. TCO is decreased through server consolidation and lower administrative work and you experience faster response times and increased availability. In short, you experience the benefits of an autonomic middleware environment that is self-configuring, self-protecting, self-healing, and self-optimizing.

A key component of the Intelligent Management is the *on-demand router (ODR)*. The ODR supports health, application edition, and performance management features. It can manage both WebSphere and non-WebSphere environments. The ODR can queue requests for less important applications so that requests from more important applications are handled quickly.

New in V8.5.5: In V8.5, the ODR runs as a separate server. Requests that arrive at the web server are forwarded to the ODR by the WebSphere web server plug-in. The ODR then sends the request to the appropriate server based on current workload conditions, service policies, or other criteria. With V8.5.5, the ODR functionality can be optionally moved into the web server plug-in, eliminating the additional ODR tier and simplifying the topology.

The sections that follow provide an overview of the Intelligent Management capabilities that are available with WebSphere Application Server V8.5.

Application edition management

You can use application edition management to roll out new versions of applications without experiencing downtime for a maintenance window. You can use this feature to manage interruption-free production application deployments. Using this feature, you can validate a new edition of an application in the production environment without affecting users. You can then upgrade applications without incurring outages to users. You can also run multiple editions of a single application concurrently, directing different users to different editions.

The feature provides the following benefits:

- ▶ Incur no downtime when updating applications or the environment.
- ▶ Run multiple versions of applications concurrently.
- ▶ Verify that a new version of an application runs in production before directing user traffic to the application.
- ▶ Reduce infrastructure costs and decrease outages in the production environment.
- ▶ Update an operating system or WebSphere environment easily without incurring downtime to the environment.
- ▶ Perform a rollout to batch applications.

With WebSphere Application Server V8.5, you can install multiple versions of an application, called *application editions*, and define which version (or versions) of the application is active and processing traffic. An application edition can be in one of the following states:

- ▶ *Inactive*, which means it is installed but not available
- ▶ *Active*, which means it is installed and available within a running application server
- ▶ *Validation*, which is used to selectively send traffic to an application edition for testing or debugging purposes

You activate an application edition using one of the following modes:

- ▶ Rollout activation

This mode activates one edition in place of another, ensuring an interruption-free update in the process. Thus, all application requests are serviced during the rollout and no requests are lost. This mode ensures the perception of continuous application operation from the perspective of the application's customers.

- ▶ Concurrent activation

This mode activates the same edition on different servers or clusters. To use multiple editions concurrently, you must distinguish user requests from one another so that the requests can be sent to the application servers that are hosting the appropriate edition. For example, if you introduce a new edition of an application, you might want a select group of users to test the edition, rather than having all users access the edition.

► Validation activation

This mode is a special form of concurrent activation. It activates an edition on a clone of its original deployment target. The clone is created when the edition is activated. After the validation rollout to the original deployment target, the clone is removed automatically. You can perform final pre-production testing of an application edition in the actual production environment with a selected set of users.

Intelligent routing

Intelligent routing improves business results by ensuring priority is given to business critical applications. Requests to applications are prioritized and routed based on rules that are defined by the administrator. One way to define the rules is through the use of a *service policy*. A service policy specifies how to classify an incoming request based on request attributes, such as the URI, the client name, or HTTP headers. You differentiate the importance of requests with these attributes.

As incoming requests are processed, the service policies are used to determine if and how long the request is queued based on the current demand and resource utilization of the target servers. In this way, for example, you can give higher priority to a client trying to purchase an item from your website than to someone viewing an item in the catalog.

With service policies, you can classify, prioritize, and intelligently route workload. You can also adjust resources if needed to consistently achieve service policies. Service policies can be viewed as a technical implementation of service level agreements (SLAs) in place between the business area and the IT area that is running their applications.

Application server health management

You can use health management to monitor conditions automatically and take corrective actions when the conditions are observed. You can monitor the status of application servers, sense problem areas, and then respond to these problem areas before an outage occurs. The health monitoring and management subsystem monitors the operation of servers continuously against user-defined health policies to detect functional degradation that is related to user application malfunctions.

Health *policies* are designed to identify potential problems and take corrective action when an event occurs. They can help detect a problem before it causes serious problems and can either notify an administrator or perform preventive actions. Intelligent Management comes with predefined health conditions, such as excessive memory usage and request or response times, for use in building a health policy. You can also build a custom health policy using conditions that can be based on metrics gathered by the server, PMI metrics, MBean operations, and attributes.

When a health policy violation is detected, an action plan can be put into effect automatically. Actions to be taken when a monitored condition is detected are designed to bypass the problem and help in diagnosis. Predefined actions include notifying an administrator, sending an SNMP trap, restarting a server, putting a server into maintenance mode, and generating Java core files or heap memory dumps for use in diagnosing the problem. You can also define a custom action to be taken. Actions can be taken automatically, or you can have them occur in *supervised mode*. Supervised mode requires an operator to allow the action.

WebSphere Application Server V8.5 comes with the following predefined health conditions that Intelligent Management can monitor for and that you can use to create custom conditions:

- ▶ Age-based
Triggers when members who are associated with this policy reach a certain age value, such as 100 hours.
- ▶ Workload
Triggers when a defined number of requests are processed, such as 1000 requests.
- ▶ Excessive request timeout
Triggers when a percentage of requests time out before being processed, such as 20% of requests time out.
- ▶ Excessive response time
Triggers when the average response time for requests exceeds a certain amount of time.
- ▶ Excessive memory usage
Triggers when the memory usage exceeds a defined percentage of the maximum heap size for a certain amount of time.
- ▶ Excessive garbage collection
Triggers when the JVM spends more than a configured percentage of time performing garbage collections.
- ▶ Memory leak
Attempts to detect a memory leak based on a consistent downward trend in free memory available to a server in the Java heap.
- ▶ Storm drain
Attempts to detect situations where requests are shifted toward a faulty cluster member that advertises low response times.

Performance management

Performance management provides a self-optimizing middleware infrastructure. You can use dynamic clusters to scale up and out the number of running cluster members automatically as needed to meet response time goals for users. You can take advantage of overload protection to limit the rate at which the *on-demand router* forwards traffic to application servers to prevent heap exhaustion, processor exhaustion, or both types of exhaustion.

Using *dynamic clusters*, WebSphere Application Server V8.5 automatically increases or decreases the number of running cluster members as needed to meet response time goals that you set for users. In a static environment, you have dedicated servers for each application, and the environment must be able handle the expected load during peak times. This environment means that during non-peak hours your servers are underutilized. In general, a company has more than one critical application. It is likely that the second application has its peak load at a different time of the day. With dynamic clustering, you can more effectively use system resources by shifting the system resources of a non-peak application to the peak application at any time.

Overload protection is a feature that monitors the use of memory and processor usage of a server and then regulates the rate at which traffic is sent to an application server to prevent memory and processor overload. Memory overload protection is disabled by default. Enabling it requires the configuration of the *autonomic request flow manager* (ARFM).

For a dynamic cluster, you can indicate a maximum heap utilization percentage that protects against out of memory errors. For processor overload protection, you can indicate a maximum processor percentage that protects against various failures that could occur when processor capacity is consumed. You can set a rejection policy that prevents a processor from being overloaded by rejecting incoming HTTP or SIP messages that are not part of existing sessions for HTTP or SIP traffic.

Highly available deployment manager

You can configure the highly available deployment manager function to eliminate single points of failure (SPOFs) for administrative functions in a WebSphere Application Server Network Deployment V8.5 cell on distributed platforms. This feature is important in environments that have significant reliance on automated operations, including application deployment and server monitoring.

The deployment managers exist as peers. One deployment manager is considered *active*, which is also known as the *primary* deployment manager. This deployment manager hosts the administrative function of the cell. The other deployment managers are *standby* deployment managers. If the active manager fails, a standby deployment manager takes over and is designated the new active deployment manager.

Each deployment manager shares the instance of the master configuration repository and workspace area, which must be on a shared file system. The on-demand router is configured with the communication endpoints for the administrative console, the `wsadmin` tool, and scripting. The on-demand router recognizes which deployment manager instance is active and routes all administrative communication to that instance. However, if the active deployment manager is stopped or the request fails, the highly available deployment manager component recognizes the loss of the active deployment manager and dynamically switches the standby deployment manager into active mode so that it can take over for the lost deployment manager.

The active and standby deployment managers share workspaces. When a deployment manager takeover occurs, work is not lost because the on-demand router recognizes the election of the new active deployment manager automatically and reroutes administrative requests to the new active deployment manager.

Session Initiation Protocol serviceability and resiliency enhancements

Session Initiation Protocol (SIP) is used to establish, modify, and terminate multimedia IP sessions including IP telephony, presence, and instant messaging.

(New in V8.5.5) Serviceability and troubleshooting enhancements to Session Initiation Protocol (SIP) support enable more resilient processing of SIP sessions.

New PMI counters at the SIP container and proxy have been added to monitor and trigger on key performance indicators:

- ▶ New counters for the SIP container allow you to monitor for thread and message congestion issues, the number of replicated and non-replicated SIP sessions, the number of rejected requests, and SIP timers.
- ▶ New counters for the SIP proxy allow you to monitor queue statistics, the health of the SIP container and load balancer, and invalid SIP messages received.

The following new troubleshooting features have been included:

- ▶ The SIP context is now added to binary logs entries for the SIP container and SIP proxy. The new information allows you to trace the flow of a SIP call through all the SIP components.
- ▶ A new utility is provided to dump SIP application sessions and their session IDs for improved debugging of SIP container sessions. This utility can be particularly useful in production environments when fine grained tracing cannot be enabled.
- ▶ SIP proxy call logging now provides complete message logging as well as logging of rejected messages.

Application composition performance improvements have been added to allow multiple independent applications installed at a single JVM to independently process either a request or response. The number of composed applications that can be deployed is increased through avoidance of serialization and de-serialization of the request headers.

A new API has been included that provides callback when a message is not matched to an existing dialog. The API receives incoming SIP request or response messages that cannot be processed by the SIP container.

Messaging infrastructure resiliency

WebSphere Application Server V8.5 implements a powerful and flexible messaging platform within the WebSphere Application Server environment called the *service integration bus*. WebSphere Application Server applications invoke asynchronous messaging services, using the Java Messaging Service (JMS) API to interface with a messaging provider. The messaging provider can be the WebSphere MQ messaging provider, the default messaging provider (bus), or a third-party messaging provider. A *messaging engine* is the service integration bus component that is responsible for processing messages, sending and receiving requests, and hosting destinations.

WebSphere Application Server V8.5 provides the following key improvements in the area of resiliency of the messaging engine:

- ▶ The Smart Messaging Engine attempts to shut down gracefully when it encounters a problem from which it cannot recover, without bringing down the entire JVM.

This graceful shutdown avoids the scenario where the entire application server is affected if the messaging engine encounters an unrecoverable problem, such as a hang or a bad database. Thus, other applications continue to function normally while the cause of the messaging engine issue is resolved.

For example, when the active messaging engine loses connectivity to the database, the messaging engine raises a local error and notifies the HAManager that it cannot continue. The HAManager triggers the standby messaging engine to start and take ownership of the database while gracefully stopping the active messaging engine. The standby messaging engine then becomes the active messaging engine, and the previously active messaging engine enters a “disabled” state.
- ▶ The Smart Messaging Engine re-enables itself after a configurable amount of time.

The Smart Messaging Engine is available if needed for future use after the unrecoverable error is resolved.
- ▶ The Smart Messaging Engine persists the Java Message Service (JMS) redelivery count to track the number of times a message is delivered, even after a restart.

This feature determines whether the message delivery was attempted earlier, before the restart. Previously, the redelivery count was not persisted. Thus, if the Smart Messaging

Engine was restarted, the redelivery count of a message was lost and always reset to zero (0). Thus, the application would reprocess the same message.

(New in V8.5.5) Service mapping

The use of services depends on the ability of the service client to locate and communicate with the service provider. Changes to either the location of the service or to the interface can impact the applications that use the service. The new service mapping feature is designed to shield applications from minor changes in the services they use.

This feature gives administrators the ability to define a mapping service that can intercept service client invocations bound for a particular service. The mapping service can determine which service location the message should be routed to, which operation on the service provider should be invoked, and how the fields in the client and server messages should be mapped to each other. Administrators can control to which service interactions the service mapping applies. The mapping is created using a graphical interface, simplifying the task.

Edge Components

The Edge Components available with Network Deployment include the following components:

- ▶ The Caching Proxy offloads back-end servers by caching static content and content dynamically generated by WebSphere Application Server. The Caching Proxy component can be configured as a reverse and a forwarding proxy. This proxy server supports the HTTP, HTTPS, FTP, and Gopher protocols.
- ▶ The Load Balancer for IPV4 and IPV6 provides horizontal scalability. It dispatches HTTP requests among several web server or application server nodes that support various dispatching options and algorithms to assure high availability in high volume environments. Using the Load Balancer for IPV4 and IPV6 can reduce web server congestion, increase content availability, and provide scaling ability for the web server.

(New in V8.5.5) The Load Balancer for IPV4 and IPV6 has been enhanced in V8.5.5 to improve flexibility in configuration and to improve workload balancing. The new features include:

- ▶ The load balancer can now be run on the same machine as the servers it is balancing. This feature is supported on Linux and IBM AIX® only.
- ▶ The Content Based Routing (CBR) component has been added to enable load balancing based on the content of client requests, for example, the URI.
- ▶ The Site Selector component has been added to enable balancing load using domain name service (DNS) round robin or using a user-provided algorithm.
- ▶ Network Address Translation (NAT) has been added, removing the limitation that back end servers are on the same locally attached network.

The Edge Components also include a Load Balancer for IPV4, which is being deprecated. The primary capabilities of this load balancer are being migrated to the Load Balancer for IPV4 and IPV6.

Increased security and control

WebSphere Application Server offers security and control to help businesses confidently reduce costs and increase business agility. WebSphere Application Server support for security specifications and granular security controls help administrators productively secure application environments on which businesses depend.

The following features provide increased security and control with WebSphere Application Server V8.5:

- ▶ Batch enhancements
- ▶ Administrative security auditing using checkpoints
- ▶ SAML Web SSO Post binding profile
- ▶ Cross-component tracing

Batch enhancements

The Java EE applications typically hosted by WebSphere Application Server perform short, lightweight, and transactional units of work. In most cases, an individual request can be completed using only seconds of processor time and relatively little memory. Many applications, however, must complete batch work that is computational and resource-intensive.

The batch function in WebSphere Application Server extends the application server to accommodate applications that must perform grid work alongside transactional applications. Batch work might take hours or even days to finish and uses large amounts of memory or processing power while it runs.

Batch support includes a web-based application for managing jobs, called the *job management console*. Through this console, you can submit jobs, monitor job execution, perform operational actions against jobs, and view job logs.

WebSphere Application Server V8.5 builds on existing batch features by integrating capabilities from WebSphere Compute Grid V8 to deliver a complete enterprise-level Java batch processing solution. WebSphere Application Server V8.5 includes the following batch features:

- ▶ Enterprise scheduler integration

You can use special connectors for IBM Tivoli® Workload Scheduler and most non-IBM workload schedulers. You can use this integration feature to put WebSphere Batch under full control of the existing enterprise workload scheduler. Workload schedulers can use workload connector to send batch jobs to WebSphere Job Scheduler for processing.
- ▶ Parallel processing of batch jobs

Container-managed parallelization now uses multiple cores for efficiency. You can use a single job, instead of multiple jobs, to retain operational control. The container manages the parallel operations in a “divide and conquer” approach that improves elapsed time and achieves near linear runtime performance.
- ▶ Memory overload protection

WebSphere Application Server V8.5 has added memory overload protection for WebSphere Batch jobs, which provides a level of protection against over-scheduling jobs to an application server. The batch container monitors job memory demand against the available JVM heap space to determine whether there are adequate resources for the next job to begin running. This process helps to prevent OutOfMemory exceptions.

WebSphere calculates a job memory estimation automatically so you can override this value in the xJCL.

► Job log SPI

A new system programming interface (SPI) provides access to the control job log content and destination. Through this SPI, you can specify that the control job log contents be sent to the job log only, to the WebSphere Application Server server log only, or to both logs. You can also suppress the job log and send information to neither log. You can also use the SPI to modify any job log line.

► Mixed batch workloads

With WebSphere Application Server V8.5, you can mix various step types within the same job. Within a single job, you can define any combination of steps, including transactional batch, parallel, compute intensive, and native execution.

► COBOL support on z/OS

You can now reuse COBOL modules in WebSphere applications by using the COBOL support available in WebSphere Application Server V8.5. On z/OS systems, you can call standard COBOL modules from Java. COBOL and Java run in the same transaction scope, on the same thread, and in the same process. IBM DB2® connections managed by WebSphere Application Server are shareable with COBOL modules, and you can use working storage isolation at either the job step or the remote call level. COBOL support is available to both batch and online applications.

► SMF Type 120 Subtype 9 usage record

With WebSphere Application Server for z/OS V8.5, you can use the optional subtype 9 usage record for each job. The subtype 9 usage record includes zIIP, zAAP, and general-purpose processor type, given in CPU seconds and service units. Additional data, including the accounting string, job ID, and submitter, are also available. Subtype 9 records complement subtype 20 batch records by providing a richer set of information.

► OSGi batch applications

You can now use OSGi for batch application development with the full batch programming model available to the OSGi framework. This enterprise bundle archive deploys both standard and blueprint bundles.

► Record processing policies

You can define declarative policies for how to handle bad records and processing failures. When exceptions occur in job steps, you can skip the records with the exceptions and stop processing after a certain number of records are skipped. You can also define the policy for trying the job step again when an exception occurs, including how many times to try again and, optionally, how long to delay before trying again. WebSphere Application Server V8.5 also includes optional programmatic control through application listeners. You can define a SkipListener or a RetryListener to gain control during a skip or retry action.

► Record metrics

The Batch Container collects several key metrics during the processing of a job and writes the values to the job log at the end of the job step. WebSphere Application Server V8.5 includes optional programmatic control through the JobStepContext. The JobStepContext object is updated to exist for the life of the job, instead of only the life of the job step. Any step-specific context is reset at the start of each job step.

The Batch Container can collect the following key metrics:

- Skipped record count
- Retry count
- Records per second
- Total processing time

- ▶ Job and step listener

You can define a JobListener and be notified programmatically of job and step lifecycle events, including job start and end and step start and end. These events are available to the JobListener through the JobStepContext object. The JobStepContext object is updated to exist for the life of the job, instead of only the life of the job step. Any step-specific context is reset at the start of each job step.

Administrative security auditing using checkpoints

As an administrator, it is important to know of each change made to the configuration and environment to ensure consistency in multiple application server environments, if needed, and to determine problems if a failure occurs.

Repository checkpoints are saved images of the repository before a configuration change is made. You can create repository checkpoints to save snapshots of the configuration as you make changes so that you can easily undo those changes if necessary. You can use the following types of checkpoints:

- ▶ A *full checkpoint* is a copy of the entire configuration repository, including applications and connectors. Use a full checkpoint to restore the entire configuration repository back to the state it was in at the time the full checkpoint was made.
- ▶ A *delta checkpoint* is a subset snapshot of the configuration repository and is used to restore the configuration repository back to a prior state. This checkpoint is similar to minor, incremental versions.

You can use WebSphere Application Server V8.5 to track changes made to the application server configuration using checkpoints made through the extended repository service. You can restore the configuration repository to a prior state using a checkpoint. If you must determine what changed in the configuration, you can extract this information from a delta checkpoint to obtain the before and after versions of the files that were saved. You can then compare the before and after versions of the files with a file comparison tool of your choice.

You can configure the repository to create an automatic delta checkpoint each time a configuration change is made. A delta checkpoint saves a copy of the configuration documents before saving changes. You can specify the number of automatic checkpoints to save. After this limit is reached, the next checkpoint replaces the oldest checkpoint. For example, you can have the system save only the last 10 checkpoints.

Configuring the repository checkpoints requires either a configurator or administrator role. Anyone with a monitor or operator role can view the repository checkpoint information.

SAML Web SSO Post binding profile

Security Assertion Markup Language (SAML) is a standard that is based on XML. It defines the framework for exchanging security information (assertions) between systems. In WebSphere Application Server V8.5, you can enable support for a SAML 2.0 HTTP post binding profile without requiring the use of an additional product, such as IBM Tivoli Federated Identity Manager.

The following features are available when implementing an SAML service provider in WebSphere Application Server V8.5:

- ▶ Single sign-on with multiple identity providers
- ▶ Options for identity assertion and mapping the assertion identity to the user registry of the service provider

- ▶ Mapping or asserting SAML token attributes to their realm, principal, and unique ID, and then grouping them into the service provider security context
- ▶ Plug point to allow for customized identity mapping
- ▶ Option to retrieve the group membership of the identity from the registry of the service provider and populate the security context
- ▶ Identity provider selection filter that routes the request back to the correct identity provider if the request did not come from the identity provider
- ▶ RSA-SHA1 and RSA-SHA256 signature algorithms
- ▶ Preservation of the SAML token in the subject of the service provider for access by the application, which makes it available for a downstream authenticated EJB or web service call
- ▶ A business application URL that acts as an AssertionConsumerService URL so that the identity provider can send a SAMLResponse directly to the business application URL
- ▶ Auditing of key SAML assertions, including Issuer and NameID

Cross-component tracing

The cross-component trace capability enables the correlation of log and trace entries that are created by multiple threads across processes that belong to the same request or unit of work. This trace annotates the logs so that log entries that are related to a request that is serviced by more than one thread, process, or even server are identified as belonging to the same unit of work. This process helps identify the root cause of problems for components. It enables administrators and support teams to follow the flow of a request from end to end as it crosses thread or process boundaries or travels between WebSphere Application Server and stack products. It can also help resolve questions about which component is responsible for a request that fails.

Cross-component trace capability is built into the WebSphere Application Server log and trace framework. Applications built using distributed architectures, such as an SOA, can benefit from cross-component tracing because this capability helps facilitate problem determination across multiple services on different systems.

You can configure the cross-component trace feature to run in one of the following modes, depending on the level of detail that you need:

- ▶ Fully disabled
- ▶ With cross-component trace request IDs added to existing log and trace records
- ▶ With cross-component trace request IDs added to existing log and trace records and cross-component trace log records added to log files
- ▶ With cross-component trace request IDs added to existing log and trace records, cross-component trace log records added to log files, and data snapshots captured

A cross-component trace request ID is an identifier that is added to log and trace records that are produced by the server when the server is configured to use High Performance Extensible Logging (HPEL). Cross-component tracing adds the same request ID to every log or trace record if the log or trace record is a part of the same request, regardless of which thread or JVM produces the log or trace entry. When cross-component tracing is used with the HPEL log and trace infrastructure, you can view request IDs with the Log Viewer tool when logs are output in advanced format.

A cross-component trace log record is typically added to the logs to note the beginning and ending of work for a particular request on a particular thread when work is about to be

transferred to another thread or process, or to indicate when work is returned from another thread or process. It is also used to show when work moves from one major component to another, even if it is on the same thread. This tracing is useful to show the transfer of control from application server code to application code.

You can use the HPEL Log Viewer tool to filter log and trace records by request ID. Tools, such as the cross-component trace Log Viewer, can also take advantage of cross-component trace log records or cross-component trace request IDs, or both, when rendering log and trace content. The cross-component trace Log Viewer is available as a tool add-on for the IBM Support Assistant.

Packaging enhancements

WebSphere Application Server is available on a range of platforms and in multiple packages to meet specific business needs. By providing the application server that is required to run specific applications, it also serves as the base for other WebSphere products and many other IBM software products. In addition to the application server component, each package contains an appropriate combination of complementary products, for example, IBM HTTP Server, IBM Assembly and Deploy Tools for WebSphere Administration, Edge components, and other products.

Because different application scenarios require different levels of application server capabilities, WebSphere Application Server is available in multiple packaging options. Although these options share a common foundation, each provides unique benefits to meet the needs of applications and the infrastructure that supports them. At least one WebSphere Application Server product fulfills the requirements of any particular project and its supporting infrastructure. As your business grows, the WebSphere Application Server family provides a migration path to more complex configurations.

The following packages are available:

- ▶ WebSphere Application Server Express
- ▶ WebSphere Application Server Base
- ▶ WebSphere Application Server Network Deployment
- ▶ WebSphere Application Server for z/OS
- ▶ WebSphere Application Server for Developers
- ▶ WebSphere Application Server Hypervisor Edition
- ▶ **(New in V8.5.5)** WebSphere Application Server Liberty Core

In addition to these options, there are three WebSphere Application Server Tools Editions available. The Tools Editions are bundles of a WebSphere Application Server run time and development tools. For more information about the Tools Editions, see:

<http://www-01.ibm.com/software/webservers/appserv/was/tools/>

Each package includes both the full profile and a Liberty profile server, except the Liberty core edition is Liberty profile only, and the Community Edition is open source based and contains neither the full nor the Liberty profile. The features available for each profile vary among the different packaging options.

(New in V8.5.5) In addition to the Liberty profile available with the WebSphere Application Server edition packages, a new WebSphere Application Server Liberty Core edition is available. This package provides a lightweight and low-cost Liberty profile based offering, providing the capabilities to rapidly build and deliver web applications that do not require the full Java EE stack.

(New in V8.5.5) WebSphere Application Server (base edition), WebSphere Application Server Network Deployment, and WebSphere Application Server for z/OS now include WebSphere eXtreme Scale in the package and entitlements to its use. Both the Liberty profile and the full profile can take advantage of the caching abilities of WebSphere eXtreme Scale.

Figure 1 shows a high level view of the WebSphere Application Server packaging options.

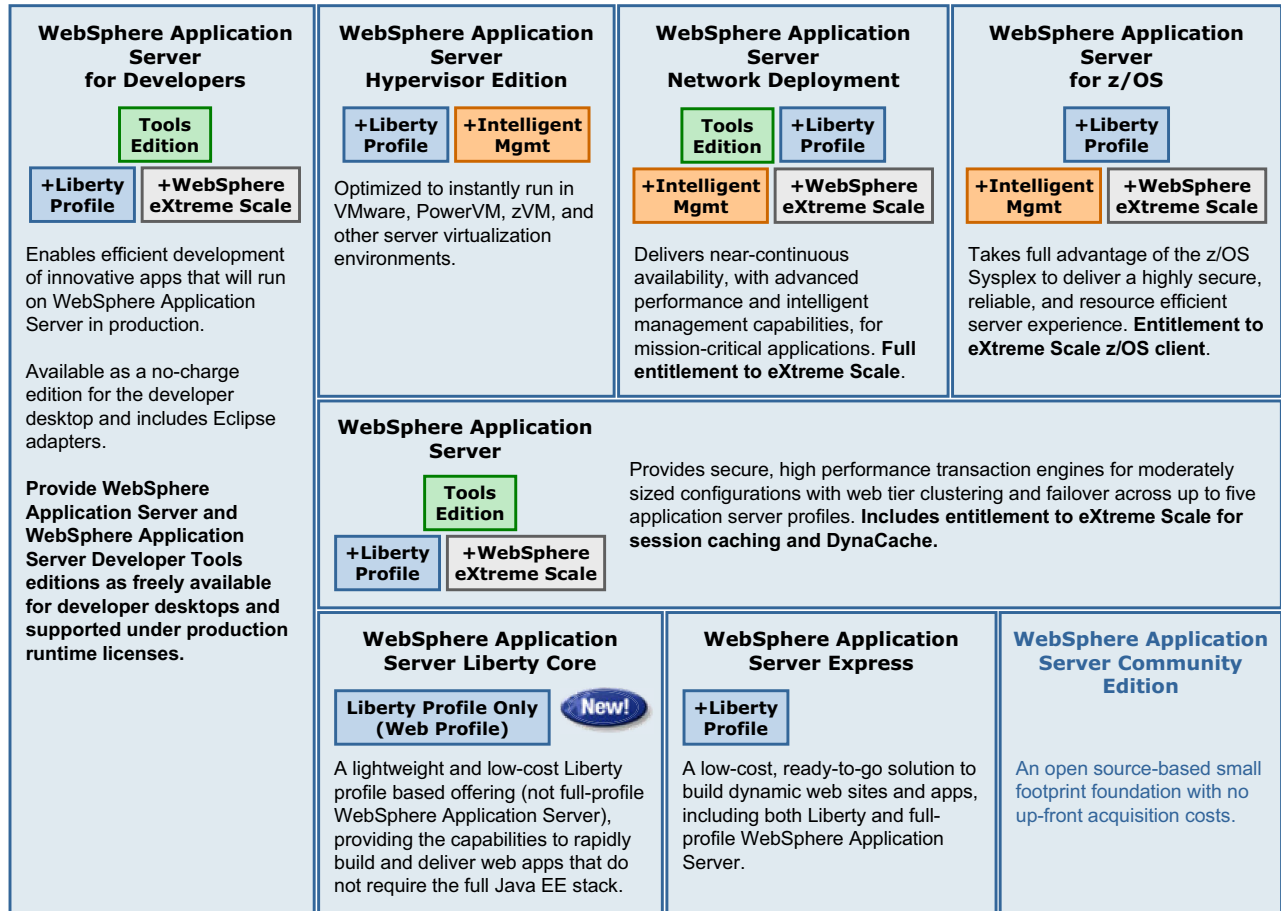


Figure 1 WebSphere Application Server V8.5 packaging overview

Authors

This paper was produced by technical specialists working at the International Technical Support Organization, Raleigh Center.

Carla Sadtler is a Consulting IT Specialist at the ITSO, Raleigh Center. She writes extensively about WebSphere products and solutions. Before joining the ITSO in 1985, Carla worked in the Raleigh branch office as a Program Support Representative, supporting IBM MVS™ customers. She has a degree in Mathematics from the University of North Carolina at Greensboro.

Susan Hanson is a member of the WebSphere Application Server foundation development team. She has 22 years of experience in developing and delivering IBM software products across the WebSphere and Tivoli brands. Her current focus products are WebSphere Application Server, WebSphere Virtual Enterprise, and WebSphere eXtreme Scale, with focus areas in release management, project management, and development process

transformation. She also works part-time in the IBM Redbooks® organization as a project leader focused on the Growth Market Unit (GMU) areas and is part of the ITSO strategy team that is focused on industries and GMU enablement. Susan holds a Bachelor's degree in Computer Science from East Carolina University and a Master's degree in Computer Information Systems from The University of Phoenix. She is based in Research Triangle Park, North Carolina, and is temporarily working and residing in Shanghai, China.

Thanks to the following people for their contributions to this project:

Margaret Ticknor, Deana Coble, Debbie Willmschen
International Technical Support Organization, Raleigh Center

Michael Cheng, Christopher Vignola, Soloman Barghouthi, Bill O'Donnell
IBM US

Ian Robinson
IBM UK

The team who created *IBM WebSphere Application Server V8.5 Concepts, Planning, and Design Guide*, SG24-8022.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks publications

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Notices

This information was developed for products and services offered in the U.S.A.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

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

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

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

This document REDP-4870-01 was created or updated on May 28, 2013.

Send us your comments in one of the following ways:


- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.



Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

AIX®	IMS™	Redbooks (logo)  ®
CICS®	MVS™	System i®
DataPower®	Rational®	Tivoli®
DB2®	Redbooks®	WebSphere®
IBM®	Redpaper™	z/OS®

The following terms are trademarks of other companies:

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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