Oracle Application Server 10g, a component of Oracle Fusion Middleware, represents the next generation of enterprise application server. With its service oriented architecture foundation and grid enabled deployment infrastructure, Oracle Application Server enables on-demand computing across an enterprise by efficiently pooling and utilizing all of the hardware resources (CPU, memory, storage) prescribed to it from within an enterprise. At the heart of Oracle Application Server 10g is Oracle Containers for J2EE (OC4J) - a fast, lightweight, highly scalable and easy-to-use J2EE container that is written entirely in Java and which executes using the Java2 Standard Edition (J2SE) 1.4.2 or 5.0 runtime.

**Overview**

Oracle Containers for J2EE (OC4J) 10g Release 3 (10.1.3.1) provides a fully J2EE 1.4 compliant environment for creating and hosting secure, scalable, highly-available applications. It provides all the containers, APIs, and services mandated by the J2EE 1.4 specification, including:

- A Web container supporting JSP page translation and execution within a servlet engine
- An Enterprise Java Beans (EJB) container providing full support for the new EJB 3.0 specification
- Support for Common Annotations (JSR-250) in both the Web and EJB containers
- Expanded JMS support, including enhanced enterprise messaging capabilities and a choice of JMS providers
- Full implementations of the remaining J2EE services such as JNDI, JDBC, JTA, JCA, and JAAS
- Extensive support for SOAP 1.1/1.2 and WSDL 1.1 based Web services including Enterprise Web Services 1.1, JAX-RPC 1.1 and SAAJ 1.2. Full support for advanced Web service capabilities includes WS-Security, WS-Reliability, REST and Web Services Metadata.

For flexibility, OC4J is distributed in two basic configurations to meet the varying needs of any organization or enterprise:

- A *standalone* configuration, in which OC4J is installed as a single, "standalone" server instance and is directly managed as a self-contained
component. This standalone distribution provides a robust platform for software development as well as for smaller-scale business application environments.

- A managed configuration, in which OC4J is installed and managed as a component of Oracle Application Server. In this configuration, OC4J is able to take advantage of the rich feature set available with the Oracle Application Server infrastructure.

J2EE Standards Support

The following table lists the J2EE specifications supported in the current OC4J release.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>2.4</td>
<td>1.1</td>
<td>3.0</td>
<td>1.2</td>
<td>JSR-160</td>
<td>1.0 (JSR-77)</td>
<td>1.1 (JSR-88)</td>
<td>1.0</td>
<td>1.1</td>
<td>1.2</td>
<td>1.2</td>
<td>3.0</td>
<td>1.0</td>
<td>1.5</td>
<td>1.1 (JSR-921)</td>
<td>1.0 (JSR-181)</td>
<td>1.1</td>
<td>1.2</td>
<td>1.2</td>
<td>1.0.5</td>
<td>JSR-94</td>
<td>JSR-250</td>
</tr>
</tbody>
</table>

New Features in Oracle Containers for J2EE 10g 10.1.3.1

The following sections provide a brief overview of the new features and functionality available with the latest release of OC4J.

Enhanced Management Capabilities

OC4J now offers a new management framework designed to lower IT management costs and increase operational efficiency for administrative staff.

For starters, OC4J now supports the following specifications, which define new standards for managing applications in a J2EE environment.

- The Java Management Extensions (JMX) 1.2 specification, which allows standard interfaces to be created for managing resources, such as services and
applications, in a J2EE environment.

- The J2EE Management Specification (JSR-77), a specification that allows standard components to be created for managing applications in a J2EE environment.

The new management functionality is implemented through the use of JSR-77-compliant MBeans. All manageable components and resources within OC4J—including J2EE services, processes, applications running within the OC4J instance and even the OC4J server itself—are fully managed using these components.

The JMX infrastructure is exposed through the Application Server Control Web-based management application, which is built on an MBean framework. For finer-grained control, the MBeans underlying Application Server Control can be accessed directly through a built-in “MBean browser”.

Custom application-defined MBeans written by application vendors can be deployed with applications into OC4J, making it possible to expose attributes and perform custom operations specific to the application through Application Server Control.

In addition to JMX support, Application Server Control now supports local and remote administration, enabling multiple OC4J instances to be centrally managed. New administrative roles are also provided, making it possible to grant varying levels of administrative access.

With the new enhancements, Application Server Control provides a comprehensive management environment with facilities for:

- Starting and stopping the server and deployed application instances
- Managing clustered environments, including performing operations on “groups” of OC4J instances within the cluster
- Deploying, configuring and managing J2EE applications and standalone modules
- Creating and managing shared resources, such as data sources and JMS providers
- Managing and testing deployed Web services
- Viewing aggregated as well as detail-level runtime performance metrics

Application Server Control is now packaged with the OC4J standalone distribution (it had previously been available with Oracle Application Server only.) This addition to the standalone distribution greatly increases overall usability.

**New Application and Standalone Module Deployment Features**

New utilities are provided to facilitate deployment of standard J2EE applications or standalone modules (EAR, WAR, EJB JAR, RAR or client JAR files) to either a single OC4J instance or across all instances within a cluster.

Deployment has also been enhanced with support for the J2EE Application Deployment API (JSR-88), which defines a standard API for configuring and deploying J2EE applications and modules into a J2EE-compatible environment. The
OC4J implementation includes the ability to create and/or edit a deployment plan containing the OC4J-specific configuration data needed to deploy a component into OC4J. Once created, a deployment plan can be re-used for successive deployments or redeployments.

OC4J supports incremental or partial redeployment of EJB modules that are part of a deployed application. This feature makes it possible to re-deploy only those EJBs within an EJB JAR that have changed, without requiring the entire module to be re-deployed. Previously deployed beans that have not been changed will continue to be used.

This functionality represents a significant enhancement over previous releases of OC4J, which treated an EJB module as a single unit, requiring that the module first be un-deployed, then re-deployed with any updates.

**New Class Loading Framework**

The OC4J class loading framework has been significantly re-engineered to offer a more flexible set of options for configuring and controlling how OC4J provides resources to deployed applications.

Inherent in the new framework is the ability to define shared libraries, each of which represents a set of code-sources (JAR files) loaded at runtime. Many libraries packaged with OC4J and inherited by deployed applications have also been defined as shared libraries.

The resulting architecture makes it possible to control which version of a particular library is used by an application or a set of applications, or even remove a library from the set imported from OC4J by default-without having to modify the packaged application. For example:

- An application can be configured to use a JDBC driver version than those inherited from OC4J or a parent application.
- An open-source XML parser could be defined as a shared library to be shared by multiple applications. To avoid class collisions with the default XML parser provided by OC4J, applications can be configured to not import the default parser at deployment time.

In the event that class collisions do occur, approximately 15 new runtime queries are provided to aid in troubleshooting and resolving common yet difficult-to-debug class loader-related errors and exceptions.

**New EJB and HTTP State Replication Capabilities**

OC4J provides a flexible framework for creating a clustered application environment for development and production purposes. In this context, a cluster is defined as two or more OC4J server nodes hosting the same set of applications. The OC4J clustering framework supports:

- Replication of objects and values contained in an HTTP session or a stateful session Enterprise JavaBean instance.
- In-memory replication using multicast or peer-to-peer communication, or
• Persistence of state data to a database.
• Load balancing of incoming requests across OC4J instances.
• Transparent failover across applications within the cluster.
• Configuration within an OC4J instance at either the global or the application level.

Application-level clustering can be configured at the time the application is deployed, or can be enabled by default for all applications that support replication within the OC4J instance.

Support for the Latest Web Tier Specifications
The latest release provides full support for the JavaServer Pages 2.0 and Servlet 2.4 specifications. This includes support for tag files, which allow JSP authors to create custom tag libraries without extensive knowledge of the Java programming language. Support for Common Annotations (JSR-250) is also provided in servlets and JSP tag libraries.

New EJB 3.0 Support
The OC4J EJB container is fully compliant with the EJB 3.0 specification – the first commercially available container to provide full support. For developers, EJB 3.0 greatly simplifies the EJB programming model and provides a standard persistence model for the J2EE platform.

For EJBQL, additional capabilities have been added to support the use of common time and date types within query statements, specifically, the use of java.util.Date, java.sql.Date and java.sql.Timestamp objects. This enables EJBQL statement to perform operations which utilize date and time fields. The use of a SQRT function has also been added.

For stateful session beans, a configurable activation and passivation capability has been implemented -- configuration options include factors such as the maximum instance count, memory threshold limits, and inactivity time-outs.

For J2EE applications which are deployed to Oracle Application Server environments and make connections to managed OC4J instances which have dynamically allocated port numbers, a new transparent lookup operation has been added to allow client side EJB components to locate the required port numbers.

Extended Web Services Support
The Oracle Application Server Web Services (OracleAS Web Services) framework is tightly integrated with J2EE so that an existing J2EE application can easily be turned into a SOAP 1.1 based Web service based on JAX-RPC 1.1, SAAJ 1.2, and Enterprise Web Services 1.1. Web services deployed to, and executed on the OC4J platform inherit all the runtime and lifecycle management elements of J2EE applications. In addition to J2EE application components, the following types of Web services can be run on OC4J

• Stateless or stateful Java classes as RPC or Document Style Web services
• Stateless session EJBs as Web services
• PL/SQL Stored Procedures as Web services
• JMS endpoints (Topics and Queues) as Document Style Web services
• Support for source code annotations to customize Web services behavior such as invocation and ending styles (RPC/literal, RPC/encoded, Doc/literal)

Beyond full J2EE 1.4 support for Web services, Oracle Application Server Web Services includes support for a number of advanced Web services features including Web Services Metadata, REST, WS-Security and WS-Reliability. A new Web Services Inspection Language (WSIL)-based tool is pre-installed in OC4J to find and expose the URLs for all deployed WSDL files within an Oracle Application Server cluster, including WSDLs for BPEL processes and ESB services.

Support for Open Source Frameworks
Open source frameworks and software has rapidly become fundamental to many organizations’ development projects. In reaction to this Oracle has invested heavily in certifying key popular open source software frameworks, libraries and products on OC4J. A short list of key open source projects that will be formally certified include:

• Spring
• Apache Struts
• Apache Axis
• Apache MyFaces
• Apache Ant
• Hibernate
• Log4J

The complete list is posted on the Oracle Technology Network along with information on configuration and setup of these and other open source projects.

New Data Source Features
Significant work has been done in Oracle Containers for J2EE with regards to data sources, particularly in the area of simplification and hardening of the underlying infrastructure. In the area of simplification, OC4J 10g R3 has focused on rationalizing the categorization of data sources to the following:

• **Managed data sources**: A managed data source is an OC4J provided data source implementation that acts as a wrapper to a JDBC driver or data source. OC4J provides critical system infrastructure, such as global transaction management, connection caching, dynamic configuration via JMX, and error handling, for managed data sources.

• **Native data sources**: A native data source is a data source implementation from JDBC driver vendors such as Oracle and DataDirect. Native data sources are not wrapped by OC4J.

In the area of hardening, OC4J provides a new connection-caching mechanism that
leverages all the new Implicit Connection Cache features in Oracle Database 10g JDBC. Some of the advantages associated with the new connection caching include:

- Driver independence
- Full compliance with the JDBC 3.0 standard
- Transparent access to the connection cache
- Multiple users and passwords per cache
- Connection recycling and refresh of stale connections
- Attribute-based search for connections
- Support for multiple cache-enabled data sources
- Connection cache callback mechanism
- Integrated support for RAC fast connection failover

Data sources are now managed dynamically using JMX, and do not require a container restart for configuration changes to take effect.

**Security Enhancements**

OC4J provides a number of enhancements to the base security infrastructure as part of J2EE 1.4 compliance as well as new capabilities for enterprise deployments. These include:

- Native support for 3rd party LDAP servers including Active Directory and Sun Java System Application Server (formerly known as iPlanet).
- Security context propagation, also known as subject propagation, between Oracle Containers for J2EE instances.
- JACC Support - This feature implements Java Authorization Contract for Containers (JSR-115).
- Digest authentication support, and client certification authentication and authorization support.

In addition, OC4J now supports ORMI over SSL (ORMIS), which enables Secure Socket Layer (SSL) RMI communication between objects across OC4J server instances.

**New Scripts and Command Line Tools**

OC4J continues to provide a comprehensive set of command line tools and scripts to facilitate management of the development and deployment environment.

- A new command line tool for performing operations, including deploying applications, on a standalone OC4J server or across one or multiple OC4J instances within a cluster. These work consistently in stand-alone as well as a fully managed Oracle Application Server environment.
- New Ant tasks for deploying applications to single OC4J instances or cluster-wide, as well as for configuring shared libraries, data sources and JMS destinations.
• New scripts for starting and stopping standalone OC4J servers

In this release full support for Apache Ant-based tasks for JSR 88 deployment and Web services development has been added to the out-of-the-box experience. Further, support for scripting using popular languages such as Groovy, Beanshell and Jython has been added, enabling the easy scripting of deployment and resource configuration tasks.

**New Middle-Tier Two-Phase Commit (2PC) Coordinator**

In this release, transaction coordination functionality is now located within OC4J itself, replacing the in-database coordination mechanism provided with earlier OC4J releases. The new architecture enables the middle-tier coordinator to be "heterogeneous", meaning that it can support any XA-compatible resource, not just Oracle databases.

The new middle tier coordinator provides the following additional features:

• Support for inter-positioning and transaction inflow

• Last Resource Commit Optimization, which allows a single non-XA-compliant resource to participate in an XA transaction

• Transaction recovery logging

**Enhanced Enterprise Messaging Capabilities**

OC4J continues to provide strong support for JMS with the introduction of the Oracle Enterprise Messaging Service (OEMS). This next-generation infrastructure is built Java 2 Enterprise Edition (J2EE) standards such as the Java Message Service (JMS) and the J2EE Connector Architecture (JCA) and is designed to reduce the time, cost, and effort required to build message-oriented integrated distributed applications.

OEMS allows you to choose from different message persistence models when architecting and integrating distributed applications. For a lightweight solution you can choose to persist messages in-memory or to the file system. For a more robust solution, Oracle Streams Advanced Queuing (AQ) in the Oracle Database can be used to store messages.

Customers who want to integrate existing messaging systems like WebSphereMQ, Tibco Enterprise JMS, or SonicMQ with an OC4J Message Drive Bean (MDB) can now accomplish this using the JMS Connector, a generic JCA based resource adapter implementation in the Oracle Application Server.

**Key Capabilities of Oracle Application Server Containers for J2EE 10g (10.1.3.1)**

With OC4J, you can take advantage of all, or even parts of the industry supported J2EE specification to create flexible, secure, portable, and high-performing applications.

**Support for Web Services**

Oracle Application Server Web Services provide a complete environment for developing, deploying and managing Web services. OC4J is the backbone of the
Web services platform, acting as the runtime engine to execute deployed Web services. Deployment and management operations are facilitated using Oracle Enterprise Manager, and UDDI services for publishing and searching for Web services are supported with the UDDI v3-compliant Oracle Application Server Service Registry. Rapid development of Web services is fulfilled with Oracle JDeveloper 10g, enabling modeling, development, deployment and testing to be carried out from one integrated environment.

**Developer Friendly, Enterprise Enabled**

OC4J is distributed in two different mediums - as a pure Java standalone distribution, and as component of the full Oracle Application Server product.

The pure Java standalone distribution is ideally suited for use in development and testing environments, as well as small scale production systems. It is distributed as a single zip file, enabling it to be easily downloaded and installed. The Web-based Application Server Control management console is provided with this distribution, a significant improvement over the previous OC4J standalone release, which was managed primarily through edits to XML configuration files.

In the standalone usage model, OC4J runs as a single process and HTTP requests are processed directly by OC4J's built in HTTP listener.

For large-scale production systems requiring the use of an HTTP server, single-sign-on capabilities for security, automatic process management for system availability, and browser based server management, OC4J is used within the full Oracle Application Server environment. In this usage model, OC4J is tightly integrated with the entire Oracle Application Server technology stack. Configuration, server management and application deployment is again performed using the Application Server Control console, which provides the ability to manage individual OC4J instances within a clustered environment. Incoming HTTP requests are serviced with Oracle HTTP Server, and processes are managed automatically by the high availability services provided in Oracle Application Server.

In both usage models the same OC4J component is used, providing a consistent level of support for J2EE and other features across the two environments. This enables developers to use the standalone distribution to develop and test applications, and then deploy completed applications to Oracle Application Server to take advantage of its advances capabilities, safe in the knowledge that the two environments are compatible with one another.

**Availability and Scalability**

OC4J provides component clustering, load-balancing, and application state replication for Web and EJB applications. It is tightly integrated with the high availability and process-monitoring infrastructure of Oracle Application Server so it can be deployed in mission critical environments with no single point of failure.

This release supports an extended set of mod_oc4j load balancing algorithms to route requests from Oracle HTTP Server to OC4J. The new algorithms provide more flexibility in the dispatching of requests from clients to waiting OC4J instances. Dynamic creation and removal of application mount points within mod_oc4j is also
Runtime Monitoring
The sub-systems of OC4J have been instrumented at key points using the Dynamic Monitoring Service (DMS) to provide runtime performance and operational data. DMS has been submitted as a Java Specification Request (JSR 138). The gathered metrics can be viewed directly using the Spy Web application supplied with OC4J, or viewed as an aggregate across all Oracle Application Server instances using the Oracle Enterprise Manager Web console.

Security
OC4J provides strong support for creating secure applications through it's implementation of the Java Authentication and Authorization Service (JAAS). This enables developers and administrators to secure the access to, and execution of, J2EE applications. To satisfy a growing requirement for single station user administration and single sign on across an organization, the Oracle implementation of JAAS can easily integrate applications with Oracle Application Server Single Sign On and Oracle Internet Directory.

Broad Platform Support
Oracle Application Server is available on the following operating systems and hardware platforms: Linux, Solaris, HP-UX, IBM AIX, Compaq Tru64 and Windows NT/2000. OC4J standalone is pure Java and is fully supported for production use on all of the above platforms.

Summary
Oracle Application Server provides the latest in grid computing and J2EE technology with its J2EE 1.4 compatible server. It delivers the performance, scalability, high availability, security, and ease-of-use needed in today’s agile, rapidly changing business world.