Oracle® Fusion Middleware

Glossary

12*c* (12.1.3)

E50329-02

July 2014

This glossary contains definitions of all terms defined in books included in the Fusion Middleware 12.1.3 library.

ORACLE®

Oracle Fusion Middleware Glossary 12*c* (12.1.3)

E50329-02

Primary Author: Oracle

Contents

1
Preface

This glossary contains definitions of all terms defined in books included in the Fusion Middleware 12.1.3 library.

Audience

This document is intended for administrators, developers, business users and end users working with Fusion Middleware products.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware12c (12.1.3) documentation set:

- *Understanding Oracle Application Development Framework*
- *Understanding Oracle Fusion Middleware*
- *Understanding Oracle SOA Suite*
- *Understanding Oracle Web Services Manager*
- *Understanding Oracle WebLogic Server*
- *Understanding Technology Adapters*
- *Understanding Web Services*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Glossary

A B C D E F G H I J K L M N O P Q R S T U V W X

3GPP

Java: Third Generation Partnership Project. A collaboration between groups of telecommunications associations, for the purpose of making a globally applicable third generation (3G) mobile phone system specification.

A

abstract

Java: A reserved Java keyword (one of 50) used to declare a class or method to be abstract.

abstract class

Java: A Java class that is declared to be abstract, meaning that it cannot be instantiated, but it can be subclassed.

abstract method

Java: A method that is declared without an implementation (without braces, and followed by a semicolon). For example: abstract void moveTo(double deltaX, double deltaY);

abstract schema

Java: In Java Persistence, the persistent schema abstraction (persistent entities, their state, and their relationships) over which queries operate.

abstract schema type

Java: In Java Persistence, the type to which the persistent property of an entity evaluates in the abstract schema.

Abstract Window Toolkit (AWT)

Java: A low-level api package in the Java language that provides graphical components, such as buttons, scrollbars, and panels. AWT is part of the Java Foundation Classes.

access control

Java: Access level modifiers determine whether other classes can use a particular field or invoke a particular method. There are two levels of access control: At the top level:

public, or package-private (no explicit modifier). At the member level: public, private, protected, or package-private (no explicit modifier).

Access Decision

WebLogic Server: Code that determines whether a subject has permission to perform a given operation on a WebLogic resource. The result of an Access Decision is to permit, deny, or abstain from making a decision. An Access Decision is a component of an Authorization provider.

action binding

JDeveloper/ADF: A binding for command components (such as buttons or links) to built-in or custom methods on the data control, or to built-in collection-level operations (such as Create, Delete, Next, or Previous). An action binding object encapsulates the details of how to invoke a method and what parameters (if any) the method is expecting.

action event

Java: An event that occurs when the user activates a JavaServer Faces component that implements the javax.faces.component.ActionSource interface. These components include buttons and hyperlinks. Action events and value-change events are types of application events.

action method

Java: In JavaServer Faces technology, a managed bean method that handles navigation processing.

activation

Java: The return of an enterprise bean by the EJB container from secondary storage to memory, caused by the invocation of one of the bean's business methods.

Active Data Service (ADS)

JDeveloper/ADF: A server-side push framework that allows you to provide real-time data updates for ADF Faces components. You bind ADF Faces components to a data source, and ADS pushes the data updates to the browser client without requiring the browser client to explicitly request it.

activity

SOA Suite: The building block of a BPEL process service component. Activities enable you to perform specific tasks within a BPEL process service component. Oracle BPEL Designer includes a set of activities that you drag into a BPEL process service component. You then double-click an activity to define its attributes (property values).

adapter data control

JDeveloper/ADF: A data control based on a framework that provides built-in support for features such as validation rules and list of value (LOV) objects. You can use the adapter framework to create a new data control type, or you can extend one of the existing data control types, such as the bean data control type.

addressing

See Web Services Addressing (WS-Addressing)

ADF application

JDeveloper/ADF: An application using one or more of the following high-level Oracle ADF technologies: ADF Business Components, ADF Model, ADF Controller, and ADF Faces. In contrast, a Fusion web application uses all of these technologies.

ADF binding

JDeveloper/ADF: An ADF application object that abstracts the details of accessing data from a data collection in a data control and of invoking its operations. ADF bindings are defined in ADF page definition files.

ADF binding container

JDeveloper/ADF: A request-scoped map of a page's bindings based on the page's page definition file. In a JSF application, the binding container is accessible during each page request using the EL expression #{bindings}.

ADF binding context

JDeveloper/ADF: A container object that holds a list of available data controls and data binding objects. The DataBindings.cpx file defines the binding context for the application.

ADF binding filter

JDeveloper/ADF: A servlet filter that ADF web applications use to preprocess any HTTP requests that may require access to the binding context.

ADF Business Components

JDeveloper/ADF: A framework that simplifies the development, delivery, and customization of business applications for the Java EE platform. You use ADF Business Components to define associations between entity objects, view objects, and application modules to reflect the foreign keys present in the underlying tables.

ADF Controller

JDeveloper/ADF: A mechanism that provides an enhanced navigation and state management model on top of JSF. This mechanism declaratively defines navigation using control flow rules.

ADF data control

JDeveloper/ADF: XML configuration files that describe a service. At design time, JDeveloper uses the metadata to declaratively bind an operation or data collection to a UI component.

ADF Data Visualization components

JDeveloper/ADF: ADF Faces components that are capable of rendering dynamic charts, gauges, and other graphics that provide a real-time view of underlying data.

ADF Data window

JDeveloper/ADF: One of the ADF application debugging windows. Displays a list of data for a given application object selected in the ADF Structure window.

ADF Desktop Integration

JDeveloper/ADF: Provides a framework for Oracle Application Development Framework (Oracle ADF) developers to extend the functionality provided by a Fusion web application to desktop applications. The ADF Desktop Integration framework

allows end users to avail themselves of Oracle ADF functionality when they are disconnected from their company network.

ADF Faces

JDeveloper/ADF: A set of standard JSF components that includes built-in AJAX functionality.

ADF Lifecycle Breakpoints window

JDeveloper/ADF: One of the ADF application debugging windows. Displays all ADF lifecycle execution points to set and check application lifecycle breakpoints. When the application is paused at an ADF lifecycle declarative breakpoint, an Execution Point icon appears next to the lifecycle phase in the ADF Lifecycle Breakpoints window.

ADF Model

JDeveloper/ADF: A module of Oracle ADF that implements service abstraction called the data control. Data controls enable a unified approach to bind any user interface to any business service, without the need to write code.

ADF region

JDeveloper/ADF: An ADF Faces component that renders the content of a bounded task flow.

ADF Security

JDeveloper/ADF: A framework that supports declarative, permission-based protection for ADF security-aware resources, such as ADF bounded task flows, top-level web pages that use ADF bindings, and attributes defined by ADF entity objects and their attributes. ADF Security is built on top of the Oracle Platform Security Services (OPSS) architecture, which in turn incorporates the Java Authentication and Authorization Service (JAAS) and Java EE container-managed security.

ADF skin

JDeveloper/ADF: A file that uses the format of CSS to specify CSS properties, ADF skin properties, global selectors, and selectors of the ADF skinning framework to customize the appearance of components that render in an ADF applications user interface.

ADF skinning framework

JDeveloper/ADF: The underlying APIs and parsers that enable the creation and registration of the ADF skins that generate the CSS to customize the appearance of ADF applications.

ADF Structure window

JDeveloper/ADF: One of the ADF application debugging windows (distinct from the JDeveloper Structure window). Displays the runtime structure of the ADF application. Works in combination with the ADF Data window to display a list of data for a given object selected in the ADF Structure window.

ADF task flow

JDeveloper/ADF: A set of ADF Controller activities, control flow rules, and managed beans that interact to allow a user to complete a task.

Adjudication provider

WebLogic Server: A WebLogic security provider that tallies the results that multiple Access Decisions return, resolves conflicts between the Access Decisions, and determines the final PERMIT or DENY decision.

administration port

WebLogic Server: A special, secure port that all WebLogic Server instances in a WebLogic domain use for administration traffic.

Administration Server

WebLogic Server: A WebLogic Server instance that provides a central point for managing a WebLogic domain. All other WebLogic Server instances in a domain are called Managed Servers.

ADS

JDeveloper/ADF: A server-side push framework that allows you to provide real-time data updates for ADF Faces components. You bind ADF Faces components to a data source, and ADS pushes the data updates to the browser client without requiring the browser client to explicitly request it.

Agent case

Identity Management: A repository for information gathered in the process of examining suspicious sessions and transactions.

Ajax

A development technique that uses asynchronous JavaScript and XML.

anchor color

JDeveloper/ADF: A category of global selector alias in an ADF skin that defines base colors for the ADF skin.

annotation

An annotation, in the Java computer programming language, is a form of syntactic metadata that can be added to Java source code. Classes, methods, variables, method parameters and packages may be annotated. Unlike Javadoc tags, Java annotations can be reflective in that they can be embedded in class files generated by the compiler and may be retained by the Java VM to be made retrievable at run-time.

annotations

Java: Annotations provide data about a program that is not part of the program itself.

anonymous access

In Java security, obtaining access to an resource without authentication.

Answer Logic

Identity Management: A mechanism to intelligently detect the correct answers in the OAAM challenge response process. It checks if the answers provided by the user matches closely to any provided during registration.

Apache HTTP Server

Apache HTTP Server is an open source web server originally derived from the National Center for Supercomputing Applications (NCSA).

applet

Java: A special kind of Java program that a browser enabled with Java technology can download from the internet and run. Applet is not used to describe JavaFX programs that run in a browser.

Application

Identity Management: A high-level container for organizing policies (and the Policy Model objects used to create them) that define the requirements for accessing this particular secure resource. An Application may correspond to a single deployed software application, a set of deployed software applications, or components of a software application (such as an Enterprise JavaBeans).

application assembler

Java: The company or person who receives application modules from component providers and may assemble them into a Java EE application.

application client

Java: A Java EE client that runs on a client machine and provides a way for users to handle tasks that require a richer user interface than can be provided by a markup language. Application clients directly access enterprise beans running in the business tier. An application client typically has a graphical user interface (GUI) created from the Swing or the Abstract Window Toolkit (AWT) API, but a command-line interface is also possible. As web client capabilities have increased, application clients are less commonly used.

application client container

Java: The container that manages the execution of application client components. Application clients and their container run on the client.

application component provider

Java: The company or person who creates web components, enterprise beans, or application clients for use in Java EE applications.

application configuration resource file

Java: In JavaServer Faces technology, an XML file named faces-config.xml that can be used to define page navigation rules and to configure beans and other custom objects, such as custom components.

Application Configurator

Identity Management: An administrative role with privileges to configure and manage Oracle Privileged Account Manager Console and servers.

application exception

Java: An exception thrown by an enterprise bean that signals an error in the business logic of an enterprise bean. If an application exception is thrown within a transaction, the EJB container does not roll back the transaction.

application failover

Coherence Suite: When an application component becomes unavailable for any reason, a copy of the failed object finishes the job. In hardware or other failures, the session state is available to other cluster members that can resume the work of the failed member.

application-managed entity manager

Java: In Java Persistence, an entity manager whose persistence context is not propagated to application components, so that the life cycle of javax.persistence.EntityManager instances is managed by the application.

application module

JDeveloper/ADF: A transactional component that Oracle ADF clients use to work with application data defined by Oracle ADF Business Components. It defines an updatable data model and top-level procedures and functions (called service methods) related to a logical unit of work. This unit of work is related to an end-user task.

application module configuration file

JDeveloper/ADF: The general term for the bc4j.xcfg configuration file. Contains metadata about ADF application module names, the database connection used by the application module, and the runtime parameters that the data model project developer has configured for the application module. The configuration allows the ADF application to interact with deployed application modules.

application module data control

JDeveloper/ADF: XML configuration file that describes a service generated from Oracle ADF Business Components. At design time, JDeveloper uses the metadata to declaratively bind an operation or data collection to a UI component.

application module data model

JDeveloper/ADF: Exposes view object instances in the ADF Business Component Model project that users will interact with through the web pages of an ADF application.

application module instance

JDeveloper/ADF: An Oracle ADF Business Components application module definition that has been checked out at runtime from the ADF application module pool for use by a client.

application module nesting

JDeveloper/ADF: Enables services exposed by one Oracle ADF application module to be consumed by multiple application modules. The application module data model may define composite services by nesting desired application modules under a root application module.

application module pool

JDeveloper/ADF: Runtime logic provided by the Oracle ADF Business Components framework that allocates an available unreferenced application module instance to an ADF client. If the client has previously requested an application module, then the application module pool will identify and allocate the instance already used by the client.

Application Overview Checklist

JDeveloper/ADF: Part of the Application Overview, it steps users through the building of a Fusion web application, according to Oracle recommended best practices.

application programming interface (API)

Java: Defines calling conventions by which an application program accesses the operating system and other services. A set of classes used by programmers to write applications that provide standard methods and interfaces and eliminate the need for programmers to reinvent commonly used code.

Application Protocol Data Unit (APDU)

Java: A communication mechanism used by SIM cards and smart cards to communicate with card reader software or a card reader device. In a TCK, a script that is sent to the test applet as defined by ISO 7816-4.

Application Role

Identity Management: Defines criteria for mapping users, groups, External Roles or other Application Roles to a particular Oracle Entitlements Server Application object. The Application Role can be granted to an external user, group, or role in an identity store, or another Application Role in the policy store.

application scope

Java: A bean scope that is shared across all users' interactions with a web application. Defined by CDI and JavaServer Faces technology.

application server-scoped Coherence cluster members

Coherence Suite: (For deployments on non-WebLogic Server application servers) With this configuration, all deployed applications in a container become part of one Coherence member. This configuration produces the smallest number of Coherence members in the cluster (one for each web container JVM) and, because the Coherence library is deployed in the container's class path, only one copy of the Coherence classes is loaded into the JVM.

application state management

JDeveloper/ADF: Runtime logic provided by Oracle ADF that supports multistep use cases in Fusion web applications. Oracle ADF application modules may passivate (store) their pending transaction state to an XML document, which is stored in the database.

application tier

Coherence Suite: Coherence client instances that are cluster members and are not responsible for cache storage.

assertion template

OWSM: An assertion template defines a policy assertion, which is the smallest unit of a policy that performs a specific action for the request and response operations. Assertion templates can be used to add assertions to web service policies, or to create new policies.

asynchronous method

Java: A session bean business method in which control is returned to the client by the EJB container before the method is invoked on the session bean instance. The client can then use the Java SE concurrency API to retrieve the result, cancel the invocation, and check for exceptions.

asynchronous web service

SOA Suite: Provides a response to an invocation that can take time to complete. The invoker of the request waits until the web service replies. Asynchronous messaging styles are useful for environments in which a service, such as a loan processor, can take a long time to process a client request.

atomic transaction

See Web Services Atomic Transaction

atomicity

Java: A state in which a particular operation is atomic. Atomicity of data updates guarantees that data are not corrupted in a power loss or other abrupt exit.

audit trail

SOA Suite: Displays execution details in Oracle Enterprise Manager about the business flow instance (for example, the activities in a BPEL process). You can scroll through the audit trail to check for errors and expand the payload links to view their contents at a given point in the flow.

Auditing provider

WebLogic Server: A WebLogic Security Framework provider that provides auditing services.

authentication

The process that verifies the identity of a user, device, or other entity in a computer system; the means by which communicating entities, such as client and server, prove to each other that they are acting on behalf of specific identities that are authorized for access. Authentication ensures that users are who they say they are.

WebLogic Server: The process of verifying the identity of a user, device, or other entity in a host system, often as a prerequisite to granting access to resources in a system. A recipient of an authenticated message can be certain of the message's origin (its sender). Authentication is presumed to preclude the possibility that another party has impersonated the sender.

Authentication provider

WebLogic Server: WebLogic Security Framework provider that enables WebLogic Server to establish trust by validating a user.

authorization

In security, the means by which interactions with resources are limited to collections of users or programs for the purpose of enforcing integrity, confidentiality, or availability constraints.

authorization constraint

Java: In Java EE security, the part of a security constraint that specifies whether authentication is to be used and names the roles authorized to perform the constrained requests.

autoboxing

Java: The automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes. For example, converting an int to an Integer, a double to a Double, and so on. If the conversion goes the other way, this is called unboxing.

Autolearning

Identity Management: A set of features in Oracle Adaptive Access Manager that dynamically profile behavior in real time. The behavior of users, devices, and locations is recorded and used to evaluate the risk of current behavior.

automatic timer

Java: An enterprise bean timer that is created upon the successful deployment of an enterprise bean that contains a method annotated with the java.ejb.Schedule or java.ejb.Schedules annotation.

B

back up

The process of creating exact duplicates of the files in your environment, so that, in the event of data loss or corruption, host failure, or media failure, you can use those copies to restore your environment. The two types of backup are offline backup and online backup. In offline backup, you must shut down the environment before backing up the files. In online backup, you do not shut down the environment before backing up the files.

backing bean

Java: In JavaServer Faces technology, a managed bean that is associated with the UI components used in a particular Facelets page.

backing bean scope

JDeveloper/ADF: ADF-defined scope in the page life cycle. An object in this scope is available from the time an HTTP request is sent until a response is sent back to the client.

Backus-Naur Form (BNF)

A notation that describes the syntax of high-level languages.

bc4j.xcfg

JDeveloper/ADF: The ADF application module configuration file. The bc4j.xcfg file contains metadata about ADF application module names, the database connection used by the application module, and the runtime parameters that the data model project developer has configured for the application module. The configuration allows the ADF application to interact with deployed application modules.

bean

Java: In the Java language, a bean is a reusable software component used to encapsulate many objects into a single object, so that it can be passed around as a single bean object instead of as multiple individual objects. See also JavaBeans. In CDI, a source of contextual objects that define application state and/or logic. A Java EE component is a bean if the life cycle of its instances can be managed by the container according to the lifecycle context model defined in the CDI specification.

bean-managed concurrency

Java: A form of concurrent access that allows clients to have full concurrent access to all the business and timeout methods in a singleton session bean. The developer of the singleton is responsible for ensuring that the state of the singleton is synchronized across all clients.

bean-managed transaction demarcation

Java: A transaction demarcation in which the code in the enterprise bean explicitly marks the boundaries of the transaction.

behavioral interface

Java: In JavaServer Faces technology, one of the interfaces that defines certain behavior for a set of components whose classes implement the interface. These interfaces are defined in the javax.faces.component package.

bidirectional

An entity relationship in which each of two entities has a relationship field or property that refers to the other entity.

big-endian

Java: A technique of storing multibyte data in which the high-order bytes come first. For example, given an 8-bit data item stored in big-endian order, the first bit read is considered the high bit.

binary compatibility

Java: Binary compatibility check mode verifies that a Java technology implementation undergoing compatibility testing and its referenced APIs are mutually binary compatible as defined in Chapter 13, Binary Compatibility, of The Java Language Specification. This assures that any application will run with any compatible API without any linkage error.. Also see source compatibility.

bind variable

JDeveloper/ADF: A placeholder variable in a SQL statement that must be replaced with a valid value before the statement can successfully execute.

binding component

SOA Suite: A component that establishes the connection between a SOA composite application and the external world. There are two types of binding components: Services: Provide the outside world with an entry point to the SOA composite application. References: Enable messages to be sent from the SOA composite application to external services in the outside world.

binding container

JDeveloper/ADF: A request-scoped map of a page's bindings based on the page's page definition file. In a JSF application, the binding container is accessible during each page request using the EL expression #{bindings}.

binding context

JDeveloper/ADF: A container object that holds a list of available data controls and data binding objects. The DataBindings.cpx file defines the binding context for the application.

BNF

Backus-Naur Form. A notation that describes the syntax of high-level languages.

boolean

Java: A reserved Java keyword used to declare a field that can store a Boolean value. Can also be used to declare a method's return value.

bootstrap user

Identity Governance: A default administrator who can create and assign users to Oracle Privileged Account Manager Admin Roles and can map users from the domain identity store to Oracle Privileged Account Manager Common Admin Roles.

bounded task flow

JDeveloper/ADF: A type of ADF task flow that has a single entry point and zero or more exit points. It contains its own set of private control flow rules, activities, and managed beans. A bounded task flow allows reuse, parameters, transaction management, and reentry. When dropped on a JSF page or page fragment, it is wrapped in an ADF Faces region component.

bounded type parameters

Java: In the Java generics feature, there may be times when you want to restrict the types that can be used as type arguments in a parameterized type. For example, a method that operates on numbers might only want to accept instances of Number or its subclasses. This is what bounded type parameters are for.

BPEL process

BPEL: A service component that integrates a series of business activities and services into an end-to-end business process flow.

break

Java: A reserved Java keyword used to resume program execution at the statement immediately following the current enclosing block or statement.

bundle patch

An iterative, cumulative patch that is issued between patch sets. Bundle patches usually include only fixes, but some products may include minor enhancements.

business event

SOA Suite: A message sent as the result of an occurrence or situation, such as a new order or completion of an order. You can raise business events when a situation of interest occurs. When an event is published, other applications can subscribe to it. Definitions for business events are published to the Event Delivery Network (EDN).

business flow instance

SOA Suite: Corresponds to an end-to-end business transaction. Business flows consist of a single SOA composite application or multiple SOA composite applications connected together to fulfill a specific business process.

business interface

Java: A means of client access to enterprise beans through a standard Java programming language interface that contains the business methods of the enterprise bean.

business logic

Java: The code that fulfills the purpose of a Java EE application, commonly implemented in enterprise beans.

business method

Java: A method that implements the business logic of an enterprise bean.

business process execution language (BPEL)

SOA Suite: Provides enterprises with an industry standard for business process orchestration and execution. Using BPEL, you design a business process that integrates a series of discrete services into an end-to-end process flow. This integration reduces process cost and complexity.

bytecode

Java: Machine-independent code generated by a Java compiler and executed by a Java Virtual Machine.

C

cache configuration file (coherence-cache-config.xml)

Coherence Suite: A file that specifies the various types of caches that can be used within a cluster. This file is also referred to as the cache configuration deployment descriptor.

cache store

Coherence Suite: An application-specific adapter used to connect a cache to an underlying data source. The cache store implementation accesses the data source by using a data access mechanism (for example, TopLink, JPA, application-specific JDBC calls). The cache store understands how to build a Java object using data retrieved from the data source, map and write an object to the data source, and erase an object from the data source.

CacheFactory

Coherence Suite: A Coherence class that is typically used to get and release an instance of a NamedCache object. This class includes methods to obtain a cluster object running Coherence services, shut down all clustered services, and to return an instance of a cache.

case

Java: A reserved Java keyword that is used to create individual cases in a switch statement.

catch

Java: A reserved Java keyword used to define an exception handler (a group of statements that are executed if an exception is thrown in the block defined by a preceding try keyword).

CDC

Java: Connected Device Configuration (CDC)

CDI

Java: Contexts and Dependency Injection for the Java EE Platform. A set of services that, used together, make it easy for developers to use enterprise beans with JavaServer Faces technology in web applications. Designed for use with stateful objects, CDI also has many broader uses, allowing developers flexibility to integrate various kinds of components in a loosely coupled but typesafe way.

certificate

Also called a digital certificate. An ITU x.509 v3 standard data structure that securely binds an identity to a public key. A certificate is created when an entity's public key is signed by a trusted identity, a certificate authority The certificate ensures that the entity's information is correct and that the public key actually belongs to that entity. A certificate contains the entity's name, identifying information, and public key. It is also likely to contain a serial number, expiration date, and information about the rights, uses, and privileges associated with the certificate. It also contains information about the certificate authority that issued it.

certificate authority

A trusted third party that certifies that other entitiesusers, databases, administrators, clients, serversare who they say they are. When it certifies a user, the certificate authority first seeks verification that the user is not on the certificate revocation list (CRL), then verifies the user's identity and grants a certificate, signing it with the certificate authority's private key. The certificate authority has its own certificate and public key which it publishes. Servers and clients use these to verify signatures the certificate authority has made. A certificate authority might be an external company that offers certificate services, or an internal organization such as a corporate MIS department.

CGI

Common Gateway Interface (CGI) is the industry-standard technique for transferring information between a Web server and any program designed to accept and return data that conforms to the CGI specifications.

char

Java: A reserved Java keyword used to declare a field that can store a 16-bit Unicode character. Can also be used to describe a method's return value.

character encoding

A mapping of a character set to units of a specific width. A character encoding also defines byte serialization and ordering rules. Many character sets have more than one encoding.

character set

A set of textual and graphic symbols, each of which is mapped to a set of nonnegative integers.

checked exception

Java: An event that occurs during the execution of a program and disrupts the normal flow of the program's instructions. A checked exception is one that a well-written program should anticipate and recover from. FILE NOT FOUND or NO PERMISSIONS TO READ FILE are examples of a checked exception.

ciphertext

Data that has been encrypted. Ciphertext is unreadable until it has been converted to plain text (decrypted) with a key. See decryption.

class

Java: A reserved Java keyword that defines the implementation of a particular kind of object.

class declaration

Java: A Java class declaration includes the following components: Modifiers such as public, private, and a number of others that you will encounter later. The class name, with the initial letter capitalized by convention. The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent. A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface. The class body, surrounded by braces, {}.

class method

Java: The Java programming language supports static methods, also called class methods. These methods can be used to access class variables.

class variable

Java: In Java, a class variable that is available to every instance of a particular class. Also called a static variable.

classpath

Java: The CLASSPATH variable is one way to tell applications, including the JDK tools, where to look for user classes.

cleartext

See plaintext.

client authentication

Authentication in which the web server authenticates the client by using the client's public key certificate. Client authentication is a more secure method of authentication than either HTTP basic authentication or form-based authentication. It uses HTTP over SSL (HTTPS).

client-side at-rule

JDeveloper/ADF: An at-rule that the ADF skinning framework passes directly to the user agent to evaluate. The user agent applies the style properties within the at-rule if the condition that the at-rule specifies is satisfied. Examples of this category of at-rule include @media, @keyframes, and @page.

cluster service

Coherence Suite: The cluster service is responsible for the detection of other cluster members, for detecting the failure of a cluster member, and for registering the availability of other services in the cluster. This service is automatically started by a Coherence cluster member; each cluster member always has exactly one service of this type running.

Coherence cache client

Coherence Suite: An application that joins a cluster to access, process, and update cached data. The application is a cluster member that is not responsible for data storage. Types of cache clients include data clients, real-time clients configured as an Extend/TCP client, and real-time clients configured as a compute client.

Coherence cache server

Coherence Suite: A Coherence cluster member that is responsible for storing and backing up cached data.

Coherence cluster

Coherence Suite: A set of Coherence instances (one instance per JVM, with one or more JVMs on each computer). The JVMs automatically join together to form the cluster.

Coherence cluster member

Coherence Suite: A Coherence JVM process that is part of a Coherence cluster. A Coherence cluster member is also referred to as a Coherence cluster node.

Coherence cluster node

Coherence Suite: See Coherence cluster member. Although Coherence cluster node may appear in the Coherence documentation, the preferred term is Coherence cluster member.

Coherence data grid

Coherence Suite: A system composed of multiple servers that work together to manage information and related operations (such as computations) in a distributed environment.

Coherence GoldenGate HotCache

Coherence Suite: A Coherence feature that allows third-party changes to the database to be propagated to objects in the Coherence cache. Coherence GoldenGate HotCache can be added to any Coherence application. Standard JPA is used to capture the mappings from database data to Java objects. The configuration can be captured in XML exclusively or in XML with annotations.

Coherence Incubator

Coherence Suite: A java.net project that contains example implementations (source code, binaries, and documentation) of advanced functionality built on top of Coherence. These examples are intended to: Illustrate how to implement bleeding-edge functionality using the publicly available Coherence releases, preferably the latest versions. Directly address the challenges and investment required when developers attempt to use or construct implementations of standard architectural patterns with Coherence for their applications. Produce a standard library of common enterprise and application pattern/framework implementations, documentation, and training material, to dramatically reduce or eliminate the effort required to adopt and integrate Coherence into nontrivial applications (that is, those that require more than "read caching").

Coherence quorum

Coherence Suite: The minimum number of service members that are required in a cluster before a service action is allowed or disallowed. Quorums are beneficial because they automatically provide assurances that a cluster behaves in an expected way when member thresholds are reached.

Coherence REST

Coherence Suite: Coherence REST provides easy access to Coherence caches and cache entries over the HTTP protocol. It is similar to Coherence*Extend, as it allows remote clients to access data stored in Coherence without being members of the cluster themselves. However, unlike Coherence*Extend, which is a proprietary protocol, Coherence REST uses HTTP as the underlying protocol and can marshal data in both JSON and XML representation formats.

Coherence Service Guardian

Coherence Suite: A mechanism that detects and attempts to resolve deadlocks in Coherence threads.

Coherence Well Known Addresses (WKA)

Coherence Suite: A feature that allows cluster members to discover and join a cluster using unicast instead of multicast. WKA is most often used when multicast networking is undesirable or unavailable in an environment or when an environment is not properly configured to support multicast. All cluster multicast communication is disabled if WKA is enabled.

Coherence*Extend

Coherence Suite: A Coherence feature that allows applications to use a Coherence cluster without becoming cluster members. Coherence*Extend consists of an extend client (Java, C++, or .NET) that connects to a cluster proxy service through a TCP-based protocol.

Coherence*Web

Coherence Suite: An HTTP session management module dedicated to managing session state in clustered environments. It brings Coherence data grid scalability, availability, reliability, and performance to in-memory session management and storage.

CohQL

Coherence Suite: A lightweight syntax (in the tradition of SQL) that is used to perform cache operations on a Coherence cluster. The language can be used either programmatically or from the command line.

collection-based components

JDeveloper/ADF: ADF Faces UI components that iterate through and display collections of structured data. Instead of containing a child component for each record to be displayed, and then binding these components to the individual records, these components are bound to a complete collection. They then repeatedly render one component (for example, an outputText component), by stamping the value for each record. Examples include table, tree, and carousel.

collection member declaration

Java: In Java Persistence, a declaration of an identification variable that uses the IN operator to access a particular member of a collection.

collection model

JDeveloper/ADF: An ADF object that extends the JSF DataModel class and adds on support for row keys and sorting. In the DataModel class, rows are identified entirely by index. This can cause problems when the underlying data changes from one request to the next. For example, a user request to delete one row may delete a different row when another user adds a row. To work around this, the CollectionModel is based on row keys instead of indexes.

compact domain

A WebLogic domain for Oracle Fusion Middleware products in which all or a subset of the application or service groups defined in the product templates are targeted to the Administration Server. Compact domains do not contain Managed Servers or clusters for the Oracle Fusion Middleware products. Also known as a single-instance domain.

Compatibility realm

WebLogic Server: Security realm that is the default (active) security realm if you are using Compatibility security. The Compatibility realm adapts your existing WebLogic Server 6.x Authentication and Authorization providers so that you can use them in WebLogic Server 7.x or later. The only security realm available in Compatibility security is the Compatibility realm.

Compatibility security

WebLogic Server: The capability to run security configurations from WebLogic Server 6.x in later releases of WebLogic Server. By using Compatibility security in WebLogic Server 7.x or later, you configure 6.x security realms; define users, groups, and ACLs; manage protection of user accounts; and install custom auditing providers. The only security realm available in Compatibility security is the Compatibility realm. The Realm Adapter providers in the Compatibility realm allow backward compatibility to the authentication and authorization services in 6.x security realms.

compensation

SOA Suite: Occurs when a BPEL process cannot complete a series of operations after some have completed, and the BPEL process service component must backtrack and undo the previously completed transactions.

component family

Java: In JavaServer Faces technology, a component or set of components that can be rendered by a renderer or set of renderers.

component-managed sign-on

Java: In Java EE security, a mechanism in which application component code manages access to an Enterprise Information System (EIS) resource by including code that performs the sign-on process to the EIS.

composite component

Java: In JavaServer Faces technology, a special type of template that acts as a component. A component is a piece of reusable code that behaves in a particular way. For example, an input component accepts user input. A component can also have validators, converters, and listeners attached to it to perform certain defined actions.

composite expression

Java: One or more expression language (EL) expressions separated or surrounded by text, which are evaluated from left to right.

composite sensor

SOA Suite: Implements trackable fields on messages in a SOA composite application. You define sensors on binding components and on service components that have subscribed to business events.

concurrent access

Java: Access by many clients to a single instance of a session bean at the same time. Concurrent access is a common use case for singleton session beans.

conditional branching

SOA Suite: Introduces decision points to control the flow of execution of a BPEL process service component. You can use switch, if, while, and repeatUntil activities to define conditional branching.

confidentiality

In security, the means used to ensure that information is made available only to users who are authorized to access it. This ensures that only authorized users can view sensitive data.

configuration override

OWSM: Ability to override an OWSM policy configuration property on a per-attachment basis.

connection filter

WebLogic Server: A programmable filter that WebLogic Server uses to determine whether the server should allow incoming connections from a network client. In addition to security policies that protect WebLogic resources based on user characteristics, you can add another layer of security by filtering based on network connections.

connector

Java: In the Java EE Connector architecture, a resource adapter.

Connector architecture

Java: A specification (Java EE Connector architecture) used by tools vendors and system integrators to create resource adapters that support access to enterprise information systems that can be plugged in to any Java EE product.

container

Java: The interface between a component and the low-level platform-specific functionality that supports the component. Before it can be executed, a web component, enterprise bean component, or application client component must be assembled into a Java EE module and deployed into its container.

container-managed concurrency

Java: A form of concurrent access in which the EJB container controls client access to the business methods of a singleton session bean.

container-managed entity manager

Java: In Java Persistence, an entity manager whose persistence context is automatically propagated by the container to all application components that use the javax.persistence.EntityManager instance within a single Java Transaction API (JTA) transaction.

container-managed sign-on

Java: In Java EE security, a mechanism in which an application component lets the container take the responsibility of configuring and managing access to an Enterprise Information System (EIS) resource.

container-managed transaction demarcation

Java: Transaction demarcation in which the EJB container sets the boundaries of the transactions.

content movement

The process of moving all or part of your environment. Content movement includes moving your environment from a test environment to a production environment.

context parameter

Java: Information provided to a web application that can be used by all components of the web application.

context path

Java: A concatenation of a slash (/) with the context root of a servlets web application.

context root

Java: The identifier of a web application in a Java EE server. A context root must start with a slash (/) and end with a string.

Contexts and Dependency Injection for the Java EE Platform (CDI)

Java: A set of services that, used together, make it easy for developers to use enterprise beans with JavaServer Faces technology in web applications. Designed for use with stateful objects, CDI also has many broader uses, allowing developers flexibility to integrate various kinds of components in a loosely coupled but typesafe way.

continuous query

Coherence Suite: A feature that combines a query result with a continuous stream of related events to maintain an up-to-date query result in real time. This capability is called continuous query, because it has the same effect as if the desired query had zero latency and the query were being executed several times every millisecond.

control binding

JDeveloper/ADF: A set of value bindings and action bindings that appears in a page definition file. The control bindings are distinct from the iterator bindings that appear in the executables section of the page definition file.

control flow

JDeveloper/ADF: An ADF Controller activity that enables navigation between other activities in an ADF task flow. The control flow links one activity to another in a task flow.

control hint

JDeveloper/ADF: ADF Business Components metadata that defines label text, tooltip, and format mask hints for entity object and view object attributes. Control hints that you define on the business service layer can be used by UI components in ADF clients.

conversation scope

Java: In CDI, a user's interaction with a JavaServer Faces application, within explicit developer-controlled boundaries that extend the scope across multiple invocations of the JavaServer Faces lifecycle. All long-running conversations are scoped to a particular HTTP servlet session and may not cross session boundaries.

conversational state

Java: The state of a unique client/bean session, as represented by the instance variables of a stateful session bean.

cookie parameter

Java: A JAX-RS request parameter that extracts information from the cookies declared in cookie-related HTTP headers. It is specified by using the javax.ws.rs.CookieParam annotation.

correlation sets

SOA Suite: Correlate asynchronous messages based on message body contents. You can use correlation sets to identify asynchronous messages to ensure that asynchronous callbacks locate the appropriate client.

CPU

The name of the program that delivers security patch updates (SPUs). The program is Oracle's program for quarterly release of security fixes. Patches released as part of this program may be patch set updates, security patch updates, and bundle patches. Regardless of the patch type, the patches are cumulative.

crawler

The software agent used by a search engine to retrieve content for indexing.

credential

An object that contains or references security attributes used to authenticate a principal. A principal acquires a credential upon authentication or from another principal that allows its credential to be used.

credential delegation

OWSM: A mechanism that provides services with the ability to access another service or server in order to complete a client request. In order to establish such a connection with the Kerberos protocol, for example, Kerberos requires the first service to be authenticated to the second service or server using the client's user account and authority level.

Credential Mapping provider

WebLogic Server: A WebLogic Security Framework provider that is used to provide credential mapping services and bring new types of credentials into the WebLogic Server environment.

Criteria API

Java: A Java programming language API that is used to create typesafe queries to query entities and their relationships.

Critical Patch Update (CPU)

The name of the program that delivers security patch updates (SPUs). The program is Oracle's program for quarterly release of security fixes. Patches released as part of this program may be patch set updates, security patch updates, and bundle patches. Regardless of the patch type, the patches are cumulative.

cross-component wiring

A method for wiring Fusion Middleware components. It automates the wiring process, and provides the ability to diagnose wirings after they are established.

Cross-Domain Single Sign-On

WebLogic Server: WebLogic Server security feature that allows users to authenticate once but access multiple applications, even if these applications reside in different DNS domains. You can use this feature to construct a network of affiliates or partners that participate in a Single Sign-On domain.

cryptography

The protection of information by transforming it (encrypting) into an unreadable format. See encryption.

custom security provider

WebLogic Server: A security provider (written by third-party security vendors or security developers) that can be integrated into the WebLogic Security Service. Custom security providers are implementations of the Security Service Provider Interfaces (SSPIs) and are not supplied with the Oracle WebLogic Server product.

D

data client

Coherence Suite: Data clients are Extend clients that can access (put, get, query) data in the cluster and also make invocation service requests using standard Coherence APIs.

data control

JDeveloper/ADF: XML configuration files that describe a service. At design time, JDeveloper uses the metadata to declaratively bind an operation or data collection to a UI component.

data control definition file

JDeveloper/ADF: A file that contains the definitions for a project's data controls. Typically, it is named DataControls.dcx.

data control structure file

JDeveloper/ADF: An XML file that provides metadata for one of the objects that is encompassed by a data control.

Data Controls panel

JDeveloper/ADF: A panel in JDeveloper that lists all the data controls that have been created for the application's business services and exposes all the collections (row sets of data objects), methods, and built-in operations that are available for binding to UI components.

data integrity

In security, the means used to prove that information has not been modified by an entity other than the source of the information.

data-model event

Java: In JavaServer Faces technology, an event that occurs when a new row of a javax.faces.component.UIData component is selected.

data source type

The grouping of data sources based on the protocol used to access the data.

data tier

Coherence Suite: Cache server instances dedicated to caching application data and storing a replicated session state.

database delegator

WebLogic Server: An intermediary class that mediates initialization calls between a security provider and the security provider's database.

Database Management System Authentication provider

WebLogic Server: A WebLogic Security Framework provider that accesses user, password, group, and group membership information stored in databases for authentication purposes. Optionally, WebLogic Server can be used to manage the user, password, group, and group membership information.

declarative security

Java: A type of Java EE security that expresses an application component's security requirements by using either deployment descriptors or annotations.

declarative SQL mode

JDeveloper/ADF: The preferred way of creating view objects in the Oracle ADF Business Components project. At design time, Oracle ADF developers can create entity-based view objects that contain no SQL statements. This mode provides an alternative to creating view objects that specify a SQL statement.

decoding

Java: In JavaServer Faces technology, converting incoming request parameters to the local value of a component.

decorator

Java: In CDI, a Java class that performs cross-cutting tasks associated with method invocation and with the life cycles of beans, but that does not perform any business logic. A decorator is annotated javax.decorator.Decorator and has a corresponding decorators element in the beans.xml file.

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

default

Java: A reserved Java keyword that can optionally be used in a switch statement to label a block of statements to be executed if no case matches the specified value.

default realm

WebLogic Server: The active security realm. In WebLogic Server you can configure multiple security realms in a WebLogic domain; however, only one can be the default (active) security realm.

deferred evaluation expression

Java: An expression that can be evaluated later by the underlying technology using the expression language.

dehydration store

BPEL: A database that automatically maintains long-running asynchronous processes and their current state information while they wait for asynchronous callbacks. Storing the process in a database preserves the process and prevents any loss of state or reliability if a system shuts down or a network problem occurs.

dependency injection

Java: In CDI, the ability to inject components into an application in a typesafe way, including the ability to choose at deployment time which implementation of a particular interface to inject. In the Java EE platform, dependency injection can be applied to all resources that a component needs, effectively hiding the creation and lookup of resources from application code.

dependent scope

Java: In CDI, the default scope if none is specified; it means that an object exists to serve exactly one client (bean) and has the same life cycle as that client (bean).

deployer

Java: The company or person who configures and deploys application clients, web applications, Enterprise JavaBeans components, and Java EE applications.

deployment

Java: The process of installing Java EE application components in the Java EE containers.

deployment descriptor

Java: An XML document with an .xml extension that describes the deployment settings of an application, a module, or a component. At runtime, the Java EE server reads the deployment descriptor and acts upon the application, module, or component accordingly.

deployment descriptor properties

SOA Suite: BPEL process properties used at runtime by Oracle WebLogic Server, Oracle Enterprise Manager, or both. There are two types: Configuration Partner link binding

deprecated

Java: Occasionally an API is deprecated, meaning it can no longer be used. This should be done only in very rare circumstances. The @Deprecated annotation is provided for this purpose.

derivative color

JDeveloper/ADF: A category of global selector alias that defines color properties for use in an ADF skin. These global selector aliases derive the hue value for their color properties from anchor colors.

destination

Java: In the JMS API, the object a client uses to specify the target of messages it produces and the source of messages it consumes. A destination can be either a queue or a topic.

device fingerprinting

Identity Management: A process by which OAAM captures information about the device being used in an access request or transaction. A device is a computer, laptop computer, PDA, cell phone, kiosk, or other web-enabled device. Device fingerprinting data is gathered from multiple sources including secure cookies, flash shared objects, user agent strings, custom agents, mobile applications, browser header data, and other data.

diagnostic context

WebLogic Server: Contextual information added by the WebLogic Diagnostics Framework (WLDF) to all requests when they enter the WebLogic Server system. You can use this information to reconstruct transactional events and reconstruct a thread of execution from request to response.

Diagnostic Framework

Diagnostic Framework aids in detecting, diagnosing, and resolving problems. It targets critical errors such as those caused by code bugs, metadata corruption, customer data corruption, deadlocked threads, and inconsistent state.

diagnostic patch

An interim patch created specifically to diagnose a problem and not to fix a bug.

diamond

Java: When creating instances of generic objects in the Java language, you can replace the type arguments required to invoke the constructor of a generic class with an empty set of type parameters () as long as the compiler can infer the type arguments from the context. This pair of angle brackets is informally called the diamond. Note that the diamond is not a Java operator. Referring to it as an operator (as in "the diamond operator") is incorrect. It is acceptable to say "diamond notation" or simply "the diamond."

digest authentication

A type of authentication in which, instead of sending user passwords over the network, the client sends a one-way cryptographic hash of the password and additional data. Although passwords are not sent on the wire, digest authentication requires that clear-text password equivalents be available to the authenticating container so that it can validate received authenticators by calculating the expected digest.

digital certificate

See certificate.

digital signature

A value encrypted by a server using a private key, which a client can then decrypt using the server's public key and compare to its own computed value. If the two values match, then the client can trust that the signature is authentic, because only the private key could have been used to produce such a signature.

direct binding

SOA Suite: Enables Java clients to directly invoke composite services, bypassing the intermediate conversion to XML required with web service binding.

direct policy attachment

OWSM: A mechanism for associating an OWSM policy with a single policy subject.

disaster recovery

The ability to safeguard against natural or unplanned outages at a production site by having a recovery strategy for moving applications and data to a geographically separate standby site.

disconnected property

JDeveloper/ADF: Property on an ADF Faces UI component that can be set on the client, but is not propagated back to the server. These properties have a life cycle on the client that is independent of the life cycle on the server.

discriminator column

Java: In Java Persistence, for the single table per class inheritance hierarchy, a column that contains a value that identifies the subclass to which the instance represented by the row belongs.

disposer method

Java: In CDI, a method that removes an object that was generated by a producer method.

distributed (or partitioned) cache

Coherence Suite: A collection of data that is distributed (or partitioned) across any number of cluster members such that exactly one member in the cluster is responsible for each piece of data in the cache, and the responsibility is distributed (or load-balanced) among the cluster members.

distributed application

Java: A Java EE multitiered application that is distributed over three locations: client machines, the Java EE server machine, and the database or legacy machines at the back end.

DistributedCache service

Coherence Suite: A service that allows cluster members to distribute (partition) data across the cluster so that each piece of data in the cache is managed (held) by only one cluster member.

distribution

A downloadable archive with an installer that installs a predefined set of Oracle Fusion Middleware products and feature sets; customers can obtain distributions through the Oracle Software Delivery Cloud, the Oracle Technology Network (OTN), or My Oracle Support.

do

Java: A reserved Java keyword used in conjunction with while to create a do-while loop, which executes a block of statements associated with the loop and then tests a Boolean expression associated with the while.

document root

Java: The top-level directory of a web module, where XHTML pages, client-side classes and archives, and web resources are stored.

domain

Java: A set of one or more Java EE server instances managed by one administration server. The domain is commonly associated with port numbers for both the server and the administration server.

Domain Template Builder

A tool for creating a template of a WebLogic domain that has been customized and that you want to propagate throughout a target environment.

domain value maps

SOA Suite: A set of value mappings that enables you to associate values from one application with values from another. For example, one value can represent a city with a long name (Boston), while another value can represent a city with a short name (BO). In such cases, you can directly map the values by using domain value maps.

double

Java: A reserved Java keyword used to declare a field that can hold a 64-bit double precision IEEE 754 floating-point number. Can also be used to declare a method's return type.

durable process

SOA Suite: A type of BPEL process that incurs one or more dehydration points in the database during execution. Dehydration is triggered by the following activities. Receive activity OnMessage branch in a pick activity OnAlarm branch in a pick activity Wait activity Instances of durable processes can be saved in-flight (whether they complete normally or abnormally). These processes are typically long-living and initiated through a one-way invocation. Because of out-of-memory and system downtime issues, durable processes cannot be memory-optimized. The asynchronous process you design in Oracle JDeveloper is an example of both a transient and durable process.

durable subscription

Java: In the JMS API, a mechanism that allows clients in a publish/subscribe messaging system to receive messages sent while the subscribers are not active. Without a durable subscription, a client must remain active to consume messages. Durable subscriptions provide the flexibility and reliability of queues but still allow clients to send messages to many recipients.

dynamic query

Java: In Java Persistence, a query that is defined directly within an application's business logic.

dynamic server

WebLogic Server: A server instance that is generated by WebLogic Server when it creates a dynamic cluster. Configuration is based on a shared server template.

E

eager initialization

Java: The initialization of a singleton session bean upon application startup, before the EJB container delivers client requests to any enterprise beans in the application. Eager initialization is specified by the javax.ejb.Startup annotation.

EAR file

Java: A file with the suffix .ear, for Enterprise Archive, that contains a Java EE application. The application may contain application clients, web modules, and EJB modules and, optionally, deployment descriptors.

EAR-Scoped Coherence cluster members

Coherence Suite: For deployments on non-WebLogic Server application servers: with this configuration, all deployed applications within each EAR file become part of one

Coherence member. This configuration produces one Coherence member for each deployed EAR file. Because the Coherence library is deployed in the application's classpath, only one copy of the Coherence classes is loaded for each EAR file.

EclipseLink

Toplink: An open-source persistence framework from the Eclipse Foundation. EclipseLink includes development tools and runtime capabilities for providing transformation, mapping, data binding, and persistence operations in Java SE and Java EE environments. EclipseLink provides support for a number of persistence standards, plus extensions to those standards. The standards include the Java Persistence API (JPA), Java API for XML Binding (JAXB), Java Connector Architecture (JCA), and Service Data Objects (SDO) The EclipseLink project is led by Oracle, and EclipseLink provides the core functionality in Oracle TopLink.

EclipseLink DBWS

Toplink: A development tool and a runtime for providing Java EE-compliant, client-neutral access to relational database artifacts through a web service. The development tool, DBWS Builder, is a command-line utility that generates the necessary deployment artifacts. (DBWS Builder is integrated into Eclipse Dali Java Persistence toolset and into Oracle JDeveloper.) The runtime provider component takes a service descriptor (along with related deployment artifacts) and realizes it as a JAX-WS 2.0 Web service. The runtime uses EclipseLink to bridge between the database and the XML SOAP messages used by web service clients.

EclipseLink EIS

Toplink: A facility for enabling the use of data stores through Java Connector Architecture (JCA) resource adapters. Using XML metadata, the interactions and their exchanged data are configured and mapped onto a domain model. The interactions data can be mapped from either the Common Client interface (CCI) or by using XML schemas. This use is intended for nonrelational data stores where no JDBC or SQL access is provided.

EclipseLink JAXB

Toplink: The EclipseLink implementation of the Java Architecture for XML Binding (JAXB) specification. It includes EclipseLink extensions to the standard, such as XPath-based mappings and a mapping file. EclipseLink JAXB is the JAXB implementation used by Oracle TopLink.

EclipseLink JPA

Toplink: The EclipseLink implementation of the Java Persistence API (JPA) specification, plus EclipseLink extensions to the standard. EclipseLink JPA is the Reference Implementation of the JPA 2.0 and JPA 2.1 specifications, and the JPA implementation used by Oracle TopLink.

EclipseLink Mapping Workbench

Toplink: A graphical tool (graphical user interface) for configuring descriptors and for mapping between an object model and a data model in an EclipseLink project, using native TopLink. (The Workbench cannot be used for mapping the Java Persistence API (JPA).) EclipseLink Mapping Workbench is a separate component available in EclipseLink distributions. EclipseLink Mapping Workbench is based on Oracle TopLink Mapping Workbench, which is no longer available. However, native TopLink mapping projects are still supported in Oracle JDeveloper.

EclipseLink MOXy

Toplink: EclipseLink Java Architecture for XML Binding (JAXB), and support for JavaScript Object Notation (JSON) as an alternative to XML in JAXB-type operations. EclipseLink supports all object/XML options when reading and writing JSON. MOXy also includes support for the native EclipseLink object/XML API.

EclipseLink Service Data Objects (SDO)

Toplink: The reference implementation of the Service Data Objects (SDO) 2.1.1 specification, including a programming architecture and API for heterogeneous data access. EclipseLink SDO is implemented as a thin layer on EclipseLink's object/XML layer, which is also the base for EclipseLink JAXB. EclipseLink SDO can be used alone or as a means to expose POJOs as Data Objects, so components like EclipseLink JPA and EclipseLink JAXB can be used.

EJB component

Java: Enterprise JavaBeans component. A server-side component that encapsulates the business logic of an application. Also called an enterprise bean, an EJB component is either a session bean or a message-driven bean.

EJB container

Java: The container that manages the execution of enterprise beans for Java EE applications. Enterprise beans and their container run on the Java EE server.

EJB module

Java: A module that contains class files for enterprise beans and, optionally, an EJB deployment descriptor. EJB modules are packaged as Java Archive (JAR) files with a .jar extension.

EL

Java: expression language. A simple language designed to enable the presentation layer in web applications to communicate with the application logic. The EL is used by both the JavaServer Faces and the JavaServer Pages technologies.

Elastic Data

Coherence Suite: Elastic Data is a Coherence feature which is used to seamlessly store data across memory and disk-based devices. This feature is especially tuned to take advantage of fast disk-based devices such as Solid State Disks (SSD) and enables near memory speed while storing and reading data from SSDs.

else

Java: A reserved Java keyword that is used in conjunction with if to create an if-else statement, which tests a Boolean expression.

embedded LDAP server

WebLogic Server: A server included with WebLogic Server that is the default store for user, group, security role, security policy, and credential information.

encoding

Java: In JavaServer Faces technology, converting the current local value of a component into the corresponding markup that represents it in the response.

encryption

The process of converting a message thereby rendering it unreadable to any but the intended recipient. Encryption is performed by converting data into code that cannot be understood by unauthorized people or systems. There are two main types of encryption: public-key encryption (also known as asymmetric-key encryption) and symmetric-key encryption.

enterprise application

A distributed, transactional, and portable application that provides the business logic for an enterprise.

enterprise bean

Java: A server-side component that encapsulates the business logic of an application. Also called an Enterprise JavaBeans (EJB) component. An enterprise bean is either a session bean or a message-driven bean.

enterprise deployment

An Oracle best practices blueprint based on proven Oracle high availability and security technologies and recommendations for a particular Oracle product, such as Oracle Fusion Middleware. The best practices described in these blueprints span all Oracle products across the entire technology stack, such as Oracle Database, Oracle Fusion Middleware, and Oracle Enterprise Manager Fusion Middleware Control. An enterprise deployment considers various business service-level agreements (SLA) to make high availability best practices as widely applicable as possible. It provides highly resilient, lower cost infrastructure. It ensures that the high availability architecture is optimally configured to perform and scale to business needs. It enables control over the length of time to recover from an outage and the amount of acceptable data loss from a natural disaster.

Enterprise JavaBeans (EJB) component

Java: A server-side component that encapsulates the business logic of an application. Also called an enterprise bean, an EJB component is either a session bean or a message-driven bean.

entity

Java: In Java Persistence, a lightweight persistence domain object. Typically, an entity represents a table in a relational database, and each entity instance corresponds to a row in that table.

entity association

JDeveloper/ADF: An Oracle ADF Business Components object that defines the relation between two ADF entity objects based on the attributes on each side of the associated entity objects. Associations are useful to access the destination entities from a source entity. At runtime, the entity object uses the association to automate working with related sets of entities.

entity object

JDeveloper/ADF: An ADF Business Components object that represents a row in a database table and simplifies modifying its data by handling all data manipulation language (DML) operations for you. It can encapsulate business logic to ensure that your business rules are consistently enforced. You associate an entity object with others to reflect relationships in the underlying database schema to create a layer of business domain objects to reuse in multiple applications.

entity provider

Java: A JAX-RS interface that supplies mapping services between representations and their associated Java types. The two types of entity providers are javax.ws.rs.ext.MessageBodyReader and javax.ws.rs.ext.MessageBodyWriter.

entity variable

SOA Suite: Used with an Oracle Application Development Framework (ADF) Business Component data provider service using service data object (SDO)-based data. The entity variable enables you to specify BPEL data operations to be performed by an underlying data provider service. The data provider service performs the data operations in a data store behind the scenes and without use of other data store-related features provided by Oracle BPEL Process Manager (for example, the database adapter). This action enhances Oracle BPEL Process Manager runtime performance and incorporates native features of the underlying data provider service during compilation and runtime.

entry aggregator

Coherence Suite: An entry aggregator performs operations against a subset of entries to obtain a single result. Aggregations are performed in a map-reduce manner with local results on each cache server sent to the originating client, which then performs the final reduce step.

entry processor

Coherence Suite: An entry processor can update cache entries in multiple caches within a single process or processAll operation. The caches must be managed by the same service, and the entries must be located in the same partition. An entry processor facilitates a highly efficient, lockless, and highly scalable compute environment where compute is moved to the data rather than moving data to a client application.

enum

Java: A reserved Java keyword used to declare an enumerated type.

enumerated type (enum)

Java: A type whose field consists of a fixed set of constants. Also called an enum.

environment

The set of resources made available to the user of a system, as distinct from the system itself.

event delivery network (EDN)

SOA Suite: Stores the published business events. The EDN runs within every SOA instance. Raised events are delivered by EDN to the subscribing service components.

event interceptor

Coherence Suite: An event interceptor explicitly defines which events to receive and what action, if any, to take. Any number of event interceptors can be created and registered for a specific cache or for all caches managed by a specific partitioned service. Multiple interceptors that are registered for the same event type are automatically linked together and executed in the context of a single event.

event root component

JDeveloper/ADF: An ADF Faces UI component that determines boundaries on the page, and so allows the lifecycle to run just on components within that boundary.

Certain components are inherently event root components, such as the popup. When an event happens within a popup, only the popup and its children are rerendered, and not the whole page. Additionally, certain events indicate event root components. For example, the disclosure event sent when expanding a showDetail component indicates that the showDetail component is an event root.

executable binding

JDeveloper/ADF: One of a category of ADF bindings that includes iterator bindings, which simplify the building of user interfaces that allow scrolling and paging through collections of data and drilling down from summary to detail information, bindings that allow searching and nesting a series of pages within another page, and bindings that cause operations to occur immediately.

expression language (EL)

Java: A simple language designed to enable the presentation layer in web applications to communicate with the application logic. The EL is used by both the JavaServer Faces and the JavaServer Pages technologies.

Extend client

Coherence Suite: Runs outside the cluster and communicates with an extend proxy service running in the cluster hosted by one or more storage-disabled cache servers. An Extend client retrieves a Coherence clustered service using a cache factory. After a service is obtained, a client uses the service in the same way as if it were part of the Coherence cluster. Operations sent to a remote cluster member is transparent to the client application.

Extend client tier

Coherence Suite: One or more computers running Coherence*Extend clients.

extends

Java: A reserved Java keyword used in a class declaration to specify the superclass; used in an interface declaration to specify one or more superinterfaces.

Extensible Markup Language (XML)

Java: A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C and in several other related specifications. These are all open standards.

F

Facelets

Java: The view declaration language for JavaServer Faces technology. Facelets is a powerful but lightweight page declaration language that is used to build JavaServer Faces views using HTML-style templates and to build component trees.

failover

WebLogic Server: The ability to reconfigure a computing system to utilize an alternate active component when a similar component fails.

Fast Infoset

Compressed binary encoding format that provides a more efficient serialization than the text-based XML format. Fast Infoset optimizes both document size and processing

performance.The Fast Infoset specification, ITU-T Rec. X.891 and ISO/IEC 24824-1 (Fast Infoset) is defined by both the ITU-T and ISO standards bodies.

fault policy bindings file

SOA Suite: Associates the policies defined in the fault policy file.

fault policy file

SOA Suite: Defines fault conditions and their corresponding fault recovery actions. Each fault condition specifies a particular fault or group of faults, which it attempts to handle, and the corresponding action for it. A set of actions is identified by an ID in the fault policy file.

feature set

A logical grouping of Oracle Fusion Middleware features that can be used as a building block for distributions.

federated identity management

Access Management: The agreements, standards, and technologies that make identity and entitlements portable across autonomous domains.

field

Java: A class, interface, or enum with an associated value.

filter

Java: An object that can transform the header and content (or both) of a servlet request or response.

final

Java: A reserved Java keyword used to define an entity once, and that entity cannot be changed nor derived from later. More specifically: a final class cannot be subclassed, a final method cannot be overridden, and a final variable can occur only once as a left-hand expression.

finally

Java: A reserved Java keyword used to define a block of statements for a block defined previously by the try keyword.

float

Java: A reserved Java keyword used to declare a field that can hold a 32-bit single precision IEEE 754 floating-point number.

fork/join framework

Java: An implementation of the ExecutorService interface that helps you take advantage of multiple processors. It is designed for work that can be broken into smaller pieces recursively. The goal is to use all the available processing power to enhance the performance of your application.

form-based authentication

A type of authentication that allows the developer to control the look and feel of the login authentication screens by customizing the login screen and error pages that an HTTP browser presents to the end user.

form parameter

Java: A JAX-RS request parameter that extracts information from a request representation that is of the MIME media type application/x-www-form-urlencoded and that conforms to the encoding specified by HTML forms. This parameter is useful for extracting information sent by POST in HTML forms.

Fusion Middleware Configuration Wizard

A tool that creates the appropriate directory structure for a WebLogic domain, a domain configuration file (config.xml), and scripts you can use to start the servers in the domain.

Fusion Middleware Control

A web browser-based, graphical user interface that you can use to monitor and administer your Oracle Fusion Middleware environment.

Fusion web application

JDeveloper/ADF: An application that uses the entire Oracle ADF web stack: ADF Business Components, ADF Model, ADF Controller, and ADF Faces.

G

GAR file

Coherence Suite: grid archive file. The name of the application module supported by managed Coherence servers. The GAR file contains the artifacts of a Coherence application and includes a deployment descriptor. A GAR file is deployed and undeployed in the same way as standard Java EE modules and the application lifecycle is decoupled from the container.

garbage collection

Java: A process by which dynamically allocated memory storage is automatically reclaimed during the execution of a program.

GCF

Java: Generic Connection Framework

generic type

Java: A generic class or interface that is parameterized over types.

generics

Java: Enable types (classes and interfaces) to be parameters when defining classes, interfaces, and methods. Much like the more familiar formal parameters used in method declarations, type parameters let you reuse the same code with different inputs. The difference is that the inputs to formal parameters are values, whereas the inputs to type parameters are types.

global policy attachment

OWSM: A mechanism for associating a policy with a range of policy subjects of the same type, regardless of deployment state. Policies are attached globally using policy sets, which can contain multiple policy references.

grantee

Identity Governance: A user, group, or role that was granted access to a privileged account.

GridLink data source

WebLogic Server: A means to connect WebLogic Server to an Oracle Database service, which may include multiple Oracle RAC clusters.

group

Java: In Java EE security, a set of authenticated users, classified by common traits, defined in the Java EE server.

GSM

Java: Global System for Mobile Communications (GSM)

H

header parameter

Java: A JAX-RS request parameter that extracts information from the HTTP headers. It is specified by using the javax.ws.rs.HeaderParam annotation.

heap

Java: A common pool of free memory usable by a computer program. A heap is part of the computer's memory used for dynamic memory allocation, in which blocks of memory are used in an arbitrary order.

high availability

An architecture that ensures that users can access a system without loss of service. Deploying a high availability system minimizes the time when the system is down (unavailable) and maximizes the time when it is running (available).

host name verifier

WebLogic Server: A class that ensures the host name in the URL to which a client connects matches the host name in the digital certificate that the server sends back as part of the SSL connection.

HTTP Analyzer

JDeveloper/ADF: Allows users to examine the content of HTTP request/response package pairs from within JDeveloper.

HTTP basic authentication

A type of authentication in which the server requests a user name and password from the web client and verifies that the user name and password are valid by comparing them against a database of authorized users in the specified or default realm.

I

idempotent activity

SOA Suite: An activity that can be safely retried. Idempotent activities are applicable to both durable and transient processes. You can manage idempotence at the operation

level of a partner link. You define applicable operations as idempotent, which enables these operations to be called multiple times.

identification

In security, the process that enables recognition of an entity by a system.

identification variable

Java: In Java Persistence API, an identifier that is declared in the FROM clause of a query.

Identity Assertion provider

WebLogic Server: WebLogic Security Framework provider that performs perimeter authentication, which is a special type of authentication using tokens. Identity Assertion providers also allow WebLogic Server to establish trust by validating a user. Thus, the function of an Identity Assertion provider is to validate and map a token to a user name.

identity federation

Access Management: The linking of two or more accounts a principal may hold with one or more identity providers or service providers within a given circle of trust.

identity propagation

Identity Governance: Process in which the OPSS Trust Service Asserter examines and validates a token, and then asserts that the identity performing a RESTful call against the Oracle Privileged Account Manager server is the one contained in the token.

IMlet

Java: An application written for IMP-NG. An IMlet is similar to an MIDP 2.0 MIDlet, except that an IMlet cannot refer to MIDP classes that are not part of IMP(-NG). An IMlet can use only the APIs defined by the IMP(-NG) and CLDC specifications.

immediate evaluation expression

Java: In Expression Language, an expression that is evaluated at once by the underlying technology, such as JavaServer Faces technology.

implicit navigation

Java: A set of built-in rules that allow developers to quickly configure JavaServer Faces page navigation. The built-in rules reduce the manual configuration process for applications.

inbound transaction

SOA Suite: A transaction initiated by an inbound adapter. For example, a transaction entering the SOA system from a JMS system.

initial request

Java: The first request for a JavaServer Faces page that a user makes.

initialization parameter

Java: The information provided to an individual servlet that is used when the servlet is initialized.

install

The process of populating your system with the Oracle Fusion Middleware software files. These software files are usually contained in a ZIP, JAR, or TAR archive that must be downloaded onto your system. After the contents of the archive file are extracted, an installer program is run to verify your system environment, then create the proper directory structure and place the Oracle Fusion Middleware software files in their proper locations. If needed, the installer also runs scripts to set environment variables and permissions. Installing also includes configuring the system. When the process is complete, Oracle Fusion Middleware components are deployed and ready to use.

interceptor

Java: A Java class or method that is used to interpose in method invocations or lifecycle events that occur in an associated target class. An interceptor performs tasks, such as logging or auditing, that are separate from the business logic of the application and that are repeated often within an application. An interceptor is annotated javax.interceptor.Interceptor. An interceptor can be used with Java EE managed objects, including managed beans. Originally designed to be used with enterprise beans, it can also be used with CDI.

interceptor binding types

Java: In CDI, annotations that associate an interceptor with target beans or methods.

interceptor class

Java: An interceptor that is defined as a Java class.

interceptor method

Java: An interceptor that is defined as a method within a target class.

interim patch

A patch that contains one or more fixes made available to customers who cannot wait until the next patch set or new product release to receive the fix.

internationalization

The process of preparing an application to support more than one language and data format.

invocation service

Coherence Suite: A service that provides clustered invocation and supports grid computing architectures. This service allows applications to invoke agents on any member in the cluster, or any group of members, or across the entire cluster. The agent invocations can be request/response, fire and forget, or an asynchronous user-definable model.

iterator binding

JDeveloper/ADF: A binding to an iterator that iterates over view object collections. There is one iterator binding for each collection used on the page. All of the value bindings on the page must refer to an iterator binding in order for the component values to be populated with data at runtime.

J

JAAS

Java: Java Authentication and Authorization Services. A Java security package that provides subject-based authorization on authenticated identities.

JAAS control flag

WebLogic Server: The JAAS control flag that determines how the login sequence uses the Authentication providers when a security realm has multiple Authentication providers.

JAD

Java: Java Application Descriptor (JAD) file

JAR

Java: Java Archive file

Java 2D

Java: The Java 2D API provides two-dimensional graphics, text, and imaging capabilities for Java programs through extensions to the Abstract Window Toolkit (AWT). This comprehensive rendering package supports line art, text, and images in a flexible, full-featured framework for developing richer user interfaces, sophisticated drawing programs, and image editors.

Java API for RESTful Web Services (JAX-RS)

Java: A Java programming language API designed to make it easy to develop web service applications that use the REST architecture.

Java API for XML Processing (JAXP)

Java: Processes XML data using applications written in the Java programming language. JAXP leverages the parser standards Simple API for XML Parsing (SAX) and Document Object Model (DOM) so that you can choose to parse your data as a stream of events or to build an object representation of it. JAXP also supports the Extensible Stylesheet Language Transformations (XSLT) standard, giving you control over the presentation of the data and enabling you to convert the data to other XML documents or to other formats, such as HTML. JAXP also provides namespace support, allowing you to work with DTDs that might otherwise have naming conflicts.

Java API for XML Web Services (JAX-WS)

Java: A technology for building web services and clients that communicate using XML. JAX-WS allows developers to write message-oriented as well as Remote Procedure Call-oriented (RPC-oriented) web services.

Java Architecture for XML Binding (JAXB)

Java: A fast and convenient way to bind XML schemas and Java representations, making it easy for Java developers to incorporate XML data and processing functions in Java applications. As part of this process, JAXB provides methods for unmarshalling (reading) XML instance documents into Java content trees, and then marshalling (writing) Java content trees back into XML instance documents. JAXB also provides a way to generate XML schema from Java objects.

Java Archive format (JAR) file

Java: A file format that enables you to bundle multiple files into a single archive file. Typically a JAR file contains the class files and auxiliary resources associated with applets and applications.

Java Authentication and Authorization Services (JAAS)

Java: A Java security package that provides subject-based authorization on authenticated identities.

Java Card

Java: A technology that allows Java-based applications (applets) to be run securely on smart cards and similar small memory footprint devices.

Java Compatibility Test Tools (Java CTT)

Java: Tools, documents, templates, and samples that can be used to design and build TCKs. Using the Java CTT simplifies compatibility test development and makes developing and running tests more efficient.

Java component

An Oracle Fusion Middleware component that is deployed as one or more Java EE applications and a set of resources. Java components are deployed to an Oracle WebLogic Server domain as part of a domain template. Examples of Java components are the Oracle SOA Suite, Oracle WebCenter Portal, and Oracle WebCenter Content components.

Java CTT

Java: Java Compatibility Test Tools. Tools, documents, templates, and samples that can be used to design and build TCKs. Using the Java CTT simplifies compatibility test development and makes developing and running tests more efficient.

Java Database Connectivity (JDBC)

Java: The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a relational database. JDBC helps you to write Java applications that manage these three programming activities: Connect to a data source, such as a database. Send queries and update statements to the database. Retrieve and process the results received from the database in answer to a query.

Java DB

Java: Oracle's supported distribution of the Apache Derby open source database. It supports standard ANSI/ISO SQL through the JDBC and Java EE APIs. Java DB is included in the JDK.

Java EE

Java: Java Platform, Enterprise Edition. A standard platform for hosting enterprise applications. The Java EE platform is developed through the Java Community Process (JCP), which is responsible for all Java technologies. Expert groups, composed of interested parties, have created Java Specification Requests (JSRs) to define the various Java EE technologies. The goal of the Java EE platform is to provide developers with a powerful set of APIs while shortening development time, reducing application complexity, and improving application performance.

Java EE application

Java: An application that consists of one or more components, which can be application clients, web components, and EJB components, and that can be deployed on client machines, a Java EE server machine, and/or the database or legacy machines at the back end.

Java EE component

Java: A self-contained functional software unit that is assembled into a Java EE application with its related classes and files and that communicates with other components.

Java EE deployment descriptor

Java: A deployment descriptor that is defined by a Java EE specification and that can be used to configure deployment settings on any Java EE-compliant implementation.

Java EE module

Java: An archive file that consists of one or more Java EE components for the same container type and, optionally, one component deployment descriptor of that type. An enterprise bean module deployment descriptor, for example, declares transaction attributes and security authorizations for an enterprise bean. A Java EE module can be deployed as a standalone module.

Java EE product provider

Java: The company that designs and makes available for purchase the Java EE platform APIs and other features defined in the Java EE specification. Product providers are typically application server vendors that implement the Java EE platform according to the platform specification.

Java EE server

Java: The runtime portion of a Java EE product, to which modules and applications are deployed. A Java EE server provides EJB and web containers.

Java EE web service

A web service implemented according to the Web Services for Java EE specification that defines the standard Java EE runtime architecture for implementing web services in Java.

Java Generic Security Services (Java GSS-API)

Java: A means to securely exchange messages between communicating applications. The Java GSS-API contains the Java bindings for the Generic Security Services Application Program Interface (GSS-API) defined in RFC 2853. GSS-API offers application programmers uniform access to security services atop a variety of underlying security mechanisms, including Kerberos.

Java Management Extensions (JMX)

Java: A simple, standard way of managing resources such as applications, devices, and services. Because the JMX technology is dynamic, you can use it to monitor and manage resources as they are created, installed, and implemented. You can also use the JMX technology to monitor and manage the Java Virtual Machine (JVM).

Java ME

Java: Java Platform, Micro Edition

Java Message Service (JMS)

Java: A Java API that allows applications to create, send, receive, and read messages using reliable, asynchronous, loosely coupled communication.

Java Naming and Directory Interface (JNDI)

Java: An API specified in Java technology that provides naming and directory functionality to applications written in the Java programming language. It is designed especially for the Java platform using Java's object model. Using JNDI, applications based on Java technology can store and retrieve named Java objects of any type. In addition, JNDI provides methods for performing standard directory operations, such as associating attributes with objects and searching for objects using their attributes.

Java Native Interface (JNI)

Java: A standard programming interface for writing Java native methods and embedding the Java Virtual Machine into native applications. The primary goal is binary compatibility of native method libraries across all Java Virtual Machine implementations on a given platform.

Java Network Launch Protocol (JNLP)

Java: A protocol used by the Java Web Start framework to remotely download and launch Java applications and applets. JNLP files are defined with an XML schema, which specifies how to launch Java Web Start applications.

Java Persistence API

Java: An API that provides Java developers with an object/relational mapping facility for managing relational data in Java applications. Java Persistence consists of four areas: the Java Persistence API, a query language, the Java Persistence Criteria API, and object/relational mapping metadata.

Java Persistence query language (JPQL)

Java: A simple, string-based language similar to SQL that is used to query entities and their relationships.

Java Platform, Enterprise Edition (Java EE)

Java: A standard platform for hosting enterprise applications. The Java EE platform is developed through the Java Community Process (JCP), which is responsible for all Java technologies. Expert groups, composed of interested parties, have created Java Specification Requests (JSRs) to define the various Java EE technologies. The goal of the Java EE platform is to provide developers with a powerful set of APIs while shortening development time, reducing application complexity, and improving application performance.

Java Platform, Standard Edition (Java SE)

Java: An application development environment for the Java language, a runtime environment for Java applications, and the mechanism for deploying Java applications in a cross platform environment. Elements of the Java platform include the Java SE Runtime Environment (JSE), the Java SE Development Kit (JDK), the Java Virtual Machine (JVM), and the Java languge specification. The Java language is managed by the Java Community Process (JCP) program.

Java Plug-in

Java: Java Plug-in technology, included as part of the Java Runtime Environment, Standard Edition (JRE), establishes a connection between popular browsers and the Java platform. This connection enables applets on websites to be run within a browser on the desktop.

Java Reflection API

Java: Reflection is commonly used by programs that require the ability to examine or modify the runtime behavior of applications running in the Java Virtual Machine.

Java Remote Method Invocation (RMI)

Java: Allows an object running in one Java Virtual Machine to invoke methods on an object running in another Java Virtual Machine. RMI provides for remote communication between programs written in the Java programming language.

Java Remote Method Invocation over Internet Inter-ORB Protocol (RMI-IIOP)

Java: A protocol that allows a program to access remote objects by using the Internet Inter-ORB Protocol (IIOP).

Java SE

Java: Java Platform, Standard Edition. An application development environment for the Java language, a runtime environment for Java applications, and the mechanism for deploying Java applications in a cross platform environment. Elements of the Java platform include the Java SE Runtime Environment (JSE), the Java SE Development Kit (JDK), the Java Virtual Machine (JVM), and the Java languge specification. The Java language is managed by the Java Community Process (JCP) program.

Java SE Development Kit (JDK)

Java: A development product provided by Oracle for building Java applications, applets, and components using the Java programming language. The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Java SE Runtime Environment (JRE)

Java: The libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language. In addition, two key deployment technologies are part of the JRE: Java Plug-in, which enables applets to run in popular browsers; and Java Web Start, which deploys standalone applications over a network.

Java Secure Socket Extension (JSSE)

Java: Enables secure Internet communications and provides a framework and an implementation for a Java version of the SSL and TLS protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication.

Java Simple Authentication and Security Layer (SASL)

Java: An Internet standard (RFC 2222) that specifies a protocol for authentication and optional establishment of a security layer between client and server applications. SASL defines how authentication data is to be exchanged but does not itself specify the contents of that data. It is a framework into which specific authentication mechanisms that specify the contents and semantics of the authentication data can fit.

Java Specification Request (JSR)

Java: The actual description of a proposed or final Java specification for the Java platform under the Java Community Process (JCP). JSRs are documents that have a title and are identified by a unique number, such as JSR 135: Mobile Media API.

Java Transaction API (JTA)

Java: A Java EE API that provides a standard interface for demarcating transactions.

Java Transaction Service (JTS)

Java: A Java API that is commonly used to implement Java EE transaction managers but is not directly called by applications, which instead use the Java Transaction API (JTA).

Java Virtual Machine (JVM)

Java: An implementation of the Java Virtual Machine specification, sits on top of a computer's operating system and interprets compiled Java applets and applications. A JVM is part of the Java platform.

JavaBeans

Java: In the Java language, JavaBeans (also referred to as beans) are reusable software components used to encapsulate many objects into a single object, so that they can be passed around as a single bean object instead of as multiple individual objects.

JavaMail

Java: An API used by Java EE applications to send email notifications.

JavaServer Faces lifecycle

Java: The stages through which a JavaServer Faces application passes, beginning when the client makes an HTTP request for a page and ends when the server responds with the page, translated to HTML. The lifecycle can be broadly divided into an execute phase and a render phase. The specific phases are the Restore View phase, the Apply Request Values phase, the Process Validations phase, the Update Model Values phase, the Invoke Application phase, and the Render Response phase.

JavaServer Faces technology

Java: A server-side component framework for building Java technology-based web applications. JavaServer Faces technology consists of the following: An API for representing components and managing their state; handling events, server-side validation, and data conversion; defining page navigation; supporting internationalization and accessibility; and providing extensibility for all these features Tag libraries for adding components to web pages and for connecting components to server-side objects

JavaServer Faces UI component

Java: A user interface component, such as a button or text field, that is specified by an HTML tag library and can be placed on an HTML page.

JavaServer Faces UI component class

Java: A class in the javax.faces.component package (for example, UICommand or UIInput) that provides the API for a JavaServer Faces UI component.

JavaServer Pages (JSP)

Java: A technology that lets you put snippets of servlet code directly into a text-based document. A JSP page is a text-based document that contains two types of text: static data (which can be expressed in any text-based format such as HTML, WML, and XML) and JSP elements, which determine how the page constructs dynamic content. JSP pages have been superseded by JavaServer Facelets technology but can still be used.

JavaServer Pages Standard Tag Library (JSTL)

Java: A library of tags that encapsulates core functionality common to many JSP applications.

JavaTest harness

Java: A test harness developed to manage test execution and result reporting for a Technology Compatibility Kit (TCK). The harness configures, sequences, and runs test suites. The JavaTest harness provides flexible and customizable test execution. It includes everything a test architect needs to design and implement tests for implementations of a Java specification.

JAX-RS

Java: See Java API for RESTful Web Services

JAX-WS

Java: See Java API for XML Web Services

JAXP

Java: Java API for XML Processing. Processes XML data using applications written in the Java programming language. JAXP leverages the parser standards Simple API for XML Parsing (SAX) and Document Object Model (DOM) so that you can choose to parse your data as a stream of events or to build an object representation of it. JAXP also supports the Extensible Stylesheet Language Transformations (XSLT) standard, giving you control over the presentation of the data and enabling you to convert the data to other XML documents or to other formats, such as HTML. JAXP also provides namespace support, allowing you to work with DTDs that might otherwise have naming conflicts.

JCP

Java: Java Community Process

JDBC

Java: Java Database Connectivity. The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a relational database. JDBC helps you to write Java applications that manage these three programming activities: Connect to a data source, such as a database. Send queries and update statements to the database. Retrieve and process the results received from the database in answer to a query.

JKS keystore

The default JDK implementation of Java keystores. All Oracle Fusion Middleware Java components and Java EE applications use the JKS-based keystore and TrustStore.

JMS

Java: See Java Message Service

JMS administered object

Java: A preconfigured JMS object created by an administrator for the use of clients. The two kinds of JMS administered objects are destinations and connection factories.

JMS API local transaction

Java: A transaction specified by the JMS session commit and rollback methods.

JMS application

Java: An application that uses a JMS provider, one or more JMS clients, messages, and JMS administered objects.

JMS client

Java: In the JMS API, a program or component, written in the Java programming language, that produces or consumes messages. Any Java EE application component can act as a JMS client.

JMS connection

Java: In the JMS API, an object that encapsulates a virtual connection with a JMS provider.

JMS connection factory

Java: In the JMS API, the object a client uses to create a connection to a provider.

JMS provider

Java: A messaging system that implements the JMS interfaces and provides administrative and control features. An implementation of the Java EE platform includes a JMS provider.

JMS session

Java: In the JMS API, a single-threaded context for producing and consuming messages.

JNDI lookup

Java: A means by which a Java EE application can obtain access to resources. Also, a means by which the client of an enterprise bean obtains a reference to an instance of an enterprise bean by using the Java Naming and Directory Interface syntax.

joining messages

Java: In the JMS API, a design pattern in which a JMS client waits for several messages from various sources and then uses the information in all these messages to assemble a message that it then sends to another destination. Such a task must be transactional, with all the receives and the send as a single transaction.

JRE

Java: Java Runtime Environment

JSR

Java: Java Specification Request

JTA transaction

Java: A transaction that is controlled by the Java EE transaction manager and that uses the Java Transaction API (JTA). The transaction manager is implemented with the Java Transaction Service (JTS).

SOA Suite: Each step of a process related to transactional adapter operations is executed within the context of a JTA transaction. A JTA transaction ensures that one or more operations execute as an atomic unit of work.

K

Kerberos

OWSM: A network authentication service that enables computers (clients and servers) communicating over a non-secure network to prove their identity to one another in a secure manner, with the help of shared secrets and a trusted third party. In Kerberos, this trusted third party is the Key Distribution Center (KDC), which contains key information for clients and servers, called principals.

Kerberos ticket

OWSM: A sequence (of a few hundred bytes in length) that is used to control access to physically insecure networks. Kerberos tickets are based on the Kerberos protocol.

key pair

Two pieces of encryption information, one private and one public. Data encrypted with one key can be decrypted only with the other key of the pair. This property is fundamental to establishing trust and privacy in transactions.

keystore

A repository of objects necessary for security, such as private key certificates, digital certificates, and trusted CA certificates, as well as objects for message security. For example, SSL communication uses keystores.

Java: In Java security, a repository of certificates used for identifying a client or a server. Typically, a keystore is a file that contains one client's or one server's identity.

L

lazy child creation

JDeveloper/ADF: For ADF Faces UI components, when child components are created only when they are disclosed. Once disclosed and the associated child components are rendered, they remain created in the component tree. Certain components can set their children to be lazily created and then uncached, meaning that once the child component is rendered, if it is subsequently hidden, it is destroyed (it does not remain created in the component tree).

lazy content delivery

JDeveloper/ADF: In ADF Faces, a method of fetching data for a UI component. Initially, the component goes through the standard lifecycle. However, instead of fetching the data during that initial request, a separate partial page rendering (PPR) request is run and the data is then returned. Because the page has just been rendered, only the Render Response phase executes, allowing the corresponding data to be fetched and displayed.

LDAP Authentication provider

WebLogic Server: WebLogic Security Framework provider that uses a Lightweight Data Access Protocol (LDAP) server to access user and group information, for example, iPlanet's Active Directory and Novell's OpenLDAP.

Lifecycle Events

Coherence Suite: A set of events that represent the activation and disposal of a ConfigurableCacheFactory instance.

literal expression

Java: An expression language (EL) expression that consists only of text.

Live Events

Coherence Suite: An event framework that allows your applications to react to operations performed in the data grid. The framework uses an event-based model where events represent observable occurrences of cluster operations. The supported events include partitioned service, cache, and application events. These events can be consumed by registering event interceptors either programmatically or by using the cache configuration.

local business interface

Java: An interface that defines the business and lifecycle methods of an enterprise bean in a way that allows local access (where the client and the enterprise bean both run in the same application), usually by means of the @Local annotation.

localization

The process of adapting an internationalized application to support a specific region or locale.

lockbox target

Identity Governance: A target type that does not interact with Oracle Privileged Account Manager, but still provides a secure mechanism for storing passwords (or any kind of sensitive information) associated with privileged accounts in a deployment.

login module

Java: In Java security, a Java class provided by an application client to gather authentication data. The class must implement the javax.security.auth.callback.CallbackHandler interface.

loosely coupled

Java: A form of communication enabled by messaging in which the sender and the receiver do not have to be available at the same time in order to communicate and do not have to know anything about each other.

lower bounded wildcards

Java: In the generics feature of the Java language, a lower bounded wildcard restricts the unknown type to be a specific type or a super type of that type.

lvalue expression

Java: An expression language (EL) expression that can both read a value and write it to an external object, as opposed to an rvalue expression.

M

machine

The logical representation of the computer that hosts one or more WebLogic Server instances.

Maintenance Lead

Java: In the Java Community Process (JCP), the person or organization responsible for maintaining an existing Java specification and related reference implementation (RI) and Technology Compatibility Kit (TCK). The ML manages the TCK appeals process, the related exclude list, and any revisions needed to the specification, TCK, or RI.

managed bean

Java: A lightweight container-managed object (a Plain Old Java Object, or POJO) with minimal requirements, which supports a small set of basic services, such as resource injection, lifecycle callbacks, and interceptors. Managed beans represent a generalization of the managed beans specified by JavaServer Faces technology and by Contexts and Dependency Injection for the Java EE Platform (CDI) and can be used anywhere in a Java EE application, not just in web modules.

managed Coherence server

Coherence Suite: A means to leverage the WebLogic Management Framework. Managed Coherence servers can be configured and controlled from an Administration Server. Managed Coherence servers can be explicitly associated with a Coherence cluster, or they can be associated with a WebLogic server cluster that is associated with a Coherence cluster. Managed Coherence servers are typically setup in tiers based on their type: a data tier for storing data, an application tier for hosting applications, and a proxy tier for allowing access to external clients.

Managed Server

WebLogic Server: A WebLogic Server instance that hosts business applications, application components, web services, and their associated resources.

Management Pack for Oracle Coherence

Coherence Suite: A feature of Oracle Enterprise Manager that provides comprehensive monitoring and management capabilities for Oracle Coherence. This pack provides complete cluster visibility by supplying detailed metrics of various cluster artifacts and their interdependencies. You can monitor more than one Oracle Coherence cluster from a single console.

many-to-many

An entity relationship in which entity instances can be related to multiple instances of each other.

many-to-one

An entity relationship in which multiple instances of an entity can be related to a single instance of the other entity.

Map events

Coherence Suite: Cache events using the JavaBean event model. The implementation allows applications to receive the events when and where they are needed, regardless of where the changes are actually occurring in the cluster.

master-detail relationship

JDeveloper/ADF: A hierarchical relationship in the data model of an ADF application established when a view link is created to associate two view object instances. Usually, but not necessarily, the master-detail relationship is based on a foreign-key relationship between the underlying database tables.

matrix parameter

Java: A JAX-RS request parameter that extracts information from URL path segments. It is specified by using the javax.ws.rs.MatrixParam annotation.

MBean Definition File

WebLogic Server: An XML file used by the WebLogic MBeanMaker to generate files for an MBean type.

MBean implementation file

WebLogic Server: One of several intermediate Java files generated by the WebLogic MBeanMaker utility to create an MBean type for a custom security provider. You edit this file to supply your specific method implementations.

MBean information file

WebLogic Server: One of several intermediate Java files generated by the WebLogic MBeanMaker utility to create an MBean type for a custom security provider. This file contains mostly metadata and therefore requires no editing.

MBean interface file

WebLogic Server: One of several intermediate Java files generated by the WebLogic MBeanMaker utility to create an MBean type for a custom security provider. This file is the client-side API to the MBean that your runtime class or your MBean implementation will use to obtain configuration data, and requires no editing.

MBean JAR file

WebLogic Server: JAR file that contains the runtime classes and MBean types for a security provider. MBean JAR files are created by the WebLogic MBeanMaker.

MBean type

WebLogic Server: A factory for creating the MBeans used to configure and manage security providers. MBean types are created by the WebLogic MBeanMaker.

menu model

JDeveloper/ADF: In ADF Faces, a tree data model that knows how to retrieve the rowKey of the node that has the current focus (the focus node). It allows you to define the hierarchical tree of navigation in XML metadata.

message

Java: In the JMS API, an object that communicates information between JMS clients.

message consumer

Java: In the JMS API, an object that is created by a session and used for receiving messages sent to a destination. It implements the javax.jms.MessageConsumer interface.

message-driven bean

Java: An enterprise bean that combines features of a session bean and a message listener, allowing a business component to receive messages asynchronously. Commonly, these are Java Message Service (JMS) messages.

message listener

Java: In the JMS API, an object that acts as an asynchronous event handler for messages. A message listener must implement the onMessage method.

message producer

Java: In the JMS API, an object that is created by a session and used for sending messages to a destination. It implements the javax.jms.MessageProducer interface.

message security

A form of security that works with web services and incorporates security features, such as digital signatures and encryption, into the header of a SOAP message, working in the application layer, ensuring end-to-end security.

Message Transmission Optimization Mechanism (MTOM)

A method for optimizing the transmission of XML data of type xs:base64Binary or xs:hexBinary in SOAP messages, as defined in the SOAP Message Transmission Optimization Mechanism specification.

metadata label

A means of selecting a particular version of each object from an MDS repository partition. Conceptually, it is a collection of document versions, one version per document, representing a horizontal stripe through the various document versions. This stripe comprises the document versions which were the tip versions (latest versions) at the time the label was created.

Metadata Services (MDS) repository

A particular type of repository that contains metadata for certain types of deployed applications. This includes custom Java EE applications developed by your organization and some Oracle Fusion Middleware component applications, such as Oracle B2B. An MDS repository can be file-based or database-based.

method

Java: A procedure or routine associated with one or more classes in object-oriented languages.

method expression

Java: An expression language (EL) expression that invokes a method, as opposed to a value expression.

method iterator

JDeveloper/ADF: A binding to an iterator that iterates over the collections returned by custom methods in the data control. An iterator binding that is always related to a method action binding object. The method action binding encapsulates the details about how to invoke the method and what parameters the method is expecting. The method action binding is itself bound to the method iterator, which provides the data.

method permission

Java: In Java EE security, a permission to invoke a specified group of methods of an enterprise bean's business interface, home interface, component interface, and/or web service endpoints. After method permissions are defined, user name/password authentication will be used to verify the identity of the user.

migration

The process of moving from a third-party (non-Oracle) product to an Oracle product.

Monitoring Dashboard

WebLogic Server: WebLogic Diagnostics Framework (WLDF) component that graphically presents the current and historical operating state of WebLogic Server and hosted applications.

monolithic session state model

Coherence Suite: A session state model used by Coherence*Web. This model stores all session state as a single entity, serializing and deserializing all attributes as a single operation.

MTOM

See Message Transmission Optimization Mechanism (MTOM)

multi data source

WebLogic Server: Abstraction around a group of data sources that is bound to the JDNDI tree or local application context just like data sources are bound to the JNDI tree.

mutual authentication

A type of authentication in which the server and the client authenticate each other. Mutual authentication is of two types: certificate-based and user name/password-based.

N

named criteria

JDeveloper/ADF: Predefined and reusable WHERE clause definitions that are dynamically applied to an ADF view object query.

NamedCache

Coherence Suite: The Coherence NamedCache object represents a data cache. This object is designed to hold resources that are shared among members of a Coherence cluster. These resources are managed in memory, and are typically composed of data that is also stored persistently in a database, or data that has been assembled or calculated. Thus, these resources are referred to as cached.

namespace

Java: A set of names in which all names are unique.

naming context

Java: A container's implementation of a components naming environment that is provided to the component.

naming environment

Java: An environment, normally provided by the Java Naming and Directory Interface (JNDI), that allows a component to be customized without the need to access or change the components source code.

native

Java: A reserved Java keyword used in method declarations to specify that the method is not implemented in the same Java source file, but rather in another language.

native EclipseLink

Toplink: The API underlying EclipseLink JPA and EclipseLink JAXB. It replaces the obsolete native TopLink API. Users can migrate from native TopLink to native EclipseLink through a package renaming utility included in all EclipseLink and TopLink distributions.

native TopLink

Toplink: The APIs, libraries, configuration files, and tools that constituted Oracle TopLink before the Java Persistence API (JPA) specification was written and implemented by TopLink (as TopLink Essentials). Native TopLink provided its own implementations of object/relational (ORM), object/XML (OXM), and other mapping/persistence technologies before the standards-based specifications were developed. Native TopLink is no longer available. However, applications can easily upgrade to native EclipseLink.

navigation

Java: In JavaServer Faces technology, a set of rules for choosing the next page or view to be displayed after an application action, such as when a button or hyperlink is clicked. In Java Persistence, the traversal of relationships in a query language expression; the navigation operator is a period.

near cache

Coherence Suite: A near cache is a hybrid cache; it typically fronts a distributed cache or a remote cache with a local cache. Near cache invalidates front cache entries, using a configured invalidation strategy, and provides excellent performance and synchronization. This feature is intended to provide local access in cache clients for repeated reads.

new

Java: A reserved Java keyword used to create an instance of a class or array/an object.

New I/O (NIO) / (NIO.2)

Java: Java I/O support is included in the java.io and java.nio packages. Together these packages include the following features: Input and output through data streams, serialization and the file system. Charsets, decoders, and encoders, for translating between bytes and Unicode characters. Access to file, file attributes and file systems. APIs for building scalable servers using asynchronous or multiplexed, non-blocking I/O.

nibble

Java: Four bits.

no-interface view

Java: A means of client access to enterprise beans in which the public methods of the enterprise bean implementation class are exposed to clients. Clients using the no-interface view of an enterprise bean may invoke any public methods in the enterprise bean implementation class or any superclasses of the implementation class.

Node Manager

WebLogic Server: A WebLogic Server utility that enables you to start, shut down, and restart Administration Server and Managed Server instances from a remote location.

non-repudiation

In security, the means used to prove that a user who performed some action cannot reasonably deny having done so. This ensures that transactions can be proved to have happened.

O

obfuscation

Java: A technique used to complicate code by making it harder to understand when it is decompiled. Obfuscation makes it harder to reverse-engineer applications and therefore, steal them.

object

Java: A unique instance of a data structure defined according to the template provided by its class. Each object has its own values for the variables belonging to its class and can respond to the messages (methods) defined by its class.

observer method

Java: In CDI, a method used by an event handler to consume events.

one-to-many

An entity relationship in which an entity instance can be related to multiple instances of the other entities.

one-to-one

An entity relationship in which each entity instance is related to a single instance of another entity.

OPatch

A Java-based utiltiy used to to patch an existing Oracle Fusion Middleware installation It runs on all supported operating systems and requires installation of the Oracle Universal Installer.

operational configuration file (tangosol-coherence.xml)

Coherence Suite: A file that provides operational and runtime settings and is used to create and configure cluster, communication, and data management services. This file is also referred to as the operational deployment descriptor.

optimistic locking

Java: In Java Persistence, a locking scheme in which, before committing changes to the data, the persistence provider checks that no other transaction modified or deleted the data since the data was read.

OptimisticCache service

Coherence Suite: An optimistic-concurrency version of the ReplicatedCache service that fully replicates all of its data to all cluster members and employs an optimization similar to optimistic database locking to maintain coherency. All servers end up with the same current value, even if multiple updates occur at the same time from different servers.

Oracle ADF

JDeveloper/ADF: Oracle Application Development Framework. An end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications.

Oracle ADF Model Tester

JDeveloper/ADF: A Java application launched from JDeveloper that allows the ADF Business Components developer to interact with business objects of the data model project. The Oracle ADF Model Tester runs outside of JDeveloper and provides a full UI for testing and examining the data model project.

Oracle Application Development Framework (Oracle ADF)

JDeveloper/ADF: An end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications.

Oracle Common directory

The directory that contains the binary and library files that are common to all Oracle Fusion Middleware products and features installed in the Oracle home. In particular, the Oracle Common directory includes the files required for Oracle Enterprise Manager Fusion Middleware Control, WLST, the Configuration Wizard, upgrade tools, and Oracle JRF. There can be only one Oracle Common directory within each Oracle home.

Oracle Diagnostic Logging (ODL)

An Oracle standard for the naming of log files and the formatting of the contents of the log files. This is often referred to as ODL.

Oracle Enterprise Manager Fusion Middleware Control

A Web browser-based, graphical user interface that you can use to monitor and administer your Oracle Fusion Middleware environment.

Oracle Fusion Middleware Infrastructure

A Oracle Fusion Middleware distribution that provides Oracle WebLogic Server and Oracle JRF.

Oracle Fusion Middleware Repository Creation Utility (RCU)

A utility used to install the schemas required for specific Oracle Fusion Middleware components into a supported database.

Oracle home

The Oracle home that is created for all the Oracle Fusion Middleware products on a host computer. This read-only directory contains binary and library files, the Oracle common directory, and the individual product directories for each Oracle Fusion Middleware product you install.

Oracle Infrastructure web services

A category of web services supported in Oracle Fusion Middleware that encompasses SOA, Application Development Framework (ADF), Oracle Service Bus, and Oracle Enterprise Scheduler services.

Oracle JRF

Oracle JRF (Java Required Files) consists of those components not included in the Oracle WebLogic Server installation and that provide common functionality for Oracle business applications and application frameworks. Oracle JRF consists of several independently developed libraries and applications that are deployed into a common location. The components that are considered part of Oracle JRF include Oracle Application Development Framework shared libraries and ODL logging handlers.

Oracle Metadata Services

A consistent means to define, use, manage, and customize application metadata. Application metadata is characterized by its ability to define the behavior of an application.

Oracle TopLink

Toplink: A persistence framework including development tools and runtime capabilities for providing transformation, mapping, and persistence operations in Java SE and Java EE environments. The core technology of TopLink is provided by EclipseLink, the open-source persistence framework from the Eclipse Foundation. TopLink includes additional features beyond EclipseLink, including TopLink Grid and TopLink Data Services. TopLink is included in Oracle distributions of Oracle WebLogic Server, Oracle Glassfish, and Oracle JDeveloper.

Oracle Wallet

A container that stores your credentials, such as certificates, trusted certificates, certificate requests, and private keys. You can store Oracle wallets on the file system or in LDAP directories such as Oracle Internet Directory. Oracle wallets can be auto-login or password-protected wallets. An Oracle Wallet is a type of keystore.

Oracle Web Services Manager (OWSM)

OWSM: Oracle Web Services Manager (OWSM) offers a comprehensive and easy-to-use solution for policy management and security of service infrastructure. It provides visibility and control of the policies through a centralized administration interface offered by Oracle Enterprise Manager.

OTP Anywhere

Identity Management: A risk-based challenge solution consisting of a server-generated one-time password delivered to an end user via a configured out of band channel.

outbound transaction

SOA Suite: An outbound transaction from the SOA system (and hence from an adapter). For example, a transaction that is made against a database outside the SOA system.

overloading

Java: A means to create more than a single method with same name with different signatures.

overriding

In Java, an instance method in a subclass with the same signature (name, plus the number and the type of its parameters) and return type as an instance method in the superclass overrides the superclass's method. The annotations feature in Java also supports an @Override annotation. While it's not required to use this annotation when overriding a method, it helps to prevent errors. If a method marked with @Override fails to correctly override a method in one of its superclasses, the compiler generates an error.

OWSM

OWSM: Oracle Web Services Manager (OWSM) offers a comprehensive and easy-to-use solution for policy management and security of service infrastructure. It provides visibility and control of the policies through a centralized administration interface offered by Oracle Enterprise Manager.

OWSM Agent

Identity Management: The OWSM component responsible for policy enforcement, execution, and gathering of run-time statistics. It is available on all Oracle Fusion Middleware Managed servers. The Agent is configured on the same server as the application it protects. It consists of two components: the Policy Access Point (PAP)

which communicates with the Policy Manager, and the Policy Interceptor which is responsible for policy enforcement.

OWSM policy

OWSM: A web service policy provided by OWSM. OWSM provides a set of predefined policies based on common best practice policy patterns used in customer deployments.

OWSM Policy Manager

OWSM: The OWSM component responsible for reading and writing OWSM documents, including predefined and custom policies and assertion templates, from the OWSM Repository.

OWSM Repository

OWSM: The OWSM Repository stores OWSM metadata, such as policies, policy sets, assertions templates, and policy usage data. The OWSM Repository is available as a database (for production use) or as files in the file system (for development use in JDeveloper).

P

package

Java: A reserved Java keyword used to declare a group of types.

page definition file

JDeveloper/ADF: A file that defines the binding objects that populate the data in UI components at runtime. For every page that has ADF bindings, there must be a corresponding page definition file that defines the binding objects used by that page.

page flow scope

JDeveloper/ADF: An ADF-defined scope in the page lifecycle. An object in this scope is available as long as the user continues navigating from one page to another. If the user opens a new browser window and begins navigating, that series of windows will have its own page flow scope.

parallel flow

SOA Suite: Enables a BPEL process service component to perform multiple tasks at the same time. Parallel flow is especially useful when you must perform several time-consuming and independent tasks.

partial page rendering (PPR)

JDeveloper/ADF: When the JSF page request life cycle (including conversion and validation) is run only for certain components on a page. The event root component determines on which components the life cycle is run.

partition

SOA Suite: Separate sections of the SOA Infrastructure into which you can deploy SOA composite applications. Deploying to partitions enables you to logically group SOA composites and perform bulk lifecycle management tasks on large numbers of composites.

partition transaction events

Coherence Suite: A set of events that are related to changes that may span multiple caches and are performed within the context of a single request.

partitioned cache events

Coherence Suite: A set of events that represent the operations being performed against a set of entries in a cache. Partitioned cache events include both entry events and entry processor events. Entry events are related to inserting, removing, and updating entries in a cache. Entry processor events are related to the execution of entry processors.

partitioned service events

Coherence Suite: A set of events that represent the operations being performed by a partitioned service. Partitioned service events include both partition transfer events and partition transaction events. Partition transfer events are related to the movement of partitions among cluster members.

partner link

SOA Suite: Characterizes the conversational relationship between two services in a BPEL process by defining the roles played by each service in the conversation and specifying the port type provided by each service to receive messages within the conversation.

passivation

Java: The deactivation of an enterprise bean by the EJB container by moving it from memory to secondary storage.

Password Validation provider

WebLogic Server: WebLogic Security Framework provider that can be configured with an authentication provider to enforce a set of password composition rules.

patch

A collection of files usually associated with a particular version of an Oracle product and involves updating from one minor version of the product to a newer minor version of the same product (for example, from version 12.1.2.1.0 to version 12.1.2.2.0).

patch set

A single patch that contains a collection of patches designed to be applied together.

patch set update

A quarterly patch that contains the most critical fixes for the applicable product, allowing customers to apply one patch to avoid many problems.

patching

Copying a small collection of files over an existing installation.

path expression

Java: In Java Persistence, an expression that navigates to a entity's state or relationship field.

path info

Java: The part of the request path for a servlet that is not part of the context path or the servlet path.

peer

JDeveloper/ADF: In the ADF Faces framework, an intermediary between a client component and the document object model (DOM). Peers interact with the DOM generated by the Java renderer and handle updating that state and responding to user interactions.

per domain Node Manager

WebLogic Server: The version of Node Manager that controls all WebLogic Server instances belonging to the same domain. The server instances need not reside on the same machine. Note that this is the default Node Manager type that is configured in a domain.

per host Node Manager

WebLogic Server: The version of Node Manager that controls all WebLogic Server instances configured on a particular machine. The server instances need not all be members of the same domain.

persistent attribute

JDeveloper/ADF: An attribute of an ADF Business Components entity object or view object that is derived from the underlying data source, such as a database table.

personally identifiable information (PII)

OWSM: PII refers to Social Security numbers, addresses, bank account numbers, and other similar information that is typically associated with one specific user and must generally be protected.

pessimistic locking

Java: In the Java Persistence API, a locking scheme in which the persistence provider creates a transaction that obtains a long-term lock on the data until the transaction is completed, which prevents other transactions from modifying or deleting the data until the lock has ended.

PII

OWSM: See personally identifiable information (PII)

plug-in

WebLogic Server: A module that adds a specific feature or service to a larger system.

POF (Portable Object Format)

Coherence Suite: A proprietary data format for high-performance serialization. POF format is significantly faster than the standard Java Serializable interface format and the serialized result set is smaller.

POF configuration file (coherence-pof-config.xml)

Coherence Suite: Custom data types when using Portable Object Format (POF) to serialize objects. This file is also referred to as the POF configuration deployment descriptor.

point-to-point messaging system

Java: In the JMS API, a messaging system built on the concept of message queues, senders, and receivers.

policy assertion

OWSM: The smallest unit of a policy that performs a specific action for the request and response operations. Assertions, like policies, are grouped into categories, such as Reliable Messaging, Management, WS-Addressing, Security, Configuration, and MTOM.

policy subject

OWSM: The target resource to which OWSM policies are attached. As defined in the Web Services Policy 1.5 - Framework, a policy subject is an entity (for example, an endpoint, message, resource, or operation) with which a policy can be associated.

postback request

Java: The request for a JavaServer Faces page. This request occurs when a user submits the form contained on a page that was previously loaded into the browser as a result of executing an initial request.

PPR

JDeveloper/ADF: partial page rendering. When the JSF page request life cycle (including conversion and validation) is run only for certain components on a page. The event root component determines on which components the life cycle is run.

predefined assertion template

OWSM: A read-only document included with your OWSM installation consisting of policy assertions that can be used to construct OWSM policies.

predefined policy

OWSM: OWSM policy provided with your Fusion Middleware installation that can be used to secure and configure your environment. Predefined policies are read only and cannot be modified.

presentation-oriented

A web application that generates interactive web pages containing various types of markup language (HTML, XHTML, XML, and so on) and dynamic content in response to requests.

primary key

A unique object identifier that enables clients to locate a particular entity instance.

primitive type

Java: A type that is predefined by the language and is named by a reserved keyword. Java has eight primitive types: byte, short, int, long, float, double, boolean, and char.

principal

An entity that can be authenticated by an authentication protocol in a security service. A principal is identified by using a principal name and authenticated by using authentication data.

private

Java: A reserved Java keyword used in the declaration of a method, field, or inner class; private members can be accessed only by other members of their own class.

privileged account

Identity Governance: An account that can access sensitive data, can grant access to sensitive data, or can both access and grant access to that data. Privileged accounts are a company's most powerful accounts and they are frequently shared. These accounts are typically associated with elevated privileges, are used by multiple end users on a task-by-task basis, and must be controlled and audited. For example, UNIX root, system, or database administrators.

producer field

Java: In CDI, a field of a bean that generates an object that can then be injected.

producer method

Java: In CDI, a method that generates an object that can then be injected.

product directory

A directory within the Oracle home that contains the binary files associated with a logical product or feature set. The name of each product directory within the Middleware Oracle home is predefined by the installer and cannot be changed.

profile

Java: A set of APIs added to a configuration to support specific uses of a mobile device. Along with its underlying configuration, a profile defines a complete and self-contained application environment.

programmatic security

Java: A type of Java EE security that is embedded in an application and is used to make security decisions.

programmatic timer

Java: An enterprise bean timer that is set by explicitly calling one of the timer creation methods of the javax.ejb.TimerService interface.

protected

Java: A reserved Java keyword used in the declaration of a method, field, or inner class; protected members can be accessed only by members of their own class, that class's subclasses or classes from the same package.

proxy service

Coherence Suite: A service that allows connections (using TCP) from clients that run outside the cluster. While many applications are configured so that all clients are also cluster members, there are many use cases that have clients running outside the cluster. Remote clients are especially useful in cases where there are hundreds or thousands of client processes, where the clients are not running on the Java platform, or where there is a greater degree of decoupling.

proxy tier

Coherence Suite: One or more computers running Coherence*Extend proxy servers.

public

Java: A reserved Java keyword used in the declaration of a class, method, or field. Public classes, methods, and fields can be accessed by the members of any class.

public key certificate

An electronic document that uses a digital signature to bind a public key with an identity. The certificate can be used to verify that a public key belongs to an individual.

public key cryptography

Cryptography that is based on key pairs.

publish/subscribe messaging system

Java: In the JMS API, a messaging system in which clients address messages to a topic, which functions somewhat like a bulletin board. The system distributes the messages arriving from multiple publishers to its multiple subscribers.

push replication

Coherence Suite: The ability to replicate cache operations from one cluster to another by using an external protocol (TCP/IP or JMS). Operations are optimistically replicated, and in the event that communication between the clusters breaks down, they are queued up until communication is reestablished.

Q

query parameter

Java: A JAX-RS request parameter that is extracted from the request URI query parameters and specified by using the javax.ws.rs.QueryParam annotation in the method parameter arguments.

query root

Java: In the Java Persistence API, the base entity from which the query will navigate to find the entity's attributes and related entities. In the Criteria API, the query root is created by calling the from method of the query object. In the Java Persistence query language, you can use a range variable declaration to designate the query root.

queue

Java: In the JMS API, a destination used in point-to-point messaging. Each message in a queue has only one consumer. Each message is addressed to a specific queue, and the receiving clients extract messages from the queues established to hold their messages. Queues retain all messages sent to them until the messages are consumed or until the messages expire.

R

range paging

JDeveloper/ADF: An access mode set on Oracle ADF view objects when a large result set is possible. The range paging mode fetches and caches only the current page of rows in the view object and allows the Oracle ADF application to page back and forth through data more efficiently than if the application used scrollable access.

range variable declaration

Java: In the Java Persistence API, a declaration of an identification variable that can range over the abstract schema type of an entity.

RAR

Java: A Resource Archive file that contains a resource adapter (.rar) module.

RDBMS security store

WebLogic Server: An external RDBMS containing a data store that, when configured in a domain, is used by select security providers for storing security data.

read-through caching

Coherence Suite: A technique of updating the cache: When an application asks the cache for an entry that is not in the cache, Coherence automatically delegates to the cache store to load the entry from the underlying data source. If the entry exists in the data source, then the cache store loads it and returns it to Coherence. Coherence then places it in the cache for future use and finally returns the entry to the application code that requested it.

read-write-backing-map scheme

Coherence Suite: A map that provides a size-limited cache of a persistent store, such as a relational database.

real time client configured as a compute client

Coherence Suite: A server-class client that provides key manageability, monitoring, Quality of Service, and performance capabilities. See also real time cluster member client.

real time clients configured as Extend/TCP clients

Coherence Suite: Clients that connect to the Coherence cluster over TCP/IP through specially configured cluster members that are called extend proxies. Real time extend/TCP clients are multilanguage clients and currently Java, .NET (C#) and C++ real time extend/TCP clients are available. In contrast to data clients, real time extend/TCP clients can sign up for notifications for events taking place in the grid and can take advantage of Coherence near caches and continuous queries.

real time cluster member client

Coherence Suite: Possibly a part of the Grid Edition cluster with all the capabilities of the Grid Edition except those associated with managing partitioned data and those associated with being an extend proxy. Real time cluster member clients may be Grid Edition members configured to have all their partitioned cache services storage-enabled attributes set to false, but unable to host connections from extend/TCP clients. Real time cluster member clients are always Java clients and cannot be configured as extend proxies for other real time clients (only Server Edition members can be configured as proxy servers for use by multilanguage data clients and real time extend/TCP clients). Real time cluster member clients are free in Grid Edition but must be paid for in Enterprise Edition.

realm

Java: In Java EE security, a security policy domain defined for a web or application server. A realm contains a collection of users, who may or may not be assigned to a group.

recovery

The process of recovering your environment in the event of data loss or corruption, host failure, or media failure, by restoring files that you previously backed up.

reference implementation (RI)

Java: The prototype or proof-of-concept implementation of a Java specification. All new or revised specifications must include an RI. A specification RI must pass all of the TCK tests for that specification.

refresh-ahead caching

Coherence Suite: A technique of updating the cache. Coherence lets you configure a cache to automatically and asynchronously reload (refresh) any recently accessed cache entry from the cache loader before its expiration. The result is that after a frequently accessed entry has entered the cache, a read against a potentially slow cache store when the entry is reloaded due to expiration has no effect on the application. The asynchronous refresh is triggered only when an object that is sufficiently close to its expiration time is accessed. If the object is accessed after its expiration time, then Coherence performs a synchronous read from the cache store to refresh its value.

region

JDeveloper/ADF: An ADF Faces component that renders the content of a bounded taskflow.

relationship field

Java: In the Java Persistence API, a persistent field of an entity whose type is the abstract schema type of the related entity.

reliable messaging

See Web Services Reliable Messaging (WS-ReliableMessaging).

remote interface

Java: An interface that defines the business and lifecycle methods of an enterprise bean in a way that allows remote access (where the client can run on a different machine and a different JVM from the enterprise bean it accesses), usually by means of the @Remote annotation.

Remote Method Invocation (RMI)

Java: An optional Java mechanism for calling instance methods on objects located on remote virtual machines (meaning, a virtual machine other than that of the caller).

RemoteCache service

Coherence Suite: A means of routing cache operations to a cache on the cluster.

RemoteInvocation service

Coherence Suite: A service that executes tasks on the remote Coherence cluster. At runtime, a connection is made to an extend proxy service and an InvocationService implementation is returned that executes synchronous Invocable tasks within the remote cluster JVM to which the client is connected.

render kit

Java: A set of javax.faces.render.Renderer classes that defines how JavaServer Faces component classes map to component tags that are appropriate for a particular client.

renderer

Java: A Java class that converts the internal representation of a JavaServer Faces UI component into the output stream associated with the response being created for a particular request.

replicated cache

Coherence Suite: Data that is replicated to each of the application server members in the cluster. This type of cache is recommended if faster read access is required but it is not suitable for write operations, because the data must be written to each of the members. The drawback of replicated caches is that they require a large amount of memory because every member has a copy of every object.

ReplicatedCache service

Coherence Suite: A synchronized replicated cache service that fully replicates all of its data to all cluster members that run the service. Replicated caches are often used to manage internal application metadata.

report configuration file

Coherence Suite: A report definition that results in the creation of a report file that displays management information for a particular set of metrics. Report configuration files must be referenced in a report group configuration file to be used at run time. The default report configuration files are located in the /reports directory of the coherence.jar file and are referenced by the default report group configuration file.

report group configuration file (report-group.xml)

Coherence Suite: List of the name and location of report definition files and the output directory where reports are written.

Representational State Transfer (REST)

REST is an architectural style that provides a set of design rules for creating stateless services that are viewed as resources, or sources of specific information (data and functionality). Each resource can be identified by its unique Uniform Resource Identifiers (URIs). REST is a simple interface that transmits data over a standardized interface (such as HTTP) without an additional messaging layer, such as SOAP.

request method designator

Java: A JAX-RS runtime annotation with @HttpMethod or with @GET, @PUT, @POST, or @DELETE. These annotations are used to identify the HTTP request method to be handled by a resource method.

request scope

Java: A bean scope whose duration is a user's interaction with a web application in a single HTTP request. The request scope is defined by CDI and JavaServer Faces technology.

resource

Java: A program object that provides connections to systems, such as database servers and messaging systems. (A Java Database Connectivity resource is sometimes referred to as a data source.) This should not to be confused with a web resource.

resource adapter

Java: A software component that allows Java EE application components to access and interact with the underlying resource manager of an enterprise information system (EIS).

resource adapter module

Java: A resource adapter packaged in a Resource Archive (RAR) file. A resource adapter module may be packaged in an EAR file, but it is more commonly deployed separately.

resource method

Java: A method of a JAX-RS resource class annotated with a request method designator that is used to handle requests on the corresponding resource.

resource scope

Identity Management: For global policy attachments in OWSM, the resource scope identifies the location within the resource where the policy set containing policies should be applied. It is mapped to the attachTo attribute of the policy set and is used for conflict resolution when multiple policy sets exist.

resource type

OWSM: For global policy attachments in OWSM, the resource type, or subject type, identifies the type of endpoint to which the policy set applies.

Resources window

JDeveloper/ADF: A manager of resources available through application server, database, and file system connections.

REST

See Representational State Transfer (REST).

RESTful web service

Java: A web service that uses the Representational State Transfer (REST) architectural style. The REST style specifies constraints, such as the uniform interface, that when applied to a web service, enhance the performance, scalability, and modification properties for those services.

return

Java: A reserved Java keyword used to finish the execution of a method.

RI

Java: reference implementation (RI)

role (development)

Java: One of several functional areas into which the creation of Java EE products can be divided.

role (security)

An abstract name for the permission to access a particular set of resources in an application.

role mapping

Java: In Java EE security, mapping of security roles to users or groups.

Role Mapping provider

WebLogic Server: WebLogic Security Framework provider that determines what security roles apply to the principals stored in a subject when the subject is attempting to perform an operation on a WebLogic resource. Because this operation usually

involves gaining access to the WebLogic resource, Role Mapping providers are typically used with Authorization providers.

root application module

JDeveloper/ADF: An Oracle ADF application module that contains (logically) one or more other application module instances, as well as view object instances. The outermost application module is referred to as the root application module. At runtime, the root application module defines the Transaction object.

root resource class

Java: In JAX-RS, a POJO that is either annotated with @Path or has at least one method annotated with @Path or a request method designator.

row set iterator

JDeveloper/ADF: The manager of the current object and current range information in the Oracle ADF application. Iterator bindings directly bind to the default row set iterator of the default row set of any view object instance.

runtime deployment descriptor

Java: A deployment descriptor that is used to configure Java EE implementation-specific parameters.

rvalue expression

Java: An expression language (EL) expression that can only read a value, as opposed to an lvalue expression, which can read a value and write it to an external object.

S

SAAJ

See SOAP with Attachments API for Java (SAAJ).

SAML

See Security Assertion Markup Language (SAML).

SAML Token Profile

The SAML Token profile, defined in the Security Assertion Markup Language (SAML) Token Profile 1.1 specification, is part of the core set of WS-Security standards. It specifies how SAML assertions can be used for Web services security.

SAR file

SOA Suite: A SOA archive deployment unit. A SAR file is a special JAR file that requires a prefix of sca_. (For example, sca_OrderBookingComposite_rev1.0.jar). The SAR file packages binding components and service components, such as BPEL processes, business rules, human tasks, and mediator routing services, into a SOA composite application.

scope

The context within which an object can be used.

second-level cache

Java: In the Java Persistence API, a local store of entity data managed by the persistence provider to improve application performance.

secure conversation

See Web Services Secure Conversation Language (WS-SecureConversation)

secure property

JDeveloper/ADF: A property on an ADF Faces UI component that must not ever be set on the client.

Secure Sockets Layer (SSL)

Java: A protocol for transmitting data over the Internet using encryption and authentication, including the use of digital certificates and both public and private keys.

Security that is implemented at the transport layer. SSL allows web browsers and web servers to communicate over a secure connection. In this secure connection, the data is encrypted before being sent and then is decrypted upon receipt and before processing. Both the browser and the server encrypt all traffic before sending any data.

Security Assertion Markup Language (SAML)

An XML standard for exchanging authentication and authorization data between security domains as defined in the Security Assertion Markup Language (SAML) specification.

security attributes

A set of attributes associated with every principal. The security attributes have many uses, for example, access to protected resources and auditing of users. Security attributes can be associated with a principal by an authentication protocol.

security constraint

Java: In Java EE security, information that is used to define the access privileges to a collection of web resources using their URL mapping. A security constraint can be defined by using either annotations or a deployment descriptor.

security context

Java: The context provided by the EJB container or web server that allows components to access security information. For example, the components may use methods provided by the javax.ejb.EJBContext interface.

security domain

Java: See security policy domain; security realm.

security patch update

An iterative, cumulative patch consisting of security fixes.

security policy domain

Java: In Java EE security, a scope over which a common security policy is defined and enforced by the security administrator of the security service. Also called a realm.

security provider database

WebLogic Server: A database that contains the users, groups, security policies, roles, and credentials used by some types of security providers to provide security services. The security provider database can be the embedded LDAP server (as used by the WebLogic Server security providers), a properties file (as used by the sample security providers), or a production-quality database that you may already be using.

security realm

WebLogic Server: An entity that acts as a scoping mechanism. Each security realm consists of a set of configured security providers, users, groups, roles, and security policies. You can configure multiple security realms in a domain; however, only one can be the default (active) security realm.

security role reference

Java: In Java EE security, a mapping between the name of a role that is called from a web component using the isUserInRole(String role) method and the name of a security role that has been defined for the application.

Security Service Provider Interfaces (SSPIs)

WebLogic Server: A set of WebLogic Server packages that enables custom security providers to be developed and integrated with the WebLogic Server Security Service. These interfaces are implemented by the WebLogic security providers and custom security providers. The WebLogic Security Framework calls methods in these interfaces to perform security operations.

security view

Java: A set of security roles that is used to define method permissions and authentication mechanisms for a Java EE application.

seeded search

JDeveloper/ADF: In ADF Faces, a search whose criteria are already determined and from which the user can choose, or you can allow the user to add criteria and then save those searches.

Selector Tree

JDeveloper/ADF: A feature in the ADF Skin Editor that displays the list of style classes, global selector aliases, selectors, and at-rules for which you can configure properties to change the appearance of ADF Faces and ADF Data Visualization components.

self-referential

Java: In the Java Persistence API, a relationship between relationship fields in the same entity.

server certificate

A public key certificate that is associated with a server.

server-side at-rule

JDeveloper/ADF: An at-rule that the ADF skinning framework evaluates and then determines the style properties to render. Examples of this category of at-rule include @agent, @accessibility-profile, and @locale.

service component

SOA Suite: A component that implements the business logic or processing rules of a SOA composite application. Service components are the basic building blocks of SOA composite applications. Service components include BPEL processes, business rules, human tasks, BPMN processes, mediator routing services, and spring.

service component architecture (SCA)

SOA Suite: Provides the service details and their interdependencies to form composite applications. SCA enables you to represent business logic as reusable service components that can be easily integrated into any SCA-compliant application. The resulting application is known as a SOA composite application.

service-enabled application module

JDeveloper/ADF: An ADF application module published with a service interface connection to make it available as an external web service in an Oracle Fusion web application. The service interface makes view object instances of the application module available to service consumers.

service endpoint implementation

Java: A Java class that implements the methods that a client can call on a JAX-WS web service.

service endpoint interface

Java: A Java interface that declares the methods that a client can call on a JAX-WS web service.

service engine

SOA Suite: Executes the business logic of the respective service component within the SOA composite application (for example, a BPEL process). There are service engines for a BPEL process, human workflow, decision service, Oracle Mediator, BPMN process, and spring.

Service Infrastructure

SOA Suite: Provides the internal message transport infrastructure for connecting components and enabling data flow. The service infrastructure is responsible for routing messages along the wire connections between services, service components, and references.

service method

Java: Any method in a servlet class that provides a service to a client. The general pattern for a service method is to extract information from the request, access external resources, and then populate the response, based on that information.

service-oriented

A description of a web application that implements the endpoint of a web service. Presentation-oriented applications are often clients of service-oriented web applications.

service-oriented architecture (SOA)

SOA Suite: Provides an enterprise architecture that supports building connected enterprise applications to provide solutions to business problems. SOA facilitates the development of enterprise applications as modular business web services that can be easily integrated and reused, creating a flexible, adaptable IT infrastructure.

Service-oriented architecture (SOA) bundle

SOA Suite: A ZIP file that includes multiple SOA archive (SAR) profiles that you can deploy to an application server at the same time.

service request

Coherence Suite: The round-trip required to complete a task. A request begins the moment a client starts a task and includes the following: The time it takes to deliver the request to an executing member (server), the interval between the time the task is received and placed into a service queue until the task starts, the task execution time, and the time it takes to deliver a result back to the client.

service table

A database table that provides a way for service providers to publish endpoint information about their services, and for clients of these services to query and bind to these services. A service table is a single table in a database schema. There is one row for every endpoint that is published to it. The service table schema is initially created by the Oracle Fusion Applications Repository Creation Utility.

service task

Coherence Suite: A called object that executes on one or more members. The objects include filters, invocation agents (entry processors and aggregators), or single-pass agents (Invocable objects).

servlet

Java: A Java programming language class that dynamically processes requests and constructs responses.

Servlet Authentication filter

WebLogic Server: A unique implementation of the Java EE filter object that replaces container-based authentication. Servlet Authentication filters control the authentication conversation with the client by redirecting to a remote site to execute the login, extracting login information out of the query string, and negotiating a login mechanism with the browser.

servlet path

Java: For a servlet, the path section that corresponds to the component alias that activated a particular request. This path starts with a forward slash (/).

session

Java: For a servlet, the state maintained between an HTTP client and an HTTP server. The session persists for a specified time period, across more than one connection or page request from the user. A session usually corresponds to one user, who may visit a site many times.

session bean

Java: An enterprise bean that represents a transient conversation with a client. When the client stops executing, the session bean and its data are gone. A session bean encapsulates business logic that can be called programmatically by a client over local, remote, or web service client views.

Session Reaper

Coherence Suite: In Coherence*Web, the means of ending a session. The Session Reaper provides a service similar to the JVM Garbage Collection (GC) capability: the Session Reaper is responsible for destroying any session that is no longer used, which is determined by when that session times out.

session scope

Java: A bean scope whose duration is a user's interaction with a web application across multiple HTTP requests. Session scope is defined by CDI and JavaServer Faces technology.

shared application module

JDeveloper/ADF: A grouping of view object instances when reuse of static data across the Oracle ADF application is required. For example, a shared application module can group view instances that access lookup data, such as a list of countries.

short

Java: A reserved Java keyword used to declare a field that can hold a 16-bit signed two's complement integer.

signature test

Java: A TCK test for a Java technology implementation to check that all the necessary API members are present and that there are no extra members that violate the extending of the API.

SIM

Java: Subscriber Identity Module or Subscriber Identification Module (SIM)

Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)

OWSM: A standard that enables a client and a service to negotiate a method to use for authentication.

Simple Object Access Protocol (SOAP)

Java: An XML-based protocol that enables objects of any type to communicate in a distributed environment, it is most commonly used to develop web services.

singleton session bean

Java: A session bean that is instantiated once per application and exists for the life cycle of the application. Singleton session beans are designed for circumstances in which a single enterprise bean instance is shared across and concurrently accessed by clients.

smart card

Java: A card that stores and processes information through the electronic circuits embedded in the silicon substrate of its body. Unlike magnetic stripe cards, smart cards carry both processing power and information. They do not require access to remote databases at the time of a transaction.

smart list

JDeveloper/ADF: A subset of the drop-down list of values of an inputComboboxListOfValues component. It is an additional filter applied to the full list to display only the most likely choices.

SOA composite application

SOA Suite: An assembly of service binding components, service components, and reference binding components designed and deployed in a single application. Wiring between the components enables message communication. The details for a composite are stored in the composite.xml file.

SOA Infrastructure

SOA Suite: A Java EE-compliant application running in Oracle WebLogic Server. The application manages SOA composite applications and their lifecycle, service engines, and binding components. You deploy SOA composite applications to a partition of your choice on the SOA Infrastructure.

SOAP

Java: Simple Object Access Protocol

SOAP over JMS transport

Typically, web services and clients communicate using SOAP over HTTP/S as the connection protocol. Using SOAP over JMS transport, web services and clients communicate using JMS destinations instead of HTTP connections, offering the following benefits: reliability, scalability, and quality of service. As with web service reliable messaging, if the server goes down while the method invocation is still in the queue, it will be handled as soon as server is restarted. When a client invokes a web service, the client does not wait for a response, and the execution of the client can continue.

SOAP with Attachments API for Java (SAAJ)

Implementation that developers can use to produce and consume messages conforming to the SOAP 1.1 specification and SOAP with Attachments notes, as defined in SOAP with Attachments API for Java (SAAJ) specification.

Sockets Direct Protocol (SDP)

Java: A networking protocol developed to support stream connections over the InfiniBand fabric. SDP support was introduced to the JDK 7 release of the Java Platform, Standard Edition (Java SE Platform) for applications deployed in the Oracle Solaris operating system (Solaris OS) and on Linux operating systems.

source compatibility

Java: The ability of an application to compile without errors with any compatible API. While binary compatibility is important, it cannot guarantee that an application in binary form as a set of class files can be recompiled without error with any other binary-compatible API. The signature-test, source-compatibility check verifies that any application that compiles without error with a compatible API will compile without error with all other source compatible APIs. This is a stricter check than the binary-compatibility check and the tool performs it by default.

speciality color

JDeveloper/ADF: A category of global selector alias that define color properties for ADF skins. The global selector aliases in this category do not derive the hue value from anchor colors and are not anchor colors for other colors.

specialization

Java: In CDI, the extension of one bean class by another so that at runtime the specialized bean completely replaces the other bean. A specialized bean has the annotation javax.enterprise.inject.Specializes.

Split session state model

Coherence Suite: A session state model used by Coherence*Web. This model extends the traditional model, but separates the larger session attributes into independent physical entities.

SPNEGO

OWSM: See Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)

spring framework

SOA Suite: Integrates components that use Java interfaces instead of WSDL files into SOA composite applications.

SSL

Java: Secure Sockets Layer

SSL handshake

In security, a client browser's acceptance of a server certificate.

SSPIs MBean

WebLogic Server: The interfaces used by Oracle to generate MBean types for the WebLogic security providers, and from which you generate MBean types for custom security providers. Security Service Provider Interface (SSPIs) MBeans may be required (for configuration) or optional (for management).

stale read

Java: In the Java Persistence API, a situation in which the underlying data may have changed in the database tables, but the value in the second-level cache has not.

standalone domain

A container for system components, such as Oracle HTTP Server. It has a directory structure similar to an Oracle WebLogic Server domain, but it does not contain an Administration Server or a Managed Server. It can contain one or more instances of system components of the same type, such as Oracle HTTP Server, or a mix of system component types. WebLogic Management Framework, which includes tools, such as the Configuration Wizard, pack and unpack, WLST, and Node Manager, can operate on standalone domains.

state field

Java: In the Java Persistence API, a persistent field of an entity.

stateful session bean

Java: A session bean that maintains a conversational state with the client through its instance variables.

stateless session bean

Java: A session bean that does not maintain a conversational state with the client. When a client calls the methods of a stateless session bean, the bean instance variables may contain a state specific to that client, but only for the duration of the call.

static

Java: A reserved Java keyword used to declare a field, method, or inner class as a class field.

static query

Java: In the Java Persistence API, a query that is defined in metadata by using the javax.persistence.NamedQuery annotation.

stereotype

Java: In CDI, a kind of annotation applied to a bean that incorporates other annotations. It specifies a default scope and zero or more interceptor bindings. A bean annotated with a particular stereotype will always use the specified annotations, so that you do not have to apply the same annotations to many beans.

Structure window

JDeveloper/ADF: A display of a structural view of the data in the document currently selected in the active Oracle JDeveloper window of those windows that participate in providing structure. These windows include the diagrams, the editors and viewers, and the Properties window.

structured attribute

JDeveloper/ADF: In a data control, a returned object that is neither a Java primitive type (represented as an attribute) nor a collection of any type. An example of a structured attribute would be a domain, which is a developer-created data type used to simplify application maintenance.

subresource locator

Java: A JAX-RS resource class method that is annotated with @Path and is used to locate subresources of the corresponding resource. It returns an object that will handle an HTTP request.

subresource method

Java: A JAX-RS resource class method that is annotated with @Path and is used to handle requests on a subresource of the corresponding resource. It handles an HTTP request directly.

super

Java: A reserved Java keyword used to access members of a class inherited by the class in which it appears.

Swing

Java: A Java GUI package that extends the AWT toolkit to create lightweight (platform-independent) components. Most AWT components, which are platform-dependent, have a corresponding Swing component: for example the Swing JButton class corresponds to the AWT Button class. The most recent Java GUI package is JavaFX.

switch

Java: A reserved Java keyword used in conjunction with case and default to create a switch statement, which evaluates a variable, matches its value to a specific case, and executes the block of statements associated with that case. If no case matches the value, then the optional block labeled default is executed, if provided.

synchronized

Java: A reserved Java keyword used in the declaration of a method or code block to acquire the mutual exclusion (mutex) lock for an object while the current thread executes the code.

synchronous web service

SOA Suite: Provides an immediate response to an invocation. A BPEL process service component can connect to synchronous web services through a partner link, send data, and then receive the reply in the same synchronous invocation.

system component

A manageable process that is not deployed in a Java application container. A system container does not run in an application server such as Oracle WebLogic Server. Oracle HTTP Server is an example of a system component.

system event

Java: In JavaServer Faces technology, an event that is generated by an Object rather than by a UIComponent. System events are generated during the execution of an application at predefined times. They are applicable to the entire application rather than to a specific component.

system exception

Java: An exception thrown by an enterprise bean that indicates a problem with the services that support an application. For example, a connection to an external resource cannot be obtained, or an injected resource cannot be found. If a system exception occurs within a transaction, the EJB container rolls back the transaction.

T

Tangosol Cluster Management Protocol (TCMP)

Coherence Suite: An IP-based protocol that is used to discover cluster members, manage the cluster, provision services, and transmit data. The default TCMP configuration uses a combination of UDP/IP unicast and multicast. However, TCMP can be configured to use UDP/IP unicast only, TCP/IP, SDP/IP, and SSL over TCP/IP or SDP/IP. Moreover, on Exalogic systems, TCMP can be configured to use reliable message bus transport protocols such as IMB, LWMB, TMB, TMBS, SDMB, or SDMBS.

target class

Java: A class associated with a Java EE interceptor.

target component

JDeveloper/ADF: In ADF Faces, the UI component that is rerendered when any event occurs on another component, referred to as the trigger component.

TCK

Java: Technology Compatibility Kit

TCMP

Coherence Suite: Tangosol Cluster Management Protocol. An IP-based protocol that is used to discover cluster members, manage the cluster, provision services, and transmit data. The default TCMP configuration uses a combination of UDP/IP unicast and multicast. However, TCMP can be configured to use UDP/IP unicast only, TCP/IP, SDP/IP, and SSL over TCP/IP or SDP/IP. Moreover, on Exalogic systems, TCMP can be configured to use reliable message bus transport protocols such as IMB, LWMB, TMB, TMBS, SDMB, or SDMBS.

TCP/IP

Java: Transmission Control Protocol/Internet Protocol (TCP/IP)

Technology Compatibility Kit (TCK)

Java: The suite of tests, tools, and documentation that allows an implementor of a Java specification to determine if the implementation complies with the specification.

template

Java: A Facelets page that acts as the base for the other pages in a JavaServer Faces application. By using templates, you can reuse code and avoid re-creating similarly constructed pages.

tenant isolation

The ability to prevent multiple clients (tenants) that are all using a single application from seeing each other's data. In a Software as a Service (SaaS) environment, a single application can be deployed for multiple clients, or tenants. Each tenant is isolated from the others in that each tenant may have access to different features in the application, private access to their own data in a shared data store, or access to other tenant-specific functionality. For example, a large company may develop a single payroll application to be used by multiple divisions. Each division has access to its own data and to shared data, but no division can see any other division's data.

test to production

The process of copying your environment from a source environment, such as a test environment, to another environment, such as a production environment. It does not move transactional data.

thin client

See web client.

this

Java: A reserved Java keyword used to represent an instance of the class in which it appears.

throws

Java: A reserved Java keyword used in method declarations to specify which exceptions are not handled within the method, but instead are passed to the next higher level of the program.

tool provider

Java: The company or person who creates development, assembly, and packaging tools used by Java EE component providers, assemblers, and deployers.

topic

Java: In the JMS API, a destination used in publish/subscribe messaging, in which each message can have multiple consumers. A topic functions somewhat like a bulletin board. Publishers and subscribers are generally anonymous and can dynamically publish or subscribe to the content hierarchy. The system takes care of distributing the messages arriving from multiple publishers to its multiple subscribers. Topics retain messages only as long as it takes to distribute them to current subscribers.

TopLink Data Services

Toplink: A Java Persistence API for RESTful Services (JPA-RS), plus additional functionality and integration.

TopLink Essentials

Toplink: The Reference Implementation of the EJB 3.0 Java Persistence API (JPA), developed by Oracle and contributed to the open source community. TopLink Essentials provided the JPA 1.0 functionality for the EJB 3.0 Reference Implementation. It is superseded by TopLink's implementation of JPA, provided by EclipseLink.

TopLink Grid

Coherence Suite: An integration between Oracle Coherence and Oracle TopLink that allows some or all of a domain model to be stored in a Coherence data grid and also allows Coherence to be used as the TopLink level 2 cache.

Toplink: A facility for integrating the TopLink JPA implementation, provided by EclipseLink, with Oracle Coherence, an In-Memory Data Grid. TopLink Grid provides the means for scaling JPA applications using Oracle Coherence. Applications can be scaled in a number of ways, ranging from using Coherence as a distributed shared (L2) cache to directing Java Persistence query language (JPQL) queries to Coherence for parallel execution across the grid to reduce database load.

traditional session state model

Coherence Suite: A model used by Coherence*Web that stores all session states as a single entity, but serializes and deserializes attributes individually.

transaction

An indivisible unit of work into which you can group a series of operations, so that the operations either all succeed or all fail. A transaction ends with either a commit or a rollback operation.

transaction attribute

Java: An attribute of an enterprise bean class or method that controls the scope of a transaction. The javax.ejb.TransactionAttribute annotation can have any of the following values: REQUIRED, REQUIRES_NEW, MANDATORY, NOT_SUPPORTED, SUPPORTS, NEVER.

transformation

SOA Suite: The process of mapping source schema elements to target schema elements in an XSL map file with the XSLT Map Editor.

transient

Java: A reserved Java keyword used to declare that an instance field is not part of the default serialized form of an object.

transient attribute

JDeveloper/ADF: An attribute of an ADF Business Components entity object or view object that is not derived from the underlying data source, such as a database table. The value of a transient attribute is derived from other attributes either by using Groovy expressions or through Java code. Transient attributes are used as temporary value holders and to implement business logic.

transient process

SOA Suite: A type of BPEL process that does not incur any intermediate dehydration points during process execution. If there are unhandled faults or there is system downtime during process execution, the instances of a transient process do not leave a trace in the system. Instances of transient processes cannot be saved in-flight (whether they complete normally or abnormally). Transient processes are typically short-lived, request-response style processes. The synchronous process you design in Oracle JDeveloper is an example of a transient process.

transition component

JDeveloper/ADF: In ADF Faces, with regard to geometry management of the components, a component that can be stretched but does not stretch its children. A transition component must always be used between a component that stretches its children and a component that does not stretch.

Transmission Control Protocol/Internet Protocol (TCP/IP)

Java: A fundamental Internet protocol that provides for reliable delivery of streams of data from one host to another.

trigger component

JDeveloper/ADF: In ADF Faces, the UI component whose event causes another component (referred to as the target component) to be rerendered.

trinidad-config.xml file

JDeveloper/ADF: A configuration file that determines the ADF skin for a Fusion web application. It can also configure other application features such as, for example, accessibility and localization.

trinidad-skins.xml file

JDeveloper/ADF: The registry of ADF skins available for use in an Oracle Fusion web application.

true

Java: A reserved Java boolean literal value.

truststore

Java: In Java security, a repository of certificates from parties with which you expect to communicate or from Certificate Authorities that you trust to identify parties. The truststore is used by the client to verify the certificate that is sent by the server.

try

Java: A reserved Java keyword that defines a block of statements that have exception handling.

try-with-resources

Java: A try statement that declares one or more resources. A resource is an object that must be closed after the program is finished with it. The try-with-resources statement ensures that each resource is closed at the end of the statement. Any object that implements java.lang.AutoCloseable, which includes all objects that implement java.io.Closeable, can be used as a resource.

type erasure

Java: In the generics feature of the Java language, type erasure is a compiler process that does the following: Replaces all type parameters in generic types with their bounds or Object if the type parameters are unbounded. The produced bytecode, therefore, contains only ordinary classes, interfaces, and methods. Inserts type casts if necessary to preserve type safety. Generates bridge methods to preserve polymorphism in extended generic types.

U

UDDI

See Universal Description, Discovery, and Integration (UDDI).

UI hint

JDeveloper/ADF: ADF Business Components metadata that defines label text, tooltip, and format mask hints for entity object and view object attributes. UI hints that you define on the business service layer can be used by UI components in Oracle ADF clients. This is sometimes referred to as a control hint.

unattended account

Identity Management: An account that is never used by an end user.

unbounded taskflow

JDeveloper/ADF: A set of activities, Oracle ADF control flow rules, and managed beans that interact to allow a user to complete a task. An unbounded taskflow has one or more points of entry.

unbounded wildcards

Java: In the generics feature of the Java language, the type that is specified using the wildcard character (?), for example, List?. This is called a list of unknown type.

unboxing

Java: The automatic conversion that the Java compiler makes when converting an instance from a wrapper class (Integer, Double, Character, and so on) to its corresponding primitive type (int, double, char, and so on).

unchecked exception

Java: An event that occurs during the execution of a program and that disrupts the normal flow of the program's instructions. An unchecked exception, also called a runtime exception, is one that a program does not catch because there is little the program can do to recover from such a problem. OUT OF MEMORY is an example of an unchecked exception.

Uniform Resource Identifier (URI)

Java: A compact string of characters used to identify or name an abstract or physical resource. A URI can be further classified as a uniform resource locator (URL), a uniform resource name (URN), or both.

unit test

SOA Suite: Automates the testing of SOA composite applications by simulating the interaction between a SOA composite application and its web service partners before deployment to a production environment. You can also create test cases for testing

BPEL process service components included in the SOA composite application. This testing helps to ensure that a process interacts with web service partners as expected by the time it is ready for deployment to a production environment.

Universal Description, Discovery, and Integration (UDDI)

Standard for describing a web service; registering a web service in a well-known registry; and discovering other registered web services as defined in the Universal Description, Discovery, and Integration (UDDI) specification.

upgrade

The process of moving from a previous major version to a new major version. For example, an upgrade would be required to move from Oracle Fusion Middleware 11g to Oracle Fusion Middleware 12c.

Upgrade Assistant

A utility used to upgrade schemas and configuration information from one major version to another, such as Oracle Fusion Middleware 11g to Oracle Fusion Middleware 12c.

upper bounded wildcards

Java: In the generics feature of the Java language, wildcards that are used to relax the restrictions on a variable.

URI

Java: Uniform Resource Identifier

URI path parameter

Java: A JAX-RS request parameter that is extracted from the request URI. The parameter names correspond to the URI path template variable names specified in the @Path class-level annotation. A URI parameter is specified by using the javax.ws.rs.PathParam annotation in the method parameter arguments.

URI path template

Java: In JAX-RS, a URI that has variables embedded within the URI syntax.

URL path

Java: The Uniform Resource Locator (URL) for a web application, which contains the context root and, optionally, a URL pattern.

user (security)

Java: In Java EE security, an individual or application program identity that has been defined in the Java EE server. In a web application, a user can associate that identity with a set of roles that entitle the user to access all resources protected by those roles. Users can be associated with a group.

user data constraint

Java: In Java EE security, the part of a security constraint that specifies how data is protected when transported between a client and a server.

using page

Java: In JavaServer Faces technology, the web page that uses a composite component.

V

value binding

JDeveloper/ADF: A binding used by UI components that display data. Value bindings range from the most basic variety that work with a simple text field to more sophisticated list and tree bindings that support the additional needs of list, table, and tree UI controls. Types of value bindings include: attribute, list, table, tree, graph, and button bindings.

value-change event

Java: An event that occurs when the user changes the value of a JavaServer Faces component represented by javax.faces.component.UIInput or one of its subclasses. Action events and value-change events are types of application events.

value expression

Java: An expression language (EL) expression that references data, as opposed to a method expression.

value extractor

Coherence Suite: A means of accepting as input a value object and returning an attribute of that object.

value iterator

JDeveloper/ADF: An ADF Model iterator pointing to a collection that contains only one data object whose attributes are the binding container variable.

variable iterator

JDeveloper/ADF: A binding to an iterator that exposes all the variables in the binding container to the other bindings. While there is an iterator binding for each collection, there is only one variable iterator binding for all variables used on the page.

view

Java: The tree of components on a JavaServer Faces page.

view accessor

JDeveloper/ADF: An ADF Business Components object that points from an entity object attribute (or view object) to a destination view object or shared view instance in the same application workspace. The view accessor returns a row set that by default contains all the rows from the destination view object.

view criteria

JDeveloper/ADF: An ADF Business Components object that allows developers to adapt a query to a particular usage by filtering the row set of the view object instance in an Oracle ADF application.

view link

JDeveloper/ADF: An Oracle ADF Business Components object that represents the relationship between two view objects, which is usually, but not necessarily, based on a foreign-key relationship between the underlying database tables.

view link accessor

JDeveloper/ADF: An Oracle ADF Business Components object that identifies an accessor attribute that the master collection uses at runtime to return the detail collection row set in an Oracle ADF application.

view link consistency

JDeveloper/ADF: Runtime logic provided by the Oracle ADF Business Components framework. This logic ensures when a new row created in one view object instance will be automatically added to the row sets of others view instances based on the same entity object. This behavior ensures that the Oracle ADF client consistently reflects new rows in different application pages for a pending transaction.

view object

JDeveloper/ADF: An ADF Business Components object that represents a SQL query and simplifies working with its results. The SQL query is used to join, project, filter, sort, and aggregate data into the shape required by the end-user task being represented in the user interface.

view scope

JDeveloper/ADF: An ADF-defined scope in the page life cycle. An object in this scope is available until the ID for the current view changes. Use view scope to hold values for a given page. Unlike the JavaServer Faces view scope, objects stored in the ADF Faces view scope will survive page refreshes and redirections to the same view ID.

virtual authentication device

Identity Management: A personalized device for entering a password or PIN or an authentication credential entry device to protect users while interacting with a protected web application. The virtual authentication devices harden the process of entering and transmitting authentication credentials and provide users with verification that they are authenticating on the valid application.

void

Java: A reserved Java keyword used to declare that a method does not return any value.

volatile

Java: A reserved Java keyword used in field declarations to specify that the variable is modified asyncronously by concurrently running threads.

W

W3C

Java: World Wide Web Consortium

WADL

See Web Application Description Language (WADL)

WAP

Java: Wireless Application Protocol

WAR-scoped Coherence cluster members

Coherence Suite: For deployments on non-WebLogic Server application servers, each deployed web application becomes its own Coherence member. This configuration produces the largest number of Coherence members in the cluster (one for each deployed WAR file that uses Coherence) and because the Coherence library is deployed in the web application's class path, there will be as many copies of the Coherence classes loaded as there are deployed WAR files.

web application

A dynamic extension of a web server or application server. A web application can be presentation-oriented or service-oriented.

Web Application Description Language (WADL)

An XML-based file format that describes a RESTful web services application.

web client

A client that consists of two parts: Dynamic web pages that contain various types of markup language (HTML, XML, and so on), which are generated by web components running in the web tier A web browser, which renders the pages received from the server A web client is sometimes called a thin client. Thin clients usually do not query databases, execute complex business rules, or connect to legacy applications. When you use a thin client with a Java EE application, such heavyweight operations are off-loaded to enterprise beans executing on the Java EE server, where they can take advantage of the security, speed, services, and reliability of Java EE server-side technologies.

web component

Java: A component that provides the dynamic extension capabilities for a web server. Java EE web components can be Java servlets, web pages implemented with JavaServer Faces technology, web service endpoints, or JavaServer Pages.

web container

Java: The container that manages the execution of web pages, servlets, and some EJB components for Java EE applications. Web components and their container run on the Java EE server. The web container provides such services as request dispatching, security, concurrency, and lifecycle management. It also gives web components access to such APIs as naming, transactions, and email.

web module

Java: The smallest deployable and usable unit of web resources. A web module contains web components, static web content files such as images, and (usually) a deployment descriptor. Web modules are packaged as JAR files with a .war (web archive) extension.

web resource

Java: A static web content file, such as an image file, that is included in a web module.

web resource collection

Java: In Java EE security, the part of a security constraint that specifies a list of URL patterns (the part of a URL after the host name and port that you want to constrain) and HTTP operations (the methods within the files that match the URL pattern that you want to constrain) that describe a set of resources to be protected.

web service

Web services are loosely coupled, distributed environments that allow companies to integrate heterogeneous applications within the enterprise or expose business functions to their customers and partners over the Internet. Because you access web services using standard Web protocols such as XML or HTTP, the diverse and heterogeneous applications on the Web (which typically already understand XML and HTTP) can automatically access web services and communicate with each other.

A client and server application that communicates over the World Wide Webs (WWW) HyperText Transfer Protocol (HTTP). A web service can be a software component provided through a network-accessible endpoint.

web service endpoint

The web service address, specified by a Uniform Resource Identifier (URI), that may be used to communicate with an instance of a web service.

web service port

See web service endpoint.

Web Services Addressing (WS-Addressing)

Transport-neutral mechanisms to address web services and messages as described in the Web Services Addressing (WS-Addressing) specification. In particular, the specification defines a number of XML elements used to identify Web service endpoints and to secure end-to-end endpoint identification in messages.

Web Services Atomic Transaction

The web services atomic transaction framework enables interoperability with other external transaction processing systems, such as WebSphere, Microsoft .NET, and so on through the support of the following specifications: Web Services Atomic Transactions (WS-Atomic Transaction) and Web Services Coordination (WS-Coordination). These specifications define an an extensible framework for coordinating distributed activities among a set of participants.

Web Services Definition Language (WSDL)

An XML-based language providing a model for describing SOAP-based web services as described in the Web Services Description Language (WSDL) 1.1 document submission.

Web Services Federation Language (WS-Federation)

Mechanisms that allow different security realms to federate, such that authorized access to resources managed in one realm can be provided to security principals whose identities and attributes are managed in other realms. This includes mechanisms for brokering of identity, attribute, authentication and authorization assertions between realms, and privacy of federated claims as defined in the Web Services Federation Language (WS-Federation) Version 1.2 specification.

Web Services Inspection Language (WS-Inspection)

Provides an XML format for assisting in the inspection of a site for available services as described in the Web Services Inspection Language (WS-Inspection) 1.0 specification.

Web Services Metadata Exchange (WS-MetadataExchange)

Part of the WS-Federation roadmap which allows retrieval of metadata about a web service endpoint as described in the Web Services Metadata Exchange (WS-MetadataExchange) specification.

web services policy

An XML document that defines the capabilities and requirements of a web service such as whether and how a message must be secured, whether and how a message must be delivered reliably, and so on. A policy is expressed as one or more policy assertions representing a Web service's capabilities or requirements. For example, a policy assertion may stipulate that a request to a Web service be encrypted.

Web Services Policy Attachment (WS-PolicyAttachment)

Mechanisms for associating policies with the subjects to which they apply as defined in the Web Services Policy 1.5 - Attachment and the Web Services Policy 1.2 - Attachment (WS-PolicyAttachment) specifications.

Web Services Policy Framework (WS-Policy)

General purpose model and corresponding syntax to describe and communicate the policies of a Web service as described in the Web Services Policy 1.5 - Framework and Web Services Policy 1.2 - Framework (WS-Policy) specifications.

Web Services Reliable Messaging (WS-ReliableMessaging)

Implementation that enables two endpoints (web service and client) running on different server instances to communicate reliably in the presence of failures in software components, systems, or networks. WS-ReliableMessaging is defined in the Web Services Reliable Messaging (WS-ReliableMessaging) 1.1 and Web Services Reliable Messaging Protocol (WS-ReliableMessaging) 1._0 specifications.

Web Services Reliable Messaging Policy Assertion (WS-RM Policy)

Policy assertion for reliable messaging for use with WS-Policy and WS-ReliableMessaging, as defined in Web Services Reliable Messaging Policy Assertion (WS-RM Policy) specification.

Web Services Secure Conversation Language (WS-SecureConversation)

Standard built on top of the WS-Security and WS-Policy models to provide secure communication between services. The Web Services Secure Conversation Language (WS-SecureConversation) specification defines mechanisms for establishing and sharing security contexts, and deriving keys from security contexts, to enable a secure conversation.

Web Services Security (WS-Security)

Standard set of SOAP [SOAP11, SOAP12] extensions that can be used when building secure Web services to implement message content integrity and confidentiality, as described in the WS-Security 1.1 OASIS standard.

Web Services Security Policy (WS-SecurityPolicy)

Set of security policy assertions for use with the WS-Policy framework, as defined in Web Services Security Policy (WS-SecurityPolicy) 1.3, 1.2, and 1.1 specifications.

WebLogic component

WebLogic Server: One of a number of Java EE component technologies, which include servlets, JavaServer Pages, and Enterprise JavaBeans. To build a WebLogic Server

application, you must create and assemble components, using the service APIs when necessary. Components are executed in the WebLogic Server web container or EJB container. Web components provide the presentation logic for browser-based Java EE applications. EJB components encapsulate business objects and processes.

WebLogic container

WebLogic Server: A WebLogic Server component that provides the lifecycle support and services defined by the Java EE specifications so that the components you build do not have to handle underlying lifecycle details.

WebLogic domain

A logically related group of Oracle WebLogic Server resources, which include an Administration Server and, typically, one or more Managed Servers and logically related resources and services that are managed collectively as one unit.

WebLogic home directory

WebLogic Server: The root directory of the WebLogic Server installation.

WebLogic Java EE service

WebLogic Server: One of a number of Java EE services, which include access to standard network protocols, database systems, and messaging systems. To build a WebLogic Server application, you must create and assemble components, using the service APIs when necessary. Web applications and EJBs are built on Java EE application services, such as JDBC, Java Messaging Service (JMS), and Java Transaction API (JTA).

WebLogic JDBC data source

WebLogic Server: A means of database access and database connection management. Each data source contains a pool of database connections that are created when the data source is created and at server startup.

WebLogic Management Framework

A framework that provides heterogeneous management capabilities for Oracle Fusion Middleware products that require basic administrative capabilities. Its capabilities include start, stop, configuration settings, and other such basic product lifecycle operations through a common command line, API, and user interface. WebLogic Management Framework includes specific WebLogic Server and Coherence features. WebLogic Management Framework manages both WebLogic Server domains, which can contain Java components and system components, and standalone domains, which contain system components.

WebLogic MBeanMaker

WebLogic Server: A command-line utility that takes an MBean Definition File (MDF) as input and output files for an MBean type.

WebLogic Messaging Bridge

WebLogic Server: A forwarding mechanism that provides interoperability between WebLogic JMS implementations, and between JMS and other messaging products.

WebLogic resource

WebLogic Server: An entity that is accessible from WebLogic Server, such as events, servlets, JDBC connection pools, JMS destinations, JNDI contexts, connections, sockets, files, and enterprise applications and resources, such as databases.

WebLogic Scripting Tool

WebLogic Server: A command-line scripting interface used for managing and monitoring active or inactive WebLogic Server domains.

WebLogic Security Framework

WebLogic Server: Interfaces in the weblogic.security.service package that unify security enforcement and present security as a service to other WebLogic Server components.

WebLogic security provider

WebLogic Server: Software modules that can be used by a WebLogic Server security realm to provide security services (such as authentication, authorization, auditing, and credential mapping) to applications. A security provider consists of runtime classes and MBeans, which are created from SSPIs and MBean types, respectively. Security providers are WebLogic security providers (provided with WebLogic Server) or custom security providers.

WebLogic Security Service

WebLogic Server: A WebLogic Server subsystem that implements the security architecture that comprises three major components: the WebLogic Security Framework, the Security Service Provider Interfaces (SSPIs), and the WebLogic security providers.

WebLogic Server

WebLogic Server: A scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. The WebLogic Server infrastructure supports the deployment of many types of distributed applications and is an ideal foundation for building applications based on service-oriented architecture (SOA).

WebLogic Server Administration Console

WebLogic Server: A web application hosted by the Administration Server that is used for managing and monitoring an active domain.

WebLogic Store-and-Forward

WebLogic Server: A means of delivering messages reliably between applications that are distributed across WebLogic Server instances.

WebLogic web service policy

A web service policy provided by Oracle WebLogic Server. It is recommended that you use OWSM policies over WebLogic web services whenever possible. You cannot mix your use of OWSM and WebLogic web service policies on the same web service.

weblogic.Deployer

WebLogic Server: A Java-based deployment tool that provides administrators and developers command-line deployment operations.

welcome files

Java: A list of files that the web container will use for appending to a request for a URL (called a valid partial request) that is not mapped to a web component.

wildcard capture

Java: A supposition by the compiler about the type of wildcard intended in a Java program that uses generic types. When compiling a Java program that uses generics,

in some cases the compiler infers the type of a wildcard. For example, a list may be defined as List? but, when evaluating an expression, the compiler infers a particular type from the code. This scenario is known as wildcard capture.

wildcarded host name verifier

WebLogic Server: A class that works the same as the default WebLogic Server host name verifier; however, the wildcarded host name verifier also accepts the asterisk character in the host name that is obtained from the certificate's Subject CommonName attribute; and SubjectAlternativeName dnsName (SAN) certificates.

Windows NT Authentication provider

WebLogic Server: WebLogic Security Framework authentication provider that uses Windows NT users and groups for authentication purposes

wire

SOA Suite: Wires connect service binding components, service components, and reference binding components into a complete SOA composite application.

WKA

Coherence Suite: Coherence Well Known Addresses. A feature that allows cluster members to discover and join a cluster using unicast instead of multicast. WKA is most often used when multicast networking is undesirable or unavailable in an environment or when an environment is not properly configured to support multicast. All cluster multicast communication is disabled if WKA is enabled.

WLST

WebLogic Server: WebLogic Scripting Tool

World Wide Web Consortium (W3C)

Java: The main international standards organization for the World Wide Web (abbreviated WWW or W3).

wrapper class

Java: The placement of a primitive data type in an object. Often, the wrapping is done by the compiler. If you use a primitive data type where an object is expected, then the compiler boxes the primitive data type in its wrapper class for you. Similarly, if you use a number object when a primitive data type is expected, then the compiler unboxes the object for you.

write-behind caching

Coherence Suite: A technique of updating the cache. Modified cache entries are asynchronously written to the data source after a configured delay. Note that this applies only to cache inserts and updates. Cache entries are removed synchronously from the data source.

write-through caching

Coherence Suite: A technique of updating the cache. When the application updates a piece of data in the cache, then the operation does not complete until Coherence has gone through the cache store and successfully stored the data to the underlying data source.

WS-Addressing

See Web Services Addressing (WS-Addressing).

WS-Federation

See Web Services Federation Language (WS-Federation)

WS-Inspection

See Web Services Inspection Language (WS-Inspection)

WS-MetadataExchange

See Web Services Metadata Exchange (WS-MetadataExchange)

WS-Policy

See Web Services Policy Framework (WS-Policy).

WS-PolicyAttachment

See Web Services Policy Attachment (WS-PolicyAttachment).

WS-ReliableMessaging

See Web Services Reliable Messaging (WS-ReliableMessaging).

WS-RM Policy

See Web Services Reliable Messaging Policy Assertion (WS-RM Policy)

WS-SecureConversation

See Web Services Secure Conversation Language (WS-SecureConversation)

WS-Security

See Web Services Security (WS-Security).

WS-SecurityPolicy

See Web Services Security Policy (WS-SecurityPolicy).

WS-Trust

Extensions that build on [WS-Security] to provide a framework for requesting and issuing security tokens, and to broker trust relationships as defined in the WS-Trust specification.

WSDL

See Web Services Definition Language (WSDL)

X

XA transaction support

SOA Suite: Enables multiple resources (such as databases, application servers, message queues, transactional caches) to be accessed within the same transaction.

XML

Java: Extensible Markup Language

XML Encryption

Process for encrypting data and representing the result in XML as defined in the XML Encryption Syntax and Processing specification.

XML Schema

Java: A set of rules to which an XML document must conform to be considered valid.

XML Signature

Rules and syntax for processing XML digital signatures as described in the XML Signature Syntax and Processing specification.