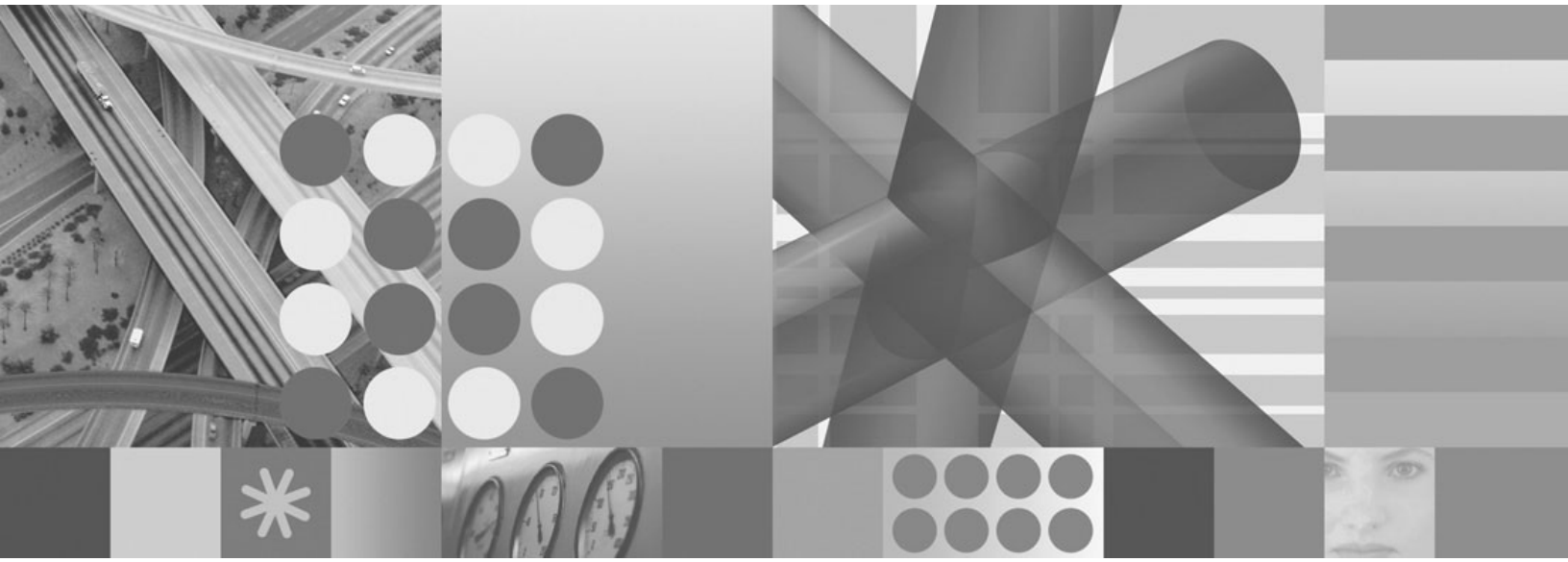
Tivoli® Autonomic Deployment Engine

Version 1.3

Autonomic Deployment Engine Glossary

Tivoli® Autonomic Deployment Engine

Version 1.3



Autonomic Deployment Engine Glossary

Note
Before using this information, read the information in "Notices" on page 9.

Fourth Edition (June 2006)

© Copyright International Business Machines Corporation 2003, 2006. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface v

Autonomic Deployment Engine Glossary 1

Notices 9

Trademarks 10

© Copyright IBM Corp. 2003, 2006

Preface

This glossary includes terms that are used by the IBM® Autonomic Deployment Engine technology. Terms and definitions that refer to Autonomic Deployment Engine by name cite it by its short name, *Deployment Engine*.

The following cross-references are used among terms:

Contrast with
Refers the reader to a term that has an opposed or substantively different meaning.

See Refers the reader to (a) a term that is the expanded form of an abbreviation or acronym or (b) a synonym or more preferred term.

See also
Refers the reader to a related term.

Obsolete term
Indicates that the term should not be used and refers the reader to the preferred term.

© Copyright IBM Corp. 2003, 2006

Autonomic Deployment Engine Glossary

A

action. In an action descriptor, the XML representation of a task that needs to be processed by the touchpoint of a hosting environment. Actions include tasks like creating or removing directories, installing or removing application files, and updating registries, configuration properties, environment variables, and paths. See also action descriptor. Contrast with change management operation.

action descriptor. In a software package, an XML document that defines a series of actions to be processed by the touchpoint of a particular hosting environment. See also action. Contrast with deployment descriptor and media descriptor.

aggregate (n.). In a unit hierarchy or deployment descriptor, an installable unit or configuration unit that subsumes (or includes) other, subordinate units.

aggregate (vb.). To accumulate into a unit hierarchy.

aggregated (adj.). Subsumed by (or included as part of) another installable or configuration entity in a unit hierarchy or deployment descriptor.

artifact. Obsolete term. See action descriptor.

application. In Deployment Engine, any script, component, feature, product, product suite, solution, maintenance, or other software deliverable that can be (or already is) deployed in an operating environment. See also software, component, feature, product, solution, and maintenance.

application package. A collection of all the software that the exploiter provides for an application. It includes all the software packages for the application. It can include Deployment Engine itself, if Deployment Engine is not already part of the operating environment. It can also include other things, such as a Java™ Runtime Environment or a software deployment program that a customer of the exploiter runs in order to deploy the software packages. See also payload files. Contrast with software package.

B

backing resource. A physical, managed resource in a hosting environment that is represented in the Deployment Engine installation database by the database entry for an installable unit. There is a one-to-one correspondence between the backing resource and the database entry for the installable unit. Deployment Engine monitors the backing resource to ensure that nothing other than Deployment Engine itself modifies it.

base software package. A software package that contains the original, or base, version of an application. A base software package is used to deploy an application for the first time. Contrast with full update software package, incremental update software package, and fix software package.

bootstrap. A small program that loads larger programs during system initialization.

bound file. A file that is defined in the installable unit deployment descriptor and is fully defined in the media descriptor. Contrast with unbound file.

C

change management. The process of planning and controlling software changes in a hosting environment. See also change request.

change management operation . An object that the change manager component of Deployment Engine constructs from an incoming change request. The change request identifies a deployment descriptor for the application that needs to be changed. The change manager constructs one change management operation for each installable unit and configuration unit that it finds in the deployment descriptor. Create, delete, update, and undo are examples of a change management operation. See also change request and change manager. Contrast with action.

change manager. The Deployment Engine component that reads the information in a deployment descriptor and then coordinates the change request to be implemented in the hosting environment. See also change request.

change request. An object that is passed by a software deployment program to Deployment Engine in order to request some kind of software change in one or more hosting environments. The change request includes the deployment or configuration operation to be performed as well as the location of the software package. The change request supplies values for all the variables defined in the software package (such as the value for the installation location). See also software deployment program and change management operation.

check (n.). An element of a deployment descriptor that defines some property criteria that the installation database or hosting environment must meet. For

© Copyright IBM Corp. 2003, 2006

example, you could provide a check to determine if the target hosting environment is suitable for deploying the installable units defined in the installable unit deployment descriptor. A check evaluates to `true` or `false`.

check (vb.). To evaluate whether the installation database or hosting environment meets some required property criteria defined in a deployment descriptor. For example, the dependency checker component of Deployment Engine evaluates a software check to determine whether a target computer has the necessary software installed.

component. A reusable object or program that performs a specific function and is designed to work with other components and applications. See also application. Contrast with feature.

conditioned installable unit (CIU). Obsolete term. See subordinate unit.

configuration unit (CU). A general term for the entities in a deployment descriptor that describe or define the hosting environments of the software package to be configured and the steps required to configure that software package. One type of configuration unit, called the root configuration unit, acts as a container for defining the other type, the smallest configuration unit. See also root configuration unit and smallest configuration unit. Contrast with installable unit.

configuration unit deployment descriptor. An XML document that defines a configuration task, which might include multiple steps. The document content includes smallest configuration units, the basic entities of configuration. There can be more than one configuration unit deployment descriptor in a software package. For example, there can be one configuration unit deployment descriptor for adding another database user and password, another for setting the port, and yet another for changing the trace level. See also configuration unit. Contrast with installable unit deployment descriptor.

configure. To set up or customize software for a particular use or environment. This activity refers to the one-time initial setup of an application or to the subsequent reconfiguration or setup of that application, which can be repeated. Contrast with deploy and package.

contained container installable unit. An installable unit that references the root installable unit of another installable unit deployment descriptor. The purpose of the contained container installable unit is efficiency. The contained container installable unit reuses another installable unit deployment descriptor, causing the installable units in the referenced root installable unit to be deployed as if they were part of the current root installable unit. The installable units in the referenced root installable unit are deployed to a single targeted hosting environment. Contrast with contained installable unit.

contained installable unit. An installable unit that references the root installable unit of another installable unit deployment descriptor. The purpose of the contained installable unit is efficiency. The contained installable unit reuses another installable unit deployment descriptor, causing the installable units in the referenced root installable unit to be deployed as if they were part of the current root installable unit. The installable units in the referenced root installable unit are deployed to one or more targeted hosting environments. Contrast with contained container installable unit.

container installable unit. An installable unit that encapsulates some combination of smallest installable units, smallest configuration units, contained container installable units, or other container installable units whose payload files are to be deployed in a single hosting environment. The purpose of a container installable unit is to be a wrapper for multiple installable units that need to be deployed together on the same target managed resource. Contrast with solution module.

corequisite. A component or service that is needed in parallel with a component. In other words, the components, resources, or services listed as corequisites of a component must be installed and configured in conjunction with the component. Contrast with prerequisite and exrequisite.

CU. See configuration unit.

D

dependency. A requirement that one installable unit has on another installable unit or managed resource.

dependency checker. The Deployment Engine component that performs dependency checking and integrity checking. See also dependency.

dependency checking. To use the dependency information defined in an installable unit deployment descriptor, combined with information in the Deployment Engine installation database and with properties of the target hosting environment, to determine whether the requirements that one installable unit has on another or on the target hosting environment are met. See also dependency and dependency checker. Contrast with integrity checking.

deploy. To place files or install software into an operational environment. Contrast with configure and package.

deployment descriptor. A general term for an XML document that contains instructions and data that

Deployment Engine requires to deploy or configure a software package. See also installable unit deployment descriptor and configuration unit deployment descriptor.

Deployment Engine bootstrap program. A program that deploys the Deployment Engine run-time environment on a computer. See also application package.

Deployment Engine run-time environment. The Deployment Engine code that is installed in the operating environment by invoking the Deployment Engine bootstrap program. The Deployment Engine run-time environment is installed before anything else.

Deploys. In Deployment Engine, a relationship, registered in the installation database, that indicates a direct relationship that an installable unit has with managed resource (in particular, a backing resource) in its hosting environment. For example, when an installable unit with a backing resource specification is deployed to specific hosting environment, that installable unit has a Deploys relationship with its backing resource.

descriptor. A general term for the three types of XML documents used by Deployment Engine for software change management. See also action descriptor, deployment descriptor, and media descriptor.

discriminant. A unique identifier that is used to distinguish multiple installations of the same application from one another in the Deployment Engine installation database.

E

Eclipse Platform. An open-source, standard platform for building integrated development environments (IDEs) that can be used to create applications, such as Web sites, embedded Java programs, or Enterprise JavaBeans™. The platform discovers, integrates, and runs the integrated modules called plug-ins that exist within its environment.

effector. An interface that enables an external agent to perform operations or change the state of a managed resource. See also touchpoint. Contrast with sensor.

elementary check. A request by the software deployment program to evaluate one or more specific checks. Generally, an elementary check is performed prior to dependency checking, in order to determine some specific knowledge about the environment. The software deployment program uses this information to decide which particular windows on a graphical user interface it should display to a user.

exrequisite. A component or service that must not be present. In other words, the components, resources, or services listed as exrequisites of a component must not be installed in conjunction with the component. Contrast with prerequisite and corequisite.

F

feature. A set of installable or configuration units that represents some specific functionality of a larger application, and whose deployment is optional. Samples, language packs, or even products in a suite are considered features. See also installation group. Contrast with maintenance.

Federates. In Deployment Engine, a relationship, registered in the installation database, that indicates that an installable unit can be shared. “Federates” indicates a parent-child relationship among software entities where one parent (a feature) allows another parent (another feature) to share its child (an installable unit). Specifically, when a feature is deployed, the feature has a Federates relationship with its child installable units only (not all descendents). Because of this Federates relationship, multiple features can contain the same installable unit and each be a parent of that installable unit. Contrast with HasComponents.

Fixes. In Deployment Engine, a relationship, registered in the installation database, that indicates that an installable unit in a fix software package has changed or corrected another installable unit (that is not itself a fix), while leaving its version, release, modification, and level unaltered. When an installable unit in a fix software package is deployed, the installable unit has a Fixes relationship with the installable unit it changes or corrects.

fix installable unit. Obsolete term. See fix software package.

fix software package. A software package that contains critical software changes or corrections (such as an interim fix or test fix) that have yet to be included in either a full update or incremental update. A fix software package is used to modify a previously deployed application. Contrast with base software package, full update software package, and incremental update software package.

fresh installation. A software deployment that installs an application for the first time.

full installable unit. Obsolete term. See full update software package.

full update software package. A software package that contains a major upgrade to an application, such as a manufacturing refresh. A full update software package includes every installable unit and therefore can be used to install an application for the first time as well as upgrade a previously deployed application. Contrast with base software package, incremental update software package, and fix software package.

H

HasComponents. In Deployment Engine, a relationship, registered in the installation database, that indicates that a feature or installable unit cannot be shared. “HasComponents” indicates a parent-child relationship among software entities where the parent is the sole parent of its child and allows no other parent to be a parent of that child. When deployed, the parent entity has a HasComponents relationship with its children only (not all descendents). For example, an installable unit in an installable unit hierarchy has a HasComponents relationship with each of its child installable units. Contrast with Federates.

hosted resource. A managed resource that is contained or supported by a hosting environment. For example, an application can be a hosted resource in an operating system hosting environment. See also hosting environment.

hosting environment . A managed resource that is a container for, or presides over, any number of hosted resources. A hosting environment cannot itself be a hosted resource. See also hosted resource.

Hosts. In Deployment Engine, a relationship, registered in the installation database, that indicates that a particular hosting environment is a container for, or presides over, a hosted resource or an installable unit deployed in that hosting environment. For example, when an installable unit is deployed to specific hosting environment, that hosting environment has a Hosts relationship with the installable unit.

I

incremental update software package. A software package that contains an application upgrade, such as a refresh pack or fix pack. An incremental update software package is used to upgrade a previously deployed application. Contrast with base software package, full update software package, and fix software package.

installable unit (IU). A general term for several entities in an installable unit deployment descriptor that describe or represent the installable parts of an application. Deployment Engine deploys installable units to a hosting environment in order to create new capabilities in that environment. One type of installable unit, called the root installable unit (root installable unit), is not deployed but acts as a container for defining all the other types. To define any type of installable unit, you specify a set of XML elements and attributes in an installable unit deployment descriptor document. Contrast with configuration unit.

installable unit deployment descriptor. An XML document that defines the content of a software package. Among other things, the document content includes installable units and, optionally, configuration units. A software package contains exactly one installable unit deployment descriptor, but it can refer to additional installable unit deployment descriptors in other software packages. See also installable unit. Contrast with configuration unit deployment descriptor.

installation. The process of adding a component to a computing environment, ensuring that it can run and that it interacts properly with all other affected components in the system.

installation database. The Deployment Engine component that retains the information about, and the installable unit deployment descriptor for, each installable unit instance deployed on the local computer. The installation database also saves information about the relationships between installable units.

installation group. A set of application features that are preselected for the user. An installation group can provide a set of features based on user roles, usage criteria, or any other criteria deemed relevant or beneficial to application users. An installation group is optional. See also feature.

instance. A particular occurrence or example of something.

integrity checking. To use relationship and other information in the installation database to ensure that deploying an installable unit will not violate the dependencies defined by its associated installable units. See also dependency checker. Contrast with dependency checking.

IU. See installable unit.

L

launcher. The executable file that starts an installation or uninstallation program on a particular platform.

M

maintenance. A general term for any separately deployable software package that represents one or more application updates or migration tasks. Fix, incremental update, and full update software packages are forms of maintenance. See also application. Contrast with feature.

managed resource. An entity that exists in the run-time environment of an IT system and that can be managed.

media descriptor. In a software package, an optional XML document that identifies the location of one or more of the package’s action descriptors or deployable

payload files. When the location of a file is supplied in a media descriptor, it overrides the original location of that file as defined in the deployment descriptor. Contrast with deployment descriptor and action descriptor.

media reader. A Deployment Engine component that encapsulates all the file location information specified in the deployment and media descriptors of a software package in order to retrieve files during certain operations. In response to a query from another Deployment Engine component, the media reader finds and opens software packages and returns the files inside them, including compressed files saved within archive files. In the latter case, the media reader can find and open Java archive (JAR) compressed files, ZIP compressed files, or other popular archive files and return their contents.

multiuser mode. An installation condition where Deployment Engine is installed on a computer and is available to all users of that computer for the purpose of deploying other software. Multiuser mode is established whenever the installer is determined by Deployment Engine to be a root user. A multiuser installation of Deployment Engine provides some additional scheduling and database functionality that is made possible by the special operating system authorities possessed by the installing root user. Contrast with single-user mode.

P

package (n). See software package.

package (vb). To reorganize your application files into a Deployment Engine–compatible software package. Contrast with configure and deploy.

package format. The description of the physical aspects of how to organize a software package. This description includes, but is not limited to, required content, directory structures, file locations and names, file formats, and package types (for example, JAR files). See also software package.

payload files. The files in a software package that are deployable; that is, the application files. These are the files that Deployment Engine actually *installs* in a hosting environment. JAR files, ZIP archives, and configuration files are among the valid types of payload files. The actual data content of these payload files is immaterial to Deployment Engine.

port type. An element in a WSDL document that comprises a set of abstract operations, each of which refers to input and output messages that are supported by the Web service.

prerequisite. A component or service that is needed before a component can be installed. In other words, the components, resources, or services listed as prerequisites of a component must be installed before the component can be installed. Contrast with corequisite and exrequisite.

product. A software application that is made up of various components. See also application and solution.

R

registration. The process of creating an object in the installation database that uniquely identifies the managed resource in the hosting environment.

relationship. An association between two components that allows management applications to perform or assist in operations, such as problem determination, based on an understanding of that association. Types of relationships include Federates, HasComponents, Hosts, Supersedes, and Uses. See also relationship registry.

relationship registry. A registry in the installation database component of Deployment Engine that stores information about the relationships between installable units.

requisite. In Deployment Engine, synonymous with corequisite. Contrast with prerequisite and exrequisite.

resource manager. The Deployment Engine component that, in response to a query from the software deployment program or Deployment Engine, checks for the presence of installable units in a hosting environment. The resource manager is used to help verify the synchronization and health of an installable unit within the installation database and the hosting environment.

root configuration unit. The top-level configuration unit in a configuration unit deployment descriptor. The root configuration unit defines the hosting environments of the software package to be configured and the overall configuration task to be accomplished by the smallest configuration units. It also acts as a container for all the smallest configuration units in the configuration unit deployment descriptor. See also smallest configuration unit. Contrast with root installable unit.

root installable unit. The top-level installable unit in an installable unit deployment descriptor. The root installable unit defines the software package to be deployed. It also acts as a container for all the installable units and configuration units in the installable unit deployment descriptor. Contrast with root configuration unit.

S

sensor. An interface that exposes information about the state and state transitions of a managed resource. See also touchpoint. Contrast with effector.

single-user mode. An installation condition where a private copy of Deployment Engine is installed on a computer for the purpose of deploying other software. This copy of Deployment Engine in single-user mode is for the installer's personal use only. Single-user mode does not require the installer of Deployment Engine to have as many operating system authorities as multiuser mode. Contrast with multiuser mode.

smallest configuration unit. A basic unit for setting up or customizing a newly or previously installed software package for a particular use or environment. A smallest configuration unit references action descriptors that describe how to configure the newly or previously installed software package in a single hosting environment. See also action descriptor.

smallest installable unit. A basic unit for deploying all or part of an application in a single hosting environment. A smallest installable unit references action descriptors that describe how to deploy its related payload files. You specify a smallest installable unit either as part of a root installable unit, a container installable unit, or a solution module. See also action descriptor.

software. Any computer programming that provides instructions to the computer hardware to tell it what to do. See also application.

software configuration. The processing required to set up or customize software for a particular use or environment. Contrast with software deployment and software packaging.

software deployment. The processing required to place files or install software into an operational environment and make that software available for use. Contrast with software configuration and software packaging.

software deployment program. An interactive or silent installation program that uses Deployment Engine APIs to implement software change requests in one or more hosting environments. See also change request.

software life cycle. An end-to-end series of states that characterize the condition of a software entity in a hosting environment. States used by Deployment Engine include Created, Usable, and Updated.

software package. A collection of files that includes the payload files to be deployed plus all the instructions and data that Deployment Engine requires to deploy the payload files. Deployment Engine and a software deployment program use this software package to install and configure all or part of an application in its target hosting environments. See also payload files. Contrast with application package.

software packaging. The process of combining packaging objects, such as descriptors, with application objects, such as payload files, to form a software package. See also software package. Contrast with software configuration and software deployment.

solution. A combination of products that addresses a particular customer problem or project. See also product and application.

Deployment Engine bootstrap program. A program that deploys the Deployment Engine run-time environment on a computer. See also application package.

Deployment Engine run-time environment. The Deployment Engine code that is installed in the operating environment by invoking the Deployment Engine bootstrap program. The Deployment Engine run-time environment is installed before anything else.

solution module. An installable entity that encapsulates some combination of smallest installable units, smallest configuration units, contained installable units, contained container installable units, container installable units, or other solution modules whose payload files are to be deployed in the hosting environment dictated by each encapsulated unit. In other words, the solution module can target multiple hosting environments. Contrast with container installable unit.

subordinate unit. A general term for any installable unit or configuration unit that can be defined within another installable unit. In an installable unit deployment descriptor, all of its installable units and configuration units are subordinate units to the root installable unit.

Supersedes. In Deployment Engine, a relationship, registered in the installation database, that indicates that a new installable unit replaces all or part of another installable unit. A Supersedes relationship occurs when you replace all or part of an installable unit by means of an incremental update. In this case, an installable unit in the incremental update software package has a Supersedes relationship with the installable unit it replaces. A Supersedes relationship also occurs when you deploy a new fix software package that includes all of a previously deployed fix software package. In this case, any installable unit in the new fix software package that replaces an installable unit in the previously deployed fix software package has a Supersedes relationship with that replaced installable unit.

T

topology. A part of a deployment descriptor that defines the target managed resources where its installable units can be deployed, where its configuration units can be processed, or where a check is to be performed. A target managed resource can also

define the backing resource, called a *resulting hosted resource*, that will be deployed if the target is selected for an installable unit.

touchpoint. A Web service interface to an instance of a managed resource. For Deployment Engine this resource is always a specific hosting environment, such as an operating system or J2EE server. Each touchpoint consists of a sensor and effector for exchanging information with the managed resources (called *hosted resources*) in the hosting environment. The touchpoint enables management operations—in this case, software deployment or configuration—to be performed in the hosting environment, including any of its hosted resources. See also sensor and effector.

U

unbound file. A file that is defined in the installable unit deployment descriptor but is not defined in the media descriptor. Contrast with bound file.

universally unique identifier (UUID). A 128-bit numeric value that is used to identify a component with the intent that no two components should have the same universally unique identifier.

update installable unit. Obsolete term. See incremental update software package.

Uses. In Deployment Engine, a relationship, registered in the installation database, that indicates that one installable unit depends on another installable unit. For example, a deployed component has a Uses relationship with each of its prerequisite and corequisite components. A Uses relationship can only be established with installable units that are already deployed.

UUID. See universally unique identifier.

W

Web service. A self-contained, modular application that can be described, published, located, and invoked over a network (generally the Internet). Web services go beyond software components, because they can describe their own functionality and look for, and dynamically interact with, other Web services. Web services use open protocols and standards, such as HTTP, SOAP, and XML. Web services provide a means for different organizations to connect their applications with one another to conduct dynamic e-business across a network, regardless of their application, design, or run-time environment.

Web Services Description Language (WSDL). An XML-based specification for describing networked services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

WSDL. See Web Services Description Language.

X

XML schema. A means for defining the structure, content, and semantics of XML documents as defined by a World Wide Web Consortium (W3C) recommendation.

XML Schema Definition Language (XML schema language). A language for describing XML files that contain XML schema.

XML schema language. See XML Schema Definition Language.

Notices

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

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku Tokyo 106, Japan

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

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

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

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

© Copyright IBM Corp. 2003, 2006

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

Trademarks

IBM and the IBM logo are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Printed in USA