

PLANETJ CORPORATION

WOW Version and Configuration Guide

Table of Contents

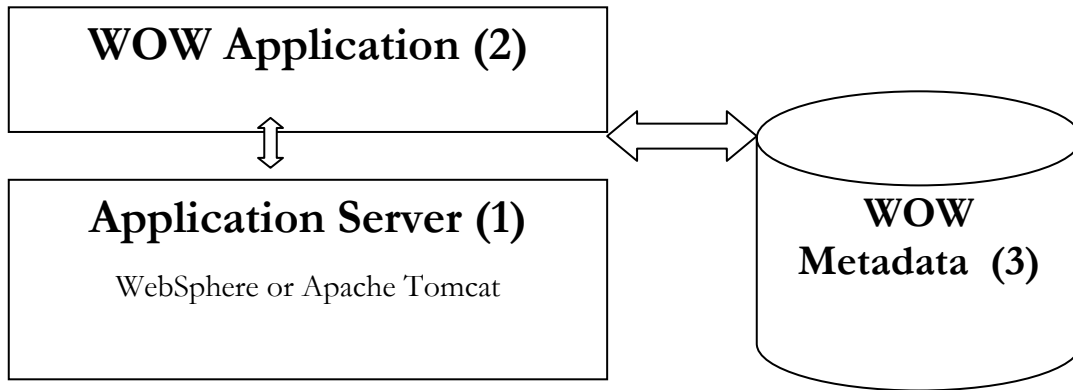
WOW Overview:	2
WOW Components:	3
1) Application Server	4
2) WOW Application	4
3) WOW Metadata	4
WOW Development Configuration Scenarios	4
1. Single Code, Single Metadata library	4
2. Single Code, Multiple Metadata library	5
3. Multiple Code, Single Metadata library	6
4. Multiple Code, Multiple Metadata library	6
5. Multiple Code, Multiple Metadata library and Multiple Metadata Servers	6
ISV Application Installation Scenarios	7
Simple Installs	7
Advanced Installation Program Options	7
Planning Your Deployment Configuration	7
Installing Fixes or Enhancements to Existing Installations	7
Handling Upgrades to Existing Systems	7

WOW Overview:

WOW (Web Object Wizard) is a 100% Java™ based web application development tool and runtime engine that creates "data centric" Internet/Intranet/Extranet applications. If you need to put data on the web, WOW is your answer. Using a Web based development interface, applications are created by specifying JDBC/SQL operations and configuring application metadata. Customers can easily access and manipulate data stored in any relational database including DB2, Oracle®, and SQL Server. Applications execute within WebSphere™, WebLogic™, or Tomcat™ application servers. Interactive product demonstrations can be viewed at: <http://www.planetjavainc.com/support.htm>

For additional information about WOW, or to report problems with WOW, please contact support@planetjavainc.com or call (760) 432-0600 for customer support.

WOW Components:



1) Application Server

WOW produces browser/web-based applications. An underlying application server is needed to provide the infrastructure and web Servlet services that WOW utilizes at runtime. WOW supports both IBM's WebSphere application server as well as Apache's Tomcat server. IBM WebSphere is a chargeable product and supported by IBM while Apache Tomcat is a free open source product supported by Apache open source community. The application server component can run any virtually any Java enabled platform including Windows, OS/400, LINUX, UNIX, and OS/390.

2) WOW Application

WOW is a development and runtime product that runs on top of an application server. WOW is written in 100% pure Java and can run on any Java enabled platform including Windows, OS/400, LINUX, UNIX, and OS/390. WOW was developed and is supported by Planetj Corporation.

3) WOW Metadata

WOW stores all application-based data in a relational database. The WOW Metadata can be stored in either iSeries DB2 or MYSQL. Additional databases may be supported depending on industry demands.

Users have their choice of where each component will run based on their environment.

WOW Development Configuration Scenarios

Development of WOW applications can involve various scenarios depending on complexity, the number of developers, and required business procedures. This section will describe the options and provide recommendations regarding the appropriate scenario you should consider. There are two important items in terms of development: Metadata and Java code. Metadata is a set of database files that store WOW applications such as operations, and field descriptors. Java code is the WOW runtime code and any customized Java classes that you may have created.

1. Single Code, Single Metadata library

This is the simplest and quickest to develop within. In this configuration, WOW is running on a single server using the standard WOW metadata library. (NOTE: The metadata may actually be stored on a different machine). Applications are developed using the WOW Builder and available immediately to users. Enhancements and fixes are reflected immediately. Development and production are on the same machine. A typical scenario would be a small development team develops application A and makes it available to the end users when it is completed. Development may proceed to application B. Application B may be secured to prevent premature usage. Enhancements and fixes for application A can be made while the application is live

or during normal maintenance times. To minimize negative impacts during fixes, production operations can be copied to a test application and modified without affecting the original. Additional steps can be taken to ensure this environment has integrity such as removing update and delete authority to the metadata to prevent unintended changes (essentially locking the files). Metadata can be journaled, copied, and saved using traditional data management.

2. Single Code, Multiple Metadata library

This configuration involves the use of multiple schemas (libraries) to hold metadata. If you have the need to deploy a single application or a set of applications to a user that is running WOW, this is a good option. In this option, you could develop application C in a schema called “APP_C”. All WOW metadata is stored in the WOW tables within schema “APP_C”. The WOW Builder can be directed to use this specific non-standard library using the URL: http://localhost/wowXX/wowBuilder? pj lib=APP_C

Once complete, application C can be distributed to a different WOW environment by restoring the schema/library “APP_C” to the database (provided there are no naming conflicts). This is called “user library” support. Reference the WOW Builders Guide for more information on user library support.

For multiple applications that have different deployment schedules but want to share resources, a single metadata schema can be deployed with an alternate name. For example: consider the following scenario where 2 applications are developed but have different needs and schedules relative to deployment from development to production.

Time Line	Application	Development Lib	Production Lib	Version	Comments
Day 1	App A	PJUSER6x*	APP_A	1.0	App_A completed and deployed to a different schema for production.
Day 1	App B	PJUSER6x	None	.1	App B started but is not ready for production.
Day 2	App A	PJUSER6x	N/A	1.x	Some new operations under development but not complete.
Day 2	App B	PJUSER6x	APP_B	1.0	App B completed and deployed without affecting App A.
Day 3	App A	PJUSER6x	APP_A	2.0	App A version 2 completed and moved to production without affecting App B.

*Depending upon your user level, your development library may be PJUSER 63, 64, etc.

In this scenario, applications A and B are able to share resources but can be deployed independently.

3. Multiple Code, Single Metadata library

This configuration should be used if WOW development requires significant development of custom Java classes or JSPs. You would setup a development machine running Eclipse or WebSphere Studio. You would then import the WOW runtime jar files into this development environment. Consult the “Using_WOW_With_Studio.doc” document for specific instructions. The development environment would reference the metadata effectively operating just like the production WOW environment. The advantage is that the development environment allows easy development of custom Java classes, incremental compiles and debug, as well as full testing without effecting the production environment. Java skills are required in this configuration. Once customized Java classes and JSPs have been developed and tested, these JSPs and custom classes can be moved to the production server and made available for end users.

4. Multiple Code, Multiple Metadata library

This is the same as #3 except development can occur in user libraries to aid in transferring to other systems.

5. Multiple Code, Multiple Metadata library and Multiple Metadata Servers

This configuration involves having development metadata and production metadata often on different metadata systems. The typical development scenario would be:

- a. Development on Application A is done using a development version of WOW and development metadata.
- b. Metadata libraries and any custom code would be migrated or copied into production after Application A is completed and tested.
- c. Enhancements to A or new development on application B would be done on the development servers and copied into production after completion. Using user library support would allow for applications to be moved independently of each other.

This development configuration could be used if mandated by corporate procedures or if the application involves updates to production data that must be tested on development machines prior to production usage.

ISV Application Installation Scenarios

Simple Installs

WOW applications can be deployed to new environments by simply installing the appropriate WOW version followed by installation of any custom metadata libraries and custom Java code. Optionally, more advanced options are also available as discussed below.

Advanced Installation Program Options

WOW provides a Windows based installation program, which is responsible for installing WOW along with Tomcat. This installation program can be adapted to allow ISVs to package their libraries and custom code for installation at customer sites. This is done by configuring property files to include your new resources.

Planning Your Deployment Configuration

ISV applications often involve the development of custom features specific to a particular customer and as well as allowing customers to develop additional features or customize supplied features. Here is a typical scenario:

- a. ISV develops a base package with metadata in the standard user library (PJUSER6x). NOTE: This could also be a user library such as ISVL6x. This contains the standard package of WOW application(s) and custom JSPs and Java code.
- b. Customer XYZ requests custom features, which the ISV builds into metadata library ISV6x. This library and any custom Java are supplied to customer XYZ.
- c. Customer XZY builds their own applications and operation in library XYZUSER6x.

In this scenario, customer XYZ has 3 metadata libraries. Since the libraries are all separate, no conflicts occur if a particular library is restored or enhanced except as noted below:

- a. The customer would have to record and reapply any changes made to metadata in libraries PJUSER6x or ISVL6x. They would only have to do this if changes are made and reapplied to the base package or ISV customization libraries.
- b. ISV enhancements or fixes can be simply restored to the customer system if all customer enhancements are in their own library.

Installing Fixes or Enhancements to Existing Installations

Fixes or enhancements can be applied to existing installations in the current release by simply providing updated libraries and code that can contain changes. For example, if ISV fixes or enhances a feature in the base package, a custom could apply the fix by installing a newer version of library PJUSER6x or ISVXY6x

Handling Upgrades to Existing Systems

WOW typically provides new releases (upgrades) by supplies a new library such as PJUSER70 (release 7) along with a utility, which copies metadata from a previous release to the current. ISVs and customers can use the same utility to upgrade customized user libraries. Multiple releases execution is accommodated by

installing an upgraded release in a new application server context. For example: customer XYZ may be concurrently running version 6 and version 7 of their applications.

www.xyz/wow60/run?id=2 “Runs the version 6 metadata (PJUSER60, ISV60, XYZUSER60) and 6 code”

www.xyz/wow70/run?id=2 “Runs the version 7 metadata (PJUSER70, ISVZY70, XYZUSER70) and 7 code”

This allows easy integration and migration to new releases. Prior releases can be left in place until the new release is installed and tested. Users migrate to the new release by simply changing their URL link used to reference their applications.

© PlanetJ Corporation
info@planetjavainc.com
Phone: 760-432-0600