OSM CVS
Core Project Structure

address	:pserver:<username>@home.osm.net:F:\system\cvs
project	core

Project Structure

lib	Contains dependent jar files required in the development and runtime phases. These include the Xerces XML parser, OpenORB platform and services, the Apache open source Avalon and Phoenix server management frameworks, component models and utilities, the InstandDB database, and jar files from Sun supporting certificate creation, certification and certificate vault management.	
src	conf	Contains server configuration files "assembly.xml" and server.xml". The server configuration file declares generic low level properties for thread and log management and handles security policy declarations for application classes. The assembly file contains the configuration of the OSM Gateway components.
	doc	General documentation concerning the platform including product schedule, feature-benefit documents, and technical notes.
	dpml	Contains the base DPML DTD and a number of example DPML documents (negotiation, contract engagement, and a simple "hello world" process description).
	etc	Additional files used in gateway deployment.
	idl	IDL source files for the ORB, common services, the Task and Session IDL, and the EC Domain IDL for the community and collaboration frameworks.
	java	The complete Java source directory.
	manifest	Contains manifest files for all bar files and demonstration appliances.
	psdl	Contains the PSDL definitions for the hub and audit server and supporting utilities for list and event storage.
build.xml	The build.xml file is ANT 1.3 project definition that contains targets for building and deploying the Gateway. Principal targets are listed below.	
	gateway	Builds and runs the gateway server (including the audit, time, registration and discover, and collaboration hub).
	test	Builds and runs a test application using the Hello World demonstration appliance.
	doc	Generates the full javadoc for all of the packages.

External Libraries

Avalon	**phoenix-loader.jar**	Bootstrap classes used to load the Pheonix engine.
	phoenix-engine.jar	Handles the loading and activation of the gateway.sar file, executing the top-level component lifecycle operations, logger association and so forth.
	phoenix-client.jar	Interface definitions within the phoenix package.
	avalonapi.java	Interfaces and abstract class implementations of a component model used by gateway components.
OpenORB	**openorb_ins-X.jar**	Interoperable Naming Service (no currently used).
	openorb_ots-X.jar	Transaction Service, used by the PSS implementation.
	openorb_pss-X.jar	Persistent State Service (including PSDL compiler) used as the underlying framework for object persistence within all gateway components.
	openorb_time-X.jar	Implementation of the CosTime service used by the net.osm.time gateway block.
	openorb_tools-X.jar	IDL compiler used during the build process and required by the PSS runtime.
	openorb -X.jar	The ORB engine used by all gateway blocks and the test applications.
JSSE	**jsse.jar** **jcert.jar** **jnet.jat**	The Java^{TM} Secure Socket Extension (JSSE) is a Java package that enables secure Internet communications. It implements a Java version of SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication. http://java.sun.com/products/jsse/
InstantDB	**idb.jar**	Fully java relational database used behind the PSS implementation when built and configured for database persistence.
	idbexampl.jar	Contains classes needed to post executiobn review of the contents of a database (db browser).
Other	**xerces-1.3.0**	XML parser from Apache.
	jta1.0.1.jar	Sun implementation of the XA transaction model, required by the OpenOPRB OTS implementation and the InstantDB implemetation.

Java packages

net.osm.appliance	Definition of an abstract business process controller than can be used to wrap existing legacy applications in terms of a business process.
net.osm.audit	The audit package defines a plug-in block in the server architecture that handles the persistent registration of business events under a business object adapter. The **home** sub-package is generated from the ${idl}/audit.idl IDL source and defines the interfaces for Adapter and the audit Manager types. The persistence sub-package contains the implementation of catalog storage object. The pss sub-package contains the generated PSS classes based on the ${psdl}/audit.psdl PSDL source.
net.osm.base	The base package contains low-level base classes and set of classes providing cascading exceptions.
net.osm.discovery	Contains the generated stubs, skeletons, base POA classes and abstract valuetypes of the Registration and Discovery services supporting the registration of artefact descriptions, publication of descriptions under directory services, and description repackaging within the context of referral service provision. The package is generated from the ${idl}/discovery.idl IDL source and implemented by classes in the **net.osm.portal** package.
net.osm.dpml	Provides support for extensions dealing with the Digital Product Modelling Language (DPML) document content, conversion to valuetypes and DPML presentation tools.
net.osm.event	PSS generated sources from the ${psdl}/event.psdl PSDL source used to define persistent storage of structured events.
net.osm.hub	Contains the implementation of a business object framework based on the OMG Task and Session model and process management services based on the OMG EC Domain Community and Collaboration Frameworks. The package provides support for the creation of business resources dealing with people, places and things, work in-progress, and process management services including back-end appliance hosting. The package provides support for User, Workspace, Desktop, Message, Task, Processor, and GenericResource. Additional types defined under the Community and Collaboration framework that be progressively added to this package include CollaborationProcessor, EngagementProcessor, VoteProcessor, Community and Agency.
net.osm.list	PSS generated sources from the ${psdl}/list.psdl PSDL source used to define persistent storage of linked lists used within the audit and collaboration hub blocks.
net.osm.portal	Contains an implementation of the Registration and Discovery service (refer net.osm.discovery) supporting the registration or artefact descriptions, publication of descriptions under a directory services, and description repackaging within the context of referral service provision.
net.osm.sample	Contains a set of example Appliance implementations that can be used as templates for the creation of new back-end integration wrappers and/or creation of new business process applications.
net.osm.test	Contains a test framework used for the purpose of validating conformance and operational integrity of the implementation.

net.osm.time	Contains an implementation of the org.omg.CosTime.TimeService block, structured as a server within the gateway framework.
net.osm.util	Contains general utilities dealing with data conversion and convenience utilities such as UtcT manipulation, IOR internalisation and externalisation.

OSM Libraries

osm-appliance.jar	Contains the implementation of AbstractAppliance, related interfaces and implementation classes. In conjunction with osm-common-client, the two jars serve as the base for back-end development.
osm-audit.client.jar	Client stubs for the audit framework.
osm-base.jar	General base utilities together with the list and event persistent storage implementations. Used as a shared library by the server implementation (audit and hub).
osm-common.jar	Full Task and Session, Community and Collaboration Frameworks stubs, skeletons, POAs and valuetypes – used as a library by server components.
osm-common-client.jar	Client stubs for the Task and Session, Community and Collaboration Frameworks together with respective valuetype implementations.
osm-dpml.jar	PDML parser.
osm-hub-client.jar	Stubs for client access to factory and factory finder services.
osm-portal-client.jar	Stuns for client access to the Registry and Directory types, together with the valuetypes supporting construction of resource descriptions, receipts and referrals.
test.jar	Contains the test client.

Gateway Components

gateway.sar	The gateway.sar file is the deployment unit for the Gateway server. It contains the default configuration files, external libraries and a set of server component building blocks list below.	
	osm-audit.bar	Structured event consumer and persistent store.
	osm-hub.bar	Business object and process management server.
	osm-portal.bar	Registration and discovery service.
	osm-time.bar	Time service.