

Microsoft Project Server Components and Authentication

Summary

This article describes the server- and client-side components for Microsoft Project Server 2002 and the application-level authentication layer for Microsoft Project Server 2002. The article also includes diagrams showing the interaction between Microsoft Project Server 2002, Microsoft Project 2002 (both Professional and Standard editions), and Microsoft Internet Explorer.

Microsoft Project and Project Server Components

The following tables list the business object components of Microsoft Project Server 2002 for both the server and the client including the components used to interact with Microsoft Project 2002.

Server Components

Component	Description
MkRetail.dll	Manages session state.
MSPNav.dll	Performs data access for the portfolio modeling engine.
MSPJWMS.dll MSPWSS.dll MSPWIF.dll MSWarp.dll	Performs portfolio modeling.
PDS.dll	Microsoft Project Data Services; handles XML-based requests to perform functions and return data.
PJOLEDB.dll	Microsoft Project OLE DB provider.
PJDBComm.dll	Receives Publish messages to synchronize Microsoft Project Web Access and Microsoft Project data.
PJMsgHlr.dll	Decodes and publishes project data.
PJNPE.dll	Sends reminders via SMTP.
PJQuery.dll	Executes queries and stored procedures.
PJSchSvc.exe	Invokes scheduled processes including reminders and cube generation.
PJState.exe PJLKR.dll	Persists user session state.
PJSvrSecurity.dll	Checks permissions to data and functionality based on user context.
PJTextConv.dll	Manages number and text formatting.
PJTrace.exe PJTrcPrv.dll	Microsoft Project Server trace service; manages generated trace and debug events.
ProjObjProv.dll	Links documents to projects and tasks.
ProjOlap.dll	Manages the process of building an OLAP cube.

Component	Description
PVWMgr.dll PCSVIEWS.dll	Publishes Microsoft Project Web Access view data from Microsoft Project data.
VWNotify.exe	Invokes view data creation on publish; invokes the OLAP cube generation process.

Client Components

Component	Description
PJ10Euc.dll	Language-dependent resources.
PJCalendar2.ocx	Calendar control.
PJGrid2.ocx	Grid/Gantt control.
PJMsgMgr.dll	<ul style="list-style-type: none"> • MESSAGING MANAGER: Determines server version (Project Central or Project Server 2002). • SENDER: Sends the publish information to the server and waits for a reply. • TRANSLATOR: Translates XML calls to Project Server 2002 into Project Central's proprietary format (or vice versa) as required.
PJOffline2.ocx	Manages Microsoft Project Web Access offline.
PJOutlook2.ocx	Microsoft Outlook integration object.
PJQuery2.ocx	Client-side data access (RDS) object.
PJRes10C.dll	Contains the client-control boilerplate text.
PJSecurity2.ocx	Login/authentication object.
PJSpool.exe PJSpooli.dll	Manages the queue of asynchronous calls to PJDBComm.dll for Microsoft Project 2002.
PJTextConv2.dll	Manages number and text formatting.
PJUpdate2.ocx	Updates projects in Microsoft Project.
RichEd20.dll	Rich text editor control.

Microsoft Project Server 2002 Authentication

Regardless of whether the client accesses Microsoft Project Server 2002 through Microsoft Project Server or Microsoft Project Web Access, authentication is based on Windows user accounts provided by Microsoft Internet Information Services (IIS), and also supports application-level authentication, which encrypts users' passwords using the Microsoft Crypto API and then stores them in the MSP_WEB_RESOURCES table in the Microsoft Project Server 2002 database. Authentication and logon sequence for Microsoft Project Server is as follows (see also Figure 1, which illustrates the page relationships):

1. Authentication begins on one of several pages and is dependent on the type of authentication and the originating application, where:

Authentication Type	Internet Explorer	Microsoft Project	Project Data Service
Integrated	LGNINT.ASP	LGNINTPJ.ASP	LGNINTAU.ASP
Application	LGNPS.ASP	LGNPSPJ.ASP	LFNPSAU.ASP
Basic	LGNBSC.ASP	n/a	n/a

2. All logon files include the file AUTHLIB.ASP, which authenticates the user as well as returns authentication and logon-related error codes. Each page navigates to the session manager page (SESSTART.ASP), passing parameters as required.
3. This session manager page includes PJSECURITY.ASP, which checks user permissions. IIS authenticates both integrated and basic authentication types by looking up the logged on user name in the MSP_WEB_RESOURCES table. If the user does not have permissions or if an error occurs an error message displays to the user, or in the case of the automation entry points, an error code returns in XML.
4. If authentication is successful, the Session Manager creates a session, which is referenced at every page request, for the user. The session must be created at the time of user authentication to eliminate the ability to deceive the system into creating a session without proper authentication. Thus, the session acts as the security context and the session cookie is used to obtain the session information during subsequent interaction with Microsoft Project Web Access. The session times out if not used within a given time limit or is destroyed when the user logs off of Microsoft Project Web Access.

When navigating to the session start page, the "ref" query string parameter is also appended to the URL. This parameter specifies the URL to access when an error occurs. Because logon error messages only display on certain pages (e.g., LGNPS.ASP) the URL with parameter can differ from the redirection page. For example, if an error occurs in LGNINT.ASP the user is routed back to LGNPS.ASP, the error code is supplied as a query string parameter, and the LGNPS.ASP page displays the error message.

5. Once the session information is properly created and persisted, the session start page consults the query string parameter to determine if the request is from the automation or Microsoft Project entry points. If the request is from the automation entry point, the page redirects to the automation page that creates the required XML. If the request is from a Microsoft Project entry point, the page redirects to MSPJLOGINDONE.ASP (i.e., Figure 1, 2a) to notify Microsoft Project that the logon sequence is successful. Microsoft Project then captures the session cookie for subsequent communication with Microsoft Project Web Access. Next, the starting application page is determined and stored in the session, and the session start page redirects to the prerequisite page (PREREQ.ASP).
6. Client-side validation is performed and client-side locale information is gathered at the prerequisite page (PREREQ.ASP). Here, a function is called to determine the

existence of client-side controls. It returns the control versions. If the function call fails or the version is earlier than the server version, the page redirects to the download End-User License Agreement (EULA) page (i.e., Figure 1, 3a: DLEULA.ASP) to accept the EULA and download the controls. If the EULA is accepted, the page redirects to the download page (i.e., Figure 1, 3b: DOWNLOAD.ASP) and the controls are downloaded. This page also calls the version check to ensure that the controls load properly. When the download is complete, the page redirects back to the prerequisite page. The prerequisite page verifies that the download was successful, then gathers the locale information and passes it as a query string parameter to the redirection page (REDIRECT.ASP).

7. The final step of the logon sequence is accessing the redirection page (REDIRECT.ASP). The redirection page uses the session cookie to obtain the user's session then stores the locale information in the session. Next, retrieval of the session's starting application page occurs, the query string parameters are added to the URL, and the page redirects to the application start page.

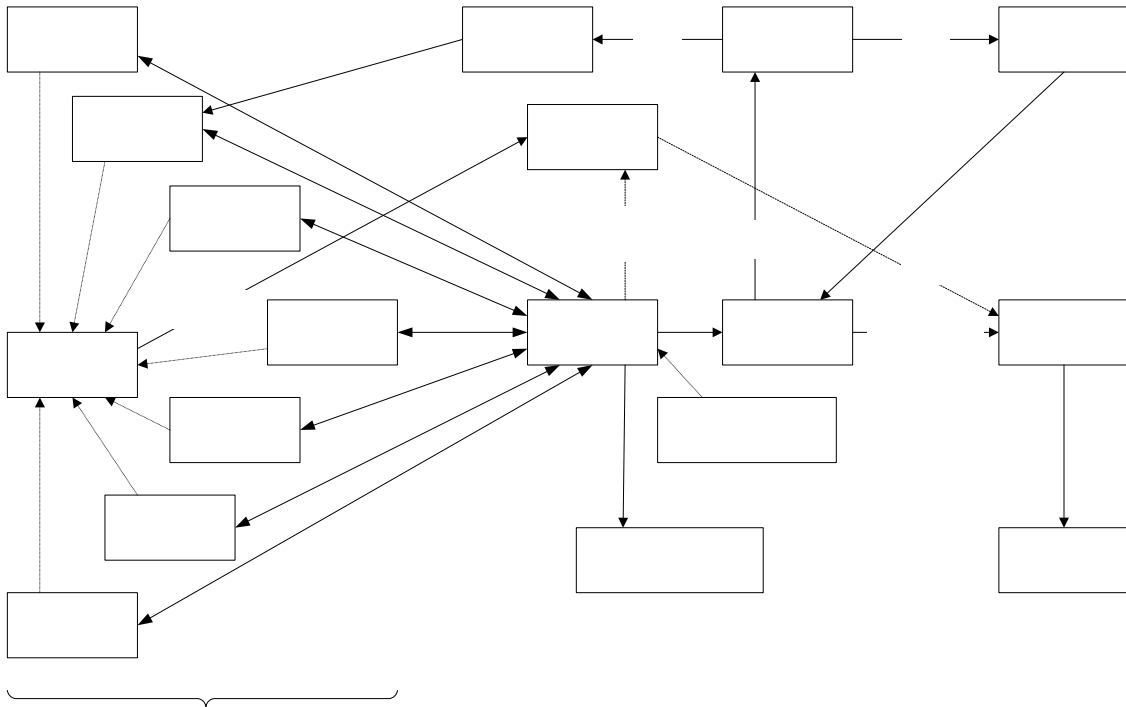


Figure 1.

Microsoft Project Server Subsystems

This section describes the various Microsoft Project Server subsystems that rely on other server products including SharePoint Team Services from Microsoft Server, Microsoft Analysis Services, and Microsoft's SMTP host.

Views Processing Subsystem

The process of publishing a large project to Microsoft Project Server can be processor- and database-intensive. In Project Central, project views were created using the Microsoft Project OLE DB provider when views were requested. To improve the performance and scalability of the view generation process, the Microsoft Project Server views processor was enhanced to:

- Occur once when the project was published.
- Write the view data to SQL Server tables to be read when project views are requested.
- Be deployed on a separate application server for performance and scalability.

Subsystem Components

Component	Description
VWNOTIFY.DLL	The Views Notification component service, which listens for operating system events to know when an .MPP and associated XML metadata file are added to the \ViewDrop folder on the server by Microsoft Project Server when the project is published (See Figure 2). The service opens the XML file, validates permissions and, if valid, passes it to the views processor. Security for this component is controlled through access permissions that are assigned to the directory from which events are generated.
PCSVIEWS.DLL	The View Processor, which removes duplicate published projects for the same project and user, reducing the impact of a denial of service attack.
PJVWMGR.DLL	The View Manager, which is responsible for reading data from either the published project file or an enterprise project database. Also transforms the data using the Microsoft Project OLE DB provider and writes the view data to the view tables in the Microsoft Project Server database.

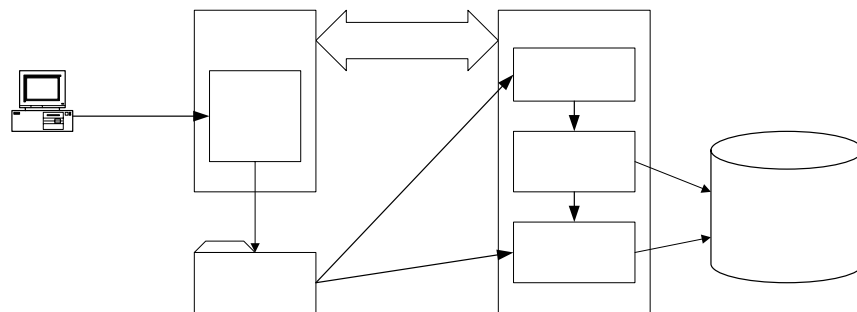


Figure 2.

Document Management

To make communicating and sharing project information easier, SharePoint Team Services from Microsoft Server can be integrated with Microsoft Project Server to provide both Web publishing and collaboration features. When integrating SharePoint Team Services from Microsoft Server with Microsoft Project Server, SharePoint Team Services from Microsoft Server acts as a component of the server rather than a standalone program. Microsoft Project Server setup automatically configures SharePoint Team Services from Microsoft Server; however, some manual configuration and administration may be required. When integrating SharePoint Team Services from Microsoft Server and Microsoft Project Server, Windows NT™ accounts—domain and local—must be used.

Subsystem Components

Component	Description
Microsoft Project Web Access	Microsoft Project Server displays SharePoint Team Services from Microsoft Server pages in the Microsoft Project Web Access document management pages within an IFrame. Documents are rendered either as HTML or within a hosted ActiveX server such as Microsoft Word or Microsoft Excel (see Figure 3). SharePoint Team Services from Microsoft Server is administered from Microsoft Project Web Access using an HTTP protocol; administrative tasks include creating and deleting subwebs and adding users to roles.
Object Link Provider (ProjObjProv.dll)	Residing in the middle layer of Microsoft Project Server, this object exposes a programmatic interface for linking documents and issues to projects and tasks. The link information is stored in the Microsoft Project Server database. The Object Link Provider can be used to integrate Microsoft Project Server with another document management system if the unique identifier of the documents can be expressed in URL, XML, or table IDs. In addition, an external object can be easily associated with projects and tasks, or with other external objects via the object link provider.
Link Information Tables	These tables reside in the Microsoft Project Server database and store information on links between documents and issues.
SharePoint Team Services from Microsoft Server	Microsoft Project Server setup modifies certain default templates and XML schemas of the SharePoint Team Services from Microsoft Server to include information relevant to project management. This server can reside on the same physical machine as Microsoft Project Server or on a different one.
Microsoft SharePoint Team Services from Microsoft Server Data Stores	Microsoft SharePoint Team Services from Microsoft Server uses a combination of a SQL database and the local file system to store SharePoint data. All interaction with these stores is performed by Microsoft SharePoint Team Services from Microsoft Server.

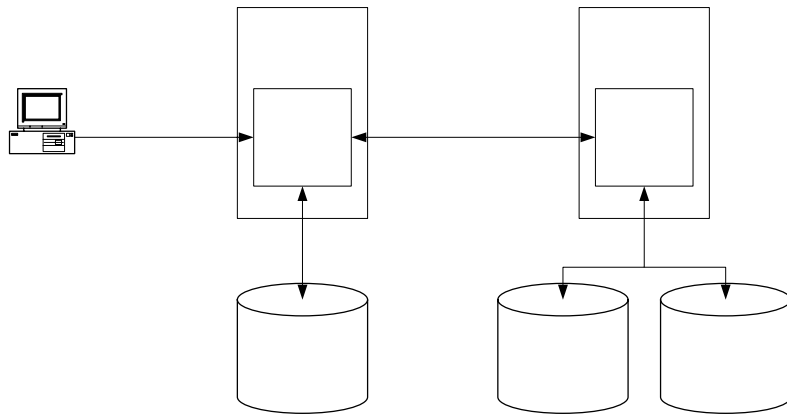


Figure 3.

OLAP Cube Generation Subsystem

Microsoft Project Server 2002 provides Online Analytical Processing (OLAP) tools for portfolio analysis. The cube-building process consists of two stages: (1) staging tables (i.e., dimension and fact tables) are populated from the view tables, and (2) the cube is built from the staging tables. Cube generation can be either scheduled or invoked on demand. Once built, Microsoft Project Web Access users can view the cube using Office Web Controls such as pivot tables and charts. For Microsoft Analysis Services to work with Microsoft Project Server, Windows NT™ accounts—domain and local—must be used.

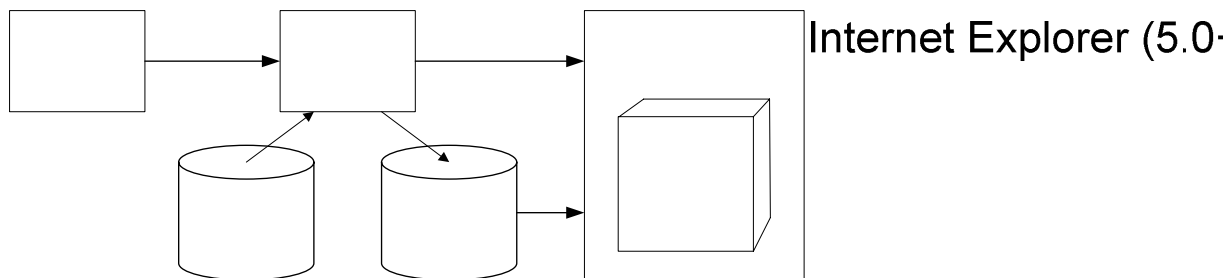


Figure 4.

Notification Subsystem

Microsoft Project Server allows project managers and team members to schedule e-mail reminders to their resources and themselves, respectively. The notification system can also send e-mail notifications based on user actions such as publishing a project.

System Components

Component	Description
PJSCHSVC.EXE	The Scheduler component service, which periodically checks for reminder e-mail messages and, if extant, calls the notification component.
PJNPE.DLL	The Notification component, which sends reminder e-mail messages when invoked by the Scheduler component or the Microsoft Project Server message handler.

PJMSGHLR.DLL	The message handler, which calls the Notification component to generate notification e-mail messages upon publication.
--------------	--

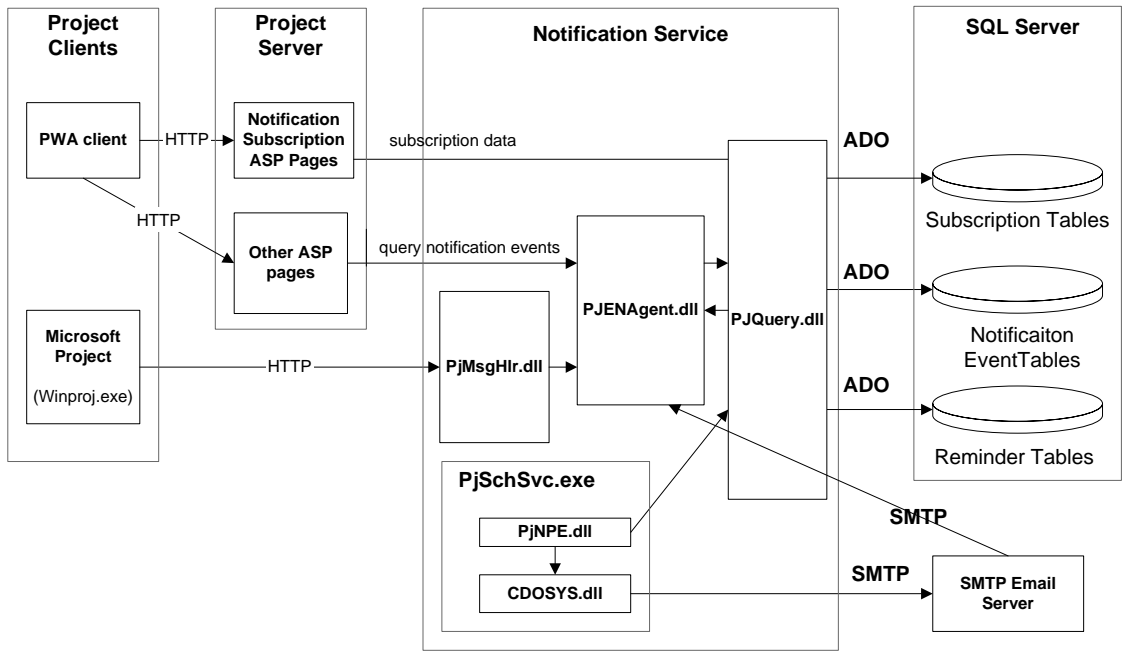


Figure 5.