ORACLE | **PRIMAVERA**

An Oracle White Paper
November 2009

Oracle Primavera P6 EPPM Integrations with Web Services and Events

ORACLE

INTRODUCTION

Primavera Web Services is an integration technology that extends P6 functionality and business objects. Based on open standards such as SOAP, XML and WSDL, Primavera Web Services enable developers to leverage standard interfaces to create integrated software solutions that interoperate with a wide variety of enterprise software applications running on a diversity of hardware and operating system platforms.

By adopting industry-standard Web services as the primary interface for programmers with the P6 6.2 platform, it becomes easier to fit Primavera into your existing IT environment. Combining the power of P6 project portfolio capabilities along with your existing enterprise investments in areas such as ERP, ticketing, MRO, reporting and portals and you can interconnect systems such as SAP, Oracle, Microsoft and legacy systems natively into Primavera. With Web services, you can build these integrations regardless of your organization's standard programming language or operating system platform. For example, you could snap together an integration of a Siebel CRM implementation running on a Sun Solaris server with Primavera P6 running on Windows Server 2003 by utilizing any .NET programming language (VB, C#), to connect to P6 Web Services. Or you could use Oracle BEA WebLogic to perform the integration. Or you could write the integration using Java via Eclipse or JDeveloper. Since Web services programmers are already versed in using their favorite programming languages with Web services, discovering available services published by P6 is easier than learning a completely new and different API environment.

ARCHITECTURE

P6 Web Services Employs Web-based Technology

The Primavera P6 Web Services platform employs Web-based technology to handle requests from external programs. The external client programs use P6 Web Services by creating a request and sending it to the application server using the SOAP protocol which is essentially XML over HTTP. Having received the request, Primavera's Project Management software invokes whatever business logic is required to service the request. The client application need not understand the semantics of this processing. Responses or requests from Primavera's Project Management software simply follow the same path in reverse.

Services Oriented Architecture for Integration

A Services-Oriented Architecture (SOA) is an approach in which services are exposed for consumption via loose coupling to create new services and applications without regard to operating systems, programming languages and other technologies that underlie applications. SOA separates functions into services in order to create distinct units accessible over a network, which can be combined and reused in the production of applications. These services

communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services.

At Oracle, SOA is a key strategy for enabling enterprise integration. The Primavera EPPM applications such as P6, make use of Web services and an event-driven architecture (ERA) to allow integration to ERP systems, desktop tools, portals, etc.

In the SOA world, WSDL is a critical element. WSDL provides a standard XML document format for describing services. Services are defined in WSDL files as collections of network endpoints, sometimes referred to as ports. The WSDL specification provides an XML format for documents for this purpose. The abstract definition of ports and messages are separated from their concrete use or instance, allowing the reuse of these definitions.

Contract First Approach

P6 Web Services uses a contract-first approach in which WSDL files are used to describe itself to requesting applications. The WSDL uses the Document/Literal Wrapped style to describe the services and their operations. The Document/Literal Wrapped style indicates that Web services exchange messages as SOAP envelopes that contain a message body and an optional message header. The message body is comprised of an XML document that is constrained by a WSDL description of the web service. Furthermore, the message body contains an operation name that defines the outer wrapper element for both the request and response messages. The contract first approach is supported by a broad-based set of tools, promotes stability, and enables you to generate your own API.

Protocols and Processing Modes

P6 Web Services supports both asynchronous and synchronous processing of requests over either of the HTTP or HTTPS protocols. Your client program use any combination of HTTP, HTTPS, asynchronous, or synchronous protocols and processing modes to invoke any of the operations. Other advantages of connecting P6 Web services with an enterprise environment include scale-out of your environment as well as high-availability through the use of clustering for load balancing. Attachments are also supported in P6 Web services using MTOM attachments for import/export operations.

P6 Web Services v.7.0 supports passing authentication tokens using UsernameToken Profile and timestamping messages. With the UsernameToken, the P6 user's credentials are passed in the SOAP header with each call, eliminating the need to use a separate Authentication service with HTTP cookies. When utilizing P6 Web services in a BPEL orchestration, you will not need to do any custom processing to handle a cookie.

MTOM is utilized here because it is the W3C standard method for sending binary data to and from Web services. MTOM uses XOP (XML-binary Optimized Packaging) to transmit binary data to be used with SOAP messages. MTOM allows more efficient sending of binary data in a SOAP request or response.

Event Driven Architecture (EDA)

P6 v7 exposes a specified set of business objects & operations passively via Web services where you can access P6 through polling and querying. Or you can subscribe to event notifications via the publisher/subscriber model utilizing a message queue. This model allows external components to subscribe to published events emanating from P6 and to trigger actions such as workflow in BPEL or BPM, automatically in an active fashion. For example, this technique would be useful to update a .NET service that is registered to listen to updates to an activity in a specified project to send updates to SharePoint Web parts, Outlook Tasks or work orders.

Event-driven architecture (EDA) in P6 v7 allows external components to register or "subscribe" to event notifications when specified objects are created or updated. Pub/sub can be used via an enterprise service bus (ESB) for large implementations, or native JMS.
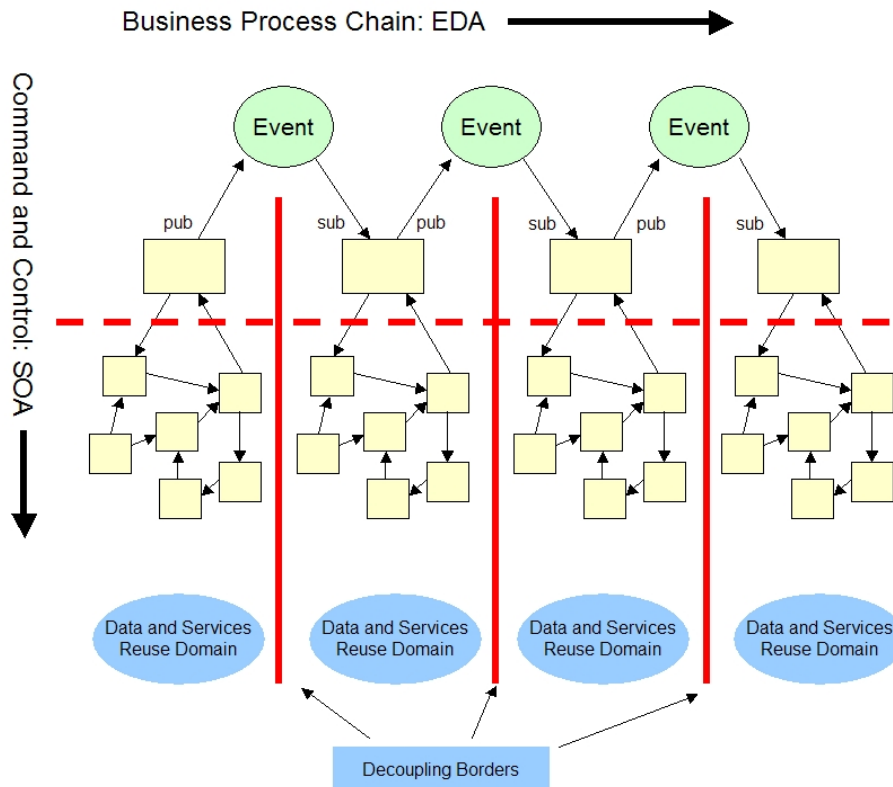

Figure 1 The pub/sub event architecture available from P6 utilizing a Message Queue

Standard Web Services Specifications

New for Oracle Primavera P6 v7 Web Services is support for these WS-* standards:

- **WS-Security** provides ways to secure your services above and beyond transport level protocols such as HTTPS. Through a number of standards such as XML Encryption, and headers defined in the WS-Security standard, it allows you to:

  - Pass authentication tokens between services

  - Encrypt messages or parts of messages

  - Sign messages

  - Timestamp messages

- The **WS-Policy** framework provides a general purpose model and syntax to describe the policies of a Web Service. For example, your environment may require transport-level security for protecting web service messages. This requirement can be defined by using a policy. For detailed information on WS-Policy and the syntax for writing policy expressions, please visit http://www.w3.org/Submission/WS-Policy. P6 Web Services has the ability to enforce policy assertions via the underlying CXF services framework. The policy assertions are defined in a single file called policies.xml (see user guide for enabling WS-Policy). In P6 Web Services v7, Oracle officially supports policy assertions for enforcing WS-Addressing and transport level security. When transport level security is enabled (by enabling the policy), P6 Web Services will only service requests that are invoked via an HTTPS connection. You will be able to enable WS-Policy on a per-service basis using policy assertions.

- **WS-Addressing** is a standardized way of including message routing data within SOAP headers. When WS-Addressing is enabled, P6 Web Services will only accept client requests that contain addressing information in the SOAP headers. One benefit of using WS-Addressing with P6 Web Services would be a case where asynchronous processing of messages is required. In this case, clients will send messages to P6 Web Services and can specify the endpoint where P6 Web Services will send the response. This endpoint can be the original endpoint that sent the request or a different endpoint. You will be able to enable WS-Addressing on a per-service basis using policy assertions.

Web Services are designed on open standards such as XML & SOAP for messages, HTTP for transport and a series of specifications for security, addressing and notifications.

CONCLUSION

Oracle Primavera P6 v7 provides enterprise-level integration capabilities to connect your project portfolio management system into your enterprise. With P6 Web Services, you can connect P6 to your ERP system, to your financial systems, to desktop productivity tools like Microsoft Outlook and Microsoft Excel, and even create Internet mashups or add P6 EPPM portlets to your enterprise portal on SharePoint. The event-driven architecture in P6 v7 will provide real-time project portfolio updates to your work order system, ticketing systems, inventory systems, or any other line-of-business enterprise application. These capabilities provide a complete enterprise-wide integration strategy that will increase your business productivity and project success by bringing your project teams together through collaboration and integration.

ORACLE®

Oracle Primavera P6 EPPM Integrations with
Web Services and Events
November 2009
Authors: Mark Kromer, Lee Goldschmidt, Pete
Angstadt

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment