

Collaborate 16 - Primavera Gateway Out of the Box

Session ID: 10702

Daniel Williams, PhD

Director of Integration, Calance

[Abstract](#)

[Primavera Integration History](#)

[Gateway Goals](#)

[Out of the box providers](#)

[Providers and Deployments](#)

[Business Flows](#)

[Modifying Business Flows](#)

[Groovy Mapping](#)

[Gateway Future](#)

Abstract

Primavera Gateway is a recent product, released in 2014, designed to simplify integration between P6 and other systems. But what exactly is it, and what do you get out of the box with Gateway? In this presentation we discuss the history of Gateway, the architecture, how to use it, and the future of this product.

Primavera Integration History

The story of Primavera is the story of multiple software systems. Joel Koppelman and Dick Faris started with Primavera with a product called Primavera Project Planner (P3) in 1983. Over the lifetime of the company, Primavera purchased many other companies and products, and from the very first days it was a struggle to make them all work together and appear as a unified “Primavera” brand. That struggle continues today, and in some ways is even more pronounced under Oracle. But first let’s step back to discuss how systems can communicate.

In the 1908’s the idea of systems communicating with each other was not on many people’s radar. Personal computers were new, and the internet basically non-existent. In the 1990s we first got access to Application Programming Interfaces (APIs), which were entry points allowing software to communicate, but there were no clean standards and the variety of APIs proliferated. Some systems had APIs, most did not. In the 2000’s the influence of the internet inspired APIs which worked over the fairly simple HTTP protocol. Yet again, the specific ways

of using that protocol varied wildly. During this time companies such as Microsoft and IBM began producing integration platforms such as BizTalk and Tivoli. Oracle too began its Fusion Middleware/SOA Suite family of products.

At Primavera, there was a Java API for P6 in the 1990s, and a COM-based API for Primavera Contract Management (PCM). Once both systems went to web-based products in the 2000s, both began to get HTTP-based interfaces, which collectively are called “Web Services”. These were ways of communicating with systems, but did not in and of themselves integrate any specific data.

Primavera used these APIs to develop integrations for their products. They created TGIF for integrating Timberline Accounting with PCM, and also created Inspire for integrating SAP and P6 (Technically Bristlecone developed Inspire, but Primavera bought the product).

When Oracle acquired Primavera in 2008, they began trying to integrate Primavera with their own systems using the Fusion Suite of products. But due to the complexity of that software, integration projects tended to be either so simple as to be unhelpful, or so complex to be single, large, one-off, projects for a specific company.

Rather than wait for Oracle to provide a solid solution, the Primavera General Business Unit (PGBU) decided to create an integration platform themselves. A key goal was to replace the myriad of integration technologies with one. This is Gateway.

Gateway Goals

Gateway does not eschew Fusion/SOA Suite technology. The ideas of web services, pluggable architecture, and independent communication layers is sound. But Primavera consciously decided to not build a generic integration engine. Instead, they built something that is focused on P6, and which allows the Primavera team to replace the hodgepodge of technologies with one solid, supportable product.

This technology, Gateway, was first used to replace the Inspire integration between SAP+P6, yet another acquisition from the Primavera days which Primavera had to support. Once the SAP integration was complete, Primavera continued to roll out data providers for other systems with each new release. From a support point of view, this is a big relief. Rather than supporting many integration technologies, they can support one, which also gives them the ability to continue improving on the core product.

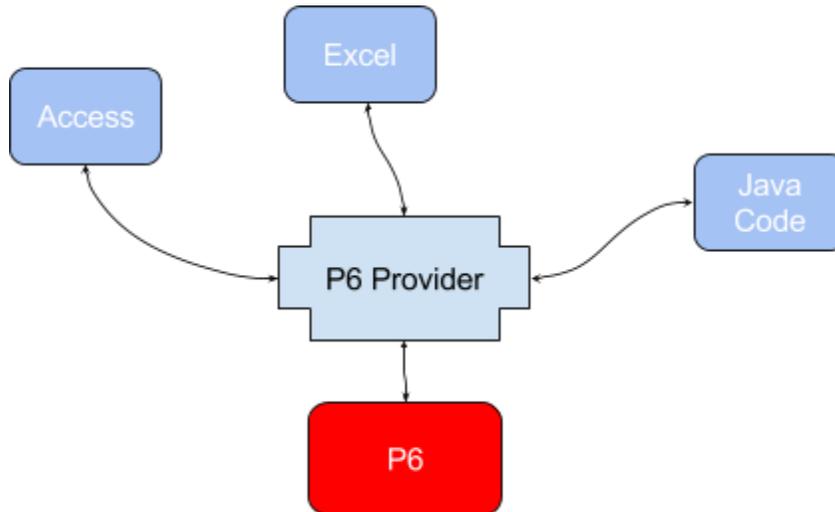
Here are some of the goals of Gateway:

- One complete platform for integration
- Tie together strategic products with P6
- Provide a set of supported integrations out of the box
- Allow extensibility through customization

These goals drive the development of Gateway, and explain many of its features.

Out of the box providers

Data providers are pieces of software that allow programs to talk to other programs. This sounds pretty confusing, right? How about a diagram:



Here we have the P6 program itself. P6 is a great product and it does its own thing. It is not a product for integrating data by itself. Instead, it has an API which something like a P6 Provider can use to communicate with P6. For example, the P6 Provider can tell P6 to give it a list of projects and activities. The Provider can also ask P6 to add, update, and delete projects and activities. Other programs can talk to the provider. Such programs could be Excel, MS Access, or a custom Java program. Really anything that can communicate over the Provider's protocol and knows the proper syntax.

The Oracle Primavera team works to produce new providers for other products. And very importantly, the team has provided a set of specifications which allow others to create their own adapters. These teams may be within Oracle, but part of another division working on other products such as VCP or EBS. They may be external companies writing providers for their own products, or they may be technology companies creating providers for yet more systems. With the specification provided by Primavera, anyone in theory can write their own providers.

In the 15.2 release of Gateway the following eight Providers are available:

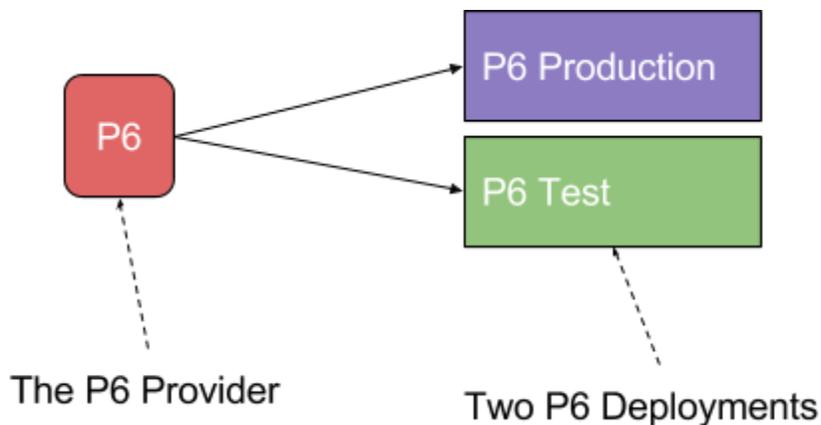
- P6
- SAP
- Unifier
- Prime
- EnterpriseTrack
- Value Chain Planning

- E-Business Suite
- Sample

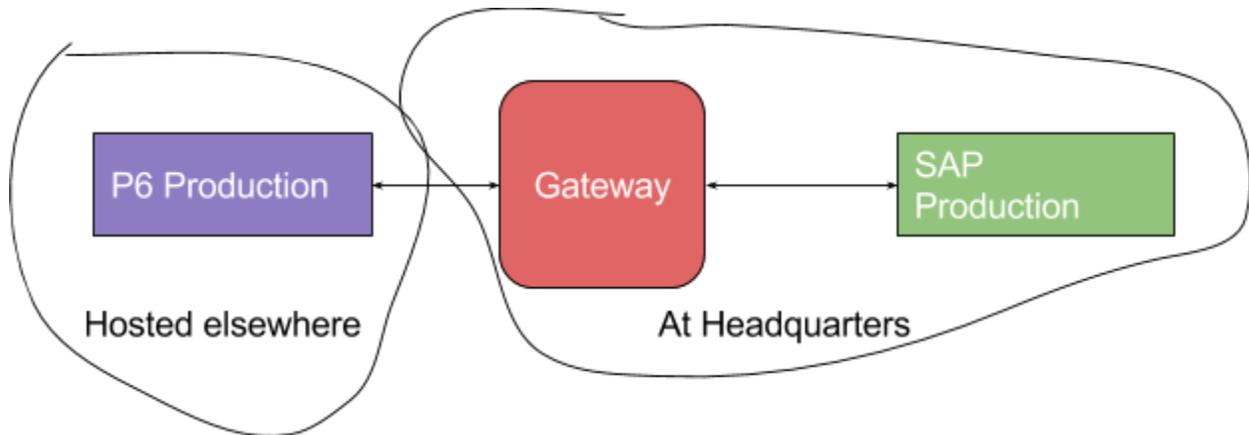
There is a pattern here. On the one hand there are providers for non-Primavera Oracle products such as VCP and EBS. But there are also ties to Unifier, Prime, and EnterpriseTrack. These last three are very important, because it helps to unify these products under the Primavera brand. Now, rather than users wondering why their different Oracle Primavera products don't talk to each other, the answer is that they do - using Gateway.

Providers and Deployments

A Provider becomes a Deployment when you give it specific information on how to communicate with its underlying system. For example, you can set up one P6 Deployment for the Production system, and one for the Testing System. So the same provider, but different instantiations of it.



Gateway will synchronize data between deployments. A key idea behind Gateway is that the systems involved do not have to reside in the same network. If a provider can grant access to P6 in another environment, such as a hosted solution, then Gateway can still integrate data. In fact, none of the systems, not even Gateway, need to be in any specific location!



A Provider also contains a list of data objects it supports. An example would be a Project in P6 or a Cost Breakdown Summary in Unifier. The Provider says what properties on each object are available, and also says what Business flows its, and its objects can participate in.

Business Flows

Gateway has the concept of data Flows, and in keeping with the P6-oriented nature of the system, these Flows are of two types: Master Business Data and Project Data. A Flow Type is basically a set of steps for integrating data.

- Load Source Data
- Convert Source to Gateway
- Load Destination Data
- Convert Destination to Gateway
- Compare Gateway-mapped source and destination data
- Convert compared data to destination format
- Review Data
- Update Destination
- Save Feedback

These are very generic steps and do not do anything in and of themselves other than set up what happens in the actual Business Flow.

A Business Flow does the actual work. It determines which Deployments are the source and destination. It determines the mapping from each of those systems into a middle-layer format called the Gateway layer. You may have noticed that there is a provider called Gateway. This is a special provider which defines the middle layer. Data objects and their properties from Providers must map into this layer. This means that if you are mapping a system which looks similar to P6, you're in good shape. But if you are mapping into a very different system, you may not be able to do what you need.

Below is an example of the out of the box business flows for Master data.

Flow Type: Master Data

Information Field Mappings **Business Flows**

| Name | Description | Source | Destination |
|-----------------------------------|--|-----------|--------------|
| Create EPS Code in Prime | Create EPS Code in Prime | P6 | Prime |
| Create Workspace Code in P6 | Create Workspace Code in P6 | Prime | P6 |
| Master Data Flow | To demonstrate how to create business flo... | P6 | P6 |
| Send Resources and Roles to Prime | Send Resources and Roles to Prime | P6 | Prime |
| Send Resources to Prime | Send Resources to Prime | P6 | Prime |
| Send Roles to Prime | Send Roles to Prime | P6 | Prime |

As you can see, there is a flow which maps EPS Codes from P6 to Prime. Another one sends Resources to Prime.

More interesting flows are available at the Project level, as shown below.

| Name | Description | Source | Destination |
|--|--|-----------|--------------|
| Create Projects from P6 to Prime | Create Projects from P6 to Prime with fiel... | P6 | Prime |
| Create Projects from Prime to P6 | Create Projects from Prime to P6 with field ... | Prime | P6 |
| Get Activity data from P6 | create P6 Activity Sheet in Unifier projects | P6 | Unifier |
| P6 Export Project Data Flow for Eventing | P6 Export Project Data Flow for eventing | P6 | Sample |
| Sample Import Project Data Flow | To demonstrate how to create business flo... | Sample | P6 |
| Send Activity data to P6 | Update P6 Activity Sheet to P6 Activity | Unifier | P6 |
| Send to Primavera Unifier | Send to Primavera Unifier with field mappin... | P6 | Unifier |
| Sync P6 data to Prime | Sync P6 data to Prime | P6 | Prime |
| Synchronize Approved Budget and Plan Dat... | Synchronize Approved Budget and Plan Dat... | Prime | P6 |
| Synchronize Planned Budget and Plan Date... | Synchronize Planned Budget and Plan Date... | P6 | Prime |
| Synchronize Project Actual Cost & progress ... | Synchronize Project Actual Cost and progre... | P6 | Prime |
| Synchronize Proposed Budget and Plan Dat... | Synchronize Proposed Budget and Plan Dat... | Prime | P6 |
| Update from Primavera Unifier | Update from Primavera Unifier with field ma... | Unifier | P6 |
| Update P6 Project with Prime data | Update P6 Project with Prime data | Prime | P6 |

Here we have functionality for sending Activities from P6 to Unifier or from Unifier to P6, for sending projects from one system to another, and sending cost information between systems.

Modifying Business Flows

You can modify Business Flows and create new ones in Gateway. In order to understand how, let's discuss Field Mappings. These are stored in Gateway as mapping between two providers, two objects, and the fields to be mapped.

Below is a screen for the mapping which takes Unifier Activity data and sends it into P6. In this case we have three fields mapped:



Edit Template

1 General 2 **Mappings** 3 Summary

Unifier: P6ActivitySheet Gateway: Activity P6: Activity

| | | | |
|----------------------|---------------------------|-------------------------------|-----|
| ElementId (String) ▼ | ActualDuration (Double) ▼ | AccountingVariance (Double) ▼ | Add |
|----------------------|---------------------------|-------------------------------|-----|

uuu_P6ActivityId : Id : Id [Delete](#)

uuu_P6ActualStart : ActualStartDate : ActualStartDate [Delete](#)

uuu_P6ActualFinish : ActualFinishDate : ActualFinishDate [Delete](#)

The Unifier object called Activity Sheet maps to the P6 object called Activity. The items in the white drop down boxes are not mapped, so please ignore them for now. The three items below this are the actual mappings. The Unifier field “uuu_P6ActivityId” maps to the Gateway field “Id” which then maps to the P6 field “Id”. Likewise, the start and finish dates are also mapped. What is shown above is called a “Direct” mapping. Fields in each system map into another. This is the simplest form of integration.

Groovy Mapping

The next out of the box mapping you can do uses a java-like programming language called Groovy. An example of a mapping using this method is shown below.

Syntax OK

| | |
|--|---|
| <p>Direction</p> <p>Provider 1 (Unifier) to Gateway</p> <p>Source Fields (Click to choose)</p> <p>uuu_P6ActivityName, uuu_P6WBSCode</p> <p>Target Fields (Click to choose)</p> <p>Name</p> <p>Groovy Expression</p> <pre>{Name} = [uuu_P6ActivityName].toUpperCase() + "-" + uuu_P6WBSCode.toLowerCase();</pre> <p>Validate Expression</p> | <p>Direction</p> <p>Gateway to Provider 2 (P6)</p> <p>Source Fields (Click to choose)</p> <p>Name</p> <p>Target Fields (Click to choose)</p> <p>Name</p> <p>Groovy Expression</p> <pre>if ({Name}.startsWith("BOB")) { {Name} = {Name} + "s your uncle"; }</pre> <p>Validate Expression</p> |
|--|---|

Here is the business rule:

The P6 Activity name should be updated from the corresponding Unifier activity, and should be the Unifier Activity Name in uppercase, plus a dash, followed by the Unifier Activity's P6 WBS code in lowercase. If the Unifier Activity Name starts with "Bob", then append "s your uncle".

So Unifier Activity named "Bobby" with WBS named "Motown" would translate into a P6 Activity named "BOBBY - motown's your uncle".

To do this we select the two Name and WBS fields from Unifier and tell them to map into the Gateway Activity Name field. This is the left side of the image.

The "Validate Expression" button will tell you if you've written valid Groovy code. It can be a challenge to get all of those parentheses, semicolons and squiggly brackets exactly right, since this is simply a text box and not an IDE by any measure!

That is for the Unifier to Gateway mapping. For the Gateway to P6 mapping, we choose the Gateway Name field as the source and the P6 Activity name field as the target. This next script looks at the Gateway Activity name, and checks to see if it starts with "BOB". If so, then adds "s your uncle" to the end.

So there is the groovy customization. As you can see, you don't want to get too complicated with this mapping, and you are limited to the fields available on the specific objects present. Also, even though the mapping is from Unifier to P6, we have to have additional mappings to the Gateway layer.

Gateway Future

Here in early 2016, the future of Gateway is murky. Oracle is well on track creating new providers for Gateway, which is a good sign. They started with SAP as the initial concept in the first release. The next releases had Unifier and VCP, two Oracle systems which naturally overlap P6. The release of the EBS adapter in 15.2 marks Oracle's first tie in with their own ERP system, and it is possible that JDE or PeopleSoft may not be far behind. Recent releases of P6 and Unifier have screens for setting up communications with Gateway, so it is certainly a key part of those products.

However, there are also a few trends which bring into question Gateway's future. Oracle has many products across many divisions.. They create databases, they create programming platforms, and they have many many products either built in-house or acquired over the years. Each of those groups tends to create their own integration platform. There is Oracle Fusion/SOA Suite, which is a suite of tools for integrating "any" system to any other. It does not do anything itself, but rather is a platform. Yet its complexity often drives users away. Another product along these lines is Oracle Cloud Integration Service. This is a friendly-looking system designed to let users set up integrations from a simple interface. It is unclear how Fusion, Gateway and OCIS will interact, but Gateway has the benefit that it has, right now, actual integrations supported by Oracle. It is not just a tool for making integrations.

Outside of Oracle there are many other competing products. Microsoft has BizTalk, Service Bus, and Azure services. Dell has Boomi, which already has interfaces for many non-Oracle systems such as SAP, as well as Oracle systems such as EBS, though nothing in the Primavera realm. And the software industry is always changing, so it's hard to predict what new products may emerge over the next few years.

Another interesting aspect of Gateway's future is the future of the Primavera products themselves. For example, while a strength of Gateway is integrating P6 and Unifier, the existence of P6 and Unifier as stand-alone products is called into question by Primavera Prime, which may eventually replace for both, unified into one product. Some functionalities of Oracle Risk Analysis is already in the process of being folded into Prime.

Yet those are just the Oracle Primavera products changing and evolving. There is also the spectre of Oracle Fusion. Note that this is the Fusion ERP system, not the Fusion/Soa Suite integration stack, though Fusion ERP relies heavily on those technologies. This project, many years in the making, was designed to take the best practices of the top three Oracle ERP systems: EBS, JDE, and PeopleSoft, and re-engineer them all into one new product built on a more modern architecture. Oracle Fusion HCM is already replacing EBS in new sales, and the accounting modules (GL, AP, and AR) are not far behind. Among the suites of this new system

is Oracle Fusion Project Portfolio Management. Will this product replace the entire Primavera suite?

My guess is that maybe, except for P6.

P6 is the one Primavera product that is most unique and most likely to remain as it is. It is specifically designed to plan and execute large-scale construction projects, with years of first-hand experience and best practices baked into the product. This is why it's so hard to get professional schedulers to use the web version - the desktop product gets the job done! So my guess is that Primavera P6 will still exist in 5 years' time. As for the other products, the future is not clear. There is a definite change in the software industry as systems become more lightweight, web and mobile capable, and user friendly. And all things are moving to something that is called the cloud. Rather than bury things under the work could, I would say that the future means software that you can access from anywhere, and which will not be harmed when your local PC dies or you drop your phone in the mud - call this cloud if you will. And these new systems by definition must talk to each other. The way that they communicate will change, but the idea that systems must communicate is not going to go away.

Note: The contents of this paper are the personal opinions and writings of Daniel Williams. They do not represent the views or opinions of Oracle, OAUG, or Quest in any way.