

Rapid Interface Development for the E-Business Suite and PeopleSoft in a SOA Environment

A Solbourne White Paper
April 2008

Rapid Interface Development for the E-Business Suite and PeopleSoft in a SOA Environment

Executive Overview

SOA, or Service Oriented Architecture, has become a proven, tested way to deliver integration of disparate systems. With Oracle's SOA Suite reaching maturity, creating a tight coupling with the E-Business Suite and /or PeopleSoft Enterprise is an important component for many large organizations. A complete replacement of existing integration using SOA tools is a massive task for most enterprises. Instead, an approach that provides incremental benefit from SOA is needed.

This White Paper will explain the key concepts behind a successful business process integration in which the E-Business Suite and/or PeopleSoft are the focal points. The primary takeaway from this paper is how to lay the foundation for a successful first foray into SOA, as well as set the stage for continued success as your organization develops a Service Oriented Infrastructure.

Introduction

For those working with ERP systems, the definition of an interface has traditionally meant simply bringing data from one system to another. This definition has begun expanding in both scope and scale as the concept of interfaces as Business Process Orchestration has taken hold. With Business Process Orchestration, the focus moves from simply transferring data to a stronger focus on *how* data is moved, emphasizing adding value to the process by complementing and enhancing the way work is done.

This enhanced concept of integrating systems is a result of the widespread adoption of Service Oriented Architecture (SOA), a platform that makes connecting to disparate file systems, databases, or business logic easier. But with an additional server and new tools and concepts to learn, the decision to use Service Oriented Architecture can be a daunting one.

So what do we mean when we talk about rapid interface development? With this added complexity, is it really possible to produce these types of value-added interfaces in a timely fashion? Absolutely. We'll talk here about what you'll need to do to prepare to develop interfaces both quickly and professionally. We'll focus on general concepts as well as some specific steps you'll need to take to prepare your environment for SOA development that will help you integrate your business data and processes more completely.

Assumptions

This paper assumes Oracle's E-Business Suite or PeopleSoft will be an integral part of the business environment where integrations are planned. A basic understanding of what SOA (Service Oriented Architecture) is and how Oracle's SOA Suite fits into an ERP system will also be helpful. A working understanding of the following list of Oracle products and technologies is assumed. They will only be covered at a higher level, as they pertain to the particular sections of the paper.

- JDeveloper
- BPEL Process Manager
- Oracle E-Business Suite technologies
- PeopleTools

Overview

The primary purpose of this paper is to help you anticipate the hurdles you'll face as you implement your first few SOA-based interfaces, and to get you up-and-running as quickly as possible.

Much more than just another IT acronym or buzzword, SOA is meant to be a platform, a place in which infrastructure is built to allow interfaces to happen at a much higher level. Its intention is to free data, free up business processes, to allow both to move more freely within any organization, or even between organizations. But with the added benefit of this increased cohesiveness comes the up-front cost of building the infrastructure it requires. Thus, building a solid foundation, a platform that can withstand the rigors of use and development for the long haul should be the first priority as any company moves into SOA development. From a solid foundation, interfaces can then be developed quickly and efficiently, as well as changed in-flight with ease.

Below you'll find discussion of some of the most significant issues you'll face as you begin to think about implementing a Service Oriented Architecture for your Business. By getting comfortable with the discussion surrounding these Business and Technology considerations, you'll find yourself better prepared for your initial foray into SOA.

Defining an Initial Project

For most organizations, it is critical that early SOA projects are successful and visible to justify future investments in SOA infrastructure. Since the benefits of a SOA architecture typically increase as the enterprise becomes more and more SOA-based, it is important to select initial projects carefully to create initial value and generate momentum towards these additional benefits.

Here are some types of projects that have been successfully used by customers to create momentum for SOA in their organization:

- **ERP projects:** ERP implementation or major upgrade projects typically provide significant resources to define business processes and develop solutions. These are also critical systems that have numerous internal and external integration points. By leveraging an ERP project, an organization can both establish a solid foundation and make a significant step towards a SOA-based enterprise. Leveraging SOA as part of these projects also prevents perpetuating legacy integration concepts, such as point-to-point integration, that the enterprise will have to retain for many years.
- **Front/back office integration:** In many organizations, the most critical integration is between the front office (order entry and/or CRM) and back office (ERP) systems. This creates an ideal opportunity for a single business process to have a high impact on the organizational efficiency.

This is reflected in Oracle's approach to developed pre-defined process integration packs (PIP) through the Applications Integration Architecture (AIA).

- **Time-critical processes:** Processes that greatly benefit from increased visibility and timeliness, especially to reduce costs and increase revenue, represent a great opportunity to leverage SOA integration and monitoring. These can often be identified by considering if a real-time Business Activity Monitoring (BAM) dashboard would help the business increase revenue or reduce exposure. Common scenarios include: real-time request backlog (orders, permits, complaints, requisitions, dispatch calls, etc.).
- **Technology-constrained processes:** Many processes remain manual because of technology constraints. The systems involved in the process typically either use legacy technologies (mainframe, client/server, etc.), are proprietary and do not expose web services (many packaged applications), or are custom applications that were not developed with external integration and composite applications in mind. Many of these systems were implemented because they are critical to the enterprise activities, and leveraging BPEL's strong connector capabilities allows for automation of these critical activities in a way that wasn't previously possible.

Regardless of the project selected, it is important that this initial project is delivered successfully, and creates a foundation that can be used for subsequent SOA projects. The remainder of this white paper focuses on recommendations to create a solid foundation and initial success.

Define Your Initial Architecture and Scalability Needs

The Oracle Application Server (current SOA release is AS 10g 10.1.3; AS 11g due in 2008) is the foundation of all Oracle SOA Suite implementations. A dedicated Oracle Home on a shared server is really all that is required for smaller installations of Oracle's SOA Suite. If you have an underutilized server, and space in an Oracle 10g database for the AS 10g metadata, you might already be closer to a SOA architecture than you know.

If, however, you have immediate plans for a larger implementation, or long-term plans for heavy processing and increased development, it's better to start early with a more robust, dedicated server. Dedicated hardware will provide better performance, reliability, and will allow for more seamless expansion of services. Key areas that contribute to heavy SOA processing requirements:

- Complex transformations and process routing
- Maintaining state information ("dehydration" and "rehydration" in Oracle BPEL PM)
- High transaction volumes
- Large transaction payloads

When considering the overall performance of a SOA architecture, it is always important to keep the 'partner' systems in mind as well. Even if the processing on the SOA server(s) is extremely fast, performance may be limited by the time required for the target systems to perform any resulting processing, particularly in a synchronous transaction.

If going the extra step to setup a dedicated machine for the SOA Suite, Oracle also recommends a dedicated Oracle DBMS system for the orabpel, orawsm, and oraesb schemas. These three schemas provide the database foundation for the SOA Suite.

In addition to system performance, availability should also be a consideration for SOA architectures. This involves evaluating the business impact if a SOA process is unavailable, and the acceptable time required for recovery. In some cases, a simple backup of the dehydration store is adequate. However, in many enterprise uses, a clustered solution is necessary to provide redundancy and high availability. Because Oracle's SOA architecture has both an application server tier and a database tier ("dehydration database"), both tiers should be considered in developing a SOA architecture.

For more information on high availability architectures for Oracle BPEL PM, refer to the Oracle white paper *Architecting BPEL Systems for High Availability*, available at http://www.oracle.com/technology/products/ias/hi_av/BPEL_HA_Paper.pdf.

Regardless of the approach you take for your initial architecture, you should also keep your future scalability needs in mind. Fortunately, the same approach that allows Oracle's SOA Suite to be deployed in a highly available environment can also be used to incrementally add processing capabilities. Rather than investing in larger and larger servers, you can add additional nodes in a grid to provide distributed and parallel processing to minimize hardware constraints on SOA performance.

Setups Are Crucial

When installing the SOA Suite server, remember that the server is built on the OC4J (Oracle Containers for Java) architecture and partitioning basic functionality according to OC4J leading practices can help maintain the integrity and robustness of your system. Specifically, it is a leading practice to follow the SOA Suite 'Advanced' install and choose the option to setup a separate home for the SOA Suite iAS Control. The iAS or EM Control is the control panel for the SOA Suite, and is itself a separate application that runs on the AS 10g server. By putting it in its own container, separate from the BPEL PM, Enterprise Service Bus, Oracle Web Services Manager (OWSM), or other primary SOA applications, you can effectively partition the administration component of the server from the functional components that run on it (Figure 1).

Figure 1
An example of an AS 10g server with a separate home for the Administrative components, separate from the WSM, BPEL PM, and ESB components.

ORACLE Enterprise Manager 10g
Application Server Control

Cluster Topology Page Refreshed Feb 25, 2008 12:22:18 AM MST • View

Overview

Hosts 1 Application Servers 1
OC4J Instances 2 HTTP Server Instances 1

Members

View By: Application Servers

[Start] [Stop] [Restart]

Select All | Select None | Expand All | Collapse All

Select	Focus	Name	Status	Type	Category	Host	CPU (%)	Memory (MB)
<input type="checkbox"/>	▼	All Application Servers						
<input type="checkbox"/>	⊕	AS3 D2YHKMB1		Application Server		D2YHKMB1		
<input type="checkbox"/>	⊕	home (JVMs: 1)		OC4J			1.33	111.16
<input type="checkbox"/>		ascontrol	↑	Application				
<input type="checkbox"/>		datatags	↑	Application	Service			
<input type="checkbox"/>		default	↑	Application				
<input type="checkbox"/>		javasso	↑	Application	Service			
<input type="checkbox"/>		WSIL-App	↑	Application	Service			
<input type="checkbox"/>		HTTP_Server	↑	Oracle HTTP Server			0.08	43.19
<input type="checkbox"/>	⊕	oc4j_soa	↓	OC4J			Unavailable	214.52
<input type="checkbox"/>		ascontrol	↓	Application				
<input type="checkbox"/>		ccore	↑	Application	WSM			
<input type="checkbox"/>		coreman	↑	Application	WSM			
<input type="checkbox"/>		datatags	↑	Application	Service			

One of the first things to have your DBA do after installing the AS10g server and Oracle's SOA Suite is to adjust several server settings, following Metalink Note 344086.1 which allows longer-running processes to complete without timing out. Out of the box, many processes running on the AS 10g server will time out

relatively quickly, which can often cause confusion and frustration. This can manifest itself as BPEL processes that 'disappear' or timeout errors that give an otherwise healthy BPEL or ESB process the appearance of failure. These timeout errors can also mask the real errors that are occurring behind the scenes.

Making these settings is also crucial for scheduled processes. BPEL processes are natively better at real time processing, using small batches of data or single transaction processing. In addition to the setups in the above Metalink Note, Solbourne has also worked with Oracle to come up with leading practices for creating large-batch, scheduled interfaces which use the SOA Suite and BPEL. As a result, we have been able to move batches of data between 5,000 and 20,000+ records, with both file and DBMS-based data sources using BPEL. In this way, if processes really are required to timeout after a certain period of time, using a timer in your processes or adding logic to timeout according to your business logic will allow you to take more control over your interfaces.

Application Server Connection Architecture

Connections are what the AS 10g server does best, and an important concept to grasp early is how connections are made to external sources both to and from the AS 10g server. JDBC is the foundation of the connections that happen on the AS10g server and Connection Pools are where the actual work is done to connect to any external system. This is where username, password, URL, and hostname are maintained and as such this largely the domain of DBAs.

From here, developers access the Connection Pool not by the name it is given, but through two different types of pointers to that Connection Pool. These two objects are called the (Managed) Data Source and Connection Factory. A Data Source is used to establish a JNDI (Java Naming and Directory Interface) name for the Connection Pool which looks like: jdbc/xxxxxx. Once this Data Source is established, it can be referred to in BPEL processes by simply providing the JNDI name and the direct connection is made to the Connection Pool as a result. This allows to important things to happen: first, the DBA never has to give away the password, and can change it at any time without affecting development. Second, the Developer can rely on the fact that, once setup, the JNDI name will never change, even if the connection details change.

But the Data Source isn't the end of the connection setup. A Connection Factory is setup and used by the Adapters (Database, File, AQ, FTP, etc.), which are really nothing more than applications deployed to the AS 10g server. This extra setup is required as the Adapters are themselves JMS (Java Messaging Service) based applications and a Connection Factory is required for them to make a connection to the Data Source. A JNDI name is also used for Connection Factories and looks like: eis/DB/xxxxxx. To sum up the relationships of these objects:

A Data Source makes a direct connection to a Connection Pool.

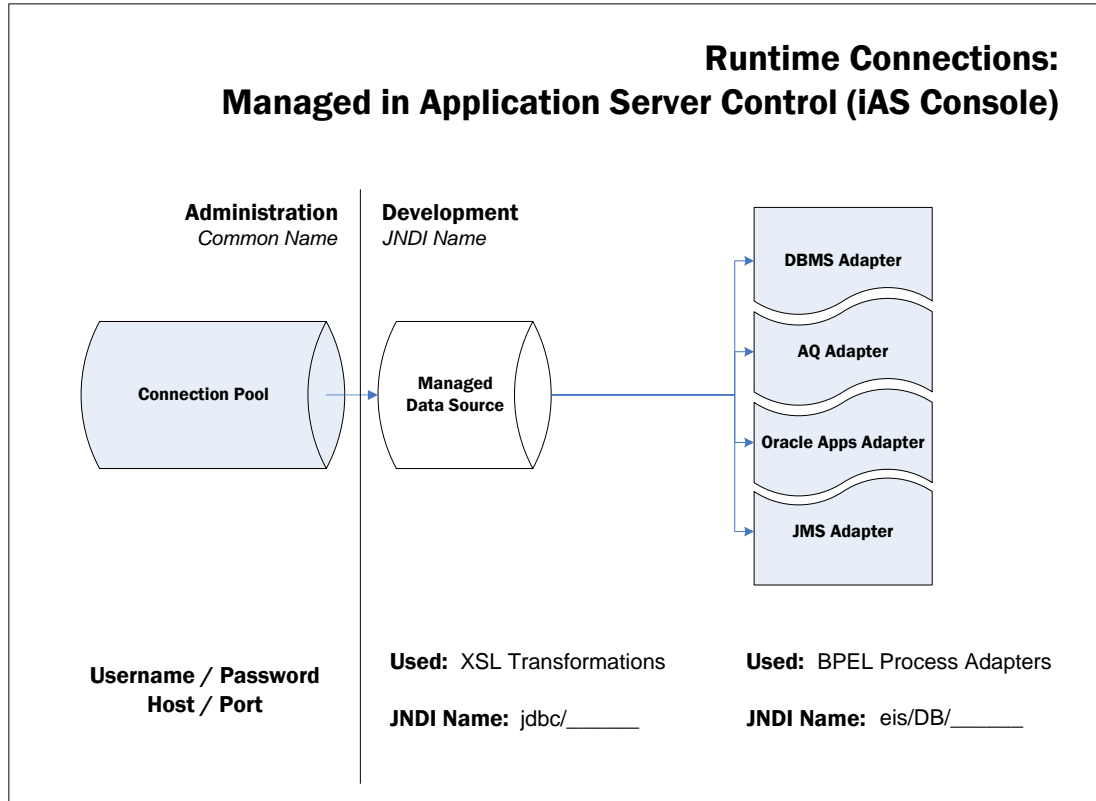
- It is possible to have more than one Data Source for any given Connection Pool, but this is not a best practice.
- While an adapter must use a Connection Factory to connect to a data source, a BPEL process can use this Data Source directly, in things like XSL Transformations to run queries on the DBMS

A Connection Factory makes a direct connection to a Data Source.

- It is necessary to have more than one Connection Factory pointing to any Data Source, as the types of adapters that use it are varied.

The above setup applies to the Database Adapter, AQ and Oracle Applications adapters primarily. When working with the File and FTP Adapters, you need only setup the appropriate Connection Factory, skipping the Connection Pool and Data Source steps (since there are no database operations involved). Establishing a JNDI naming convention for your connections, and assigning ownership for the maintenance of those connections should be done immediately post-installation and is a key component of successful Fusion Middleware implementations.

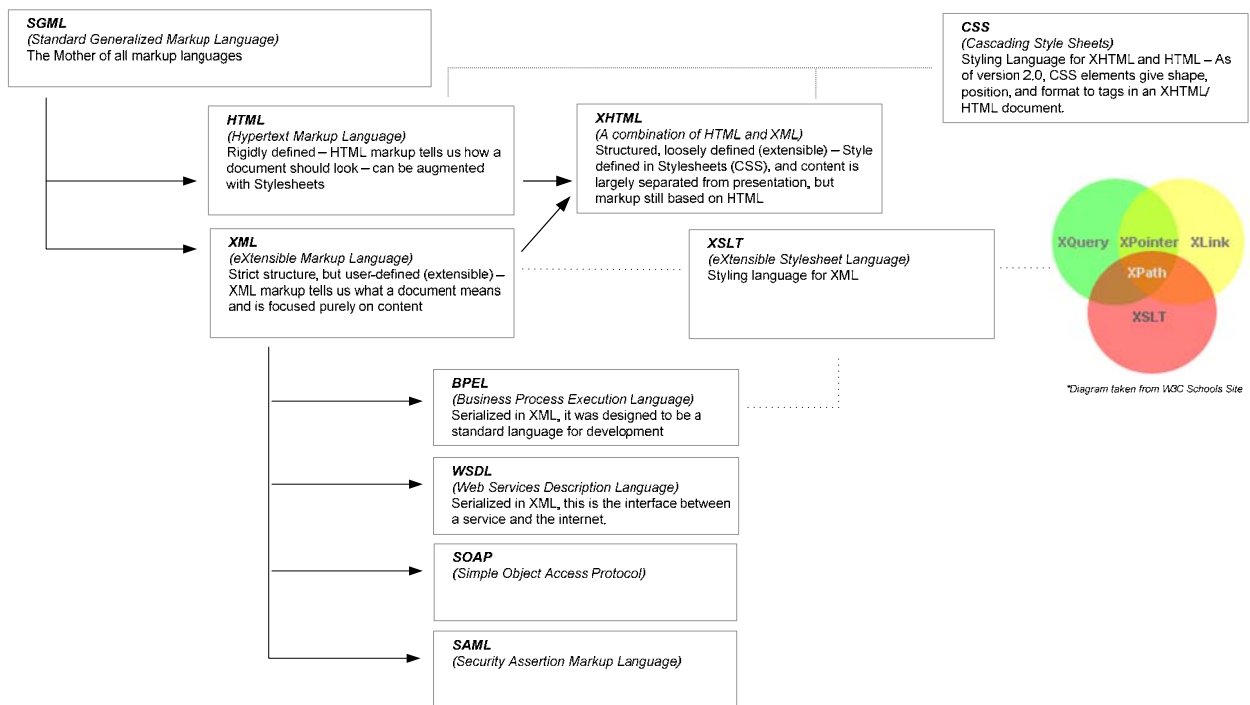
Figure 2 – Understanding the AS 10g Connection Infrastructure



Prepare Your SOA Knowledge Foundation

Whether you're doing a full-blown development project in house, hiring a consulting firm like Solbourne to help out or complete the project entirely, one of the best things you can do is become familiar with XML. Underlying everything you'll do in SOA and BPEL development is XML and its dependence on **XML schemas** and **XPath**. There are many free resources for learning about XML, XPath, and all the other acronyms that get thrown around in the SOA world, including the World Wide Web Consortium's site (<http://www.w3schools.com/>) as well as purchasable publications from Oracle Press and many others. Taking the time up-front to become fluent in the concepts surrounding XML and XML Schemas will save time and money both initially and in the long term.

Figure 3 – XML Family Tree
(expand this document to 200% for better viewing)
Solid lines indicate Parent/Child relationships
Dotted lines indicate complementary languages



Planning for Human Workflow

Human Workflow is an addition to the BPEL 2.0 standard that allows for routing decisions to humans, by means of emails (actionable or just notifications), or the built-in BPEL Worklist which comes with the BPEL PM. If you're planning to use Human Workflow in your BPEL Processes, however, decide *early* in the process where your user repository, including email addresses will be located, and begin the infrastructure setups to make that work.

Examples of user repositories include Active Directory, a Custom Identity Service plug-in, and Oracle Internet Directory (OID). Metalink Note:464060.1 discusses setting up a direct connection to Active

Directory, and Oracle has made this process relatively easy with step-by-step instructions. OID is the most common way to setup a user repository when working with BPEL, but it does require an AS 10g 10.1.4 server that is separate from the AS10g 10.1.3.3 where BPEL lives. A final way that user data can be accessed is via custom schema, using the APIs provided by Oracle to create a Custom Identity Service plug-in. Although less common, Solbourne has worked with Oracle to implement this solution for clients who have neither an LDAP directory (Active Directory is one example of LDAP), or OID.

For the Custom Identity Service plug-in, a Java-based custom Identity application is written and deployed to the AS 10g 10.1.3.3 server that eliminates the need for a more complex OID setup or connection to Active Directory. A set of custom views is then placed in the E-Business Suite, PeopleSoft, or any third party schema. These views are then called from the custom Java program. The view shown below is a portion of one of the views Solbourne has written especially for the E-Business Suite but could be customized for any database schema, as we have done on other projects as well.

Figure 4 – sample Custom Identity Service database view

```
----- WLS_POLICY_0 CANNOT BE ALSO REQUIRED IN THIS SITUATION -----  
CREATE OR REPLACE VIEW wfis_user_view  
AS SELECT  
  b.person_id AS userid  
  , g.realm AS realm  
  , h.user_name AS name  
  , b.first_name AS firstName  
  , b.middle_names AS middleName  
  , b.last_name AS lastName  
  , b.last_name||', '||first_name AS displayName  
  , ' --- ' AS description  
  , f.name AS title  
  , nvl((SELECT  
        SUP.full_name  
        FROM  
          PER_PEOPLE_X          PEO  
        , PER_PEOPLE_X          SUP  
        , PER_ASSIGNMENTS_F     ASS  
        WHERE  
          PEO.person_id = ASS.person_id  
        AND ASS.supervisor_id = SUP.person_id  
        AND PEO.employee_number = b.employee_number  
        AND PEO.business_group_id = SUP.business_group_id  
        AND PEO.business_group_id = b.business_group_id  
        AND rownum <= 1), '-- No Manager --')  
    AS manager  
  , b.email_address AS mail  
  , (select pp.phone_number from HR_LOOKUPS hrl, PER_PHONES pp  
    where hrl.lookup_code = pp.phone_type and hrl.meaning = 'Work'  
    and no parent id = b person id and rownum <= 1)
```

A final consideration for preparing your SOA environment to interact with people is to decide whether you want emails to be involved. Simple email notifications can be setup independently of more complex Human Workflows or used in conjunction with Human Workflow to provide actionable emails. Email notifications themselves do not require a user repository to be setup, but do require an SMTP server to be configured.

To setup emails, you need only setup the appropriate file on the server and restart. This file is ns_emails.xml and is located at \$ORACLE_HOME/bpel/system/services/config/ns_emails.xml. The Oracle SOA Suite Developer's Guide has more information on this and can be found at:

http://download.oracle.com/docs/cd/B31017_01/core.1013/b28764/bpel013.htm

Understanding your business needs can help focus your implementation decision. By preparing early and understanding the infrastructure that goes along with using Human Workflow, it can be much smoother to implement and a real benefit to your interfaces.

Working with PeopleSoft

Working with PeopleSoft and the SOA framework offers a unique opportunity for integration. PeopleSoft's Integration Broker, a component of PeopleTools, offers a guided setup assistant for creating web services, based on PeopleSoft functionality. Integration Broker is one of the rare tools in the ERP world that really is almost as easy to use as it first appears. A set of screens will take you quickly through the setup which will provide you with both a WSDL and XML Schema for creating or consuming web services.

The functionality setup by using the Integration Broker, for outgoing messages, works as a functional trigger pushing data out to the system that is listening on the created web service. When you Save a new entry in the PeopleSoft system, creating a new Person for example, the data would be immediately exported. Inbound interfaces act similarly, and are triggered as data they are listening for arrives. When extending your business processes and data outside of PeopleSoft, for real time interfaces, using the Integration Broker should be your first choice.

There is one hitch to using the components created with Integration Broker. As mentioned previously, schemas are the backbone of SOA/web services development. Schemas both define and validate the message types¹ that are used to pass XML data between systems. When you create the Integration Broker component to expose PeopleSoft logic as a web service, you also create the schema that will be used by any services that consume it.

The catch is that the schema you'll get directly from the Integration Broker is *not quite* ready for use in a BPEL process. The basic structure is great, and offers a very rich and complex data set. However the structure of that schema needs to be massaged to be 100% compatible with the schema requirements of a BPEL process, to create variables based on the elements in that schema.

If for any reason you decide not to use the Integration Broker, whether you want to import/export data in batch mode or simply want to pull a small subset of data from a table or file, you can use the Database Adapter, File, and FTP adapters instead. These adapters allow you to query, insert, update, delete data just as you would for any third-party system, quickly and easily. As with any system, you have only to setup the Connection Pool, Data Source, and Connection Factories that provide the connection to the database or file system. Then create your BPEL process to perform whatever operation you like, based on those connections.

For additional information on using Oracle SOA Suite with PeopleSoft Enterprise, refer to Solbourne's *Using PeopleTools for Seamless Integration with Oracle's SOA Suite and BPEL* white paper, available at <http://www.solbourne.com/downloads.html>.

Working with E-Business Suite

There are essentially three approaches to integration with E-Business Suite:

- **Leverage the Oracle Applications Adapter:** Despite the generic name, this adapter is specific to E-Business Suite. Oracle is continuing to develop capabilities through the Oracle Applications Adapter, and view this as a strategic direction for E-Business Suite SOA integration. In its current state, a project should evaluate this adapter to determine if it effectively meets the use cases. If it does meet the use cases, it can significantly reduce the integration effort required.
- **Expose services using XML Gateway to the Oracle SOA Suite:** XML Gateway is similar in purpose to PeopleSoft's Integration Broker. While on the surface they provide similar functionality, web service enabling the ERP application in which they run, in reality they are very different. Integration Broker has a single interface which allows you to quickly setup web services going into or out of PeopleSoft. XML Gateway accomplishes the same goals, but with far more moving parts. As a result, when using BPEL for integrations into the E-Business Suite, we recommend bypassing XML Gateway, as this keeps the complexity much lower.
- **Leverage Oracle technology adapters:** The third approach is to utilize the SOA Suite technology adapters. Using this approach, the interfaces into and out of the E-Business Suite rely heavily on PL/SQL inside the E-Business Suite, and Oracle's SOA Adapters are used to bridge the gap to the outside world. When working with bringing data into or out of the E-Business Suite, you'll work primarily with these basic technology tools:

¹ Message types are the variables of the SOA world. They can be simple data types (String, integer) or they can be more complex data types similar to record types in PL/SQL, Arrays or Classes in Java, or a Struct in C/C++

When creating interfaces using any of the above approaches, the following basic technologies will also be used to implement or complement their functionality:

For Outbound Interfaces:

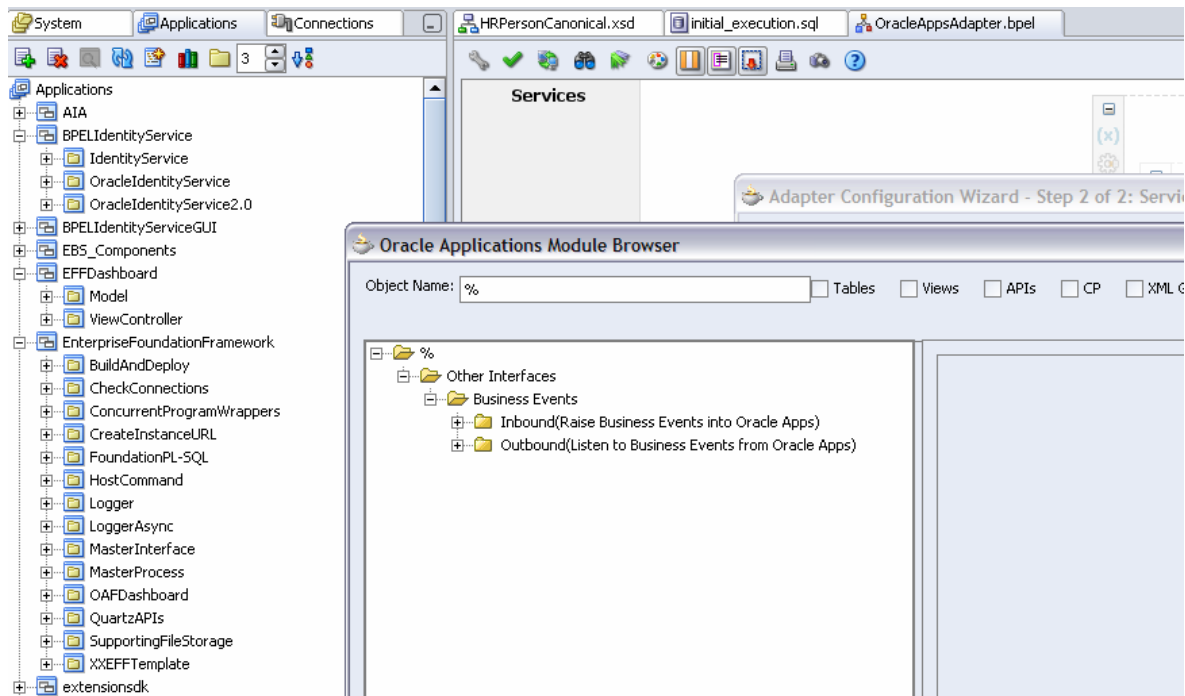
- Business Events
- Custom PL/SQL
- AQ

For Inbound Interfaces:

- Open Interface Table (OIT)
- Module specific APIs
- Custom PL/SQL

Business Events have been created for nearly all operations inside the E-Business Suite, and PL/SQL 'subscriptions' can be written to extend these logical triggers for real-time interfaces. Oracle's E-Business Suite Adapter can be used to find Business Events and create these Subscriptions. This adapter is built into JDeveloper and comes with an additional cost when deployed to the AS 10g 10.1.3 server. In addition to Business Events, the Applications Adapter can help identify and create calls to APIs, Concurrent Programs,

Figure 5 – Using the Oracle Applications adapter to find Business Events



XML Gateway functionality, and help locate specific tables and views by E-Business Suite module.

Once you have created the initial calls using the Applications Adapter, you can tweak the starter code it produces to enhance its functionality further. By providing the E-Business Suite context for Concurrent Program and API calls on inbound interfaces (using the seeded package `fn_d_global`), you can submit programs as a particular user in the Applications. By writing additional PL/SQL to capture the status of the programs you call for inbound interfaces, you can provide robust error handling and process reporting that will make your interfaces easier to monitor, test, and maintain.

Outbound interfaces in the E-Business Suite often require a bit more work as the move from normalized data to XML can lead to a bit more work in PL/SQL. For example, header and lines data, common in all ERP systems, resides in two different tables, with a one-to-many relationship. How do you move that kind of data into a single BPEL process that needs all of it in the same message variable? Performing an operation of this nature requires a translation of normalized data serializable XML. PL/SQL Objects are the answer to this dilemma and represent the single largest asset for making the transition between normalized data and XML.

The following example demonstrates how to handle a simple master-detail set of data, when pushing that data to a BPEL process.

Figure 6

Moving Master-Detail Data From the E-Business Suite to a BPEL Process: a Practical Example

Moving data from a BPEL process to the E-Business Suite is relatively easy thanks to Oracle's many different adapters which translate XML data to PL/SQL very well. In this simple example, however, we are going the opposite direction: capturing business logic in the E-Business Suite and pushing it out to a BPEL process.

The procedure below accepts a `person_id` as a parameter and returns a simplified set of the main person data, along with all assignments the person may have (again, simplified for clarity). Here, the results are stored in a PL/SQL table of type Object to return this data to the calling BPEL process. The resulting procedure parameters act as both Header and Detail information, providing a BPEL process with a clear picture of what is happening within the E-Business Suite..

```
CREATE OR REPLACE TYPE XXEFFHR_PERSON_HEADER_O AS OBJECT (  
    PERSON_ID                NUMBER  
    , PERSON_HIRE_DATE        DATE  
    , HOME_PHONE_NUMBER       NUMBER  
    , HOME_ADDRESS            VARCHAR2(240)  
);
```

```
CREATE OR REPLACE TYPE XXEFFHR_PERSON AS OBJECT (  
    PERSON_HEADER             XXEFFHR_PERSON_HEADER_O  
    , ASSIGNMENT_ID           NUMBER  
    , ASSIGNMENT_EFFECTIVE_DATE DATE  
);
```

The Object above is then used to create a PL/SQL table type:

```
CREATE OR REPLACE TYPE XXEFFHR_PERSON T AS TABLE OF XXEFFHR_PERSON;
```

The above referenced table type is used to populate the results of all processes associated with this import program, returning an E-Business Suite's concurrent request ID, and status of the concurrent request. That table type is then used as an OUT parameter in a PL/SQL procedure.

```
PROCEDURE GET_ORACLE_APPS_PERSON(  
    P_PERSON_ID                IN VARCHAR2  
    , P_PERSON_DATA            OUT XXEFFHR_PERSON T  
);
```

For additional information on using Oracle SOA Suite as an alternative or complement to Oracle E-Business Suite XML Gateway and Workflow, refer to Solbourne's *Using BPEL and Workflow to Build Integration Components for the E-Business Suite* white paper, available at <http://www.solbourne.com/downloads.html>.

Developing, Promoting, and Deploying BPEL Processes

The procedures for developing, promoting, and deploying solutions are important for all software development projects. For Oracle's SOA Suite, there are some specific considerations that should be kept in mind:

- **Concurrent development:** Oracle BPEL PM is not currently designed for multiple developers to work with the same process simultaneously. To prevent code branching and other issues, care should be taken to segregate development activities. This can most simply be done by assigning each developer a separate BPEL process. In cases where a single BPEL process is very complex and requires multiple resources to be assigned, consider compartmentalizing the process into multiple sub-processes that can be developed independently.
- **Source code control:** Oracle BPEL PM does not have built-in source code control, but can be utilized with external source code solutions such as Visual SourceSafe, SubVersion, CVS, and ClearCase. JDeveloper allows for BPEL projects to be saved as a cohesive unit called a project, which can then be checked in/out of these source code tools.
- **BPEL process promotion between servers:** A technology that is important to understand for deploying and migrating code is ANT. Put out by the Apache organization, ANT started as a Java build tool and has expanded to be able to not only compile the small bits of Java found in some BPEL processes, but also validate XML, install SQL, FTP files, and run server-side executables – all from a single script run on your laptop or on the AS10g server.

ANT is built into JDeveloper, and is also installed as part of the SOA Suite on the AS 10g server. It's an open source tool that can also be downloaded and installed on any machine. Once you have a working and tested process, ANT takes the .jar file which is the contents of your BPEL process (including all .xsd, .wsdl, etc files) and can move and register that file with any AS10g or SOA-enabled server. As part of any BPEL process, you'll see two files created that are expressly for use with ANT: build.xml (the main ANT deployment file) and build.properties, the file that you can customize to specify the settings for your particular server environment.

Books have been written about ANT and it's not possible to cover all the material here, however you can start your education right on the Apache.org site (<http://ant.apache.org/manual/index.html>) or for a more complete treatment of the topic in "ANT in Action", by Steve Loughran and Erik Hatcher.

Conclusion

Rapid interface development requires a solid foundation. With a good investment early in the process, your company will be well positioned to quickly develop new business processes, as well as support, and maintain them for years to come. When preparing for implementing a Service Oriented Architecture in your enterprise, it's important to have a clear vision of the path you'll follow, as well as the tools and expertise you'll require. Integrating smaller or larger systems with the E-Business Suite, whether it is file system data, data located in a database, or existing business logic exposed as a web service, requires a clear overall picture of your online as well as offline business processes. Understanding what you can do out-of-the-box, and the components you'll need to develop as part of a more robust SOA infrastructure your users will embrace, is essential to success.

Working with an Oracle Partner like Solbourne can help minimize your exposure initially, as well as maximize your investment in the long-term. With the expertise of over 260 Oracle Applications projects, combined with Solbourne's Enterprise Foundation Framework family of tools, we can help smooth your transition to the new world of SOA and set your company up with a foundation for integration that will serve it well immediately, and for years to come. Talk to a Solbourne representative about our 30 Day Quick Start project plan for getting your company going with its first SOA implementation quickly and cost effectively.



Rapid Interface Development for the E-Business Suite and PeopleSoft in a SOA Environment
April 2008

Author: Chad Coffman, Solbourne

[Contributing Author: Mike Brown, Solbourne](#)

Solbourne
4001 Discovery Drive, Suite 210
Boulder, CO 80303
U.S.A.

Inquiries:
Phone: +1-303-417-2800
Fax: +1-303-417-2820
www.solbourne.com

Copyright © 2008, Solbourne. All rights reserved.
This document is provided for informational purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. The Solbourne name, logo, related marks and all images on this document are owned by Solbourne and are registered and/or used in the U.S. All trademarks, service marks and trade names not owned by

Solbourne referenced in this document are the property of their respective owners.