

Integrating Oracle Application Express with Oracle Application Server 10g

*An Oracle White Paper
April 2005*

Integrating Oracle Application Express with Oracle Application Server 10g

Introduction	3
Integration Points.....	4
Oracle HTTP Server	4
Oracle Internet Directory	5
Oracle Single Sign-On.....	6
Oracle Application Express as an External Application	7
Any Oracle Application Express Application as a Partner Application	7
A Single Oracle Application Express Application as a Partner Application	7
Oracle Portal.....	8
Oracle Reports	9
Web Services (OC4J).....	10
Creating a Form and Report on a Web Service.....	11
Creating a Form on a Web Service.....	11
Creating a Process on a Web Service	12
Oracle MapViewer.....	13
Conclusion.....	15

Integrating Oracle Application Express with Oracle Application Server 10g

Oracle Application Express can take advantage of several Oracle Application Server 10g features to provide a more robust and secure application development environment

INTRODUCTION

Oracle Application Express – a feature of the Oracle Database 10g – is a powerful and easy to use web application development platform. With Oracle Application Express, you can quickly develop and deploy applications in a matter of hours, often without writing a single line of code. As these Oracle Application Express applications become more sophisticated and numerous, you may find yourself in need of a single sign-on infrastructure or want the ability to produce print-quality PDF reports, for example.

Fortunately, Oracle Application Express was designed with integration to Oracle Application Server 10g in mind. Oracle Application Express can take advantage of many of the features of Oracle Application Server 10g, often with little or no programming. This paper will discuss the various integration points between the two, how they can enhance your Oracle Application Express applications, and at the same time, save hours of development time. The following Oracle Application Server 10g features will be covered:

- Oracle HTTP Server
- Oracle Internet Directory
- Oracle Single Sign-On
- Oracle Portal
- Oracle Reports
- Web Services (OC4J)
- Oracle MapViewer

While this paper is meant to provide a technical overview of how to integrate Oracle Application Server 10g with Oracle Application Express, it is by no means intended to be a step-by-step guide. For detailed instructions on how to integrate the two products, please be sure to visit the following web site, as well as the links at the end of each section:

http://htmldb.oracle.com/app_server_integration

The Oracle HTTP Server from Oracle Application Server 10g or 9i can be easily configured to support Oracle Application Express, enabling you to consolidate servers for manageability and security purposes

INTEGRATION POINTS

Oracle HTTP Server

Oracle Application Express requires only the Oracle HTTP Server and `mod_plsql` on the server. For customers who do not own Oracle Application Server, these components are available on the Oracle Database 10g Companion CD. However, those customers who do own Oracle Application Server – either release 10g or 9i – can utilize the HTTP server in their existing infrastructure. Doing so eliminates the need to manage and secure another HTTP server in your environment.

Since there is no Java-based middle tier in Oracle Application Express, the steps to configure an existing Oracle Application Server HTTP server with it is quite simple. All that is needed is an additional entry in the Database Access Descriptor (DAD) configuration file.

In Oracle Application Server 10g, this file is called `dads.conf`, and is typically found in the `$ORACLE_HTTP_SERVER_HOME/Apache/modplsql/conf` directory. In Oracle Application Server 9i, this file is called `wdbsrv.app`, and is typically found in the same place.

An example of an entry for Oracle Application Express in the `dads.conf` file follows:

```
<Location /pls/htmldb>
SetHandler pls_handler
Order deny,allow
Allow from all
AllowOverride None
PlsqlDatabaseUsername          HTMLDB_PUBLIC_USER
PlsqlDatabasePassword          @BXUKpCh0pabnAcDAE+X2P9w=
PlsqlDatabaseConnectString    localhost:1521:sadler ServiceNameFormat
PlsqlDefaultPage               htmldb
PlsqlDocumentTablename        wwv_flow_file_objects$
PlsqlDocumentPath              docs
PlsqlDocumentProcedure        wwv_flow_file_manager.process_download
PlsqlAuthenticationMode        Basic
PlsqlNLSLanguage               AMERICAN_AMERICA.WE8MSWIN1252
</Location>
```

Another reason to use the HTTP server from Oracle Application Server is for fault-tolerant purposes. Adding a second HTTP server will provide added reliability in the event one fails. Keep in mind that there will not be a significant performance gain by adding extra HTTP servers, as all of Oracle Application Express's processing occurs in the database. If Oracle Application Express needs to scale significantly, consider Oracle Real Application Clusters (RAC). Oracle WebCache can be used with Oracle Application Express in order to cache static objects, such as images, CSS files and JavaScript libraries.

For more information on how to configure Oracle HTTP Server with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:http_server

On this page, there are links to the following documents:

- *Oracle HTTP Server Overview*
- *Install Application Express into multiple Oracle Instances Using a Single Oracle HTTP Server*
- *Install and Configure Oracle Application Express for use with Oracle Application Server 10g*
- *Install and Configure Oracle Application Express for use by a 9iR2 Oracle HTTP Server*

Oracle Internet Directory

Oracle Application Express Users and Groups can be managed in Oracle Internet Directory, which can be synchronized with other LDAP directory services

Organizations are constantly faced with the challenge of managing their employees' accounts and associated privileges. In almost every instance, organizations own and operate multiple systems which do not share a common user account repository. This makes it a challenge to manage an employee's accounts and privileges, as when they are hired or fired, they must be updated in several places.

Oracle Internet Directory – a component of Oracle Application Server 10g - is an LDAP-compliant directory service designed to solve this very issue. Oracle Internet Directory (OID) can be used to centrally manage all of your users account information for almost any application. Since it is built on the Oracle database, OID takes advantage of the security, scalability and high availability associated with Oracle. In addition to being able to integrate with Oracle Application Express, OID is integrated with Oracle Portal, Oracle Single Sign-On and Oracle Human Resources.

In addition to managing user accounts, OID can also be used to create and manage groups. These groups can then be referred to by a tool such as Oracle Application Express in order to manage what a user can and cannot do. For example, a group called *Admins* can be created in OID. This group can then be referenced by an Oracle Application Express authorization scheme by using the

HTMLDB_LDAP.IS_MEMBER API:

```
return htmldb_ldap.is_member
(:APP_USER,
null,
'cn=users, dc=somedomain, dc=com',
```

```
'oidserver.somedomain.com',  
'389',  
'SAMPLE_APP_ADMINS',  
'cn=Groups, dc=somedomain, dc=com');
```

This authorization scheme can then be associated with any Oracle Application Express component – such as pages, tabs or regions – in order to limit access only to users who are members of the OID group *Admins*. If a user is removed from the *Admins* group, their access to the items associated with the Oracle Application Express authorization scheme is also removed. If a user is removed from OID altogether, then their account will automatically be removed from all associated groups, which makes for a more secure environment.

In addition to the benefit of centralized Identity Management, Oracle Internet Directory provides both users and administrators flexible administration options. Users can manage their own personal information and reset their passwords as required or as needed. Administrators can set up rules – such as minimum password length – which users will in turn have to adhere to. Oracle Internet Directory can also be configured to synchronize with other directory services, such as Microsoft's Active Directory.

For more information on how to configure Oracle HTTP Server with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:oid

On this page, there are links to the following documents:

- *Oracle Internet Directory Overview*
- *Oracle Identity Management Overview*
- *How to use OID Groups to Manage Application Express Security*

Oracle Single Sign-On

Using Oracle Single Sign-On provides your users with the ability to authenticate once to the Oracle Login Server and then automatically be authenticated to any of their other applications – Oracle Application Express or otherwise

Oracle Single Sign-On (SSO) works closely with Oracle Internet Directory to provide users with the capability of signing on one time and gaining access to multiple applications. Oracle SSO makes use of the OID repository of user accounts and passwords.

Oracle Application Express can be configured three different ways when integrated with Oracle SSO:

1. Oracle Application Express as an External Application
2. Any Oracle Application Express Application as a Partner Application
3. A Single Oracle Application Express Application as a Partner Application

Oracle Application Express as an External Application

In the realm of Oracle Single Sign-On, an External Application is one where Oracle SSO stores account names and passwords in a password vault, and then uses those credentials to sign on to another application.

A good example of an External Application is an airline's web site. The credentials for an airline's web site are typically managed separately from internal account credentials, as they are two distinct accounts and organizations. Oracle SSO can be used to store account names and passwords for that airline's web site, and then authenticate users automatically to that site, based on the stored credentials. One drawback to this approach is that when a password is changed at the airline's web site, it will have to be updated manually in OID as well.

Oracle Application Express can be configured as an External Application, thus storing Oracle Application Express credentials in Oracle SSO. This configuration is ideal for an organization which does not have centralized directory services (Oracle OID or otherwise) or for cases when users want to seamlessly sign on to an Oracle Application Express application outside of the realm of their organizations. Since configuring an External Application is as simple as providing a username and password, no additional software is needed in order to configure Oracle Application Express to work as one.

Any Oracle Application Express Application as a Partner Application

A Partner Application can be defined as one which delegates all authentication services to the Oracle Login Server. Once a user authenticates to the Login Server, they will not have to provide their credentials to any Partner Application. Thus, a Partner Application uses the same account name as the Login Server itself.

To configure Oracle Application Express to work as a Partner Application, the Single Sign-On SDK must be installed in a common schema on the Oracle Database where Oracle Application Express is installed. After installing the SSO SDK, the Oracle Application Express application must be registered as a Partner Application with the Single Sign-On Server.

Once installed and configured, Oracle Single Sign-On can be used as the Authentication Scheme for any Oracle Application Express Application. This enables any Oracle Application Express application to use the same accounts that the Login Server uses, enabling a true single sign-on solution.

A Single Oracle Application Express Application as a Partner Application

If only a select few Oracle Application Express application need to use Oracle Single Sign-On as an authentication scheme, then the SSO SDK could optionally be installed into a specific schema instead of a common one. This will only allow Oracle Application Express applications which parse as that schema to utilize the Oracle Single Sign-On Server as an authentication scheme. All other properties and behaviors of a Partner Application apply.

For more information on how to configure Oracle Single Sign-On with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:sso

On this page, there are links to the following documents:

- *Oracle Single Sign-On Overview*
- *Configure an Application Express Application as a Partner Application in Oracle AS Single Sign-On*

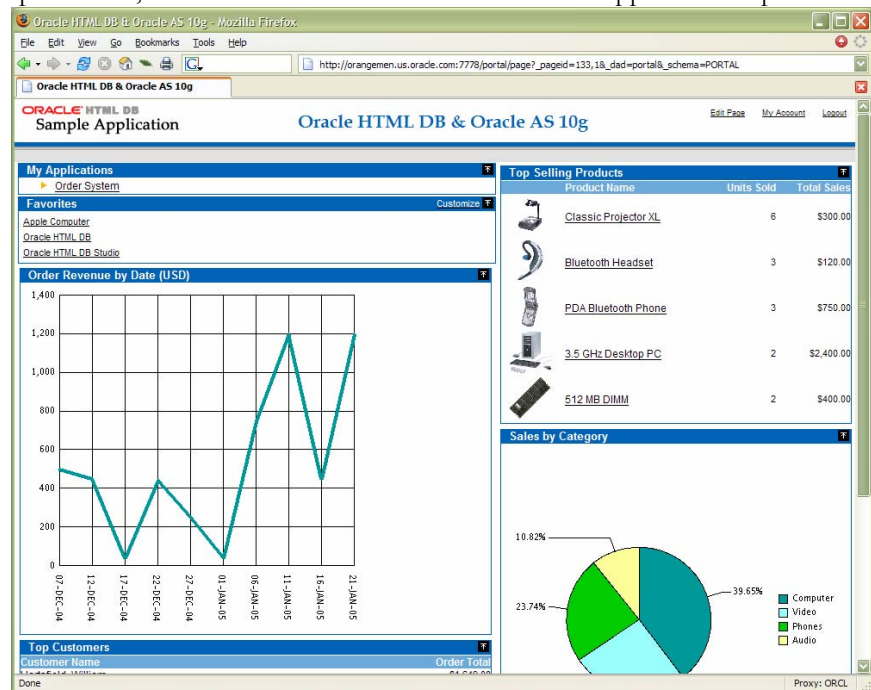
Oracle Portal

Oracle Portal's OmniPortlet can display data from any Oracle Application Express application as a chart, report or form

Oracle Portal is an ideal product to use when data from multiple sources needs to be consolidated into a single place. Its rich content management features also allows the management of documents and files. Portal can also be used to present a personalized summary of data from different applications in the form of reports and charts. These applications can be developed using JSP, PHP, or Oracle Application Express, for example. Since application development is one of Oracle Application Express's strengths, it makes sense to pair these two products into a single integrated solution.

Oracle Portal's OmniPortlet can be used to render an Oracle Application Express Report as a portlet on any Portal page. This portlet can take the form of a line, bar or pie chart, a report, or news items. All of this can be done via declarative Wizards on both the Oracle Portal and Oracle Application Express sides - no programming or configuration is required.

The OmniPortlet can accept a number of different types of data sources, such as a spreadsheet, Web Service or XML document. Oracle Application Express can



An Oracle Portal page with several OmniPortlets. Each OmniPortlet is displaying content from an Oracle Application Express application report, rendered as an XML document.

render any SQL query as an XML document, simply by selecting the appropriate report template. Thus, any SQL query in Oracle Application Express can be used as a source for an OmniPortlet.

OmniPortlets can also be made aware of page parameters for both data filtering and deep linking purposes. For instance, an OmniPortlet can filter rows of data based on the user who is viewing it. (*Note: For true data security, data should always be secured at the database level, and not with the OmniPortlet or any other application development tool*) An OmniPortlet can also be rendered with parameters as part of each data element, providing the developer with the ability to create OmniPortlets with drill-down capabilities.

It is important to note that when configuring an OmniPortlet using an Oracle Application Express Report as its source, the OmniPortlet should be set to not cache the results if a real-time view of the data is required. By default, the OmniPortlet will attempt to cache the results for 60 minutes.

For more information on how to configure Oracle Portal with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:portal

On this page, there are links to the following documents:

- *Oracle Portal Overview*
- *Publish an Application Express Component as a Portlet in Oracle Application Server Portal*

Oracle Reports

Oracle Reports is a nice complement to Oracle Application Express when high quality printed and/or PDF reports are a requirement.

Oracle Reports is a robust, mature tool for designing and deploying print-quality reports based on data from an Oracle database. Reports can be output in either PDF or HTML format, and can be integrated with Oracle Internet Directory for additional security purposes. Since Oracle Application Express Release 1.6 does not have any native support for PDF printing, Oracle Reports can be used with Oracle Application Express to include PDF printing support.

At the simplest level, Oracle Application Express can call an Oracle Report by referring to the URL of an Oracle Report. For example, the below URL will call the report `orders.rdf` and pass the value of the Oracle Application Express item `P1_ORDER_ID` to the Oracle Report's `p_order_id` parameter:

```
http://server.com/reports/rwservlet?module=orders.rdf
&userid=sspendol/oracle@syracuse&destype=cache&desformat=PDF
&p_order_id=&P1_ORDER_ID.
```

In the above example, the username and password of the Oracle Reports user is passed through the URL. Clearly this presents a security issue, and in most cases it is not the recommended method with which to call Oracle Reports from Oracle Application Express, or any other development environment.

In order to call a report without passing in the username and password in the URL, a Key Map file must be configured. By adding an entry to the Key Map file - typically found at `$ORACLE_HOME/reports/conf/cgicmd.dat` - the URL can reference that entry, instead of containing the username and password. For example, to use a Key Map for the previous URL, the entry in the `cgicmd.dat` file would look like this:

```
order: module=orders.rdf destype=cache desformat=PDF
      userid=sspendol/oracle@syracuse p_order_id=%1
```

In Oracle Application Express, item value substitution can be used in a URL which calls an Oracle Report in order to make it accept a parameter from a user. When Oracle Application Express parses the above URL, it will substitute a value for `P1_ORDER_ID` based on which row was selected, which button was pushed, etc.

In the Key Map file, each input parameter must be mapped to a position in the URL, denoted in the file itself by `%1` through `%9`. When calling the Oracle Report's URL, the values must be passed in as item tokens in the URL.

Thus, the new URL used to call the `orders.rdf` report via the Key Map would be:

```
http://server.com/reports/rwservlet?module=order&&P1_ORDER_ID.
```

For additional security, Oracle Reports can also be secured via the Oracle Single Sign-On server. For detailed information on how to do this, please refer to Chapter 10 of *Oracle Application Server Reports Services Publishing Reports to the Web 10g* (Part Number B10314-01).

For more information on how to configure Oracle Reports with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:reports

On this page, there are links to the following documents:

- *Oracle Reports Overview*
- *Integrate Oracle Reports with Oracle Application Express*

Web Services (OC4J)

Web services are fast becoming a critical part of any organization's IT architecture. Much of web services' popularity can be attributed to their open standards and extensibility. A web service can be authored in a number of different languages, such as Java, C++, or Visual Basic. Regardless of the source language, all web services share a common protocol set: XML & HTTP. This makes them extensible enough to be written once and used or consumed by a wide variety of applications.

Oracle Application Express can easily be used to create forms, reports and processes based on the results of a web service

Oracle provides products for both developing and deploying web services. Oracle JDeveloper 10g is a complete integrated development environment (IDE) for building J2EE compliant web services. Once a web service is built, Oracle JDeveloper can deploy that web service to any J2EE compliant application server, such as Oracle Application Server 10g using Oracle Application Server Containers for J2EE (OC4J). OC4J is the core J2EE runtime component of Oracle Application Server 10g, which offers easy deployment options as well as outstanding performance and scalability.

Oracle Application Express can only be used to read or “consume” a web service. It cannot be used to develop or deploy web services. In order for Oracle Application Express to be able to consume a web service, the web service must conform to a few specifications. First of all, it must provide a valid web services definition language (WSDL) document. A WSDL document describes how to access a web service and which actions it performs. The WSDL must provide Oracle Application Express with a URL to post a SOAP request to the web service, a URI which identifies the SOAP HTTP request, a list of operations, and any input/output parameters. Second, the web service must be an RPC type web service. Document Object Model (DOM) web services are currently not supported.

Both custom built and existing web services can be consumed by Oracle Application Express, so as long as they conform to the above specifications. When creating a web service reference, Oracle Application Express provides the ability to browse any UDDI repository in order to search for and use an existing web service. Web service references can also use a proxy server, if necessary.

Once a web service is consumed by Oracle Application Express, it becomes available as a Shared Component in the Application Builder. Oracle Application Express provides several wizards in order to facilitate the creation of forms, reports and processes based on a web service reference.

Creating a Form and Report on a Web Service

This wizard will create an input form and report based on a web service reference. The input parameters will be automatically mapped to the input parameters of the web service itself, while the output of the web service will be displayed as a report. This wizard is ideal for when non-scalar or multiple results are expected from the web service. An example of this would be a web service that returns a list of books based on the search criteria.

Creating a Form on a Web Service

This wizard will create a form based on a web service reference. The input and output parameters will be automatically mapped to the parameters of the web service itself. Use this wizard when you expect only a scalar or single result to be returned from a web service. An example of this would be a web service that

returns the current temperature based on the ZIP code passed in as the input parameter.

Creating a Process on a Web Service

This wizard will create an Oracle Application Express process, which will invoke the web service when executed. Input and output parameters can be mapped to existing Oracle Application Express items. Use this wizard when you want to use a web service as part of your page rendering or processing.

A good example of a process based on a web service can be seen in the Sample Application 1.6, which is installed by default in any freshly created Oracle Application Express workspace. The Sample Application 1.6 makes use of the *City & State Lookup* web service, which is hosted by Oracle. The *City & State Lookup* web service only has one operation: *getCityState*. When passed a ZIP code, this method will return the corresponding city & state.

The screenshot shows the configuration for a process named "Get ZIP Code from Web Service".

- Process:**
 - Page: 7 Add/Modify Customers
 - Name: Get ZIP Code from Web Service
 - Type: Web Service
- Process Firing Point:**
 - Sequence: 5
 - Process Point: On Submit - After Computations and Validations
 - Run Process: Once Per Page Visit (default)
- Web Service:**
 - Reference: City & State Lookup
 - Operation: getCityState
- Web Service Input Parameters:**

Parameter Name	Source	Value
zipCode	Item	P7_CUST_POSTAL_CODE
- Web Service Output Parameters:**

Store Result In: Items Collection

Parameter Name	Destination	Value
city	Items	P7_CUST_CITY
state	Items	P7_CUST_STATE
lookupResponse	Items	

An Oracle Application Express process based on a web service. Note how both the input and output parameters of the web service are mapped to Oracle Application Express items.

Using this web service, when a new customer is created in the Sample Application 1.6, it does not ask for the city & state, but rather just the ZIP code. Before it calls the process to commit the data to the customers table, Oracle Application Express invokes a page process which is mapped to the *City & State Lookup* web service. It passes the *getCityState* operation the ZIP code that the user entered, and then returns the corresponding city & state to corresponding Oracle Application Express items on the page. Once the web service executes, Oracle Application

Express then inserts the row, complete with the city and state values returned by the *getCityState* operation, into the customers table.

For more information on how to use web services with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:web_services

On this page, there are links to the following documents:

- *Oracle Application Server Containers for J2EE Overview*
- *Publish a JAX-RPC Java Web Service*
- *Amazon.com Store Demo for Oracle Application Express 1.6*

Oracle MapViewer

MapViewer is a J2EE service used to render maps based on spatial data from the Oracle Database. Combined with the database features Oracle Spatial or Oracle Location Services, MapViewer provides the Oracle Application Express developer with the tools to render, view and navigate graphical maps.

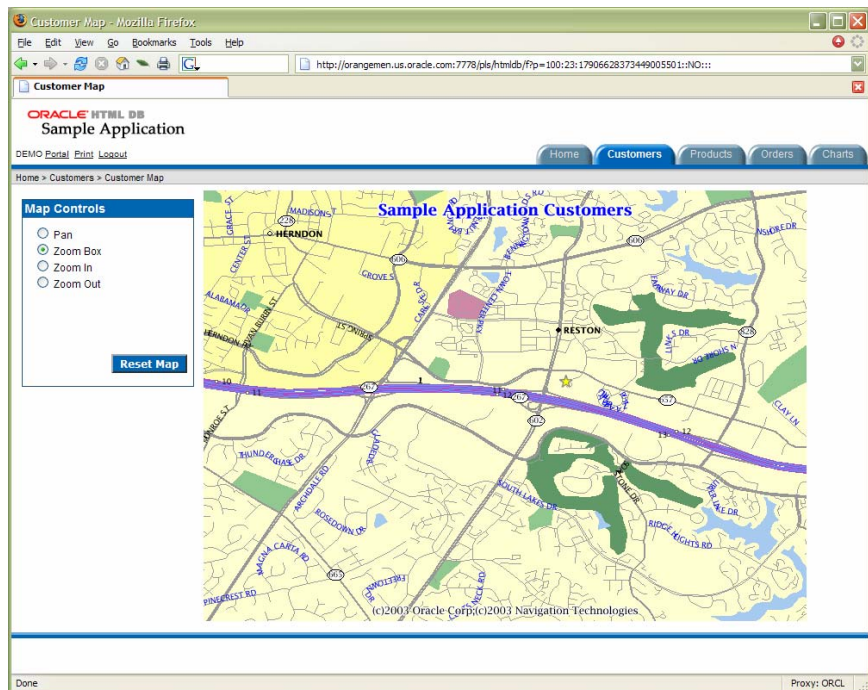
As a J2EE service, MapViewer requires and runs in an OC4J container, which is a component of Oracle Application Server 10g described in the previous section of this paper. It also requires that data be spatially indexed. Despite being a J2EE application, little to no Java knowledge is required to integrate MapViewer with Oracle Application Express, as all of the integration can be done via PL/SQL APIs.

When used together, Oracle MapViewer and Oracle Application Express can display any map which MapViewer can render. Oracle Application Express can also utilize MapViewer's functions, such as the ability pan in any direction, zoom in and out, and re-center based on a point. Oracle MapViewer themes and layers are also available when using it with Oracle Application Express.

Oracle MapViewer requires either Oracle Locator or Oracle Spatial on the database. Oracle Locator is a feature of both the Standard and Enterprise Edition, while Oracle Spatial is only supported in the Enterprise Edition. Oracle Locator provides core location functionality sufficient for most customer applications, whereas Oracle Spatial augments Oracle Locator's capabilities with more advanced features.

Currently, integrating Oracle Application Express and MapViewer is a manual process. On the Oracle Application Express side, calls to MapViewer must be assembled via PL/SQL and then passed on to MapViewer. MapViewer will then generate a map based on the parameters passed to it, and hand back a URL of that map to Oracle Application Express. Oracle Application Express will then render a page with the map displayed as an image or SVG.

Visualize your data graphically on a map
with Oracle Application Express & Oracle
MapViewer



An Oracle Application Express application integrated with Oracle MapViewer. Users can pan, zoom in and out and draw a zoom box directly in a web browser, without and Java on the client.

In order to greatly simplify the process of integration, Oracle will be providing PL/SQL APIs for MapViewer and Oracle Application Express in the near future. These APIs, when combined with the appropriate Oracle Application Express items, will allow developers to integrate MapViewer with Oracle Application Express without writing any significant code.

For more information on integrating Oracle MapViewer with Oracle Application Express, please refer to the following documents:

For more information on integrating Oracle MapViewer with Oracle Application Express, please visit the following URL:

http://htmldb.oracle.com/pls/otn/f?p=app_server_integration:mapviewer

On this page, there are links to the following documents:

- *Oracle MapViewer Overview*
- *NavTeq Sample Map Data*
- *OTN: MapViewer Quick Start*
- *Integrating Oracle MapViewer with Oracle Application Express (Coming Soon)*

CONCLUSION

Combining Oracle Application Express with Oracle Application Server 10g gives developers the best of both worlds: The ability to quickly build secure, scalable web applications while at the same time, taking advantage of enterprise-class integration, identity management and reporting services. Oracle Application Server 10g functionality, such as PDF Reporting, MapViewer and web services can easily be integrated into any Oracle Application Express application, often with little to no code. In addition to adding functionality to your Oracle Application Express applications, Oracle Application Server 10g increases the overall manageability of them by seamlessly integrating them with a single user repository and authentication services.

Oracle Application Express Free Trial
Service
<http://htmldb.oracle.com>

Oracle Application Express Home
<http://htmldb.oracle.com/otn>

Oracle Application Express How To
Documents
<http://htmldb.oracle.com/howtos>

Oracle Application Express Studio
<http://htmldb.oracle.com/studio>

Oracle Application Express Forum on OTN
<http://htmldb.oracle.com/forums>

Oracle Application Express References
<http://htmldb.oracle.com/references>



Integrating Oracle Application Express with Oracle Application Server 10g

April 2005

Author: Scott Spendolini

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

Copyright © 2005, Oracle. All rights reserved.

This document is provided for information 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, whether expressed orally or implied in law, including implied warranties and 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.

Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.