Oracle Application Server 10*g* (9.0.4) - Forms Single Sign-On

An Oracle Whitepaper
December 2003

ORACLE

Oracle Application Server 10*g* (9.0.4) - Forms Single Sign-On

Oracle Application Server 10*g* (9.0.4) – Forms Single Sign-On

INTRODUCTION TO SINGLE SIGN-ON

Current Web architectures require clients to access many applications from a single entry point: the Web Browser. Most applications require the user to login which, besides its annoying usability aspect, is a potential security risk as the user soon runs out of suitable passwords. Usually, when users forget passwords for an application, they try all the passwords that they can remember, therefore creating opportunities for crackers that are sniffing the network.

Additionally, secure password policies don't allow any sensible name to be chosen as a password and demand for using alphabetical-numerical characters of both cases mixed with special characters. How many of those passwords can you remember without writing them down?

Due to these problems, single sign-on becomes an important security tool for applications that are deployed on the Web, making authentication the first line of defense.

Single sign-on is the ability of an application to authenticate users by means of a shared authentication token or authentication authority. In this scenario, a user authenticated for one application is automatically authenticated for all other applications within the same authentication domain.

Single sign-on solves the security problem by having the end-user remember a single username and password combination. However, Single Sign-On still keeps different username and password combinations for the actual login for each application.

Did you know? All Forms applications, built with previous versions of Oracle Forms, can take advantage of single sign-on after upgrading to OracleAS Forms? No additional coding is required!

ORACLE APPLICATION SERVER 10*g* - FORMS SINGLE SIGN-ON

Throughout this Whitepaper the Oracle Application Server 10*g* (9.0.4) is also referred to as "OracleAS". Similar Oracle Application Server 10*g* Forms Services are referred to as "OracleAS Forms Services" and "Forms Services".

Oracle Application Server 10*g* (9.0.4) in its Enterprise Edition contains Oracle Single Sign-On that is used by all Oracle tools such as Oracle Forms, Oracle Reports, Oracle Portal and Oracle Discoverer to authenticate users on the Web. The Oracle Single Sign-On Server can also be enabled for applications that are not Oracle products such as custom built J2EE applications.

Oracle Forms[1] applications seamlessly integrate into a company's single sign-on architecture by leveraging OracleAS Single Sign-On for authentication. Oracle Application Server 10*g* Forms Services provides out-of-the box support for single sign-on for as many Forms applications as run by the server instance with no additional coding required in the Forms application.

Using OracleAS Forms Services, Oracle Forms applications are single sign-on enabled simply by setting parameters in a configuration file on the middle tier server.

SINGLE SIGN-ON COMPONENTS USED BY FORMS

Oracle Single Sign-On takes advantage of these components in Oracle Application Server 10*g* (9.0.4) when your Forms applications run in single sign-on mode

- *Oracle Single Sign-On Server* – an authentication Service in OracleAS that uses Oracle Internet Directory as a username and password store.

- *Mod_osso* – The HTTP module mod_osso simplifies the authentication process by serving as the sole partner application to the Single Sign-On server, rendering authentication transparent for OracleAS applications. OracleAS Forms Services and OracleAS Reports Services use mod_osso to register as a partner application to the Oracle Single Sign-On Server.

- *Oracle Internet Directory (OID)* – An LDAP v3 compliant directory server that stores information about single sign-on users and their login information. An LDAP server is a special database that is optimized for read access.

- *Forms Servlet* – The OracleAS Forms Services component that accepts the initial user request to start a Forms application. The Forms Servlet detects if an application requires single sign-on, directs the request to the Single Sign-On Server and accesses the Oracle Internet Directory to obtain the

[1] Single sign-on is not enabled for the Forms and Reports only installation option in OracleAS. You must install the full Enterprise Edition of Oracle10*g* AS to take advantage of single sign-on with Forms.

database connect information.

The OracleAS version is 10*g*, but all Forms components still use "90" in their namings and directory settings.

- *Formsweb*.cfg – The Forms configuration file that contains the parameters to enable a Forms application for single sign-on. The formsweb.cfg file is located in the forms90/ server directory of an Oracle Application Server 10*g* installation.

ORACLE FORMS SINGLE SIGN-ON WORKFLOW

Before the emergence of single sign-on, a user connected to each Oracle Forms application individually by providing the database connect information in a Forms logon screen. This connection information is used by the Forms application to connect to the database.

When single sign-on is enabled for an application, the application still connects to the database using SQL*Net logon. However, instead of the user entering the application's database connect string, it is retrieved from the Oracle Internet Directory based on the authenticated user's single sign-on username and the application that is being accessed. The user only needs to login once using an HTML Form sent to the browser. The information the user inserts in this login Form will be used to identify him for any other application in that secured domain.

Here is how an OracleAS Forms Services application uses the single sign on process:

- OracleAS Forms Services applications are accessed by a URL similar to "http(s):/ / <hostname>:<port>/ forms90/ f90servlet?config=<applicatio n>&…".
- The target OracleAS Forms application is specified by the value of the *config* parameter, referencing a "Named Configuration" in the *formsweb.cfg* file (1).
- The Forms Servlet reads the single sign-on parameters specified for the requested application and directs the user to the mod_osso module if single sign-on is enabled (2).
- The mod_osso module checks if the requested application URL was authenticated previously, and if not, passes the request to the Oracle Single Sign-On Server(3).

  The Oracle Single Sign-On Server looks for an encrypted authentication cookie that is attached to the browser session the first time the user was successfully authenticated. If the authentication cookie cannot be found, the Single Sign On Server presents an HTML logon screen for the user to provide the single sign-on credentials. These credentials are checked against Oracle Internet Directory, which is the password store used by the Single Sign-On Server (4).

- After the user is successfully authenticated, the original application request is issued back to the Forms Servlet. The Forms Servlet uses the value provided in the *config* URL parameter and the user's single sign-on username to build a unique key to retrieve the database connect information from Oracle Internet Directory. The user's application database connect information needs to be stored in OID prior to using it (5).
- The Forms Servlet generates the start HTML file for the Forms Applet. The user's database connect information is kept in the servlet session on the middle tier server and is not downloaded to the client at any time.
- The Forms Applet accesses the Forms Listener Servlet for all subsequent HTTP or HTTPS communication (7).

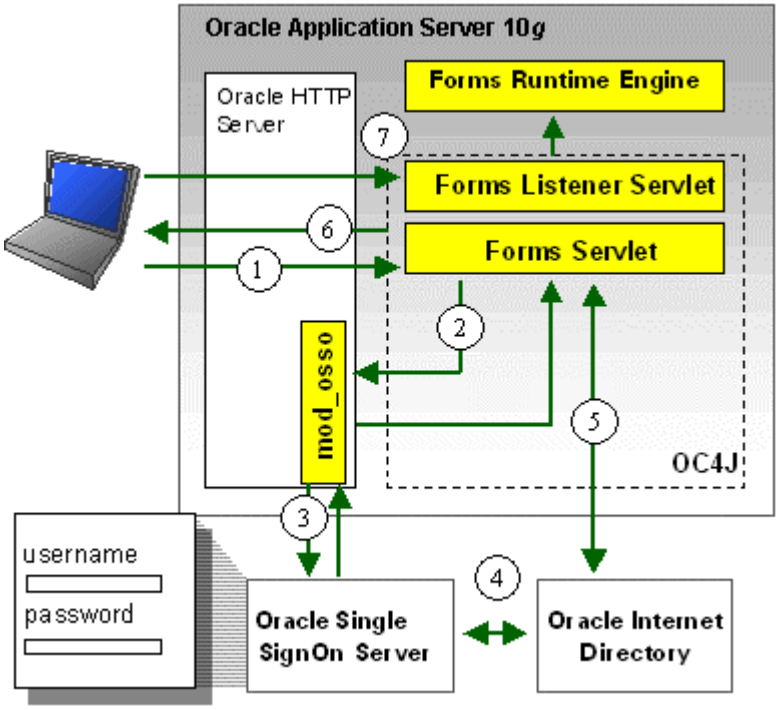

Figure 1- Oracle10*g* AS Forms Services single sign-on flow

The user's single sign-on username, the user Distinguished Name (DN) and the subscriberDN are passed to the Forms runtime engine and can be accessed by the Forms application.

The single sign-on username is unique and can be used to build a context for a virtual private database allowing restricted data access for each user even if all users use the same database connect string.

HOW TO CONFIGURE SINGLE SIGN-ON IN ORACLE FORMS

Single sign-on in Forms is configured by storing the application's database connect information in Oracle Internet Directory and by single sign-on parameters settings in the Oracle10*g* AS Forms Services *formsweb.cfg* configuration file.

Configuring Oracle Internet Directory for Forms single sign-on

The Oracle Internet Directory and Oracle Single Sign-On Server are installed and configured for you when installing Oracle Application Server 10*g* with infrastructure.

In general, single sign-on user accounts are created in Oracle Internet Directoy using either the *Delegative Administration Service (DAS)*, a Web based self service application, the *Oracle Directory Manager*, a Java based administration console or one of the provided command line utilities and programming APIs.

Resource Access Descriptors (RAD)

OracleAS Forms Services applications on the Web use SQL*Net to connect an application to the database. This also is true for Forms applications that run in single sign-on mode. To connect a single sign-on Forms application to the database, the user's database connect information is read from a named entry in Oracle Internet Directory by the Forms Servlet. These named entries, called *Resource Access Descriptor (RAD),* are uniquely identified by the combination of the user's single sign-on name and the name of the requested application. The name of an application is the name specified for the *Named Configuration* section in the *formsweb.cfg* file. The application name also is specified as the value of the Forms *config* URL parameter passed with the application request. RAD entries can easily be created using the Delegative Administration Interface or by one of the OID scripting APIs.

Default Resource Access Descriptors

Instead of creating RAD entries individually for each user and application, default Resource Access Descriptors can be created in DAS. Default RAD entries are resources that are shared among all users in OID. This means that all users connecting to a single sign-on protected Forms application use the same physical database connect information. Default Resource Access Descriptors simplify administration and work for Oracle Forms and Oracle Reports. They are recommended whenever an application has its own integrated application account management, such as where each user is identified within the application itself. The Forms single sign-on architecture and configuration is the same for individual RAD creation and default RAD creation.

Configuring single sign-on parameters in the formsweb.cfg file

The OracleAS Forms Services configuration file *formsweb.cfg* in the forms90/ server directory is used to configure a Forms application that is deployed on the Web. It

also contains specific parameters that define the single sign-on behavior for a single application or the whole Forms Services instance. Here is a list of the single sign-on relevant parameters:

- *SsoMode* –set to true or false to determine if the Forms Services or a specific application requires single sign-on authentication. The default setting for this parameter is false, not requiring single sign-on.

- *SsoDynamicResourceCreate* – set to true or false to determine if a missing RAD resource in Oracle Internet Directory shall be created by Forms on behalf of the application user. To provide the database access information for this application, an HTML form will be presented to the user. The RAD resource created contains the name specified as the value of the config parameter in the request URL. The default value of this parameter is "true".
- *SsoCancelURL* – If setting SsoDynamicResourceCreate to true, the user is presented a HTML form to create a new RAD entry whenever this entry is missing for the application he tries to run. If the user presses the cancel button of this form then the SsoCancelURL parameter can be used to direct the user to a custom help page or any other URL.
- *SsoErrorURL* – If you don't want Forms to dynamically create missing RAD entries in OID then set the SsoDynamicResource parameter to false and use this parameter to specify a URL that a user gets directed to. This parameter is useful if you want to handle the situation where a RAD entry cannot be found based on the user misspelling the value of the config parameter.

Note that configuring an application as single sign-on enabled with the value of the *ssoDynamicResourceCreate* parameter set to false, while not specifying a value for the ssoErrorURL, will cause Oracle Forms to show an error message if no RAD resource exist for the authenticated user and this application.

The single sign-on parameters in the *formsweb.cfg* file can be set in the "User Parameter" section to make them the default behavior for all Forms applications run by the server, and and in a "Named Configuration", making the settings valid for a particular application only. A single sign-on definition overrides the same definition set in the User Parameter section.

Example *formsweb.cfg* file configurations

```
# sso enabled application with dynamic resource creation if RAD
does not exist
```

```
[myApp]

form=myApp.fmx

# enable single sign on for this application

ssoMode=true

# create RAD source if not existing

SsoDynamicResourceCreate=true

# Handle user pressing cancel button in DAS HTML form

ssoCancelURL=http://my_company.com/help/what_is_wrong.html

lookAndFeel=oracle

…

# public accessible application

[myPublicApp]

form=myApp.fmx

lookAndFeel=oracle

IE=Jinit

…
```

ACCESSING SINGLE SIGN-ON INFORMATION FROM FORMS

Optionally, if you need to work with single sign-on authentication information in a Forms application, the GET_APPLICATION_PROPERTY() Built-in can be used to retrieve the follwing sso login information: sso userid, the user distinguished name (dn) and the subscriber distinguished name (subscriber dn)

```
authenticated_username := get_application_property('sso_userid') ;
userDistinguishedName := get_application_property('sso_usrdn') ;
subscriberName      := get_application_property('sso_subdn') ;
```

HANDLING DATABASE PASSWORD EXPIRY

OracleAS Forms Services handles the renewing of expired database user passwords. If the database password has expired and the Oracle Forms application, running in single sign-on mode, is used to renew it, then the new password entered by the user is used to update the Resource Access Descriptor in OID for this application. This ensures that single sign-on with Forms continues working even when a database password was changed.

SUMMARY

Single sign-on is a highly recommended feature for any Web applications as too many password compromise security. OracleAS Forms Services provides out-of-the-box and easy to configure support for single sign-on. Since all the work is done within the Oracle10*g* AS Forms Services deployment framework, there is no requirement for modifications added to existing Forms applications simplifying the

Oracle Application Server 10*g* (9.0.4) - FormsSingle Sign-On
September 2003
Author: Frank Nimphius
Contributing Authors: Robin Zimmermann, Regis Louis, Shay Shmeltzer

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
www.oracle.com

Oracle Corporation provides the software
that powers the internet.