

APPLICATION SECURITY – WHAT ARE MY OPTIONS?

Susan Behn, Solution Beacon, LLC.

Introduction

Increasing regulations, such as Sarbanes-Oxley, HIPAA and others, have made *key employees* and *managers* liable for protecting information. This paper is not just for DBAs! Security is a hot topic and options to implement security are numerous – some technical and some functional. It is important for managers and superusers as well as system administrators, DBAs and developers to understand what is available within the applications and which security options can meet the various requirements in your organization.

Security must be tight enough to protect data from both internal and external threats and security must be granular enough to protect privacy by limiting access to specific pieces of data for specific groups. According to studies, 70-80% of security breaches are internal and most result in financial loss. This document covers many components of application security-both technical and functional, primarily focusing on security that is included within the applications. This includes standard security features via profile options, menu driven security, function security as well as personalized security options via Forms6i personalization and OA Framework personalization.

Standard Application Security

Standard application security is based on three specific components – *authentication*, *authorization* and *audit trail*. Authentication verifies the user is legitimate primarily via password control. Authorization grants access to data and functions that a user needs to do their job. Audit trail reports activities and updates to the system and data after it occurs and is the last line of defense to verify a user does not misuse access privileges.

The focus of many businesses today is authentication. Great lengths are taken to verify a user is legitimate. SOX compliance and other regulations related to separation of duties and control, tend to focus on the authorization component with some focus on audit trail. However, some of these controls and compliance are dependent on asking the right questions and reporting the right information related to security.

Key questions are, “Do you know who has authority to make changes and more importantly, do you know who is allowed to grant authority?” The answer to this question is probably “yes.” Another key question is, “Can you track who made changes and who granted authority to other users?” If the answer is “no”, you have a serious security problem.

Authentication – Who are you?

Password Profile Options

Oracle Applications use profile options to control standards and policies regarding authentication.

The following chart provides recommendations for profile options related to passwords.

Profile	Default	Recommendation
Signon Password Failure Limit	None	3 (attempts)
Signon Password Hard to Guess	No	Yes
Signon Password Length	5	8 (characters)
Signon Password No Reuse	None	180 (days)
Signon Password Custom	None	Create custom java class if needed
Password Case Option	Insensitive	Sensitive

Timeout profile options

The following profile options control various timeouts in the E-Business Suite.

Profile	Purpose	Default	Recommendation
ICX:Session Timeout	Time of inactivity in a form session	None	30 (minutes)
ICX: Limit Time	Maximum connection time regardless of activity	4 (hours)	4 (hours)
ICX: Limit Connect	Max HTML page requests that can be made in a session	1000	2000
JTF_INACTIVE_SESSION_TIMEOUT	CRM session timeout for inactivity	None	30 (minutes)

There are many other timeout settings in configuration files at the database level. This includes:

- ICX Timeout Profiles
- CRM Application Profiles
- Jserv (Java) Timeout Settings
- Apache HTTP Timeout Settings
- Forms 60 Environment Timeout Settings
- Oracle Single Sign On Timeout Settings

Recommendations for database configuration files can be found in the resources identified at the end of this white paper.

Other Authentication Best Practices

In addition to the password profile option settings, consider the following best practices:

- Never use generic passwords. During the time a new employee does not change the initial password, the application is open to abuse.
- Set passwords to expire periodically – preferably 60 days.
- Regularly change system passwords and shared account passwords such as the SYSADMIN password.
- Minimize passwords in files. Run programs through the concurrent manager whenever possible to avoid the requirement to pass user names and passwords. If you must pass a user name or password, use an encrypted password file.

Authorization – what can you do?

Oracle Applications has traditionally used a menu driven security model to limit the functions available to a user. A user is assigned to one or more responsibilities which include a menu of functions and an assigned request group for concurrent programs. In 11.5.10, Oracle introduced the User Management module (UMX) which is a Role Based Access Control (RBAC) model. The goal of the RBAC model is to combine data security with function security and base access on a job rather than individual identity. This is an additional layer to compliment the existing menu driven security model.

Menu Driven Security

Menu driven security utilizes menus, responsibilities and request groups to group functions that are assigned to users. Using this model, many different responsibilities are created to limit functions by role and users must be assigned multiple responsibilities to do their job. This model is not hierarchical and requires significant maintenance as users are hired or change jobs.

MENUS

Menus include a group of submenus and functions for an application. Menus are assigned to one or more responsibilities.

REQUEST GROUPS

Request Groups define requests, request sets and concurrent programs that can be selected from the Submit Requests function. A request group is associated with a specific application simply to uniquely define the request group. Requests owned by other applications can be included in the request group. The request group is assigned to one or more responsibilities.

RESPONSIBILITIES

A responsibility loosely represents an application and is a type of role. A role represents a job function that may cross responsibilities requiring the assignment of multiple responsibilities to a user to enable them to perform all job functions required for their role in the organization. Each user is assigned at least one responsibility which defines the functions and processes available to the user. Responsibilities include a menu which may be limited by exclusion of specific functions on that menu. Access may be further limited at the responsibility level for self service applications by excluded items and securing attributes. Data security is provided at the responsibility level through security groups and security profiles which are discussed later.

FUNCTION SECURITY

Function security limits access or permissions based on menus and specific functions available on the menu. A function can be an entire form or functionality within a form, such as the ability to force approval for an invoice which is a check box on the approvals block within the invoice entry form. Menu exclusions provide the ability to exclude any function on a menu.

EXCLUDED ITEMS

Excluded items provide column level security for self service applications by allowing the exclusion of specific attributes on a self service page.

SECURING ATTRIBUTES

Securing attributes provide row level security by limiting the data that can be updated. For example, iExpenses should only allow a user to enter or update their own expense report. This is accomplished by setting TO_PERSON_ID and ICX_HR_PERSON_ID to the employee person id.

User management

User Management provides role based access control based on job functions rather than user based access control which is based on the user. Functional security is combined with data security to create roles. Roles, which are hierarchical, are granted access to specific functions. For example, a payables entry clerk for a specific organization would have access to a group of data and functionality. Any user with this job would be assigned this role. This ensures consistency for all "Payable Entry Clerks" and minimizes the risk of inappropriate access by a user. A user "inherits" the access for that role and well as roles lower in the hierarchy. The Payables Manager would be assigned the role for the manager and would automatically inherit the Payable Entry Clerk role as well as any other roles lower in the hierarchy.

With RBAC, roles can be defined to consolidate responsibilities. A role is assigned to a user as a one-time setup then roles can be changed as needed and applied to all users of that role. For example:

Approvals Management Administrator includes roles:

- Approvals Management Business Analyst – can modify grants
- Approvals Management System Administrator – can create or modify grants

In the menu driven security model prior to 11.5.10, the Approvals Management Administrator would be granted at least two responsibilities. With user management, the Approvals Management Administrator would be granted the role for that position which would include the two roles for the Approvals Management Business Analyst and the Approvals Management System Administrator. If the business decides to move a job function from the System Administrator to the Business Analyst, only the role must be changed rather than multiple users.

Similarly, data security is granted to roles using grants. Permission to act upon specific rows of data can be created for a single user or a group of users as defined by an organization or responsibility. Data access is granted to a single instance (or row) as in the example of expense reports or HR updates, or a set of rows limited by any attribute of that row. For example, you can limit access to Suppliers assigned to a specific buyer.

HRMS Security options

Standard HRMS security allows users to be assigned to security profiles which are assigned to business groups. A security group is assigned to a responsibility via profile options. This provides the row level security to limit access to a group of data. Security profiles are difficult to change and users must change responsibilities to view different groups of data

Forms Configurator and Taskflows

Forms Configurator is a new tool in Release 11i that can be used to modify base/delivered Forms in order to provide additional security. The configurator available in HRMS is similar to what can be achieved through Forms6i Personalization and OA Framework Personalization. For example a custom Template (Form) can be created by using Forms Configurator in order to exclude salary-related columns.

Taskflow is a mechanism used to link multiple Windows. A taskflow combines menus and responsibilities to define a job process. The first step in establishing a taskflow is to define its nodes. Taskflow nodes can be defined using Custom forms that are created using the Custom Forms Screen or Forms Configurator.

Security groups/Security profiles

A Security Profile is attached to a Responsibility and determines the degree of information a user can view/access. Common security profile options within HRMS include:

- Department Based Security – an organization/department hierarchy can be defined to enable the used of a security profile to limit access to a specific organization/department.
- Position Based Security – a position hierarchy can be defined to enable the used of a security profile to limit access based on an employee’s position within an organization.
- Employee/Supervisor Based Security – an employee list is generated with subordinate data at the time an employee logs in including all employees below the supervisor level. This option may create an undesirable impact on performance depending on the number of levels in the hierarchy.

Audit trail – What did you do?

The most basic requirement for achieving the third security component, audit trail, is to never allow shared accounts. A shared account is any account used by more than one person. If an account is shared, there is no audit trail available for changes made using the shared account. The most common and most abused shared account used for this purpose is the SYSADMIN user account. Only a very limited number of people should have access to the SYSADMIN user password. This is typically a senior DBA and their manager. In later releases, Oracle is making it increasingly easier to limit the number of people that require access to the SYSADMIN user. For example, the “Grant Access” functionality within Workflow allows the SYSADMIN user to grant access to the notifications owned by SYSADMIN to any user or role. This eliminates the need to require anyone to log in as SYSADMIN to monitor Workflow errors.

The number of logins for a user name can also be limited to protect against the use of shared accounts. See MetaLink Note: *270454.1, How to Limit the Number of Form User Open And The Number Of Session User Logon* for more information.

Oracle Applications also includes functionality to audit any database object at the row and column level for inserts, updates and deletions. When auditing is enabled, triggers are created to populate a shadow table with complete history. These shadow tables are identified by the name of the base table appended with *_A*. While auditing can impact performance and results are difficult to report due to the large amount of data and complexity, a simple approach to auditing only critical objects can be a good strategy. For example, consider only auditing non-transactional data objects that impact security. The tables listed below tend to be relatively static, but critical in identifying security breaches quickly prior to any negative impact resulting in financial loss.

FND_AUDIT_COLUMNS	FND_AUDIT_TABLES
FND_CONCURRENT_PROGRAMS	FND_DATA_GROUP_UNITS
FND_ENABLED_PLSQL	FND_FLEX_VALIDATION
FND_FORM	FND_FORM_FUNCTIONS
FND_GRANTS	FND_MENUS
FND_MENU_ENTRIES	FND_ORACLE_USERID

FND_PROFILE_OPTION_VALUES

FND_REQUEST_GROUP_UNITS

FND_RESP_FUNCTIONS

FND_USER_RESP_GROUPS

To enable auditing, complete the following steps:

Set the profile option, *AuditTrail:Activate*, to *Yes*

Navigate to Security → AuditTrail → Install

Check the *Audit Enabled* checkbox for schemas where auditing is to be enabled

Navigate to Security → AuditTrail → Tables to identify tables to be audited

Run the *AuditTrail Update Tables* concurrent request to activate auditing

Personalized and custom Security Options

The security strategies described to this point, cover security from the site level to a form or function level with the exception of additional options for self service applications and Human Resources. What about the rest of the applications? How can portions of a form or individual items be secured for core application forms written in Forms6i? What other options are available to secure self service forms? For example, how do I hide tax id numbers, bank account numbers and other secure information at various levels?

Oracle Applications include three tools to enable security at a more granular level – down to the individual item on a form.

1. Custom Library – CUSTOM.pll is a PL/SQL library located in \$AU_TOP/resource on the forms server. Using this library, a developer can modify the application 6i form to secure fields, blocks, tabs and other items by hiding, masking, preventing update or insert or perform other restrictions and validations. This library executes as part of the application code and does not impact support. Additionally, this code will survive patching and upgrades through Release11i.
2. Forms6i Personalization – This feature was introduced in 11.5.10 to enable an application user to declaratively alter the behavior of a 6i form. This is simply a table driven implementation of the custom library which is far easier to implement and modify than the custom library.
3. OA Framework Personalization – OA Framework Personalization is a new tool that enables a functional user to alter the behavior of a self service form. Use it to secure fields or pages, hide fields or pages, move them around, change certain attributes, and much more.

Custom library

Programmatic personalization is accomplished by adding code to the CUSTOM.pll package body using Forms Builder 6i specifying the event for which the code should execute. Typical modifications that impact security include hiding fields, making fields required, restricting insert or update, restricting values or performing additional validations.

After PL/SQL coding is complete, CUSTOM.pll must be moved to the forms server and compiled.

Figure C.pll shows some examples of CUSTOM.pll coding for security related personalizations.

```

PACKAGE BODY XXXXXAPXVDMVX
IS
  PROCEDURE event (event_name VARCHAR2) IS
BEGIN
  IF event_name = 'WHEN-NEW-FORM-INSTANCE' THEN

  \* Hide the tax payer id*\

  APP_ITEM_PROPERTY2.SET_PROPERTY('VNDR.NUM_1099_MIR', DISPLAYED, PROPERTY_OFF)
  END IF;
END event;
END XXXXXAPXVDMVX;

```

C.pll Figure 2 Package Body

Forms6i personalization

Forms personalizations declaratively alter the behavior of 6i forms delivered with the E-Business Suite. Although the declarative nature of this tool does not *require* the use of PL/SQL, it is generally recommended that the user have some understanding of PL/SQL and forms in order to understand the impact of the changes. Forms personalizations are effective immediately as opposed to CUSTOM.pll which must be compiled.

Most changes traditionally done using CUSTOM.pll can be accomplished using forms personalization; however, users may still need to utilize CUSTOM.pll for complex modifications. For the same event, the declarative forms personalizations will fire prior to the CUSTOM.pll for the same event.

Profile Options Impacting Forms Personalization

Hide Diagnostics menu entry – This profile options must be set to “No.” Setting this profile options to “Yes” hides the diagnostics menu.

Utilities: Diagnostics – If this profile options is set to “Yes” the apps password is not required to use diagnostic features including forms personalization and examine. Set this profile option to “No” in production environments.

Creating Forms Personalizations

The following example will describe a personalization for the Supplier form to hide the taxpayer id.

First access the form to be personalized. After opening the form, go to Help → Diagnostics → Custom Code → Personalize to navigate to the forms personalization tool. The forms personalization form for the suppliers form is shown below in Figure 5.

Figure 5 Form Personalizations (Suppliers)

HEADER

The function name and form name will default to the current function and form. These values may not be changed. The debug mode can be set to off, Step-by-Step or Show Debug Messages. *Step-by-Step* will display a pop-up window showing events impacted by the rule. *Show Debug Messages* will show messages where the type = debug.

Sequence numbers are set between 1 and 100 and can be reused. Rules will run in sequence.

Description is free form entry.

Level can be set to form or function. For example, you may want to create a personalization that applies only to the query-only function for the Supplier form.

Personalizations are enabled by checking the *Enabled* check box.

CONDITION

Trigger Event

The following provides some guidelines on which standard trigger event to use for specific types of personalizations related to securing items. The events included in this list are events Oracle generally provides in all forms. However, you must test to make sure the trigger event is actually firing. To find events that are firing, go to Help → Diagnostics → Custom Code → Show Custom Events. As you navigate through the form, a pop-up window will display the trigger events as they fire. You may also choose to use any other event fired in the form, however, Oracle does not guarantee that other events will be retained in patches, so these personalizations may not be protected.

- WHEN-NEW-FORM-INSTANCE or WHEN-NEW-BLOCK-INSTANCE
 - Security rules
 - Navigation rules
 - Visual attributes
- WHEN-NEW-ITEM-INSTANCE
 - Default values dependent on entry of another item
 - Skip to other items
- WHEN-VALIDATE-RECORD
 - Populate hidden fields
 - Additional validations

Trigger Object

The trigger object may be required depending on the trigger event. If the list of values is available when the cursor is in the trigger object field, than the trigger object is required.

Condition

The condition is an optional SQL code fragment used to limit the scope of the personalization. For example, you can limit a message to appear only certain times of the year based on the system date. You may reference the value of a field in the current record in the format *:blockname.fieldname*.

Processing Mode

The processing mode determines when this personalization is applicable. Options are *Only in Enter-Query Mode*, *Not in Enter-Query Mode* or *Both*.

Context

The context determines who the personalization applies to. Multiple rows are allowed and will be processed as “and” statements. Applicable levels are site, responsibility, and user. Industry exists in the list of values, but is reserved for future use. If the context is null, the personalization will apply to all.

TIP: When testing new personalizations, set the context so the personalization only applies to your own user id.

Actions

The Actions tab shows the actions on the left and details about the action on the right. There are 4 action types – property, message, builtin, and menu. The details on the right side of the tab page will change depending on the type of action.

Property action type

Property action types require an object type, target object, property name and value. Security at the item levels can be increased by changing various properties such as insert, update and delete. A list of object types/properties related to security for are listed below.

- Item – size, position, visibility, insert, update, delete, value, case restriction, LOV, format mask and more
- Block – size, position, default where clause and order by, navigation, insert, update, delete
- Tab Page – enabled, displayed, label
- Radio Button – size, position, label, prompt, visibility
- LOV – size, position, group name

Message action type

Message action types require a message type and message. Messages can be displayed at trigger events other than WHEN-NEW-FORM-INSTANCE. Valid message types include Show, Hint, Error, Debug and Warn.

Builtin action type

The following builtins types are available. Many of these built-ins can be used to create additional security.

- Launch SRS Form – launches a concurrent program
- Launch a Function – launches another form function
- DO_KEY – execute a form builtin
- Execute a Procedure – execute any procedure using syntax exactly as you would in PL/SQL code
- GO_ITEM – navigate to a specific item
- GO_BLOCK – navigate to a specific block
- RAISE FORM_TRIGGER_FAILURE
- EXECUTE_TRIGGER – call a trigger.
- Call Custom Library – call a specific event you have coded directly in CUSTOM.pll
- Create Record Group from Query – creates a record group dynamically. Assign this new record group to a list of values to limit data returned in the list.

Menu action type

Using the menu action type only displays the menu in the appropriate location on the tool bar. A builtin to execute the functionality behind this menu item is also required.

FIGURE 6 EXAMPLE 1 – CONCEAL THE TAX ID – CONCEALING DATA PLACES ASTERISKS IN THE FIELD WHERE DATA EXISTS.
Trigger Event = WHEN-NEW-FORM-INSTANCE

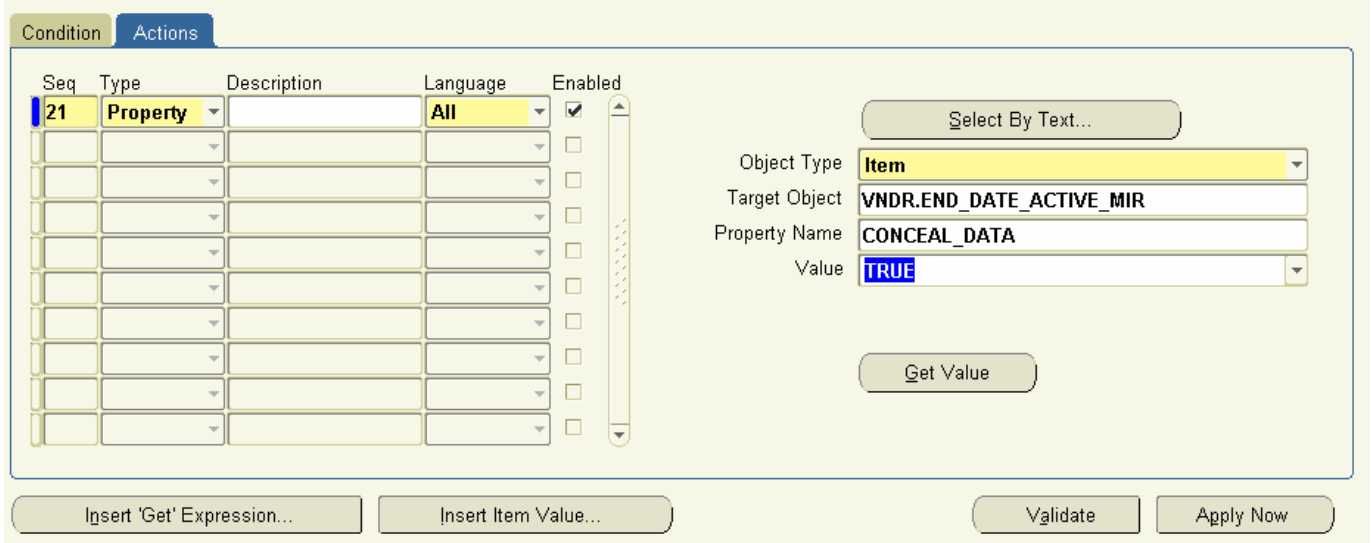


Figure 6 Example 1

OA Framework Personalization

OA Framework personalizations were introduced in 11.5.9 with patch 3323690 and are standard in 11.5.10. They apply only to self service applications and can be used to limit access at the field level. They are somewhat intuitive and this functionality is intended to be utilized by functional superusers.

Profile Options Impacting Forms Personalization

Personalize Self-Service Defn – Set this profile to *Yes* to allow personalizations.

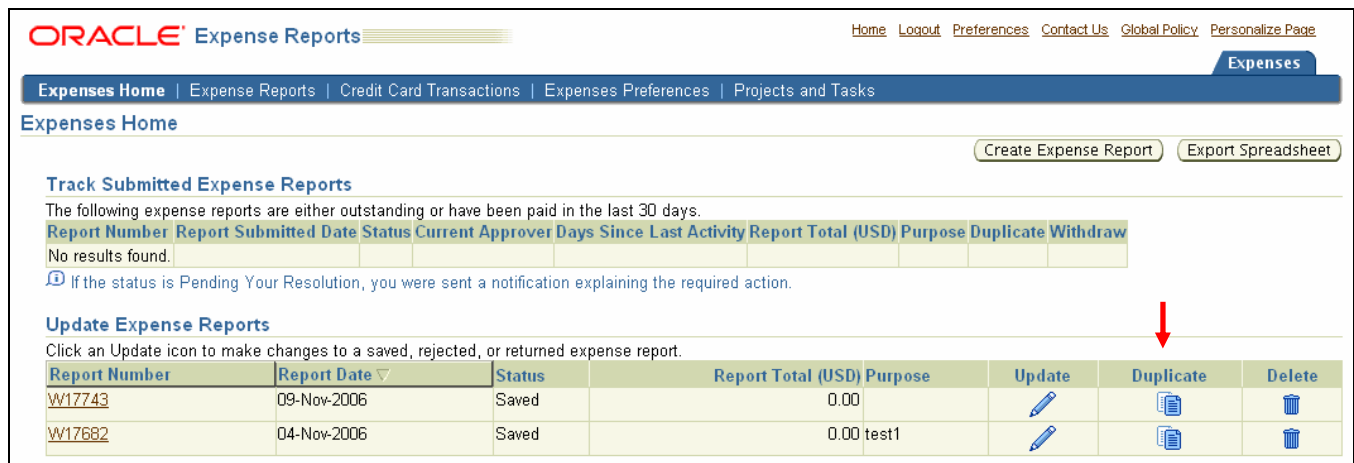
FND: Personalization Region Link Enabled – Set this profile to *Yes* to display the regional links

Disable Self-Service Person - *Yes* will disable all personalizations at any level.

FND: Personalization Document Root Path (new in 11.5.10) - Set this profile option to a tmp directory with open (777) permissions for migrating personalizations between instances.

Personalization Levels

Personalizations can be enabled at the function, site, operating unit or responsibility level. Personalizations at lower levels override personalizations at higher levels. Values inherit the definition from the level immediately above unless changed. Below is a simple example to hide the “duplicate” column on an iExpenses page to disallow the duplication of an expense report.



Set Rendered = False and apply change

The screenshot shows the Oracle Expense Reports Personalization Switcher interface. At the top, it says 'Personalize Switcher : Duplicate' with 'Cancel' and 'Apply' buttons. Below this is the 'Personalization Context' section with the following details:

- Scope: Page: Expenses Home
- Document Name: /oracle/apps/ap/oi/webui/HomePG
- Function: OIE Home Page
- Site: Include
- Organization: Vision Operations
- Responsibility: IE OAUG DEMO

The 'Personalization Properties' table is shown below. The 'Rendered' property is highlighted with a red circle, indicating it has been set to 'false'.

	Original Definition	Function: OIE Home Page	Site	Organization: Vision Operations	Responsibility: IE OAUG DEMO	Result / Source
Admin Personalization	true	Inherit	Inherit	Inherit	Inherit	true / Original Definition
CSS Class	Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Controller Class	Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Export View Attribute	Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
Post Initial Values	false	Inherit	Inherit	Inherit	Inherit	false / Original Definition
Prompt	Duplicate	Inherit	Inherit	Inherit	Inherit	Duplicate / Original Definition
Rendered	true	Inherit	Inherit	Inherit	false	false / Responsibility
Sort Allowed	no	Inherit	Inherit	Inherit	Inherit	no / Original Definition
Sort By View Attribute	Default	Inherit	Inherit	Inherit	Inherit	Default / Original Definition
User Personalization	true	Inherit	Inherit	Inherit	Inherit	true / Original Definition

Summary

At this point, it should be clear that there are many different ways within Oracle applications to impact security. The chart below summarizes options to be considered in the security plan for your organization.

Security Functionality	What is it for?	Setup and Maintained by	Release
Password Profile Options	Authentication	DBA / Sysadmin	All
Timeout Options	Authentication	DBA / Sysadmin	All
Menus, Request Groups, Responsibilities	Authorization for functions	Sysadmin / Superuser	All
Function Security in Responsibilities	Authorization for functions	Sysadmin / Superuser	All
Excluded Items/Securing Attributes in Responsibilities	Authorization for data	Sysadmin / Superuser	All
User Management	Authorization for data	Sysadmin / Superuser	11.5.10
HRMS Forms Configurator	Authorization for functions	Superuser	Release 11i
Security Groups/Security Profiles	Authorization for data	Sysadmin / Superuser	Release 11i
Audit Trail	Audit	DBA / Sysadmin	Release 11i
Custom library	Item level authorization for core forms	Developer	Release 10
Forms6i Personalizations	Item level authorization for core forms	Developer	Release 11i
OA Framework Personalizations	Item level authorization for self service	Superuser	Release 11i

Additional Resources and References

Best Practices for Securing Oracle E-Business Suite, Available for download at MetaLink Note 189367.1

30 Minute Release 11i Security, Keeping the Bad Guys Away , OAUG Insight, Fall 2006, p.12, Randy Giefer, Solution Beacon available at repo.solutionbeacon.net/insight2006-q3.pdf.

Installing, Upgrading & Maintaining Oracle E-Business Suite Applications Release 11.5.10+ - or "Teaching an Old Dogs New Tricks - Release 11i Care and Feeding" , Barb Matthews of OnCallDBA and John Stouffer, Randy Giefer, Karen Brownfield, Jeff Holt, James Morrow, Bruno Coon, Tim Sharpe, and Faun deHenry of Solution Beacon, published by Reed-Matthews, Inc., January 2007 – This document focus primarily on standard application security options available at the System Administrator level such as profile options and user management and is available at www.oncalldb.com/docs/Books/index.shtml.

Oracle Applications Concepts, Chapter 12 – This document provides a high level overview of security concepts and is available in all Oracle Applications document sets and on MetaLink.

Oracle Applications System Administrator's Guide – Security – This is a detailed document explaining all System Administrator Security Topics and is available in all Oracle Applications document sets and on MetaLink.

Oracle Application Framework Personalization Guide – This document covers OA Framework Security Topics and is available in all Oracle Applications document sets and on MetaLink.

MetaLink Note: 290996.1 Oracle User Management Supplementary Documentation

Oracle HRMS Configuring, Reporting, and System Administration Guide – This document describes how security groups and security profiles are used to limit an employee's access to data. Instructions are also provided to create templates and taskflows.

Oracle Applications Developer's Guide – This document contains instructions for the use of the Custom Library.

MetaLink Note: 290996.1 Information About the Oracle Applications Form Personalization Feature in 11i – This document includes instructions for Forms6i personalization.