



SUNGARD PUBLIC SECTOR
SECURITY

IFAS
Integrated Financial &
Administrative Solution

NOTICE

SUNGARD PUBLIC SECTOR BI-TECH LLC MAKES NO REPRESENTATIONS OR WARRANTIES, ORAL OR WRITTEN, EXPRESS OR IMPLIED, WITH RESPECT TO THE SYSTEM, SERVICES, SOFTWARE, DOCUMENTATION, OPERATING ENVIRONMENT, ANY OTHER SOFTWARE OR SERVICES PROVIDED HEREUNDER OR ANY OTHER MATTER ADDRESSED HEREUNDER, AND SUNGARD PUBLIC SECTOR BI-TECH LLC EXPLICITLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF TITLE, MERCHANTABILITY AND FITNESS FOR A SPECIFIC PURPOSE. SunGard Public Sector Bi-Tech LLC shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This documentation is proprietary and confidential information of SunGard Public Sector Bi-Tech LLC. Copying, reproduction or distribution is strictly prohibited. All rights reserved.

Copyright © 2008 by
SunGard Public Sector Bi-Tech LLC
890 Fortress Street
Chico, CA 95973

Should you wish to make a comment, or if you find an error in the text, please contact us via email:

doc@bi-tech.com

Document Change Log

| Version | Date | Change Description |
|----------------|-------------|---------------------------|
| 7.9 | June 2008 | SQL corrections |

Contents

| | |
|--|-----------|
| 1 Overview | 7 |
| 1.1 Introduction to Role-Based Security | 7 |
| 1.1.1 Security Structure | 7 |
| 1.1.2 Security Roles | 9 |
| 1.1.3 Inclusive vs. Exclusive Security | 10 |
| 1.1.4 IFAS Menu Masks | 10 |
| 1.1.5 Application Functionality | 10 |
| 1.1.6 IFAS Data Access | 11 |
| 1.1.7 Common Security | 11 |
| 1.1.8 Resulting Security – Menu Access | 12 |
| 1.1.9 Resulting Security – Application Functionality | 13 |
| 1.1.10 Resulting Security – IFAS Data | 13 |
| 1.1.11 No Simple Answers | 19 |
| 1.1.12 Common Approaches to Role-Based Security | 22 |
| 1.1.13 The Vanilla School District | 24 |
| 1.1.14 Security Planning | 51 |
| 1.1.15 Implementing Security | 61 |
| 1.2 Security System Features | 63 |
| 1.3 Getting Started | 63 |
| 1.3.1 Assign Security Roles Administrative Tool | 63 |
| 1.3.2 Column Level Security | 65 |
| 1.4 Flow Diagram | 74 |
| 1.5 Basic Steps | 74 |
| 2 Setup | 75 |
| 2.1 Basics | 75 |
| 2.1.1 Terms and Definitions | 75 |
| 2.1.2 Concepts | 75 |
| 2.1.3 Masks and Corresponding Data Set Names | 96 |

| | | |
|----------|---|------------|
| 2.2 | Intermediate | 99 |
| 2.3 | Advanced | 99 |
| 2.4 | Best Practices | 99 |
| 3 | Processes | 100 |
| 4 | Process Reference | 101 |
| 4.1 | Entry | 101 |
| 4.2 | Processing | 101 |
| 4.2.1 | GL functional Security | 101 |
| 4.2.2 | Purchasing Functional Security | 104 |
| 4.3 | Utilities | 109 |
| 4.3.1 | Security Import | 110 |
| 4.3.2 | Security Export | 120 |
| 4.4 | Reports | 128 |
| 5 | Maintaining and Troubleshooting Security | 129 |
| 5.1 | Managing Security | 129 |
| 5.2 | Security Internals | 129 |
| 5.2.1 | Database Tables | 129 |
| 5.3 | US_SECOBJ_MSTR | 129 |
| 5.4 | US_ROLE_MSTR | 129 |
| 5.5 | US_ROLESEC_DTL | 130 |
| 5.6 | US_ROLE_DTL | 130 |
| 5.7 | IFAS_DATA | 130 |
| 5.7.1 | Persisted Information | 130 |
| 5.8 | Troubleshooting Security Problems | 132 |
| 5.8.1 | User Security Audit Report | 132 |
| 5.8.2 | Rebuild User Isn't Working | 132 |
| 5.8.3 | User Lacks Expected Menu Access | 132 |
| 5.8.4 | User Lacks Expected Data Access | 133 |
| 5.8.5 | When and How to Use Tracing for Security | 134 |

| | | |
|----------|---|------------|
| 6 | Advanced/Special Configuration | 135 |
| 6.1 | Defaults Rules (NUUPDF) | 135 |
| 6.1.1 | Entity List | 137 |
| 6.1.2 | VBScript Editor | 138 |
| 6.1.3 | XML Editor | 148 |
| 6.1.4 | BT20 Tree View | 154 |
| 6.2 | Security Recovery | 155 |
| 6.3 | Rebuild Security | 156 |
| 6.4 | Forgot Login Page..... | 157 |
| 6.4.1 | Configuration | 157 |
| 6.4.2 | Login Page..... | 158 |
| 6.4.3 | Workflow Email | 158 |
| 7 | Implementation | 162 |
| 7.1 | Dependencies | 162 |
| 7.1.1 | Comparing 7.9 Security and its Predecessors | 162 |
| 7.2 | Templates | 164 |
| 7.3 | Agendas | 164 |
| 7.3.1 | Security Agenda | 164 |
| 8 | Module Integration..... | 165 |
| 9 | FAQ | 166 |

1 Overview

The IFAS Security System is used to establish user log-on capabilities, define user database access, and define user job running capabilities. This guide provides you with the information required for establishing and maintaining IFAS security.

The IFAS Security System is the means by which your Organization's internal accounting controls are established. By correctly defining user access and job running capabilities, safeguarding of the financial data, as well as the separation of duties, is greatly enhanced.

System security can only be defined by a user who has been given the capability of managing users. The ability to set system security will be given to one or two persons at the time of system installation. It will then be up to them to determine the security levels for other users. If a user should receive warnings regarding lack of access or job running capabilities on reports or during interactive Inquiry into any database, the user should contact the person in charge of security to determine if the user's capabilities are defined correctly.

1.1 Introduction to Role-Based Security

The need to control access to different aspects of the software is a central requirement for all organizations. In IFAS that involves controlling access to Menu options, Application Functionality and the information stored in the database. In addition, the security must be flexible enough to accommodate the diverse and constantly evolving nature of both our clients and the software.

1.1.1 Security Structure

All of the security in the software is represented by what we refer to as the "Security Structure". This is essentially a hierarchical organization of each type of menu mask, application functionality, database table, etc. Anything that can have security applied against it has a corresponding security object in the structure. The reason for using a hierarchal representation is that it allows security to be applied to a security higher up on the tree and then have that security ripple down to the objects below it. This allows access to be granted or restricted at an upper level without having to explicitly address every object below it. For items such as CDD Folders or Menu Masks, this hierarchy is somewhat obvious. However, for Data and Application security the structure can be a bit less intuitive.

In order to accommodate a very heterogeneous set of security objects, the type of security that can be applied to each security object changes depending on the type of object. So for CDD Folders the access is controlled based on Read, Write, Update, Delete and Execute but for Menu Masks the only access that really applies is Execute since the ability to alter records is handled by the Data access once the screen or utility is run. For items such as database tables simply controlling the type of access isn't enough because it fails to allow data to be controlled based on

specific data in the table. Therefore, filters can be applied to the security objects representing the database tables to further restrict how much or how little data will be made available to that security object.

For most of the IFAS Subsystems, the Security Structure has a somewhat predictable layout. Each subsystem has a root node, and below that are three nodes for the Menu, Data and Function access. Below the Menu Access node is the IFAS menu structure as it would appear from the 7i Menu. Below the Data Access is the list of Database Tables considered to be part of that subsystem. Below the Functions node is a list of Application Functionality specific to that subsystem.

Subsystem Root

Subsystem Menu

Subsystem Data

Subsystem Functions

The example below shows a small subset of security nodes below the Person/Entity and Click Drag & Drill root nodes in the security example.

```

Person/Entity
  Person/Entity Menu
    PE - Person/Entity Menu
      UP - Update Data Base
      PE - Person/Entity Information
  Person/Entity Data
    PE_NAME_MSTR
  Person/Entity Functions
Click Drag & Drill
  CDD Folders
    Vendor Reports
  CDD Functionality
    CDD Reports
  
```

Every user essentially has their own copy of this Security Structure. We may store it in different ways, depending on the needs of the software, but in the end all of a user's access reduces down to how much or how little of the Security Structure they have access to at any given time. In this way the setup of security is more about deciding how much of the structure they will have in their copy than it is deciding what they can or cannot do explicitly. The example below lists two possible users and their resulting access within the software.

| User | Security Structure | Result |
|-------|---|---|
| Sally | Person/Entity Person/Entity Menu PE - Person/Entity Menu UP – Update Data Base PE – Person/Entity Information Person/Entity Data PE_NAME_MSTR Click Drag & Drill CDD Folders Vendor Reports CDD Functionality CDD Reports | Sally can run PEUPPE as well as any reports within the Vendor Reports folder in CDD. |
| Joe | Person/Entity Person/Entity Data PE_NAME_MSTR Person/Entity Functions Click Drag & Drill CDD Folders Vendor Reports CDD Functionality CDD Reports | Joe can run CDD reports in the Vendor Reports folder but is unable to run PEUPPE to edit any vendors. |

For some aspects of IFAS, more than one node in the Security Structure may be required for access to a given CDD Report or Screen. In the example above access to the PEUPPE screen is controlled by Menu security on that mask. However, access to the data on that screen is controlled by access to the PE_NAME_MSTR table.

1.1.2 Security Roles

The application of security to each user is facilitated by Security Roles. Modifications to the Security Structure that can be grouped and named individually and then applied to users in layers. One role might provide access to some CDD folders that the user will run reports from.

Another role might control how much financial information they can see in IFAS at any given time. How the roles are created and managed is perhaps the most important and potentially time consuming part of the setting up the security in IFAS.

An important aspect of the security roles is the concept of "derived" permissions. This means that, unless otherwise specified, a security object will inherit its security from the security object above it in the Security Structure. If that is not specified then it will continue to ripple up the hierarchy until it either finds some explicit security or no access at all. In this way security does not need to be applied to every single security object in the Security Structure but can be granted for all nodes under a parent security object. However, as soon as a security object has its derived setting removed it will no longer be impacted by the object above it.

1.1.3 Inclusive vs. Exclusive Security

One of the concepts that may take a little time to get used to is the idea of "Inclusive Security". Inclusive Security does not specifically remove functionality from users but adds to their function abilities. The more Security Roles a user has the more access they will have to the software. To control access to sensitive data or application functionality you do not deny access to certain users but simply withhold any roles that would grant access. Understanding this is the key to developing roles that can be reused throughout the organization.

It is also important to consider the long term maintenance issues that may result from role creation. For example, by having particularly sensitive data or functionality granted in multiple roles it may become difficult to manage who can and cannot see that data later. What is considered sensitive can vary greatly from client to client. Another reason for the flexibility of this security model is that it allows each organization to structure their roles based on their individual needs.

1.1.4 IFAS Menu Masks

The main IFAS Menu is a collection of screens, reports and processes grouped by subsystem and then function. Because this list is represented to the user in a hierarchical tree it fits nicely into the Security Structure. The security is on or off for each menu option by granting the "Execute" permission within a Security Role. Access can be granted or denied at either the top level of that subsystem's menu structure or controlled at the very low levels of the menu tree. If two roles are assigned to a user and one grants a menu option and the other does not grant it, the result is that the user is granted the menu option.

1.1.5 Application Functionality

Different aspects of Application Functionality are controlled by the Security Structure. This can be anything from Executing CDD Reports without Selection Criteria to the ability to Print Purchase Orders. How an application will interpret the levels of access on each of these nodes can be very specific to that application. Therefore, we recommend consulting the individual subsystem's user guide for more information about a particular piece of Application Functionality.

1.1.6 IFAS Data Access

A complicated and a sensitive aspect of Role-Based Security is the IFAS Data Access security. Access can be granted to entire subsystem of tables by using the subsystem's data node. Alternatively, access can be granted on a table by table basis and filtered using a SQL Where Clause to provide more specific restrictions.

For each IFAS subsystem there are database tables listed below the top-level data object. For many subsystems this will include a combination of transaction, batch and other miscellaneous tables directly related to that portion of the software. Many of the tables will be granted to any user whose job requires them to have access to that subsystem. This can be done fairly easily by granting the top-level data node for that subsystem and as a result granting access to the tables below.

However, for some users the access to some of the tables within that subsystem will need to be filtered to control how much of the data within that table they have the ability to view or alter. For each of these tables security will need to be applied specifically to them by removing their "derived" status, specifying the desired access and then specifying that access with a filter.

1.1.7 Common Security

Some of the data in IFAS is considered to be common to many parts of the software. A restriction on this common information extends to all subsystems that reference it. Applying the same Data Access filter to each occurrence of that common information would make security complicated to setup and maintain. To reduce this administrative burden the concept of Common Security was made part of the Role-Based Security design.

A simplified example of this security concept is security on the General Ledger Account Keys in the software. The Account Keys themselves are stored in the GLK_KEY_MSTR. There is a security object for that database table stored within the General Ledger subsystem in the Data list. However, the Account Key is used on a lot of different tables in IFAS. To accommodate that need there is a Common Security object for the Account Keys that provides for a subset of the keys to be granted using a filter on that table.

It's important to remember that some tables will therefore exist twice in the security table. This is to allow the user's access to the Common Security to differ from access to the table itself. For example, you may want to configure a user's access to update data based on a subset of Account Keys but not grant the ability to change the Account Keys themselves.

Granting access to the Common Security object doesn't necessarily grant any access to the tables in IFAS. Alternatively, granting access to a particular table does not necessarily grant access to its related Common Security object. Without both the desired Table and its associated Common Security objects the user would not have access to any data in the table. The following are a few examples of IFAS tables with links to Common Security objects. A complete list can be generated using the "Common Security Listing" screen in the Admin Console.

| IFAS Table | Common Linkages |
|--------------|--|
| GLK_KEY_MSTR | "Ledger Security" by ledger "Account Key Security" by key |
| GLT_TRNS_DTL | "Ledger Security" by ledger "Account Key Security" by key "Object Code Security" by object |
| HR_EMPMSTR | "Employee Definition" by ID |
| HR_EMPPAY | "Employee Definition" by ID |

1.1.8 Resulting Security – Menu Access

For menu access determining what a user's resulting security will be is pretty straight forward. Any role that provides access to the IFAS Menu mask will enable that mask for the user. Initially, no access is granted to any menu options. However, once Execute access is granted to that menu option it will be accessible. Also, since the menu portion of the Security Structure is represented in the same hierarchy as it is used by the software granting access to a top-level menu will result in granting access to all child nodes of that menu. Of course, removing the derived option from a child menu will override the access granted by a parent of that menu option.

Additionally, the more roles with menu access granted assigned to a particular user the more menu options that user will have available to them. The chart below lists four masks and how two different roles would have interpreted their access. Remember that Role-Based security is inclusive so even if one role does not grant access, another role can undo that restriction by granting access to that menu. When a menu has not been granted any access at all either by itself or from a parent node in the Security Structure then the user will not have access to that menu.

| Menu Mask | Role A | Role B | Result |
|-----------|-------------|-------------|----------------|
| PEUPPE | Execute | Not Defined | Access Granted |
| PEUPPR | Not Defined | No Access | No Access |

| | | | |
|--------|-------------|-------------|----------------|
| POUPPR | Not Defined | Not Defined | No Access |
| POUPRC | No Access | Execute | Access Granted |

1.1.9 Resulting Security – Application Functionality

Application Functionality is very similar to the Menu Access. One difference is that some functions include all five types of access (Read, Write, Update, Delete, Execute). The chart below shows the resulting security for different functions depending on role assignments.

| Functionality | Role A | Role B | Result |
|-----------------------|--------|-------------|--|
| CDD Reports | RWUX | R | Read, Write, Update and Execute CDD Report Designs |
| CDD Scriptlets | R | No Access | Read CDD Scriptlets |
| Print Purchase Orders | X | Not Defined | Access Granted |

1.1.10 Resulting Security – IFAS Data

Determining Data Security is more complicated. The software has to determine if the user has Read, Write, Update, Delete or Execute access per table. This is done by checking not only the table itself under its subsystem data node but also any related common security items. Once that is done, any filters that have been written are merged to create one SQL Where Clause that reflects the application of one or more roles.

Example #1 – Simple Table Access

In this example we select a single table with no common links and apply two roles. Two roles are used in this example to show an increasing complexity of security between this example and the two that follow it. Granted, this is probably not the most practical table to select but its useful as a very simple example.

Role A: Grants read access to the CD_CODES_MSTR

Role B: Does not grant any access at all to that table.

| Table | Role A | Role B | Result |
|---------------|---------------|-------------|-------------|
| CD_CODES_MSTR | R (no filter) | Not Defined | Read Access |

This example shows the result of a single filter on the same table from the prior example.

Example #2 – A Simple Filter

Role A: Grants read access to the CD_CODES_MSTR but only for the printer codes.

Role B: Does not grant any access at all to that table.

| Table | Role A | Role B | Result |
|---------------|--|-------------|----------------------|
| CD_CODES_MSTR | R (filter: CD_CATEGORY = 'NULP') | Not Defined | CD_CATEGORY = 'NULP' |

This example will show the result of two roles that both grant Read access to the table and both have filters defined.

Example #3 – Multiple Filters

The "OR" that is added between the two filters will result in both subsets being available to the user. This is done to ensure that the result of two filtered roles will always be Inclusive.

Role A: Grants read access to the CD_CODES_MSTR but only for the printer codes.

Role B: Grants read access to the CD_CODES_MSTR but only for the seeds.

| Table | Role A | Role B | Result |
|---------------|--|---|---|
| CD_CODES_MSTR | R (filter: CD_CATEGORY = 'NULP') | R (filter: CD_CATEGOR Y = 'SYNO') | CD_CATEGORY = 'NULP' OR CD_CATEGORY = 'SYNO' |

Example #4 – Common Security

This example will show the resulting security when Common Security is defined for a particular table. To show the impact of a user's security with and without the Common Security granted on the table, the results will be shown in two parts. In the first part the user does not have any access to the GLK_KEY_MSTR because no access has been granted to its Common Security items. The second example shows how a second role has provided the required Common Security. As a result the user has Read Access to the GLK_KEY_MSTR.

Role A: Grants read access to the GLK_KEY_MSTR table under the General Ledger Data node.

Role B: Grants full access to the "Ledger Security" and "Account Key Security" Common Security items.

Without Role B:

| Common Items | Role A | | Result |
|----------------------|-------------|--|-----------|
| Ledger Security | Not Defined | | |
| Account Key Security | Not Defined | | |
| Table | | | |
| GLK_KEY_MSTR | R | | No Access |

With Role B:

| Common Items | Role A | Role B | Result |
|----------------------|-------------|-----------|-------------|
| Ledger Security | Not Defined | R,W,U,D,X | |
| Account Key Security | Not Defined | R,W,U,D,X | |
| Table | | | |
| GLK_KEY_MSTR | R | | Read Access |

**Example #5 –
Common Security
Filters**

Filters can be defined on both the IFAS Tables and the Common Security tables. The result for the user is a filter that enforces all of the restrictions within that role. Because the Common Security is enforced in addition to the individual table filters, the filters within the role are joined by an "AND" instead of the "OR" that is used between roles.

Role A: Grants read access to the GLK_KEY_MSTR table under the General Ledger Data node with a filter on the GLK_KEY column.

Role B: Grants full access to the "Ledger Security" and "Account Key Security" Common Security items. Also, a filter has been placed on part 1 of the "Account Key Security" Common security.

| Common Items | Role A | Role B | Result |
|----------------------|---|---|---|
| Ledger Security | Not Defined | R,W,U,D,X | |
| Account Key Security | Not Defined | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '01') | |
| Table | | | |
| GLK_KEY_MSTR | R (filter: GLK_KEY >= 10000 and GLK_KEY <= 19999) | | (GLK_GRP_PART_01 = '01') AND (GLK_KEY >= 10000 and GLK_KEY <= 19999) |

Example #5 – Merged Filters

When there are multiple roles with filters defined for the same Common Security or IFAS Data Table the result is a security filter with the common security filters merged with an "OR", the table filters merged with an "OR" and the two sets merged with an "AND".

Role A: Grants full access to the "Ledger Security" and "Account Key Security" Common Security items. Also, a filter has been placed on part 1 of the "Account Key Security" and a filter on the Org. Key of the GLK_KEY_MSTR table.

Role B: Grants full access to the "Account Key Security" Common Security item with a filter on part 1 of the "Account Key Security" and a filter on the Org. Key of the GLK_KEY_MSTR table.

| Common Items | Role A | Role B | Result |
|----------------------|---|---|--------|
| Ledger Security | Not Defined | R,W,U,D,X | |
| Account Key Security | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '01') | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '02') | |

| Table | | | |
|--------------|---|---|---|
| GLK_KEY_MSTR | R (filter: GLK_KEY >= 10000 and GLK_KEY <= 19999) | R (filter: GLK_KEY >= 20000 and GLK_KEY <= 29999) | ((GLK_GRP_PART_01 = '01') OR (GLK_GRP_PART_01 = '02')) AND ((GLK_KEY >= 10000 and GLK_KEY <= 19999)) OR (GLK_KEY >= 20000 and GLK_KEY <= 29999)) |

**Example #6 –
Unfiltered Override**

Role Security will merge the filters of multiple roles for the same table. However, since Role-Based Security uses the concept of inclusive security when a user has unfiltered access to the table the security removes any other filters on the table and the user has access to all records with no restrictions. The result of having access to both a filtered set of rows and all rows would always be all rows. The same is true for any Common Security items. Any role granting unfiltered access to the Common Security item overrides any other filters on that same item.

Role A: Grants full access to the "Ledger Security" and "Account Key Security" Common Security items. Also, a filter has been placed on part 1 of the "Account Key Security" and a filter on the Org. Key of the GLK_KEY_MSTR table.

Role B: Grants full access to the "Account Key Security" Common Security item with a filter on part 1 of the "Account Key Security" but all access with no filter on the GLK_KEY_MSTR table.

| Common Items | Role A | Role B | Result |
|----------------------|---|---|--------|
| Ledger Security | Not Defined | R,W,U,D,X | |
| Account Key Security | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '01') | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '02') | |

| Table | | | |
|--------------|---|-----------|---|
| GLK_KEY_MSTR | R (filter: GLK_KEY >= 10000 and GLK_KEY <= 19999) | R,W,U,D,X | ((GLK_GRP_PART_01 = '01') OR (GLK_GRP_PART_01 = '02')) |

Example #7 – Common Security Gap Common Security provides a convenient method of applying security on commonly used data without visiting security on every single table that it might reference. However, a single missing Common Security access can restrict any access to the table. The following two examples will demonstrate how access to a particular table such as the GLBA_BUDACT_MSTR not only requires access to the table but to the Common Security items the table is linked to as well. The first example will demonstrate the gap in security and the second one will demonstrate a solution to that problem.

Role A: Grants full access to the "Ledger Security" and "Account Key Security" Common Security items. Also, a filter has been placed on part 1 of the "Account Key Security" and unfiltered access to the GLBA_BUDACT_MSTR.

Role B: Grants full access to the "Account Key Security" Common Security item with a filter on part 1 of the "Account Key Security".

| Common Items | Role A | Role B | Result |
|----------------------|---|---|-----------|
| Ledger Security | Not Defined | R,W,U,D,X | |
| Account Key Security | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '01') | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '02') | |
| Object Code Security | Not Defined | Not Defined | |
| Table | | | |
| GLBA_BUDACT_MSTR | R,W,U,D,X | | No Access |

**Example #8 –
Correcting the
Common Security Gap**

To correct the gap from the previous example either security can be applied on the Common Security item to an existing Security Role or a new role can be added with that access granted.

Role A: Grants full access to the "Ledger Security" and "Account Key Security" Common Security items. Also, a filter has been placed on part 1 of the "Account Key Security" and unfiltered access to the GLBA_BUDACT_MSTR.

Role B: Grants full access to the "Account Key Security" Common Security item with a filter on part 1 of the "Account Key Security".

Role C: Grants full access to the "Object Code Security" Common Security item.

| Common Items | Role A | Role B | Role C | Result |
|----------------------|---|---|-------------|---|
| Ledger Security | Not Defined | R,W,U,D,X | Not Defined | |
| Account Key Security | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '01') | R,W,U,D,X (read filter: GLK_GRP_PART_01 = '02') | Not Defined | |
| Object Code Security | Not Defined | Not Defined | R,W,U,D,X | |
| Table | | | | |
| GLBA_BUDACT_MSTR | R,W,U,D,X | | | ((GLK_GRP_PART_01 = '01') OR (GLK_GRP_PART_01 = '02')) |

1.1.11 No Simple Answers

Role-Based Security provides a very flexible method of setting up a user's security. What it doesn't do is provide a quick way of granting access to a particular function of the software. Access to a particular mask may provide access to that screen, but by itself won't provide access

to the data. Additionally, access to the specific tables does not grant access to the Common Security those tables are linked to. The examples below show how access to a specific item in the software requires some level of access to multiple security objects in the structure.

Note: these examples are being used to illustrate the point that a single function in the software can require access to multiple security objects in the structure.

Example #1 – CDD Budget Report

A CDD Budget to Actual report is one of the more common reports. Providing access to this report requires multiple security objects to be granted to the user. To allow a user to run this type of CDD Report, access must be granted to:

- The tables on the report
- All the common security items the tables are linked to
- The ability to execute a CDD Report

- Access to the CDD Report folder the report resides in

Development Oracle
GL Budget to Actual with Encumbrances
 Fiscal Year: 2008
 Report Date: Oct 2007

| Object | Description | Budget | MTD Actual | YTD Actual | Encumbrance | Balance |
|---|---------------------------|--------|------------|------------|-------------|-------------|
| Expense Accounts | | | | | | |
| Key: 000000 - 000000 Long description | | | | | | |
| 0001 | Expense 0001 | 0.00 | \$1,666.66 | \$1,766.66 | 0.00 | -\$1,766.66 |
| 1234 | SomeTestObjectValue | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2000 | Accounts Payable | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4301 | Credit Rev 4301 | 0.00 | 1,228.64 | 1,728.64 | 0.00 | -1,728.64 |
| Total For Org Key 000000: | | 0.00 | \$2,895.30 | \$3,495.30 | 0.00 | -\$3,495.30 |
| Key: 005000 - options general | | | | | | |
| 5001 | Misc supplies | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Total For Org Key 005000: | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Key: 006000 - Ticket 373106 - Encum Key | | | | | | |
| 2005 | Customer Deposits | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6060 | Ticket 373106 - Encum Obj | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Total For Org Key 006000: | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Key: 006122 - Ticket 373106 - Pay Key | | | | | | |
| 6064 | Ticket 373106 - Pay Obj | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Total For Org Key 006122: | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Key: 006999 - Ticket 373106 - Control Key | | | | | | |
| 0001 | Expense 0001 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2005 | Customer Deposits | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6060 | Ticket 373106 - Encum Obj | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Total For Org Key 006999: | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Key: 007000 - Test Budget | | | | | | |
| 6060 | Ticket 373106 - Encum Obj | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Total For Org Key 007000: | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Ready | | | | | | |

Required Security

Data Access

- GLBA_BUDACT_MSTR
- GLK_KEY_MSTR
- GLO_OBJ_MSTR

Common Access

- Ledger Security
- Account Key Security
- Object Code Security
- Budget Version Security

CDD Functionality

- Execute CDD Reports

CDD Folders

- GL Budget Folder

Example #2 – Purchasing Data Entry Screen (POUPPR)

Providing access to the Mask of a data entry screen will allow the screen to open but it won't provide access to the data itself. Even providing access to the tables the screen references won't necessarily provide access to the Common Security objects those tables reference. And in the case of the Purchasing Data Entry screen the list of available Security Codes on the screen is also controlled by the Purchasing Security Codes Common object.

The screenshot displays a web application for purchase requests. The main window shows a list of purchase requests (PR NUMBER, PO NUMBER, V) and a detailed view of a specific request (PR: TS2, PO: P0000382). The detailed view includes fields for Vendor (JGV3), Requested By (JENNY GOWER), Date (09/13/2002), and PO Total (\$101.00). A callout box titled "Required Security" lists the necessary data, common, and menu access for this role.

Required Security

- Data Access**
 - POP_PV_DTL
 - POI_ITEM_DTL
 - POA_ASSOC_DTL
 - POT_TEXT_DTL
- Common Access**
 - Purchase Orders
 - Purchasing Security Codes
- Menu Access**
 - POUPPR Mask

1.1.12 Common Approaches to Role-Based Security

Here are some of the common setup approaches with some their different advantages and disadvantages.

1. Fewest Possible Roles

This approach attempts to accommodate security by creating the least number of Security Roles possible to meet the organizations needs. Frequently this results in somewhat large and complex roles many of which will mirror positions within the organization.

Advantages: There are much fewer roles to keep track of for maintenance. Typically this reduces the amount of time spent initially setting up applying security.

Disadvantages: These types of roles tend to be very inflexible. Any change within the organization or to a particular user's security needs can make it difficult to accommodate within the complex roles. As a result, either the roles are changed resulting in unintended security changes for other users or new roles spring up undermining the advantage that this approach provided initially.

2. One Role per User

This approach creates one Security Role per user typically using a naming convention that matches the users IFAS login.

Advantages: This approach is extremely flexible on a person by person basis. Additional functionality or movement within the organization can be facilitated by changing just that person's role without impacting any other users.

Disadvantages: With hundreds or even thousands of users within the organization this can create a maintenance nightmare. The sheer volume of roles makes any department changes or the introduction of a new module or reporting structure very labor intensive. Also, being able to answer the question of "Which users can perform function X?" can be a daunting task.

Note: Due to the significant differences between the Role-Based Security's inclusive concept and the legacy exclusive method of applying security this approach is the one selected by the 7.9 Security Conversion. Because of the disadvantages, the rollover is considered to be a short term approach to security.

3. One Role per Position

This approach takes each position type within the organization and creates a single security role for each one that handles their access to data, CDD folders, menu masks and application functionality.

Advantages: This method makes it very easy to setup a new employee. It also makes it very easy to manage the changes to a single position for a group of people sharing that position.

Disadvantages: This method also makes it very difficult to answer the "Who in our organization can perform a particular function?" question because that same function may be granted in different roles. Also, it makes it very difficult to manage all of the special cases that can exist within the organization. It seems all too common that, while a person may have a particular position description, their duties or responsibilities may stretch just outside of that position based on training or prior experience.

4. One Role per Security Element

This approach creates one Security Role for each discrete security element. This may be individual application functionality such as the ability to create CDD reports or access to a single or subset of CDD folders.

Advantages: This approach provides a very flexible way of adding or removing security objects from a user's security definition. It also provides a fairly straight forward way to determine how many users can perform a particular function within the organization because the role assignments are easy to report against.

Disadvantages: While this approach might work for an individual subsystem the total number of security objects in the Security Structure to accommodate the IFAS Menus, Database Tables and Application Functionality makes this approach extremely unmanageable. Literally an individual user could end up with hundreds of role assignments.

5. Common Security Needs

This approach involves evaluating the current security needs to look for common security access both horizontally and vertically. This requires reviewing security needs that span across positions within the department and across departments. With this information security roles can be created that grant smaller portions of security and reused across the organization. This allows access to be granted in layers to the users and reduces the need for excessively complicated or numerous roles.

Advantages: This method provides a very flexible way to manage, over time, both the introduction of new users and new functionality in the software.

Disadvantages: There is no "canned" way of setting up security and therefore the burden of creating a solid and reusable security structure within an organization can be substantial. This also requires that the rest of the organization stay committed to this approach or over time the structure will become confusing and ineffective.

While it is within each organization's discretion as to which method to use when designing security, for the purposes of this document we will be using the "Common Security Needs" approach. This is the approach that we firmly recommend.

1.1.13 The Vanilla School District

To help illustrate the process of setting up security, we will use a fictitious school district to demonstrate both the process of analyzing and setting up security. This is obviously a scaled down and somewhat skewed example of an organization's structure but should help to illustrate the security setup process.

Please keep in mind that most of this example will focus on a few areas of security in order to help keep the example as straightforward as possible. This example does not take into account applications such as Documents Online or the more specific security options within different subsystems such as the ability to print Purchase Orders.

Vanilla School District Position Descriptions

The positions listed below are extremely simplified position descriptions for the Vanilla School District. Most likely they do not represent actual positions within your organization but have been established to help give a very generic example.

| Position | General Description |
|----------|---------------------|
|----------|---------------------|

| | |
|-------------------------------|---|
| Superintendent | The Superintendent of the Vanilla School District needs to be able to find out what is going on with the district's finances at any time. This position also needs to be in a position to accommodate questions from parents or the school board. |
| Business Services | |
| Dir. of Business Services | Responsible for the finances of the school district. |
| Budget Manager | Responsible for overseeing the creation and status of the various budgets throughout the district. |
| Procurement Manager | Oversees the Purchasing, Accounts Payable and Stores Inventory functions in the District. |
| Procurement Clerk | Data entry and reporting for procurement functions within the district. |
| Internal Auditor | Responsible for internal audit process within the district. |
| Payroll Manager | Oversees Payroll process within the district. |
| Payroll Clerk | Data entry, internal reporting and contact person for employees with payroll questions. |
| Asst. Dir. Human Resources | Responsible for Human Resources within the district. |
| Teacher Representative | Responsible for setup and maintenance of employee information as well as the first contact for employees within the district with questions. |
| Information Technology | |
| Director Info. Tech | Responsible for the Information Technology needs of the district. |
| Report Writer | Responsible for the creation and maintenance of reports within the district. |
| Analyst | Responsible for the custom applications, data analysis and special projects within the district. |

| | |
|---------------------|---|
| Helpdesk | Responsible for issues within the system and working with vendor for support. |
| Schools | |
| Principal | Responsible for both the long term and day to day planning at an individual school. |
| Assistant Principal | Responsible for internal staff issues and questions within the school and to assist the Principal as needed. |
| School Secretary | Responsible for purchasing data entry and limited financial reporting within the school as well as assisting school administration as needed. |

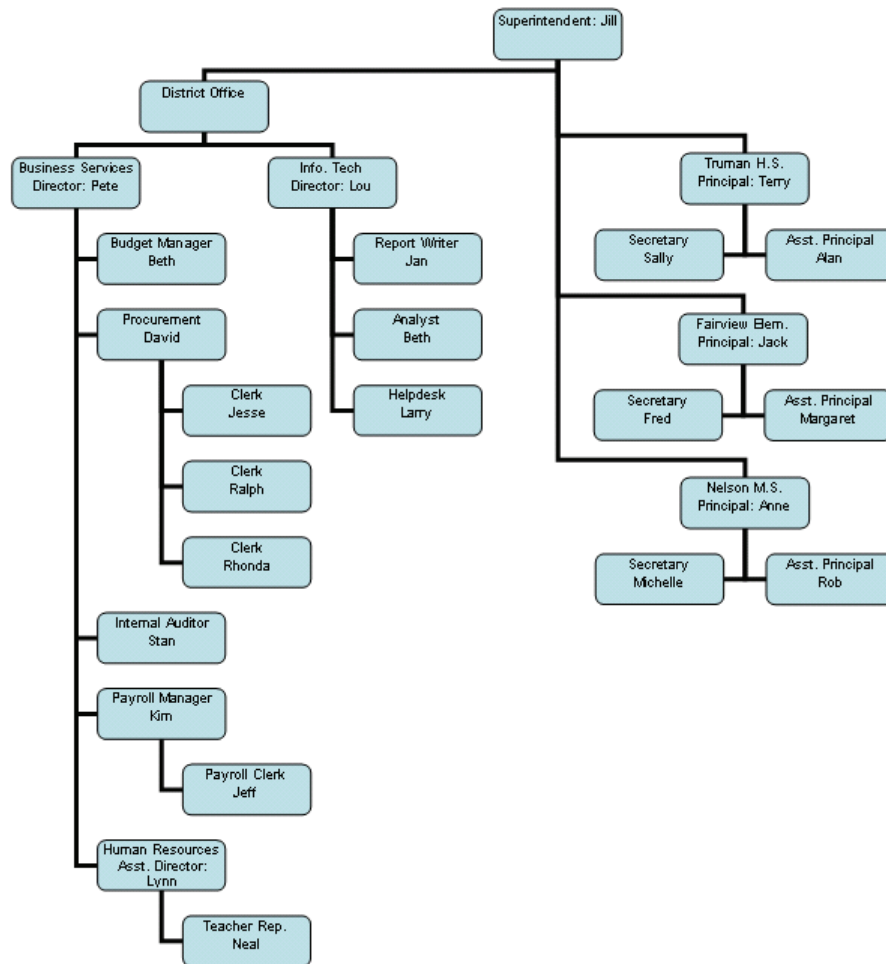
Assessing the Vanilla School District's Security Needs

For the purposes of this document the security needs of the Vanilla School District will be evaluated and configured as a comparatively brief and linear process. In any organization this process may require the input of many different people and force the evaluation to be reworked multiple times before a clear understanding of the needs are laid out. Time spent on design part of this process will yield a time savings both during creation and the long term maintenance of the security structure.

In order to help reduce the number of security variations it is important to look at an organization both vertically and horizontally. This involves looking for discrete types of software usage not just within a single department, but across departments and across position types. Example: Executing CDD Reports is a type of functionality that frequently spans both administrative and data entry positions across departments. Now, which reports can be run or which data can be viewed will change but the ability to run the reports in general is shared.

The district also chose to exploit the fact that access to a particular part of the software was based on a combination of permissions and not just one particular security object. Example: granting access to the financial reports doesn't necessarily grant access to all of the financial data.

The Vanilla School District Organization Chart



Security Needs Round One: The Initial Pass

A small subset of "power users" at the Vanilla School District got together and brainstormed about the organization's current security usage. The intent was to go through each position and figure out what security was needed for each one to fulfill their job responsibilities. Using a combination of job descriptions, contractual restrictions and obligations as well as their own first-hand experiences the following breakdown was created.

| Position | Software Usage |
|---------------------------|---|
| Superintendent | <p>Runs Summary Financial and Personnel reports for the district</p> <p>Some IQ type reports to answer questions from the School Board</p> <p>No data restrictions at all</p> |
| Business Services | |
| Dir. of Business Services | <p>Financial reports</p> <p>Personnel reports</p> <p>Limited Data Entry Access</p> <p>No data restrictions at all</p> |
| Budget Manager | <p>Financial reports</p> <p>Data Entry to all GL Screens</p> <p>No restrictions on Financial data</p> <p>No access to Personnel data</p> |
| Procurement Manager | <p>Financial reports</p> <p>Procurement reports (AP, PO, SI, AR)</p> <p>Procurement Data Entry screens (AP, PO, SI, AR)</p> <p>No restrictions on Financial data</p> <p>No access to Personnel data</p> |

| | |
|----------------------------|---|
| Procurement Clerk | Procurement reports (AP, PO, SI, AR) Procurement masks other than utilities (AP, PO, SI, AR) No restrictions on Financial data No access to Personnel data |
| Internal Auditor | Financial reports Payroll reports No data restrictions at all |
| Payroll Manager | Payroll reports Access to all PY masks No data restrictions at all |
| Payroll Clerk | Payroll reports Access to all Payroll masks other than utilities No data restrictions at all |
| Asst. Dir. Human Resources | HR reports Access to all HR masks other than utilities No restrictions to Personnel data |
| Teacher Representative | HR reports Access to all HR masks other than utilities No restrictions to Personnel data |

| | |
|------------------------|--|
| Information Technology | |
| Director Info. Tech | Access to all Reports No data restrictions at all |
| Report Writer | Access to all reports Access to develop reports Access to manage CDD folders No data restrictions at all |
| Analyst | Access to all masks Access to all reports No data restrictions at all |
| Helpdesk | Access to all masks Access to all reports No data restrictions at all |
| Schools | |
| Principal | Financial reports Personnel reports Financial data for that particular school Personnel data for that particular school |

| | |
|---------------------|--|
| Assistant Principal | Financial reports Personnel reports Financial data for that particular school Personnel data for that particular school |
| School Secretary | Financial reports Purchasing data entry screens Financial data for that particular school |

Resulting Security Types

Based on the resulting needs list the following security groupings were created. These don't necessarily translate directly to roles as much as they create a better idea about the type of security the roles need to accommodate.

Data Access

- All Data Access
- All Financial Data Access
- All Personnel Data Access
- School Specific Financial Data Access
- School Specific Personnel Data Access

Menu Access

- All Masks
- Data Entry GL Screens
- Procurement Masks other than utilities (AP, PO, SI)
- Procurement Data Entry
- Payroll Masks other than utilities
- HR Masks other than utilities

Report Access

- All Reports
- Financial Reports

- Personnel Reports
- School Financial Reports
- School Personnel Reports

Application Functionality

- Run Reports
- Create/Edit Reports

Security Needs Round Two: Blurring Security Lines

Fortunately the Vanilla School District was wise enough to realize that their first pass at the security needs of the organization was probably not comprehensive enough to create security groupings. So they circulated the first draft of the security needs document out to various people within the district and received feedback about the needs of different positions.

| Position | Name | Additional Software Usage |
|----------------------------|--------|--|
| Procurement Clerk | Jesse | Jesse is the only clerk that needs access to Stores Inventory. |
| Procurement Clerk | Rhonda | Rhonda gets the flat-file interface from our Office Supplier once a month and needs access to a custom import utility into AP. |
| Payroll Clerk | Jeff | Some staff seem to have trouble understanding the difference between HR and PY so Jeff gets a lot of questions about people's current HR setup using the HR screens. However, we don't really want Jeff changing any HR setup. |
| Asst. Dir. Human Resources | Lynn | Lynn has taken a number of CDD classes and will need to be able to make changes to existing HR reports in the system. |
| Analyst | Beth | Beth has taken a number of CDD classes and develops reports when Jan's workload gets a bit high. |
| Helpdesk | Larry | While Larry doesn't develop new reports he does make small tweaks to report designs to fix minor issues. |

| | | |
|------------------|------|---|
| School Secretary | Fred | Jack historically has struggled using the School Reports and typically just asks Fred to run the financial reports for him. Most of the time this has involved him giving Fred his login and password and we would really like to avoid that. However, that does not include granting access to personnel reports reserved for the school administrators. |
| Asst. Principal | Alan | The student government at the high school has an account they use for their activities. Since Alan is their staff representative he does a lot of the PO data entry for them and runs their reports. |
| Principals | All | Initially we had thought about giving the Principals all the Financial reports, but the menu structure made it confusing for them to find exactly the reports they wanted and a few times they ran resource intensive month end reports just to get their available budgets. It would be nice to give them a scaled down list of reports. |

Resulting Security Types

With this new information in mind we have reworked the security types needed by the district to accommodate its employee's software usage. (changes in bold) Some of the changes have resulted in the alteration of the security types while others will simply change the way the existing types are applied to individual users. This blurring of the security lines is another reason why the "One Role per Position" solution can be very problematic.

Data Access

- All Data Access
- All Financial Data Access
- **All Payroll Data Access**
- **All Human Resources Data Access**
- **Read Only Human Resources Data Access**
- School Specific Financial Data Access
- School Specific Personnel Data Access

Menu Access

- All Masks
- Data Entry GL Screens
- **Procurement Masks other than utilities (AP, PO)**

- **Stores Inventory Masks**
- Procurement Data Entry
- **AP Flat File Import Utility**
- Payroll Masks other than utilities
- HR Masks other than utilities

Report Access

- All Reports
- Financial Reports
- **Budget Reports**
- **Procurement Reports**
- Personnel Reports
- School Financial Reports
- School Personnel Reports

Application Functionality

- Run Reports
- **Create Reports**
- **Edit Reports**

Security Setup for the Vanilla School District

In theory there were probably several iterations of the security needs analysis before the district had a firm grasp of what kinds of security they needed exactly. For our purposes we will skip ahead to the setup portion of this process.

Armed with a detailed outline of the needs of the positions, the exceptions to those rules, and a breakdown of common security needs, they were ready to lay out what types of Security Roles they would create to accommodate those needs. In order to assist the long-term maintenance and planning within their security setup they attempted to document each of the roles and how it was going to be used.

Note: this example is purely fictitious and is not intended to be used as an implementation example in any way. This is merely a method of demonstrating how user access will be impacted by the configuration of the Security Roles.

Security Roles

The following roles were created by the district to accommodate the security needs of its users. While a lot of work may have gone into these initial roles everyone involved was aware that security is an ongoing process and at any time some of these roles may need to be broken out into different roles to accommodate the evolving needs of both the organization and the software.

Data Access

The first roles created by the district were designed to provide the different types of Data Access that would be needed by the employees.

| Role Name | Nodes | Description |
|-------------------------|---|---|
| Financial Data | Accounts Payable Data (RWUDX) Bank Reconciliation Data (RWUDX) General Ledger Data (RWUDX) Encumbrances Data (RWUDX) Person/Entity Data (RWUDX) Purchasing Data (RWUDX) Stores/Inventory Data (RWUDX) Common Security Accounts Payable Bank Reconciliation General Ledger Budget Version Security Ledger Security Object Code Security Purchasing Stores Inventory | This role provides access to the Financial tables. |
| Payroll Data | Payroll Data (RWUDX) | This role provides access to the Payroll tables. |
| Human Resources Data | Human Resources Data (RWUDX) | This role provides access to the Human Resources tables. |
| All Accounts | Common Security General Ledger Account Key Security (RWUDX) | This role provides unrestricted access to financial data. |
| All Employees | Common Security Human Resources Employee Definition | Provides full access to all employees. |
| Read Only Employee Data | Common Security Human Resources Employee Definition (R Only) | This role provides a read only view of the Employee data. |
| Truman Data | Common Security | This role provides access to this school's data. |

| | | |
|-------------------------------------|--|--|
| (Truman High School) | <p>General Ledger Account Key Security (RWUDX + Filter)</p> <p>Human Resources Employee Definition (RWUDX + Filter)</p> <p><i>Note: The “Account Key” Common Security object was granted and filtered for the school’s background part. The “Employee Definition” node of the Common Security was granted and filtered by the “Location” code on the table for that school. (WORKSITE)</i></p> | |
| Fairview Data (Fairview Elementary) | <p>Common Security General Ledger Account Key Security (RWUDX + Filter)</p> <p>Human Resources Employee Definition (RWUDX + Filter)</p> <p><i>Note: The “Account Key” Common Security object was granted and filtered for the school’s background part. The “Employee Definition” node of the Common Security was granted and filtered by the “Location” code on the table for that school. (WORKSITE)</i></p> | This role provides access to this school’s data. |
| Nelson Data (Nelson Middle School) | <p>Common Security General Ledger Account Key Security (RWUDX + Filter)</p> | This role provides access to this school’s data. |

| | | |
|--|--|--|
| | <p>Human Resources Employee Definition (RWUDX + Filter)</p> <p><i>Note: The "Account Key" Common Security object was granted and filtered for the school's background part.</i></p> <p><i>The "Employee Definition" node of the Common Security was granted and filtered by the "Location" code on the table for that school. (WORKSITE)</i></p> | |
|--|--|--|

Menu Access

Based on the user variations at the Vanilla School District for Menu Access on the Data Entry screens, the district decided that it would use fewer Data Access roles and more Menu roles to control access to update the data.

| Role Name | Nodes | Description |
|---------------------------|--|---|
| GL Masks | <ul style="list-style-type: none"> ● GL - General Ledger Menu (X) <ul style="list-style-type: none"> ○ UT – Utilities (Underived – no access) | By granting the top-level GL mask all of the masks below it that were marked as derived will inherit the security. The utilities was the one are underived and left with no access. |
| Procurement Masks | <ul style="list-style-type: none"> ● PE –Person/Entity (X) <ul style="list-style-type: none"> ○ UT – Utilities (Underived – no access) ● AP - Accounts Payable Menu (X) <ul style="list-style-type: none"> ○ OHUT – Utilities (Underived – no access) ● PO – Purchasing (X) <ul style="list-style-type: none"> ○ UT – Utilities (Underived – no access) | PE, AP and PO top level masks were granted access. The utilities mask below each of them was underived and left with no access. |
| Stores Inventory Masks | <ul style="list-style-type: none"> ● SI – Stores/Inventory <ul style="list-style-type: none"> ○ UT – Utilities (Underived – no access) | The SI top level mask was granted and only the utilities masks underived and left with no access. |
| Purchase Order Data Entry | <ul style="list-style-type: none"> ● POUPPR – Purchase Requests (X) ● POUPRC – Receiving Information (X) | The two main screens used by Purchase Order entry in the district were granted execute permissions. |
| AP Flat File Import | <ul style="list-style-type: none"> ● APOHCSFI – Client Specific Import (X) | The district's flat file import mask was granted execute permissions. |
| Payroll Masks | <ul style="list-style-type: none"> ● PY – Payroll Menu (X) <ul style="list-style-type: none"> ○ UT – Utilities (Underived – no access) | The top level Payroll Mask was granted execute and the utilities underived and left with no access. |
| HR Masks | <ul style="list-style-type: none"> ● HR – Human Resource Menu (X) <ul style="list-style-type: none"> ○ UT – Utilities (Underived – no access) | The top level HR mask was granted execute and the utilities underived and left with no access. |

Report Access

| Role Name | Nodes | Description |
|---------------------|---|---|
| All Reports | CDD Folders (RWUDX) | The top-level CDD Folder was granted Read, Write, Update, Delete and Execute. |
| Financial Reports | CDD Folders GL Reports (RWUDX) AP Reports (RWUDX) PO Reports (RWUDX) SI Reports (RWUDX) | The top-level folder for each subsystem's reports is granted full access. |
| Budget Reports | CDD Folders GL Reports Budget Reports (RWUDX) | The folder containing Budget reports is granted full access. |
| Procurement Reports | CDD Folders AP Reports (RWUDX) PO Reports (RWUDX) SI Reports (RWUDX) | The folders containing Procurement reports are granted full access. |
| Payroll Reports | CDD Folders PY Reports (RWUDX) | The folder containing PY reports is granted full access. |
| HR Reports | CDD Folders HR Reports (RWUDX) | The folder containing HR reports is granted full access. |
| Personnel Reports | CDD Folders HR Reports School Reports (RWUDX) PY Reports School Reports (RWUDX) | The PY and HR report folders that are intended for users at the school sites are put in one role. All reports join to the HR_EMPMSTR to ensure data security. |

Application Access

| Role Name | Nodes | Description |
|------------------|--------------------------------------|---|
| Run Reports | CDD Functionality CDD Reports (X) | Execute permission is granted on CDD Reports. |

| | | |
|----------------|--|--|
| Create Reports | CDD Functionality (RWUDX) | The top level CDD Functionality node is granted full access. |
| Edit Reports | CDD Functionality CDD Reports (RWUDX) | Full access is granted to the CDD Reports functionality. |

Special Considerations

| Role Name | Nodes | Description |
|------------|--------------------------|--|
| All Access | Application Root (RWUDX) | Full access is granted to all of the system. |

Detailed User Security Needs

- Jill (Superintendent)
In the end, Jill just needed to be able to answer the questions she got from the School Board and from the occasional outside request.
- Pete (Dir. Business Services)
- Beth (Budget Manager)
- David (Procurement Manager)
- Jesse (Procurement Clerk)
- Ralph (Procurement Clerk)
- Rhonda (Procurement Clerk)
- Stan (Internal Auditor)
- Kim (Payroll Manager)

| Role Assignments User | Position | Roles |
|--------------------------|----------|-------|
|--------------------------|----------|-------|

| | | |
|--|-------------------------------|---|
| <p>Jill Security provides access to run any financial report against all of the financial data in the system.</p> | <p>Superintendent</p> | <p>Financial Data All Accounts Financial Reports Run Reports</p> |
| <p>Pete Access to run any report against the Financial, HR or Payroll data. Limited access to the masks in IFAS.</p> | <p>Dir. Business Services</p> | <p>Financial Data Payroll Data Human Resources Data All Accounts All Employees GL Masks All Reports</p> |
| <p>Beth Access to run all Financial reports against the system and limited access to GL masks.</p> | <p>Budget Manager</p> | <p>Financial Data All Accounts GL Masks Financial Reports Run Reports</p> |

| | | |
|---|---------------------|---|
| David Access to run any of the Financial reports. | Procurement Manager | Financial Data All Accounts Financial Reports Run Reports |
| Jesse Access to run any of the Financial reports as well as the Procurement and SI masks. | Procurement Clerk | Financial Data All Accounts Procurement Masks Stores Inventory Masks Financial Reports Run Reports |
| Ralph Access to run any Financial reports as well as access to the Procurement masks. | Procurement Clerk | Financial Data All Accounts Procurement Masks Financial Reports Run Reports |

| | | |
|--|--------------------------|---|
| <p>Rhonda Access to run any Financial reports as well as access to the Procurement masks and the Flat File import.</p> | <p>Procurement Clerk</p> | <p>Financial Data All Accounts Procurement Masks AP Flat File Import Financial Reports Run Reports</p> |
| <p>Stan Access to run any reports against Financial and Payroll data in the system.</p> | <p>Internal Auditor</p> | <p>Financial Data Payroll Data All Accounts All Employees Financial Reports Payroll Reports Run Reports</p> |

| | | |
|---|------------------------|---|
| <p>Kim Access to run any Financial or Payroll data as well as the Payroll Masks.</p> | <p>Payroll Manager</p> | <p>Financial Data Payroll Data All Accounts All Employees Payroll Masks Financial Reports Payroll Reports Run Reports</p> |
| <p>Jeff Access to run any Financial or Payroll data as well as the Payroll Masks. Additionally, Read Only access to HR screens for lookups.</p> | <p>Payroll Clerk</p> | <p>Financial Data Payroll Data All Accounts All Employees Read Only Employee Data Payroll Masks HR Masks Run Reports</p> |

| | | |
|---|--|---|
| <p>Lynn Access to all HR data, reports and masks.</p> | <p>Asst. Director of Human Resources</p> | <p>HR Data All Employees HR Masks HR Reports Run Reports Edit Reports</p> |
| <p>Neal Access to all HR data, reports and masks.</p> | <p>Teacher Representative</p> | <p>HR Data All Employees HR Masks HR Reports Run Reports</p> |
| <p>Lou Access to run all reports in the system.</p> | <p>Director of IT</p> | <p>Financial Data Payroll Data HR Data All Accounts All Employees All Reports Run Reports</p> |

| | | |
|--|---------------|---|
| Jan Access to develop reports against the entire system. | Report Writer | Financial Data Payroll Data HR Data All Accounts All Employees All Reports Create Reports |
| Beth All access to the system. | Analyst | All Access |

| | | |
|--|-------------------------------|--|
| <p>Larry Access to all of the Data in the system, most of the masks and all of the reports. Also, the ability to edit CDD reports.</p> | <p>Helpdesk</p> | <p>Financial Data Payroll Data HR Data All Accounts All Employees GL Masks Procurement Masks Payroll Masks HR Masks All Reports Run Reports Edit Reports</p> |
| <p>Terry</p> | <p>Principal (Truman H.S)</p> | <p>Financial Data Payroll Data HR Data Truman Data Budget Reports Personnel Reports Run Reports</p> |

| | | |
|-------|---------------------------------|--|
| Alan | Asst. Principal (Truman H.S) | Financial Data Payroll Data HR Data Truman Data Budget Reports Personnel Reports Run Reports |
| Sally | Secretary (Truman H.S.) | Financial Data Procurement Masks |
| Jack | Principal (Fairview Elementary) | Financial Data Payroll Data HR Data Fairview Data Budget Reports Personnel Reports Run Reports |

| | | |
|----------|---------------------------------------|--|
| Margaret | Asst. Principal (Fairview Elementary) | Financial Data Payroll Data HR Data Fairview Data Budget Reports Personnel Reports Run Reports |
| Fred | Secretary (Fairview Elementary) | Financial Data Fairview Data Budget Reports Run Reports Procurement Masks |
| Anne | Principal (Nelson M.S.) | Financial Data Payroll Data HR Data Nelson Data Budget Reports Personnel Reports Run Reports |

| | | |
|----------|-------------------------------|---|
| Rob | Asst. Principal (Nelson M.S.) | Financial Data Payroll Data HR Data Nelson Data Procurement Masks Budget Reports Personnel Reports Run Reports |
| Michelle | Secretary (Nelson M.S.) | Financial Data Nelson Data Procurement Masks |

Managing Change

The test of a well planned security setup is its ability to handle the changes that occur in an organization over time. However, setting up enough roles initially such that no new roles ever need to be created would result in too many roles to be managed easily. Therefore, the real challenge comes in, not just setting up discrete enough roles to be flexible, but knowing when to accept that some roles will need to be added, changed or broken into multiple roles later. Also, change can be tricky but knowing when and how to make a change can make the difference in creating a maintainable security setup over time.

The Procurement Manager needs to be able to post AP batches.

Solution: Create the "Post AP Batches" role, grant execute permissions to APOHBTDS and assign it to the Procurement Manager.

While Jesse is on vacation Ralph will be filling in for his Stores Inventory duties.

Solution: the "Stores Inventory Masks" role was granted to Ralph. Once Jesse returns it will be removed from his Role assignments.

To help alleviate some of the burden on Beth in IT the HR department got approval to hire their own dedicated Analyst to run utilities and write CDD Reports.

Solution: Once hired, the new position will be setup basically the same as the Assistant Director of HR except that "Create Reports" will be added to the role list. This will grant the ability to change and create reports but only within the HR folders. In addition to this a new Role is created named "HR Utilities" with just the HR utility masks and is assigned to the new position.

PY Manager can now run utilities

Solution: A new role is created named "Payroll Utilities" with the PY utility masks granted and assigned to the Payroll Manager.

One of the Procurement Clerks (Jesse) can write SI reports but only in the SI folders. This is an especially tricky change because up until now the organization was operating under the premise that all the folders would grant full access and the ability to edit or create would be controlled at the functional level.

Solution: At first it looked like the "Financial Reports" role would need to be removed and replaced by two new roles to accommodate this change. However, after further analysis they found that the only people in the organization who were editing or creating financial reports were people in IT that already had access to the "All Reports" role. Therefore, the "Financial Reports" role was changed to only grant Execute access and a new role "Edit SI Reports" was created with Read, Write, Update and Delete access on the SI folders.

The Procurement Clerks writing SI reports didn't seem to grasp the concept of reusability and when creating far too many categories. Also, the categories were including tables that were not limited to Stores Inventory.

Solution: To address this need the "Create Reports" role was change to only grant Read, Write, Update and Delete on "CDD Reports". Then, a new role "Create Categories" was added that allowed access to Read, Write, Update and Delete Information Categories and Table Definitions and that role was assigned to all the same people who had "Create Reports" other than the Procurement Clerks. The result was that now that clerks could only create reports and not categories and therefore all the subsequent report designs were limited to existing categories in the SI category folders.

Due to some miscommunication the Vanilla School District decides only IT can delete reports, information categories and table definitions.

Solution: To address this need the Delete permission was removed from the "Create Reports" role and a new role was created named "Delete Reports" and that role was only assigned to the people in IT who had "Create Reports" before.

1.1.14 Security Planning

Please keep in mind that this portion of the documentation is simply intended to provide some suggestions about how to plan for security within your organization. These are definitely no formal requirements of the software but merely one possible approach of setting up security.

Breaking Down Your Organization

Once you have a solid understand of Role-Based Security the next step is to start planning for its implementation. Its important during this phase to focus on the types of security needed without considering what specific Security Roles will be needed.

The needs of your organization can typically be broken into one of three categories.

- **Departmental Security Needs**

These are the needs that are specific to a particular group within your organization. This could be a formal departmental grouping such as the Human Resources or an informal grouping such as the staff entering Purchase Orders within a decentralized environment.

- **Cross-Department Security Needs**

This is the list of needs that cross departments and positions within the organization. Some of these can be fairly obvious such as the ability to Execute CDD Reports. While users may have a different selection of folders and reports available to them, the Application Functionality itself is common among a number of different users throughout the organization. Identifying these needs can result in the most effective roles because they allow a single role to fulfill the security needs of many different users.

- **Isolated Exceptions**

Within every organization there are going to be isolated pockets of access that don't fit into common roles. And while it's important to always look for commonalities between users it's also important to accept that there will have to be some special cases along the way. Keeping these exceptions to a minimum will help make your security more maintainable over time.

Step 1: Creating Security Needs List

The first step in identifying the types of security your organization will need is to create the shopping list of all of the discrete security needed throughout the organization. To help keep the list organized it's also helpful to start with some kind of top-level grouping. In the Vanilla School District they were Data Access, Menu Access, Report Access and Application Functionality. However, those may not be the correct groupings for everyone.

Example:

1. *Data Access*
 - a. Purchasing Data
 - b. Budget Information
2. *Menu Access*
 - a. Purchasing Data Entry
 - b. Accounts Payable Data Entry
3. *Report Access*
 - a. PO Inquiry Reports
 - b. Month End Reports
4. *Application Functionality*
 - a. Create Reports
 - b. Run Reports

Unless your data filtering needs are very simple it's probably best not to focus on the type of Common Security or the types of filters you will need for your users. Instead just focus on large areas of data access needs. Later on in this process how much or how little of a particular part of the system's data will be made available will be discussed in more detail.

As you work through this list, be prepared to break it out into more detailed items. For example, if the list of people who can run POUPPR (Purchase Requests) is not the same as the users who can use POUPRC (Receiving Information) you may need two different items to accommodate the actual needs of your organization to distinguish between the two groups.

Step 2: Checking for Gaps

Initially the security needs list can be created by a subset of "Power Users". However, the true test of the list will be applying it to all or at least a significant subset of the users in the organization. To do this, simply create a list of users and add the items from your shopping list to each user to see if your list accurately covers their needs. This may actually require the input of multiple people within the organization in order to make sure that you have accommodated the needs of each user.

Example:

| User | Position | Security Needs |
|-------|----------|---|
| Sally | PO Clerk | Purchasing Data Purchasing Data Entry PO Inquiry Reports Run Reports |

During this process be prepared to add or break out items from your list of security needs. For example, as you are assigning the Application Functionality it may become apparent that some of the people on the list can delete reports, but some cannot. In that case "Create Reports" will essentially be the same as it was before but a new item "Delete Reports" will be added so that it is now identified as a separate need.

Step 3: Data Access Needs

Once you have a first pass at your Security Needs list you can start discussing the data access needs in a bit more detail. The reason for putting this off initially is to prevent the initial security list from becoming overwhelming. This stage might require you to look at your needs from a slightly more abstract viewpoint than you are used to historically.

Instead of creating a list of Data Access needs for each person it might be useful to look at how data is restricted within your organization. For example, a school district such as the Vanilla School District may need to restrict each school's principal to just the GL Key Background Part

that is set aside for their school. However, adding each school's background part at this stage in the planning is probably a bit redundant. Instead, just noting that one of the needs is a data restriction at that level for the different schools is sufficient.

Example:

1. Data Access
 - a. All Financial Information
 - b. All Employee Information
 - c. Financial Information Filtered by Fund
 - d. Employee Information Filtered by Location (worksite)

Not all of the Data Access will need to be flushed out during this stage. It can be assumed that over time changes in departmental responsibilities or special projects will necessitate distinct Security Roles to meet those needs. This stage should simply focus on the general Data Access needs and the more commonly used types of restricting that access.

Planning for Your Current Needs

While the first pass at your list of security needs is a good start it is most likely not going to be your final version of that list. In fact, over time the changing needs of both your organization and the software will necessitate a revisiting of that list and how accurately it fits your needs. Older items will become obsolete, new needs will be discovered and, in general it will continue as an organic process. With this first pass complete its time to start figuring out how to create Security Roles to meet your current needs.

The Law of Diminishing Returns

Your first instinct may be to create a Security Role for every individual security need you have already identified. While this would certainly provide a fair amount of flexibility it would also create a significant collection of Security Roles to manage. For example, imagine creating a Security Role for every folder in IFAS and then creating the proper security assignments for a user capable of running all of your CDD Financial reports. First, this would create a rather lengthy role assignment list to be managed per user. Second, anytime a new CDD folder was added a new role would also need to be added.

On the other hand, just creating one role per top-level CDD folder may not meet your needs very well. Imagine that you have one top-level folder for each subsystem in IFAS and as a result one top-level Security Role for each of those folders. Now, you are asked to grant access to a sub-folder of the "GL Reports" named "Budget Reports" to an auditor so you create a role for that folder. Then a few days later it turns out they also need three of the folders below the "Month End" folder within "GL Reports". Do you create three more Security Roles? What if next month a different auditor needs the same collection of roles? In this case it would make more sense to create one role named "Auditor Reports"

that included any of the folders that made up that particular reporting need. This also accommodates long-term changes such as a new folder being identified as useful to the audit process by only requiring that one role to be modified.

The challenge in setting up your security is to find the efficient middle ground between too few roles to accommodate change and too many roles to manage. Since this will vary from client to client there are no hard rules for finding this middle ground. Instead, this document will try to assist you in finding a good match for your needs.

Menu Security

One of the places where a single role might fulfill multiple security needs is in the area of Menu Security. For example, the act of processing Payroll involves multiple masks. In all likelihood, the masks are almost always granted in a group and a subset of them is rarely required. In this case, it might make perfect sense to create a "Payroll Processing Menu" Security Role that includes all of the IFAS Menu Masks required to complete that activity.

Data Access Security

Before trying to setup Security Roles on IFAS data it can be useful to break your data security needs into three areas.

1. Common Security Needs

Identifying those aspects of your data security needs that can be addressed by Common Security filters is important because these filters have the biggest impact on data access throughout the software. In many cases access to the Common Security items is required to view the data in other tables at all. For example, without access to "Ledger Security" a user is unable to view any data in a table linked to that common item.

2. Individual Table Filters

Some of your users will need access to a subset of the data in a particular table. For those tables with links to Common Security items this can be handled with that type of filtering. For some of the IFAS tables individual filters may need to be setup to allow a reduced portion of that data to be accessed based on the role assignments. Its important to identify these needs up front because, based on the concept of Inclusive Security, it is possible for one Security Role to override the filter of another role by providing unfiltered access to the table.

3. Unfiltered Data Access

A significant number of the tables in the software are going to fall into this category. In some cases this will be because these tables are storing data that is either already filtered as a result of a Common Security link or your users simply require all or nothing access. For example, the tables storing the HR codes are frequently granted access to anyone with access to the other HR tables without any type of filter.

Once you have the list of Data Access security needs grouped based on the type of restrictions they require the next step is to look for logical groupings within those types. For example, in the Vanilla School District the secretaries at each of the schools needed to be able to enter Purchase Requisitions as well as report against, not only their Purchase Requisitions, but the status of payments to vendors. Therefore, in their

situation there were a significant enough number of users requiring access to both Purchasing and Accounts Payable information to warrant a Security Role designed to provide what the district felt was standard data access for Purchasing Data Entry. Whether or not to combine both the Data Access needs with the Menu Access needs in this one role is up to each organization and will most likely be influenced by whether or not there are some users who will need this same Data Access but have different Menu Access needs.

However, because the secretaries would require Account Key (Common Security on GLK_KEY_MSTR) limitations based on the school they were assigned it did not make sense to extend that role to include Common Security on the "Account Key" security item. Instead, for that need they would create a grouping of Security Roles to address the Account Key security needs for each of the schools. This same Account Key class can also be used to provide the Common Security necessary for the Principals at each site to run CDD reports on their site's financial information.

A combination of the one role providing access to perform "Purchasing Data Entry" and the Security Role providing them the Account Key access for their individual school site will make up their individual data access needs. It will also provide a method of answering the "Who can perform Purchasing Data Entry?" question that may come up at a later date because that can be answered by simply retrieving a list of all of the users with that Security Role assigned to them.

Example:

| Role Name | Nodes | Description |
|-------------------------|---|---|
| Financial Data | Accounts Payable Data (RWUDX) Bank Reconciliation Data (RWUDX) General Ledger Data (RWUDX) Encumbrances Data (RWUDX) Person/Entity Data (RWUDX) Purchasing Data (RWUDX) Stores/Inventory Data (RWUDX) Common Security Accounts Payable Bank Reconciliation General Ledger Budget Version Security Ledger Security Object Code Security Purchasing Stores Inventory | This role provides access to the Financial tables. |
| Payroll Data | Payroll Data (RWUDX) | This role provides access to the Payroll tables. |
| Human Resources Data | Human Resources Data (RWUDX) | This role provides access to the Human Resources tables. |
| All Accounts | Common Security General Ledger Account Key Security (RWUDX) | This role provides unrestricted access to financial data. |
| All Employees | Common Security Human Resources Employee Definition | Provides full access to all employees. |
| Read Only Employee Data | Common Security Human Resources Employee Definition (R Only) | This role provides a read only view of the Employee data. |
| Truman Data | Common Security | This role provides access to this school's data. |

In the example above the logical groupings have become the general security needs of multiple users such as those needing access to Purchasing Data Entry along with a group of roles more specific to each school site. These become useful groupings because it not only exploits the concept of layered security but it also allows one role to be used by multiple people even though they have different common

security. This can also make it easier to manage change within an organization. For example, if a secretary in the sample organization were to move from one school site to another the only change necessary to that person's security setup would be to remove the old school's financials role and replace it with the new one. All their other responsibilities would be the same.

Application Functionality

Application Functionality is probably the area of Role-Based Security most likely to result in a number of different roles with very few actual items granted within that role. For Example, the ability to "Assign PO Number" is handled as a functional item within the Purchasing subsystem. This may not be a function that is given to every user who can enter Purchase Requisitions at an organization. As a result, a single Security Role may be created with this access only granted to it.

Another example would be CDD Functionality. In that case the "CDD Reports" item in the Security Structure below Click Drag & Drill's functionality might even be broken up into its different types of permissions. The ability to Execute a CDD Report may be granted to a large number of users but the ability to Edit, Save or Delete a CDD Report may not be nearly as widespread.

Security Reporting

One last consideration in setting up your security is whether or not it is reportable based on the needs of your organization. While generating a report from some of the setup within a role is possible, it can also be fairly complicated and possibly prone to reporting errors. This issue is best resolved by considering some of the reporting needs up front and simply putting commonly queried access information into separate roles so that it is much easier to generate a list of users who can or cannot perform a certain role within your organization.

For example, the ability to "Assign a PO Number" might only be granted to a subset of users and being able to identify that list of users quickly could be very beneficial. Balancing the need to report on every kind of security against creating too many Security Roles to be managed easily will need to be considered by each organization.

Don't Force the Model

It would be nice if all of your users fit into well structured groups and a common set of Security Roles fit everyone's need. That is probably not going to happen. There will always be users who for one reason or another need some special access to some part of the system. Not all Security Roles need to be reusable and certainly not all users within a certain department or position type need to share the same security. By using a combination of common roles and a set that addresses the exceptions should help find a balance that meets the larger need without unnecessary complexity.

Examples of Special Cases

- Unusual Utilities

Its possible that some utilities should only be accessible by a small number of users. Creating special Security Roles to address this is a good use of Role-Based Security.

- Testing Reports

For the purposes of testing one or more CDD Reports in your organization before they are made available to everyone it may be necessary to create a role for "CDD Report Testing" where access to reports being actively worked on are stored and can be made available as needed to users testing the reports.

Planning for Changes within Your Organization

Change is inevitable and changes within your organization can have a significant impact on security. However, planning for known types of change as well as being flexible with your security definitions can help minimize the impact of change. The following section is intended to help plan and address the issue of managing changes to security.

Movement within the Organization

Some movement within an organization can be predictable and somewhat easily managed. Certainly the example of a secretary moving from one school to another is probably both common and simple to address. The duties of that position most likely stayed the same and the only significant change would be the access to the data. It is a good example though of a way to predict such movements and plan for them.

Slightly more difficult to predict are those movements within the organization that are not quite as simple. For example, it's altogether possible that someone within your organization will move to a new position but still retain some of the duties from their prior position even if that is only for a short period of time.

This is an opportunity to really maximize the concept of Security Role layering. Meaning, if security is defined with a reasonable level of granularity then it would be possible for that person to simply have both the roles from their old position and a set of role assignments necessary for their new position at the same time. Remember that Inclusive Security will always add to the user's security and will never take away from it. And should the duties of their former position no longer be required the older roles would simply be stripped away and with them the Menu Masks, Data Access, etc. going away as well.

Tips for Accommodating Movement

Look for commonly restricted data filtering. For example, if departments are restricted based on a commonly filtered GL Background Part, consider keeping that security separate to accommodate both movement between departments and the need for one individual to have access to multiple departments.

Avoid creating a role to fill all of the access needs of a single position. For example, creating a single role for a Data Entry Clerk position will create an all or nothing role assignment for that position and possibly restrict its usage as people move around the organization.

Ongoing Training

Over time your organization will most likely receive additional training and you may find that some users will need increased access to the software to make use of that training. For example, some of your staff may attend CDD classes where they will learn to create or at least change CDD Reports. You may not want them to be able to edit all of the reports in the system, but with a combination of the roles already granting them CDD Folder access, and the addition of a new role granting them edit access to the CDD functionality, they would now be able to edit the reports they already have access to in the system.

Tips for Accommodating Ongoing Training

- Keep access, such as the ability to edit CDD Reports, separate from the actual CDD Folder access. This way common security, such as the ability to edit a report, can be granted without having to reconsider what folders they have access to each time.
- Don't always assume that you will need a new role or set of roles for each new subsystem that you will be adding. When implementing a new module you should be considering the different roles your staff will be fulfilling as it relates to that new module and how that crosses over to other roles already being filled. For example, the ability to run Stores Inventory Inquiry reports may also be needed by people entering Purchase Requisitions.

Software Changes

With each release of the software, the database tables, menu masks and application functionality that make up IFAS are bound to change. Therefore, the Security Structure is likely to be enhanced with each release. Most of this can't be completely planned for but there are some steps that can be taken to help minimize the impact on your Security Roles.

Tips for Minimizing the Impact of Software Changes

With the obvious exception of the Application Root node, grant access to the parent nodes within subsystems as often as possible. One of the strengths of the Security Structure's hierarchy is that child nodes can be set to derive their security from a parent node. Since the derived state is the default any new nodes inserted will also be derived. For example, any new tables added to a subsystem will inherit their access from the parent nodes. If the data parent for that subsystem were granted access and the tables below it left derived then the new table would simply inherit the permissions of that parent security object and that role would continue working after the update.

Check the release notes for new security items. Typically we try to post any new security features in the release notes. If there are references to changes, that may necessitate the use of the "Manage Security Structure" web client screen to rebuild the Security Structure to include these changes.

Special Cases

From time to time your organization will have the need to provide short-term access to parts of the software for various reasons. A staff member may be assigned to an internal project for a set period of time. One person may be filling in for another while they are out of the office. There are many different reasons. Some of this can be planned for, but much of it is going to have to be addressed on a case by case basis.

Tips for Accommodating Special Cases

As discussed in the Vanilla School District example one of the special cases is employees covering for each other. Consider creating special Security Roles designed for recurring issues such as staff members being out of the office. Since it may not be the same person providing coverage each time this can be an effective way to make the necessary assignment quickly and removing it once it is no longer required.

Consider creating temporary Security Roles for special projects. If the project is unlikely to come up again this might be the perfect place to create a temporary role. Once the project is done it can simply be deleted from the system and along with it the access it granted.

1.1.15 Implementing Security

Security Role Creation

Once you have a solid understanding of Role-Based Security and have developed a plan for your organization the next step is to begin creating the initial Security Roles based on that plan.

Existing Roles

Unless you are setting up security for the very first time you will have existing roles to contend with. These roles will most likely come from one of two places. First, if you have been using a version prior to 7.9 you most likely had Security Roles in place to provide access to CDD, Documents Online or other PC Applications. Depending on how you have setup these roles they may or may not still meet your organizational goals.

Second, if you chose to perform the Security Rollover when first installing 7.9 then you may have a series of roles with the naming convention of "USERID__R" where the "USERID" was an existing IFAS User ID. Due to the differences between the Inclusive nature of Role-Based Security and the Exclusive nature of its predecessor the only way to create a role structure truly representative of the old security was to enclose all of the user's previous security into one role.

In both cases the roles are currently being used by at least some portion of your users. It is definitely not recommended that you remove them or their assignments during normal working hours. Instead, you may need to leave them in place until the new roles have been created and assigned and then plan on removing them as part of a general clean up at a later date.

Security Role Naming Convention

To help you keep track of your roles it's a good idea to develop a naming convention. Roles are identified by two pieces of information. First, the Role ID provides the unique identifier for the role. Second, the Role Title provides a description of the role. The Role ID can be up to 16

characters long and the Role Title can be up to 30 characters long. Between the two of these you should be able to create a naming convention for your roles that makes it easier to identify their purpose without necessarily digging into the details.

If you assume that the Role Title will provide a meaningful name for your role you can then utilize the Role ID to help give an indication of what type of role you will be creating. In the Vanilla School District example they chose to group roles by Data Access, Menu Access and Application Functionality. So the naming convention they would have selected would have been based on those types. For example, each Data Access Security Role may have included "DATA" in the first four characters of each of those Role IDs.

Using a good naming convention will also make it easier to distinguish between your new roles and those that existed in previous versions or as part of the Security Rollover. Whatever naming convention you select be sure to document that convention to help ensure that it will continue to be used as your security is maintained over time.

Manage Security Roles

The screen used to create Security Roles in IFAS is the Manage Security Roles web client screen (NUUPSR). This will involve entering the ID, Title and then assigning the security for that role by altering how that role will impact the Security Structure. Please consult the help for that screen for more specific information about its usage.

At this stage since the roles are not actually assigned to anyone you will not need to use the "Rebuild Security" tool after creating the roles. Technically you will only need to run that tool when modifying an existing role and even then only if it's actually assigned to a user.

User Creation

The setup of users in the system involves a lot of other decisions such as default printers, hours codes, etc. This document is only discussing the Security Role aspect of user setup. The Manage Users web client screen (NUUPUS) has a tab specifically for Security that will allow you to check those roles you wish the user to have assigned to them. Please consult the help for that screen for more specific information about its usage.

Once you have assigned one or more roles to a particular user you may want to go ahead and launch the "Rebuild Security" tool on that screen. This will put request into the Workflow Queue to Rebuild Security for that one user so that the information necessary for the software to implement their security is generated. Until that user's security has been rebuilt the changes will not fully take effect.

In addition to using this screen to lookup a particular user's security you can also use it to locate the list of users assigned to a role by putting the screen into Find mode, checking the Security Role you are interested in and then applying the Find criteria. The list of users returned will all be users with that role assignment.

Assigning Additional Roles

The Manage Users web client screen (NUUPUS) is useful for setting up one user at a time, but it's not quite as effective when you are setting up a new role that you wish to apply to multiple users at the same time. In that situation you can use the Assign Security Roles web client screen (NUUPSA) to assign multiple users to a role all at once. Please consult the help for that screen for more specific information about its usage.

Once you have made your assignments remember to use the "Rebuild Security" tool on that screen to rebuild the users impacted by that Role Security assignment. This screen can also be used to quickly determine which users have access to a particular Security Role without the need for a separate report.

1.2 Security System Features

IFAS security provides the flexibility to customize the system to best meet your organization's needs. Additionally, once the system has been configured, the designated System Administrators at your facility can easily adjust and revise security status as necessary. What's more, SunGard Bi-Tech's IFAS provides you with a powerful security system that controls access from "Class" to "Value" levels.

The major features of the IFAS Security System are listed below:

- Ability to assign unique passwords to individual users and control password expiration and time-out parameters.
- Ability to restrict user access to only certain printers, and further to define the days and hours when the user may access the system.
- Ability to assign maximum job priority the user is allowed to specify when running jobs and specify if the user is allowed to schedule jobs.
- Ability to restrict user access at the row, field, and function level.
- Ability to define job running capabilities within subsystems.

1.3 Getting Started

1.3.1 Assign Security Roles Administrative Tool

The "Assign Security Roles" screen can be found within the Administrative Console, in the Security Admin / Setup section. Select this screen by double-clicking **Assign Security Roles** in the Options Panel. This will bring up the view shown below:

Administrative Console

Options Panel

Page 1 of 1

| RoleId | RoleTitle |
|----------|-----------------------|
| ALL | ALL - All Access |
| SYSADMIN | System Admin |
| TEST SEC | Testing Role Security |

Assign Security Roles Manage Security Structure

Assigned Users

| | |
|---------------------------|-------------------------------|
| DGWMGR (DGWMGR) | DOC (Doc Team) |
| DOUGD (DOUG DZIEDZIC) | DOUGH (Doug Hasse) |
| EDR (Ed Rose) | EQADMIN (EQADMIN User) |
| HEIDI (Heidi Genasci) | JAMESF (James Flint) |
| JENNY (JENNY GOWER) | JENNY1 (JENNY1) |
| JENNY2 (JENNY2) | BRENDA (Stowe, Brenda) |
| JENNY3 (JENNY3) | MS000005 (STEVENSON, LO...) |
| JENNY4 (JENNY4) | JUSTINO (Justin Onstot) |
| LEANN (Leann Doussett) | MICHELLE (Michelle Stenquist) |
| ROBERTC (Robert Cabral) | VICKIW (V. Webster) |
| WALTER (Walter Crane) | WFUSER01 (WF User 01) |
| 930 | MS000001 (STENQUIST, MIC...) |
| MS000003 (SMITH, HENRY H) | |

Page 1 of 2

| UsId | Name |
|----------|--------------------|
| 100 | Smith, J Opal |
| 71600 | Timecard, Larry |
| 71610 | Splt, Andrew L |
| 71620 | Splt, Tommy L |
| 74510 | Hourly, Sam |
| 74511 | Hourly, Linda |
| 75000 | Fairbanks, Larry L |
| 75001 | Splt, Tommy L |
| 790BMS | TEST, 790 |
| AMEET | Ameet Patil |
| AMIT | amit, patil |
| ANURADHA | Anuradha Bhimirec |
| AQBRENDA | BRENDA, AD |
| AOTES | PERSON, AOTES |
| APSUPVSR | Accounts Payable |
| ATBMS09 | USER, TEST |
| ATBMS11 | USER, TEST |
| ATBMS12 | USER, TEST |
| ATBMS13 | USER, TEST |
| ATBMS14 | USER, TEST |
| ATBMS15 | USER, TEST |
| ATBMS16 | USER, TEST |
| ATER111 | TEST, TEST |
| ATER12 | TEST, TEST |
| AT_ER_01 | Test, Ed |
| BALA | Balamurugan Shar |
| BEZ | Bez |
| BRENDA | Stowe, Brenda |
| BSIDBA | BSIDBA |
| CDDTEST | CDD Security Test |
| CDD_ALL | CDD User - All Acc |
| CDD_GL | CDD User - GL Se |
| CDD_GRP5 | CDD User - Accou |
| CDD_K0 | CDD User - Accou |

EntityList

Admin Plugins

Status

The **Assign Security Roles** screen is used to assign Security Roles to IFAS users.

This screen consists of the Options Panel containing the EntityList, a list of the current Role ID's and Role Titles, the Assigned Users and a panel containing all existing User ID's and Names.

Adding Users to a Role

To add an IFAS user to a role first select the desired role from the Entity List on the far left. Then locate the desired IFAS User using the "IFAS Users" list on the far right. The IFAS Users list can be paged using the forward and reverse arrows at the top of the list much like a standard entity list. Double clicking on a UsId from the list will add that user to the highlighted role. Once there are pending changes to the Assigned Users list, the "Save" button will become active. It is necessary to click the save button prior to exiting the **Assign Security Role screen** to ensure any changes have been saved. Exiting the **Assign Security Role** function or selecting a new Role for update with out saving the changes to the existing role will result in the changes/additions being lost.

Removing Users from a Role

To remove an IFAS user from a role first select the desired role from the Entity List in the far left panel. Locate the desired IFAS User from the "Assigned Users" list in the center of the screen. Selection is performed by clicking on a user. Multiple users can be selected by holding down the "Control" key to select specific users or the "Shift" key to select a group of users. Once users are selected the "Remove" button in the toolbar will be activated and clicking on that button in the toolbar will remove those users from the role assignment. Once there are pending changes for the role assignments the "Save" button will become active. Exiting the **Assign Security Role** function or selecting a new Role for update with out saving the deletions from the existing role will result in the changes/deletions being lost.

1.3.2 Column Level Security

Overview

Column level security is designed to allow restrictions on individual fields on 7i screens. You first must choose which columns on each table that will be controlled by security (by default, none are selected). In each Role, you then will have the ability to restrict the column by Read, Write or Update access (the default is derived, meaning it inherits the security its table). If you revoke read access, the corresponding field on the 7i screen will display a '~' sign in the place of the hidden data. The field will also be disabled in Find mode. If you revoke Write access, the field will be disabled in Add mode. Revoking Update access will disable the field in update mode. Any combination of Read, Write and Update access levels are permitted.

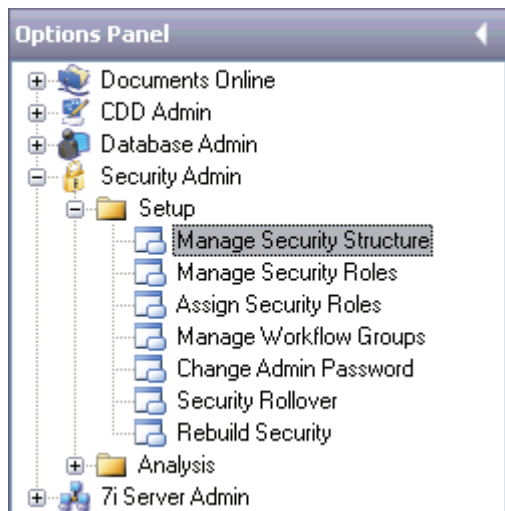
This security feature can be used to hide sensitive information from certain types of users, while allowing them to see the rest of the record. Column level security can also restrict users from changing portions of the data displayed to them in 7i screens. This capability should be used sparingly, as many fields are required by IFAS for proper business rule operation.

Column Security Setup

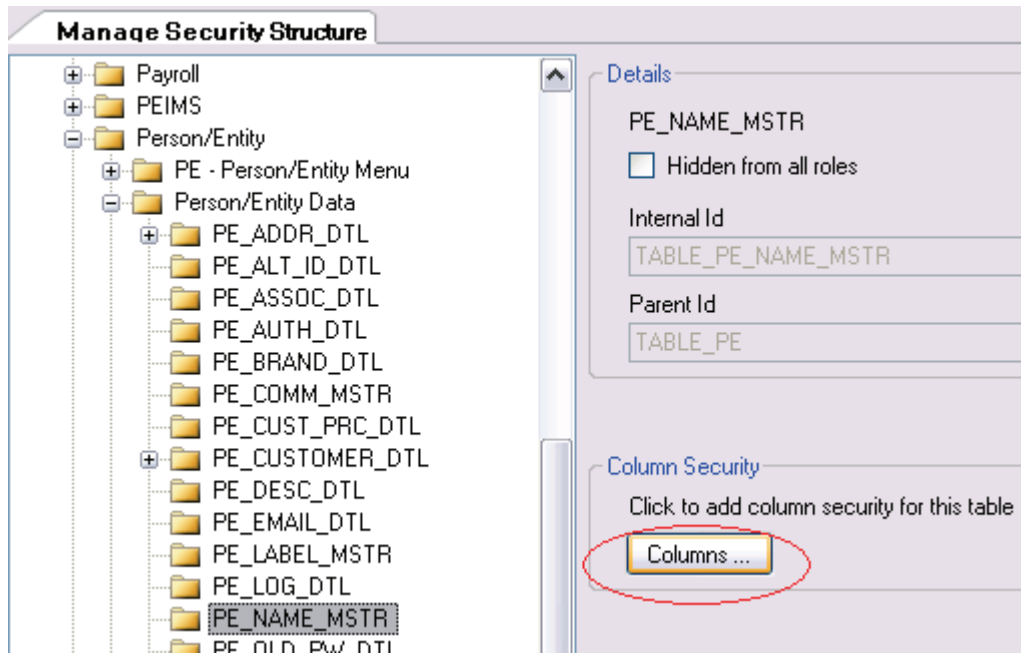
The following example will show how to hide the SSN on PEUPPE.

Security Structure

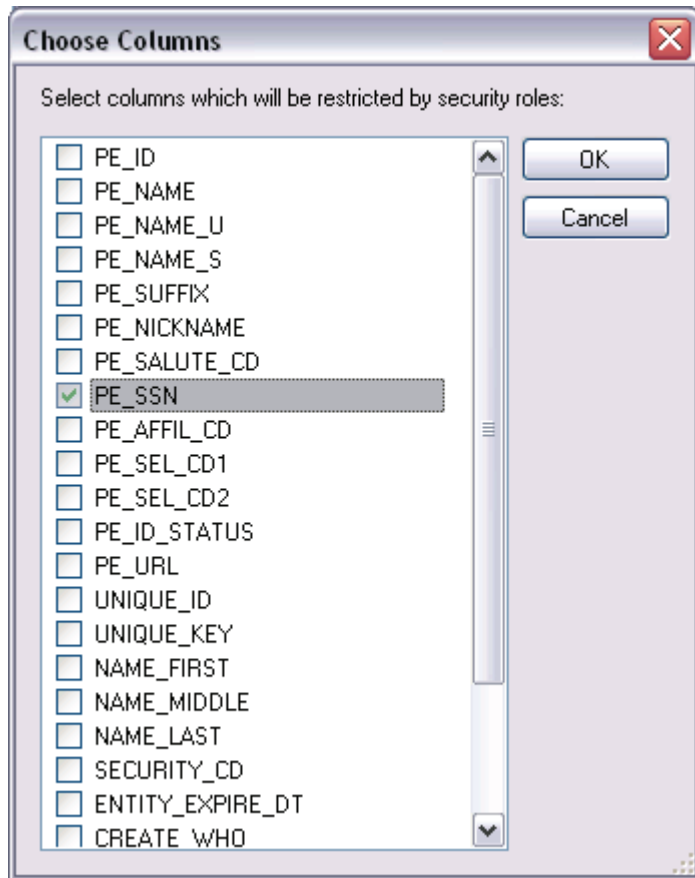
First run the Admin Console and choose 'Manage Security Structure' from the Options Panel:



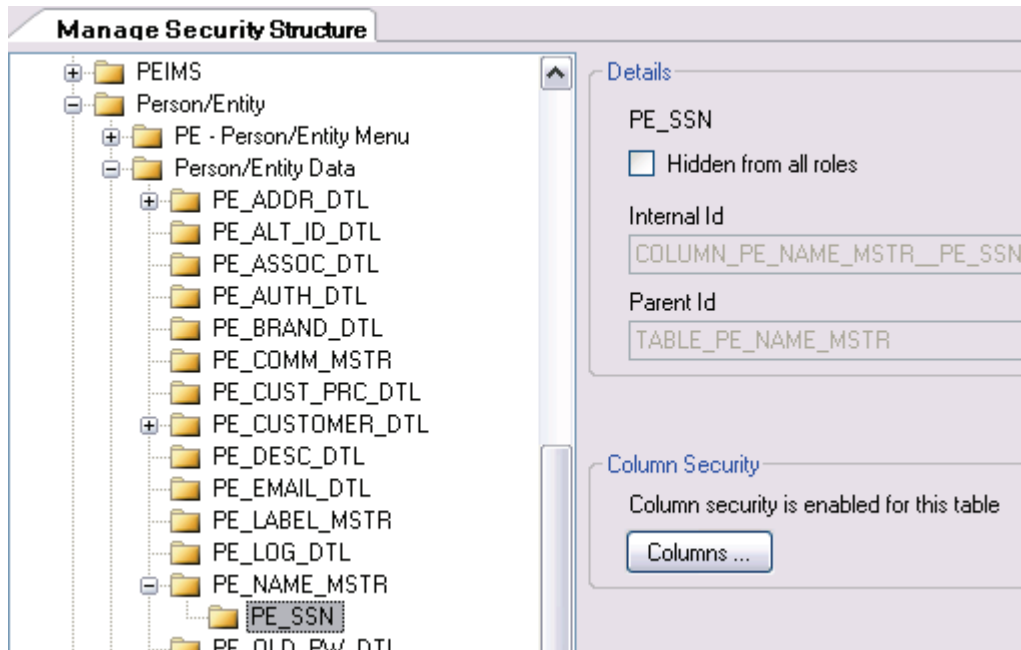
Next, we will need to navigate to the Person/Entity subsystem and locate the PE_NAME_MSTR table. Column security is not enabled on any tables by default. You must select each column that you wish to have the ability to control from each Role.



Next, click on the 'Column Security' button to bring up the 'Choose Columns' dialog to allow you to select the columns you wish to control security on. For this example, we are going to choose 'PE_SSN' and click the 'Ok' button.



After the Admin Console automatically updates the security structure, you will now see your newly added column under the PE_NAME_MSTR table.

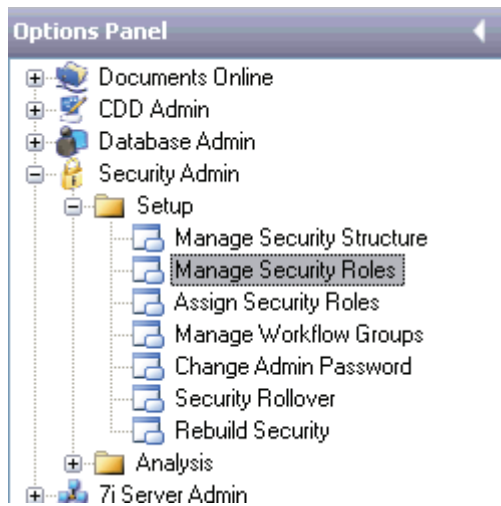


Now click the 'Update Servers' button to update the 7i Servers with the new changes made to the security structure. If you choose not to do this now, when you close the Manage Security Structure screen, it will prompt you to update the servers then.



Manage Security Roles

With the new security structure changes in place, we are ready to restrict access on the PE_SSN in a Security Role. Run the 'Manage Security Roles' screen from the Admin Console options panel.



Find the role you wish to modify and then locate the PE_SSN column under the PE_NAME_MSTR. The default behavior for all columns is derived. Uncheck the derived box and the Read, Write and Update permissions. This will restrict the user from viewing or modifying any data in this column.

Manage Security Roles

Role ID: Role Title:

| Security Object | Derived | Execute | Read | Write | Update | Delete |
|---------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Payroll | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Person/Entity | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Person/Entity Data | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PEIDINFO (PE Vendor Info) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PEIMS_CODES () | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_ADDR_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_ALT_ID_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_ASSOC_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_AUTH_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_BRAND_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_COMM_MSTR | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_CUSTOMER_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_CUST_PRC_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_DESC_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_EMAIL_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_LABEL_MSTR | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_LOG_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_NAME_MSTR | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PE_SSN | <input type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Once you have completed making your changes, click the save button to record your changes.



We now need to update the 7i servers with the changed Role information. This may seem redundant because we performed the same step in Setting up the Managed security structure. If you are just changing the Security Role, and not the structure, than this step needs to be performed. It is also possible that a user has logged in since you updated the 7i server last and now their old Role is cached and updating the server is again required to ensure your changes took place.



Column Security Usage

Now run PEUPPE and login as the user assigned to the column - restricted role. In our example we would have assigned the role COLUMN_TEST to a user and then rebuilt the user's security.

PEUPPE now shows a '~' in the SSN field and it has been disabled. The user is not allowed to click in the field or see the actual data behind the '~'. The SSN field will also be disabled in QBE if the user does not have read access. This will prevent the user from 'guessing' at the value in the field.

Allow Read-Only Access

| | | | | | | | |
|--|---------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | PE_LOG_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PE_NAME_MSTR | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | PE_SSN | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | PE_OLD_PW_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Column Security Limitations

Now lets use the Mange Security Role screen to change the Role to have only read access on the SSN. Be sure to save the changes, and update the servers from the Manage Security Structure screen.

Now when we run PEUPPE (you must close any open 7i screens and re-login to see the new changes), our user is able to view the SSN data, but not allowed to edit the field. This is because they still do not have Update access to that column. This is helpful if you wish to protect data and only allow a few select individuals make changes.

In general, any field on any 7i screen can be restricted with Column Level Security. There are however some exceptions.

Areas Where Column Security are Not Supported

All of the IFAS WebClient screens (such as NUUPUS and NUUPDF) do not currently support Column Level security.

Account Control

The Account Control that appears on many 7i screens (such as POUPPR and APOHBTUB) only works at a high level based on Ledger. If the ledger column on a table is restricted and it is part of the account control, then the entire account control will be disabled or enabled accordingly. Due to the importance of the ledger drop down on the screen, it cannot be replaced with the '~' character when read access has been removed.

Security on the key/object is used to control whether or not the data is displayed to the user or replaced with the '~' character to hide the data. It will not control the write/update security – that is handled by the ledger column as described above. Below is a screenshot of the OHB_BATCH_DTL. This is the child records table on APOHBTUB. This setup will disable the entire account control, and replace the key/object with the '~' character.

| Security Object | Derived | Execute | Read | Write | Update | Delete |
|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Application Root | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Accounts Payable | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Accounts Payable Data | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| AP_IMAGE [Image based AP 3way m... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| AP_IMG_ITEMS [item splits for ap ima... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| OHA_ACCT_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| OHB_BATCH_DTL | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| OH_GL_GR | <input type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| OH_GL_KEY | <input type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| OH_GL_OBJ | <input type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Column level security is not applied to the ability to display and enter account splits on POUPPR.

Exceptions

Dashboard applications (portals) do not support column level security. Examples are: Employee Online, Timecard Online, Applicant Online, Admissions Online, etc.

7i Screens that are 'Custom Controller' screens – meaning that they have special behavior to allow a richer User Interface, do not support column level security for any of their displayed fields. Below is a list of these screens:

APOHBTUBEX
 ARBTCRBL
 ARBTCRIC
 ARCSCOIC

BKCSMC

BKUPCA

BKUPMM

BKUPRC

GLCSPO01

HRPYPADS

PBUPEMDS

POUPRC

SIOECR

1.4 Flow Diagram

Under Construction

1.5 Basic Steps

Under Construction

2 Setup

2.1 Basics

2.1.1 Terms and Definitions

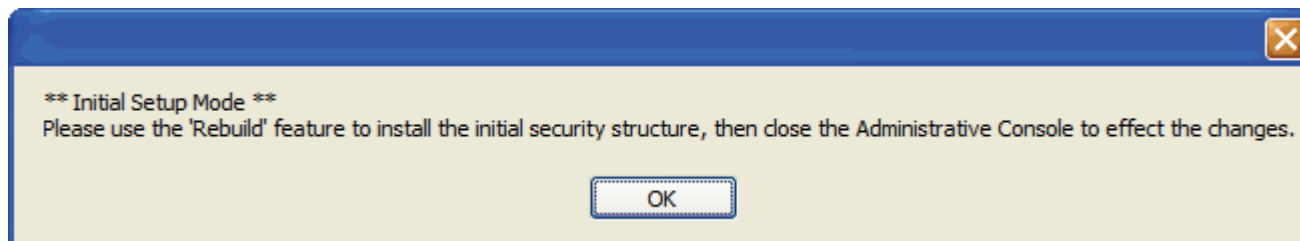
Under Construction

2.1.2 Concepts

Initial Security Configuration

In order to configure IFAS security, an IFAS user with full permissions must be available. This is necessary because the roles which control IFAS security also control access to the security configuration tools. To resolve this situation during the initial installation and configuration phase, a BSI user has been included that is loaded when the IFAS database is installed. The BSI user should be used for initial security configuration.

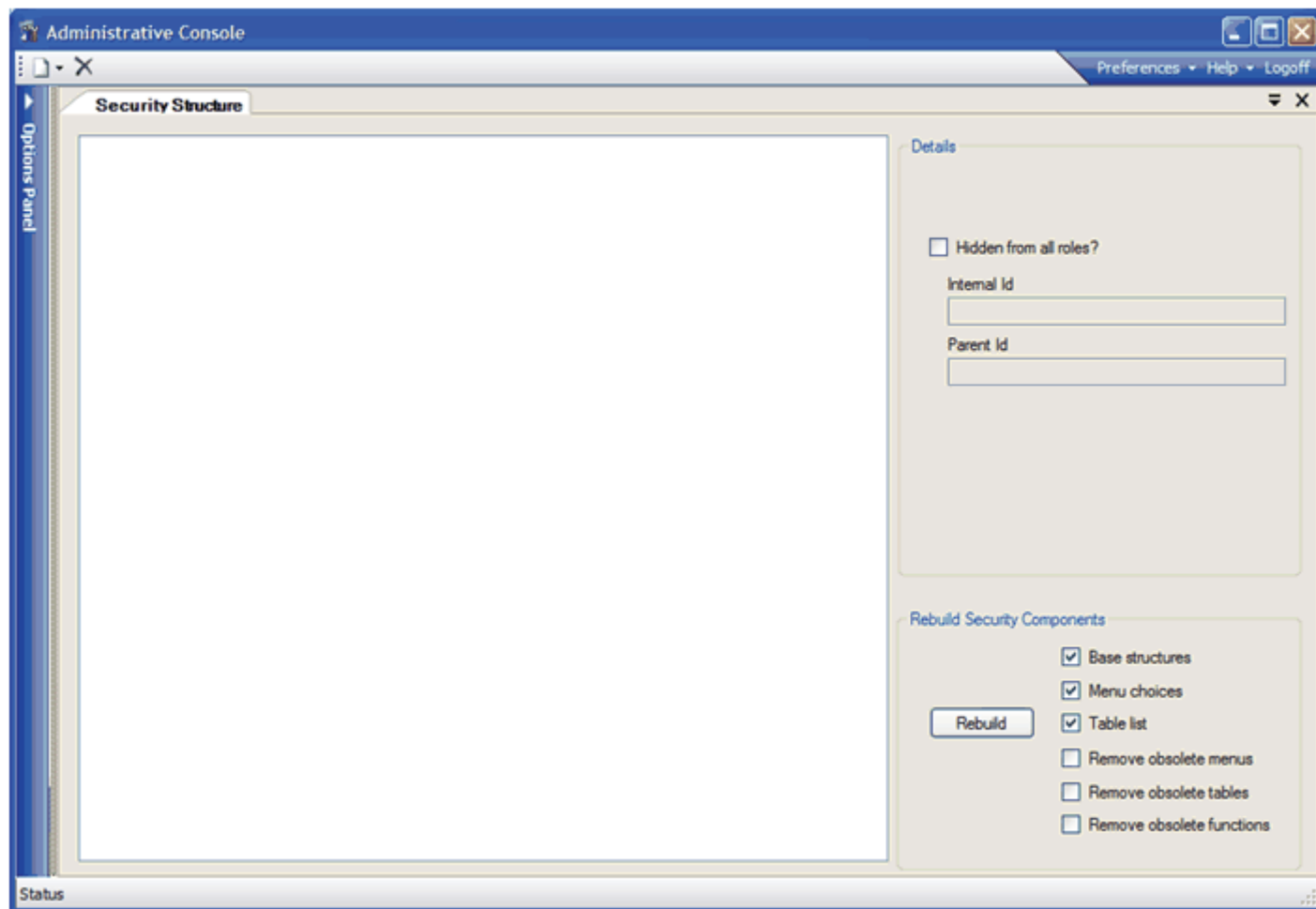
To perform these initial configuration steps, please run the Administrative Console. Choose the connection which was previously created during the install process and log in. The BSI user is recommended for this initial configuration, although this is not a requirement. The first time the Administrative Console is run, the following "Initial Setup Mode" prompt will appear:



Click "OK" to continue to the main console screen.

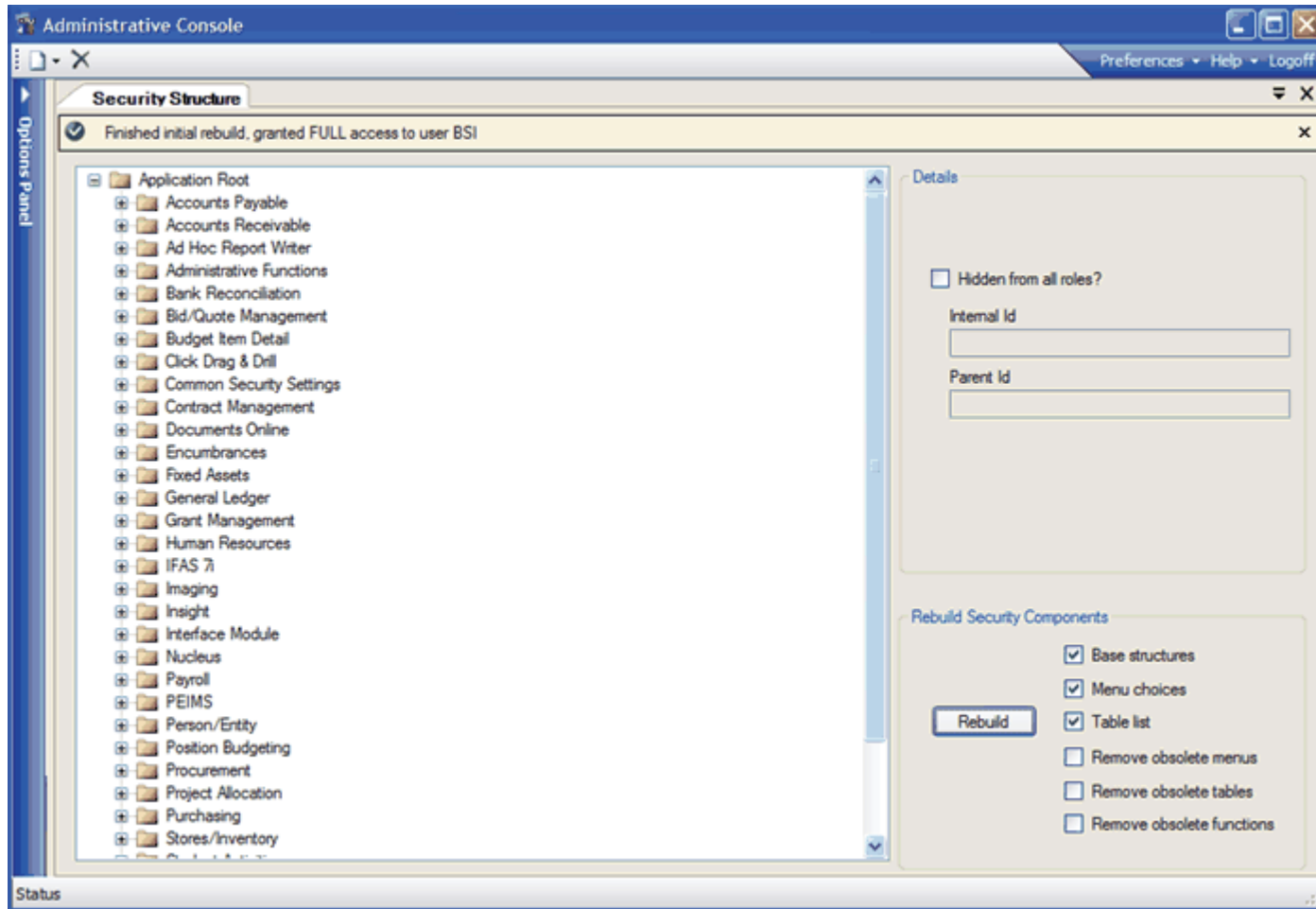
Initially no security objects are defined, so all users effectively have no security access. To remedy this, the initial setup mode will bypass the security requirements and allow the security objects to be installed.

In the initial setup mode, the "Manage Security Structures" plugin will automatically be started, presenting the following screen:



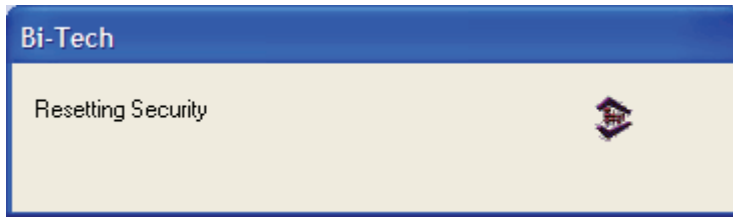
No security objects exist at this point. To install the security objects, click the "Rebuild" button in the bottom right corner of the screen. The default check boxes should be selected (Base structures, Menu choices, Table list). These options are described in the Administrative Console user manual.

The Rebuild operations may take several minutes. After this the security structure tree will reload and the current IFAS user will be granted permissions to a FULL access role. This will allow the user to perform additional configuration steps as needed, including the creation of users and the management of security roles for those users.

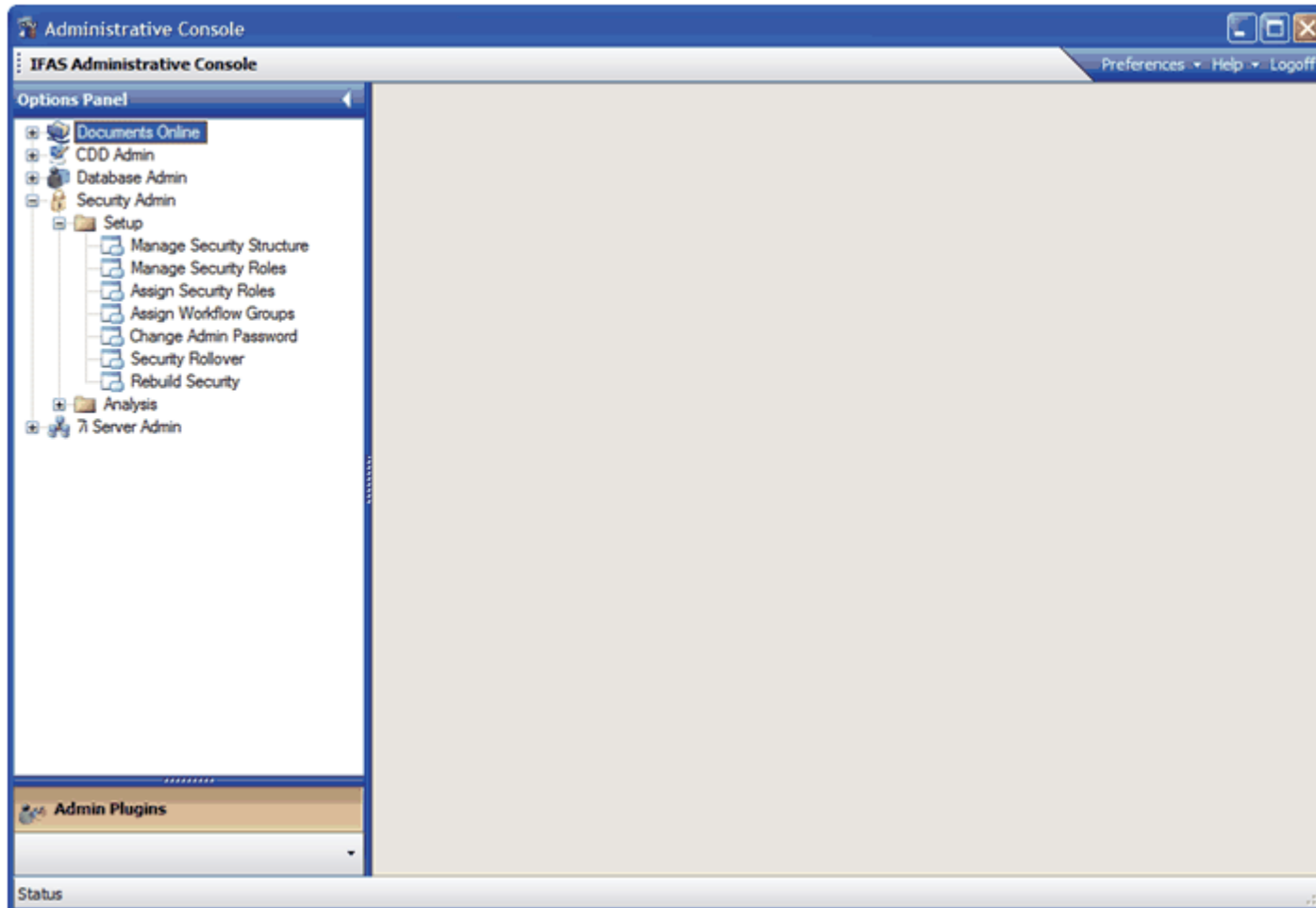


Once the security objects are installed, please close the Administrative Console to effect the changes.

Upon exit a "Resetting Security" dialog will appear. This process will also attempt to contact the 7i server. If the 7i server is not running at this point there will be a 30 second delay as the Console attempts to locate the 7i server. This is not a problem.



The next time the Administrative Console is accessed, additional Admin Plugins will appear in the Options Panel. The user who performed the initialization steps will have been granted a FULL security role which initially grants full access to all IFAS functions and data. Additional security roles and restrictions may be defined in the Manage Security Roles plugin within the Setup section of Security Admin.



For additional information about the operation of the Console, please refer to the Administrative Console user manual.

Role-Based Security

Concept

Role-based security structure facilitates implementing and maintaining security assignments. This document will discuss the role-based structure and offer information on how it can be set up to best meet your organizational needs.

To facilitate a role-based security, we had to change our concept of security. Imagine that you created a security role that only gives access to run reports. Then you create a second security role that only gives access to save reports. In a "most restrictive" environment the two would cancel each other out. However, in an inclusive security model the two would combine giving the user the ability to run and save reports. Using many well-designed security roles enables clients to more easily determine which users have access to a particular functionality. For example, with one role that grants the ability to save reports (called "Save Reports"), you can easily access a quick listing of role assignments to determine who can save a report. Also, you can quickly grant a user the ability to save reports without granting other security defined in that role that you were unaware of at the time.

Remember, the roles work in conjunction with each other and not in opposition. You may have a set of Payroll Reports that you don't want users able to see, so you put them in a CDD Folder called "Payroll Reports". Now you could create a role called "Payroll Reports Folder" that gives access to those folders. In your organization, some people create reports, others modify reports, and some only run them. However, frequently those are not the same people.

Security Roles

The security is based on individual roles within the organization. Rather than looking at security from a concept of what people can or cannot do, this method focuses on groupings of similar tasks. This method allows an organization to establish a structure of security long before it has been decided who would perform those duties.

Managing Software Change

One of the benefits of grouping similar functionality into specific roles is that as the software changes the time and energy required to maintain security is dramatically reduced. Imagine that a new feature is added to an application that qualified as "Super User" utility. Features of this type are not appropriate for all users but extremely useful for others. In most cases, making this utility available to users would entail determining the security class of the desired users and making the change to that class. If the desired users don't have the same class this might involve adding that change to multiple classes. However, you would also need to make sure that those classes were not also assigned to users who should not have this functionality. As you can see, the seemingly simple task of making a feature available to users can become a real ordeal and mistakes can be very costly. With the role-based structure, one of two methods can be used to simplify the process. If a class for "Super Users" already exists, the user can view a list of all of the users assigned to this role. If it is appropriate for those users, that security can be granted to that role. If there is no one role appropriate, a new role can be created that only gives access to that utility. Specific users can then be assigned that role and that functionality will be made available to them.

Managing User Change

Over time, the nature of peoples' jobs and duties can and frequently will change. A user's individual job requirement may change in ways that cannot be addressed in their security class without also giving all of the other people with that class the same access. Also, employee turnover can create additional security issues. The person filling a position may not have exactly the same security needs as the person who left that position. If the security class was only assigned to that one user, then it can be edited, but if not, a new security class may have to be created. The concept of security roles can help to eliminate some of that hassle. Security, such as the ability to print finance reports, can be put into a

single role with no other functionality. If an individual's job requirements are changed to include printing financial reports, that access can be granted without granting it to everyone else that has that person's current security. Also, the ability to print financial reports can be just as easily removed without impacting any of the other access that person may need.

Security Structure

The security structure is presented as an explorer "tree". This represents the security objects that will be available when configuring security roles. When defining security roles, access may be granted to any of the objects, or indirectly by inheritance from a parent object. Only the objects shown in the "tree" will appear as security options when defining roles.

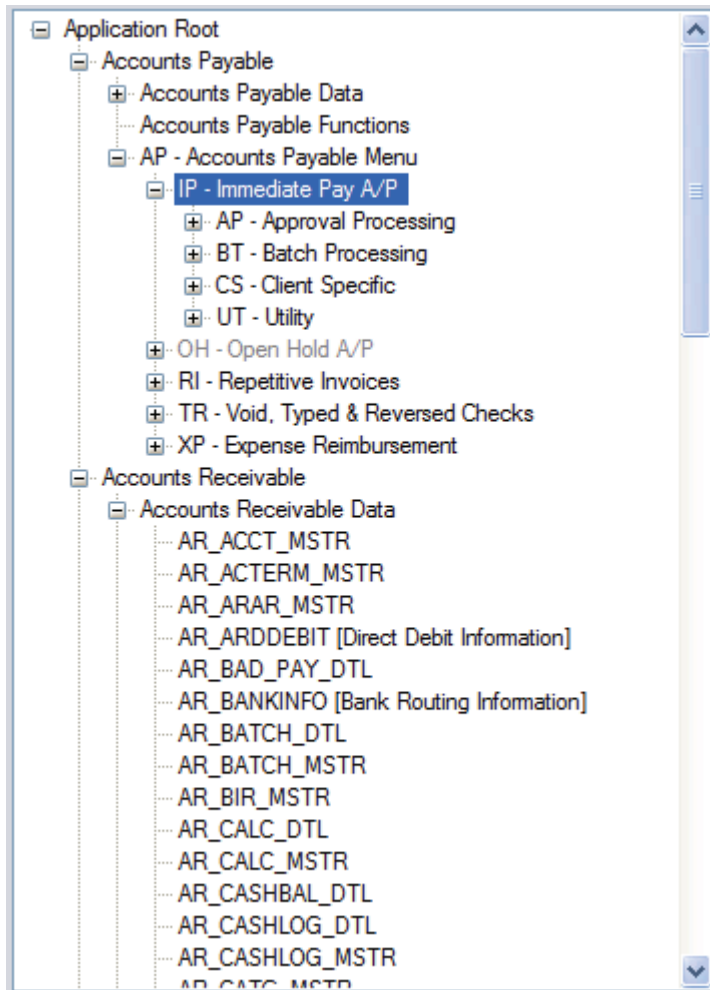
Under "Application Root", each application and subsystem will be shown. The subsystems will contain 3 main sections: Data, Functions, and Menu:

Data represents the database tables. Within the data section an entry will be presented for each table within the subsystem. Each table entry can be accessed individually and the attributes of each table set by role.

Functions represent additional user capabilities, which vary by subsystem.

Menu represents the various screens, reports, and utilities that are available. Each subsystem contains various menu functions and sub-functions, corresponding to the main menu structure.

Here is an example of the "tree" structure showing these sections:

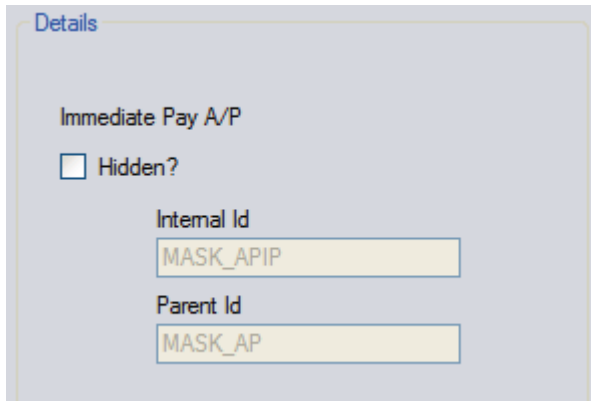


In the previous example, the OH subsystem has been hidden. This is represented by a grayed-out name (see the "OH – Open Hold A/P" line). Objects can be hidden by accessing the Details pane from the Manage Security Structure tab.

Details

The Details pane displays additional information about the currently selected security object within the tree. The additional information consists of the object description, the Hidden state, and the internal and parent ids. The internal id information is provided only for troubleshooting purposes.

The "**Hidden from all roles**" flag determines whether the current object will be shown when managing security views. Hidden objects and their child objects will not appear within the security views. This feature can be used to simplify security configuration by removing objects that will never be used by your organization. When the Hidden box is checked, the corresponding object will be hidden from all security roles.



The screenshot shows a 'Details' panel for an object named 'Immediate Pay A/P'. It contains a 'Hidden?' checkbox which is currently unchecked. Below the checkbox are two text input fields: 'Internal Id' with the value 'MASK_APIP' and 'Parent Id' with the value 'MASK_AP'. The text in the input fields is grayed-out, indicating that the object is hidden.

Hidden objects will appear with their text grayed-out. They will not appear when managing views.

Context-Sensitive User Interface

The security settings for a specific role may change depending on the type of security object you are viewing. Example: when viewing Menus you may only see one checkbox for "Execute" as opposed to Data that has checkboxes for Read, Write, Update, Delete and Execute.

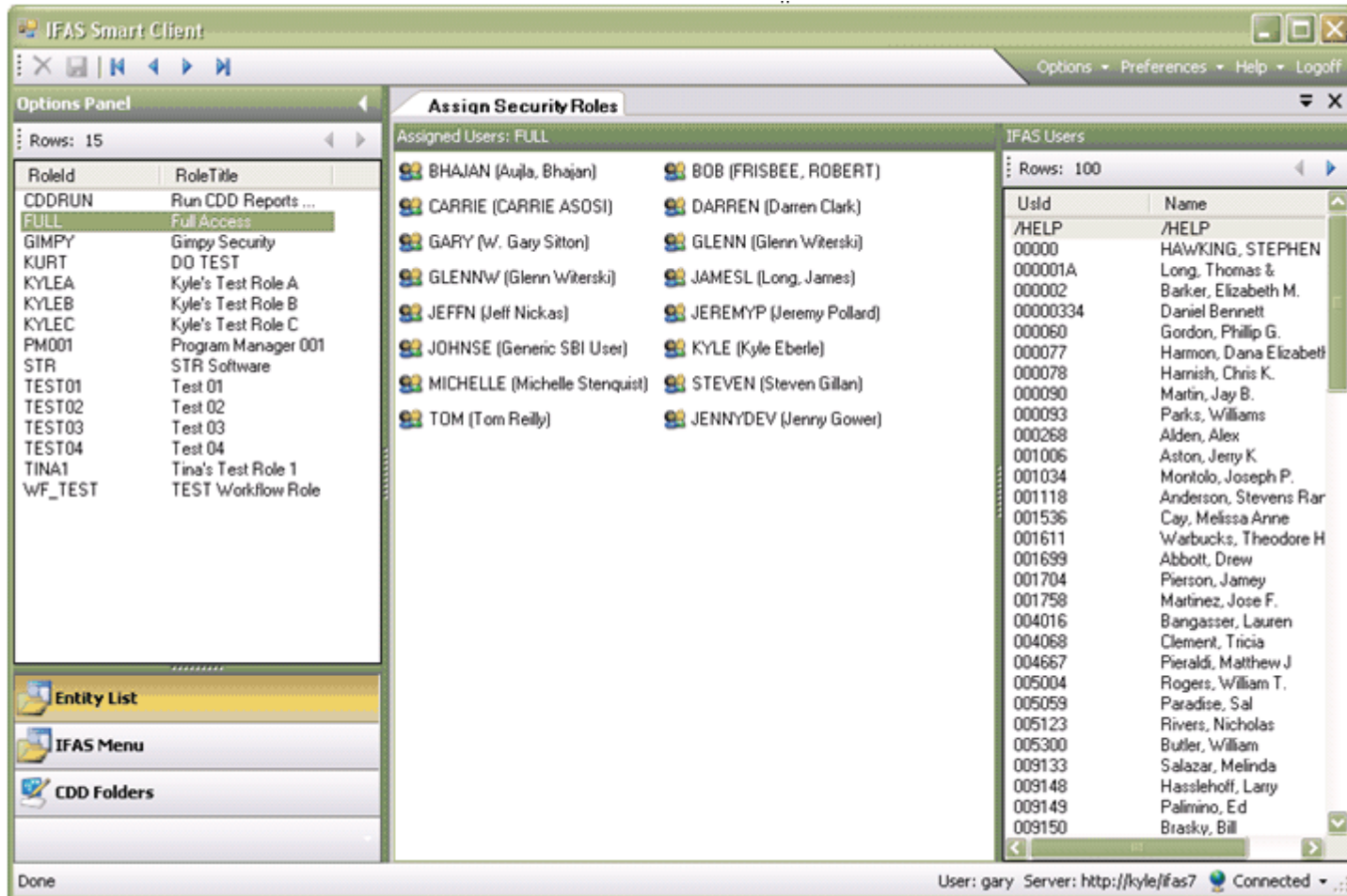
When viewing an existing role browsing to those objects in the Security Object "tree" and changing the security settings in the panel to the right of the "tree" can change the security of the objects within that role.

Table Access Restrictions

Within each application in the "tree", the data objects, the database tables that comprise the application are listed. For each table that is not hidden, the Read, Write, Update, Delete and Execute attributes are shown. A where clause can be written for each table and attribute listed. This capability allows data within a table to be restricted by any element of the table. For instance, an employee could be granted access to a specific department(s) data for viewing, reporting or updating.

Assign Security Roles

The Assign Security Roles screen is used to assign Security Roles to IFAS users. The IFAS Mask used to access this screen is NUUPSA.



Adding Users to a Role

To add an IFAS user to a role first select the desired role from the Entity List on the far left. Then locate the desired IFAS User using the "IFAS Users" list on the far right. The IFAS Users list can be paged using the arrows at the top of the list much like a standard entity list. Double clicking on the items in the list will add that user to the role. Once there are pending changes for the role assignments the "Save" button will become active.

Removing Users from a Role

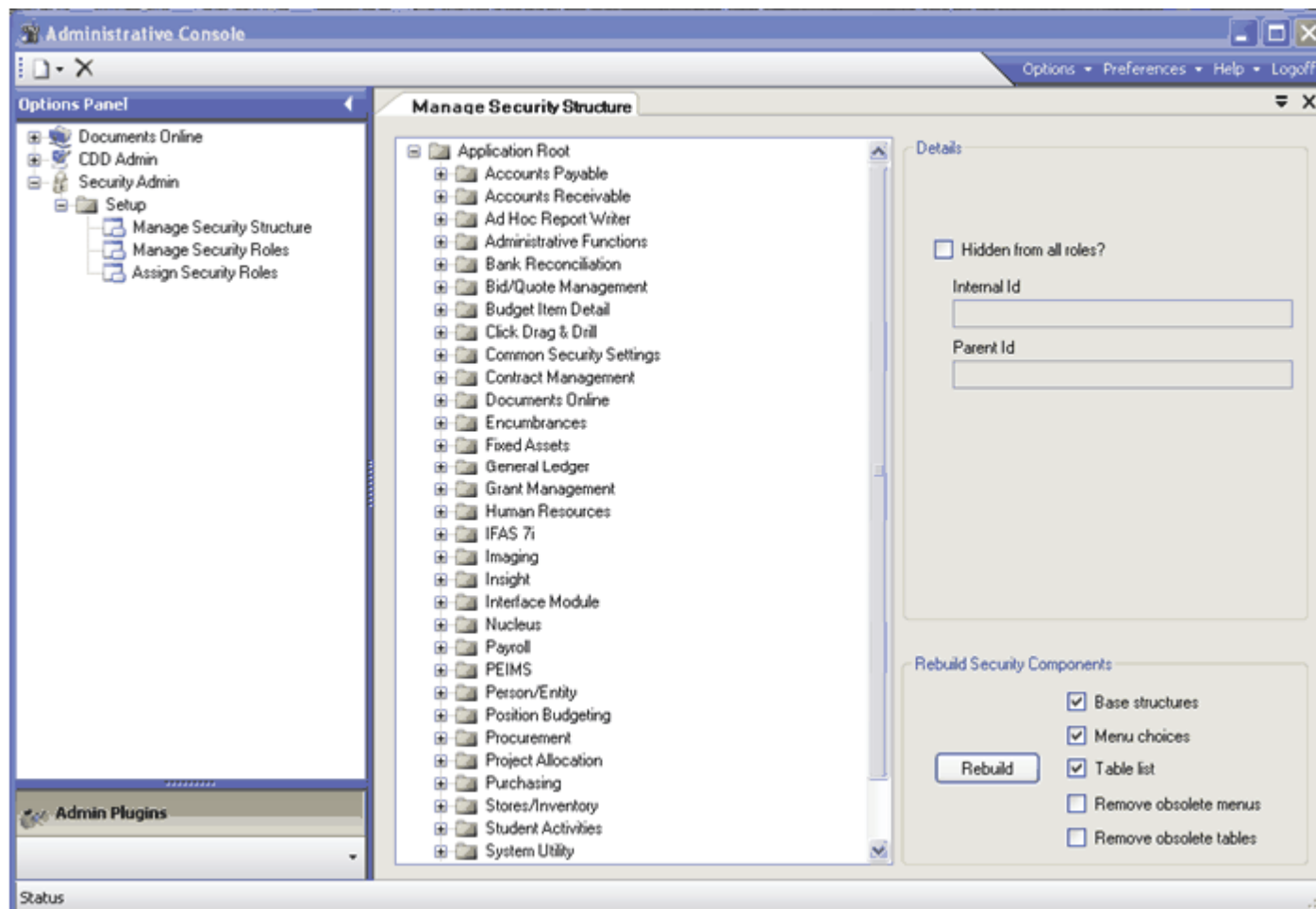
To remove an IFAS user from a role first select the desired role from the Entity List on the far left. Then locate the desired IFAS User from the "Assigned Users" list in the center of the screen. Selection is performed by clicking on a user. Multiple users can be selected by holding down

the "Control" key to select specific users or the "Shift" key to select a group of users. Once users are selected the "Remove" button in the toolbar will be active and clicking on that button in the toolbar will remove those users from the role assignment. Once there are pending changes for the role assignments the "Save" button will become active.

Manage Security Structure

Administrative Tool

The "Manage Security Structure" screen can be found within the administrative console, in the Security Admin / Setup section. Selecting this screen by double-clicking will bring up the view shown below:



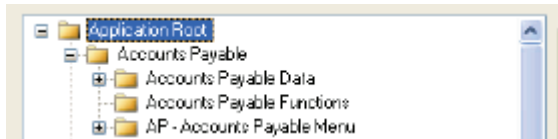
This screen consists of a security tree, a details pane, and a set of "Rebuild" options. The following sections will explain each of those regions.

Security Structure

The security structure is presented as an explorer "tree". This represents the security objects that will be available when configuring security roles under the IFAS menu options NUUPSR (Manage Security Roles) and NUUPSA (Assign Security Roles). When defining security roles, access may be granted to any of these objects, or indirectly by inheritance from a parent object. Only the objects shown here will appear as security options when defining roles.

Under "Application Root", each application and subsystem will be shown. The subsystems will contain 3 main sections: Data, Functions, and Menu:

An example of the structure for the subsystem Accounts Payable displayed below:

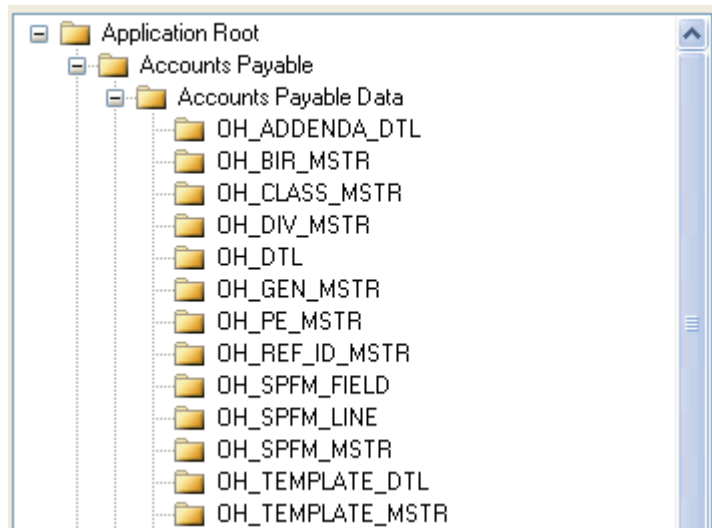
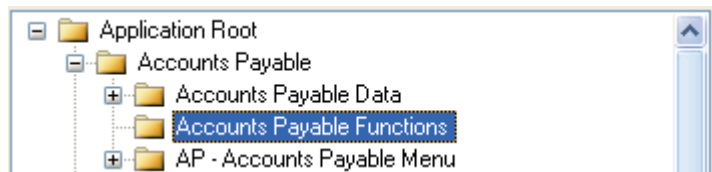


Data represents the IFAS database tables. Within the data section an entry will be presented for each table within the subsystem.

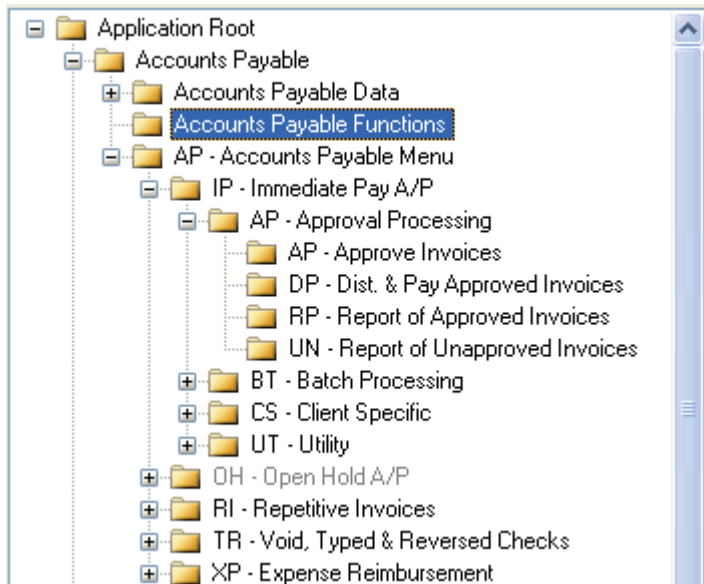
Functions represent additional user capabilities, which vary by subsystem.

Menu represents the various IFAS screens, reports, and utilities that are available. Each subsystem contains various menu functions and sub-functions, corresponding to the IFAS main menu structure.

Examples of each of the 3 Main Sections are shown below:

Data**Functions**

Menu



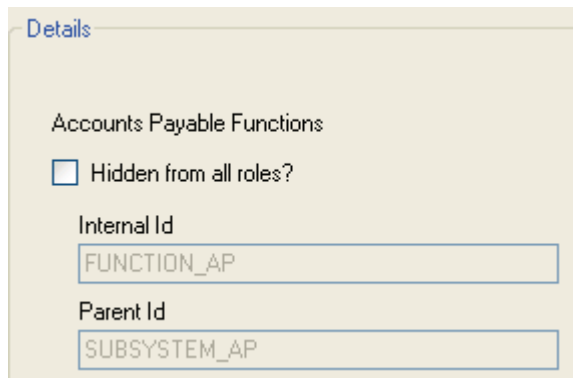
In the previous example, the OH subsystem has been hidden. This is represented by a grayed-out name (see the "OH – Open Hold A/P" line). Objects may be hidden from the Details pane.

Details

The Details pane displays additional information about the currently selected security object within the tree. The additional information consists of the object description, the Hidden state, and the internal and parent ids. The internal id information is provided only for troubleshooting purposes.

The "**Hidden from all roles**" flag determines whether the current object will be shown when managing security views. Hidden objects and their child objects will not appear within the security views. This feature can be used to simplify security configuration by removing objects which will never be used by your organization. When the Hidden box is checked, the corresponding object will be hidden from all security roles.

Details pane



Details

Accounts Payable Functions

Hidden from all roles?

Internal Id
FUNCTION_AP

Parent Id
SUBSYSTEM_AP

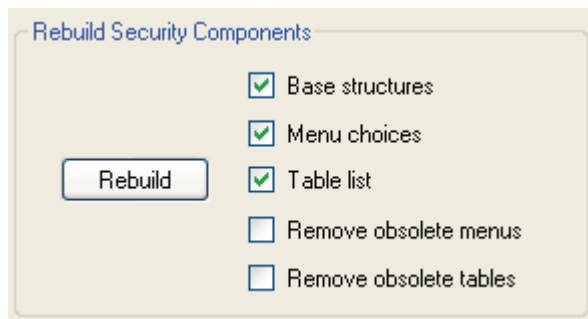
Hidden objects will appear with their text grayed-out. They will not appear when managing views through the NUUPSR screen.

Rebuild Security Components

The Rebuild Security Components pane is used to regenerate the entire list of security objects. This is normally only necessary when applying IFAS upgrades.

The security components are built from multiple components:

- Internally predefined base structures
- IFAS menus (based on current Nucleus menu definitions)
- Additional screens (predefined; 7i screens not modeled in the Nucleus menus)
- IFAS tables (based on current database tables)
- Client specific tables and masks



Rebuild Security Components

Base structures

Menu choices

Table list

Remove obsolete menus

Remove obsolete tables

Rebuild

The options are as follows:

Base structures – Reloads the predefined security objects, including all subsystems and applications.

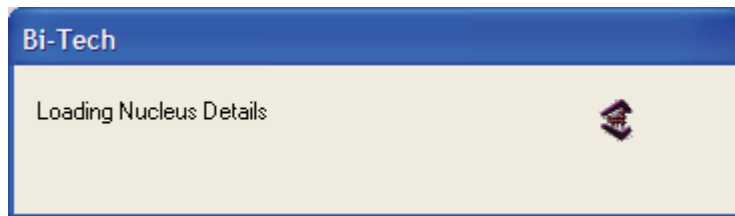
Menu choices – Reloads all menu choices defined within the current IFAS database. This is the list of masks that is maintained using NUUPJB, and presented on the IFAS menu structures (Insight, 7i Home page, Dashboard "All" page, etc). Additionally, a predefined list of 7i screens is added, to allow security on known 7i screens that are not maintained as Nucleus menu items.

Table list – Reload table names for all installed IFAS subsystems, based on internal details derived from database schemas at table creation time.

Remove obsolete menus – Compare the current security component list to the Nucleus menu definitions, removing any "mask" objects that do not exist in Nucleus or in the predefined list of 7i screen objects.

Remove obsolete tables – Compare the current security component list to the existing tables within the database server, removing any table objects that represent non-existent tables.

To rebuild security components, check all desired boxes and click the **Rebuild** button. This will present a wait dialog that displays the current phase of the process.



Depending on system/network performance, the rebuild process may take a few minutes.

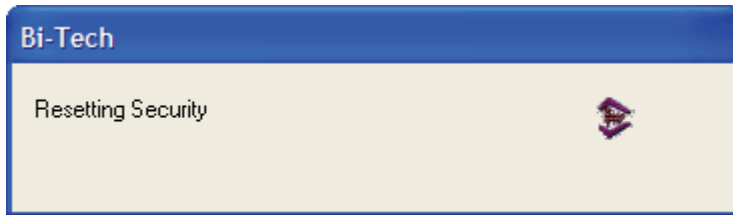
Running the rebuild functions more than once will not cause problems. The only options that will potentially remove security objects are the "Remove obsolete" functions.

Client Specific Security Objects

In addition to the standard IFAS security objects, client specific menu options and tables may be added. This is only necessary for menu options which are not defined in Nucleus (via NUUPJB), or for database tables which are not regular IFAS tables.

Reset Security

If changes are made using the Manage Security Structure tool, several internal security-related structures will be reset upon exit. This process involves the removal of cached security data stored within the IFAS database. The 7i server farm will also be requested to reload its security so that the changes will take effect for any users connecting to 7i. For users who are already connected to 7i, certain types of security changes will not take effect until the user's next web browser session.



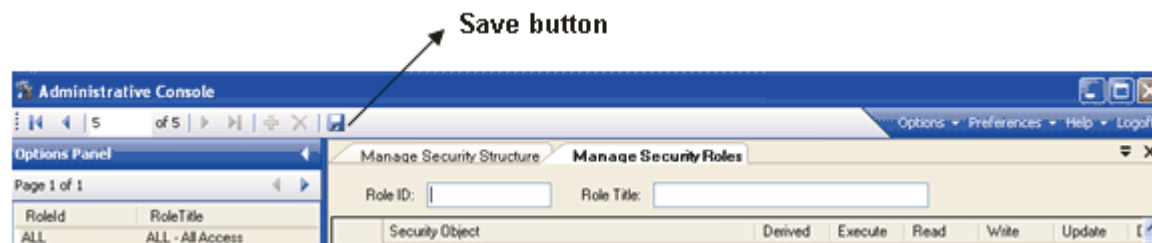
Manage Security Roles

Security Objects

Security Roles are made up of the specific security granted to the different security objects within the structure. By default all of the objects are marked as "Derived". Derived security objects inherit the security of the first parent object that isn't derived. If none are found then no access to that security node is granted with a particular role.

Creating a New Role

To create a new Security Role use the "New" icon in the toolbar. Then, enter the new Role ID and Role Description for the role. After entering those two values the specific security desired for this role can be set in the Security Object Tree. Once all of the desired changes are made the role will be created by using the "Save" button on the toolbar.



Editing an Existing Role

When viewing an existing role the security of the objects within that role can be changed by browsing to those objects in the Security Object Tree and changing the security settings in the panel to the right of the tree. If any changes to the role are made but have not been saved yet the "Save" button in the toolbar is enabled. Clicking the "Save" button will send all pending changes to the 7i Server.

Context-Sensitive User Interface

The security settings for a specific role may change depending on the type of security object you are viewing. Example: when viewing Fixed Asset Menus you may only see one checkbox for "Execute" as opposed to Fixed Asset Data that has checkboxes for Read, Write, Update, Delete and Execute.

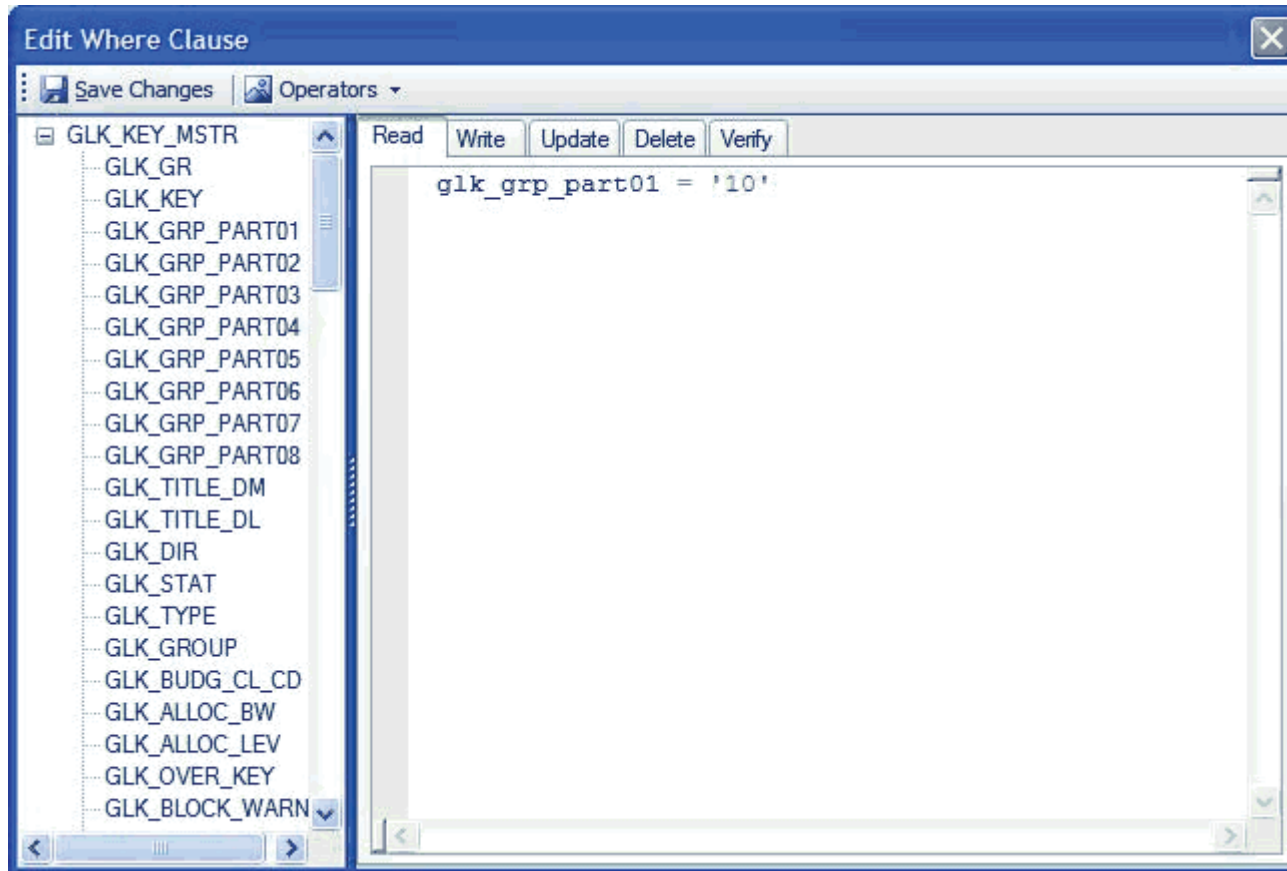
| Security Object | Derived | Execute | Read | Write | Update | Delete | Extension |
|------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-----------|
| [-] Fixed Assets | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] Fixed Assets Data | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] Fixed Assets Functions | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] FA - Fixed Assets Menu | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |
| [-] DP - Depreciation | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |
| [-] IQ - Interactive Inquiry | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |
| [-] RE - Reports | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |
| [-] UP - Update | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |
| [-] UT - Utilities | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |

Common Security Settings

Common security settings allow where clauses to be written for tables within roles. Access can be further restricted to tables and linked tables based on a combination of common settings found in all roles assigned to the user. Below is a display of the Common Security Settings tree.

| Manage Security Roles | | | | | | | | |
|---|--------------------------|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-----------|--|
| Role ID: <input type="text" value="ALL"/> | | Role Title: <input type="text" value="ALL - All Access"/> | | | | | | |
| Security Object | Derived | Execute | Read | Write | Update | Delete | Extension | |
| ▶ Common Security Settings | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Bank Reconciliation | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| Bank Definition | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| General Ledger | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| Account Key Security | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Ledger Security | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Object Code Security | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Human Resources | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| Employee Definition | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Purchasing | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| Purchase Orders | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Purchasing Security Codes | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Stores Inventory | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| Stores Inventory Orders | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Stores Inventory Security Codes | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |
| Stores Inventory Warehouse | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter | |

Below is an example of the **Where Clause** screen. Note that the where clause has been written for the **Read** attribute of the associated table, the GLK_KEY_MSTR. It is possible to include where clauses for each of the attributes of the table, read, write, update, delete and verify.



Not only are the tables listed available for restriction based on the where clause, but the linked tables are restricted as well. See the example below for reference in how linked tables might be restricted. The example shown below does not include each link possibility, but rather an example of the way tables are linked.

In the above example, General Ledger is broken down into Account Key Security, Ledger Security and Object Code Security.

Ledger Security (glg_gen_mstr)

glg_gen_mstr

glk_key_mstr

glo_obj_mstr

glba_budact_mstr

bd_eqpt

bd_misc

Account Key Security (glk_key_mstr)

glk_key_mstr

glba_budact_mstr

bd_eqpt

bd_misc

Object Code Security (glo_obj_mstr)

glo_obj_mstr

glba_budact_mstr

bd_eqpt

bd_misc

In the example shown below, the **Purchasing** common security settings are shown:

Purchase Orders PR's (pop_pv_dtl)

pop_pv_dtl

poi_item_dtl

Purchasing Security Codes (pos_sec_mstr)

pos_sec_mstr

pop_pv_dtl

Allow Entry of Set ID

The Execute object check box is the only attribute that is validated. With this box checked, the operator has the authority to override the Batch Seed value and create a unique seed value.

The screenshot shows the 'Manage Security Roles' interface for the 'ADMIN' role. The table lists various security objects and their permissions. The 'Execute' column is highlighted with a red oval, indicating that this is the only attribute validated for the 'Allow Entry of Set ID' object.

| Security Object | Derived | Execute | Read | Write | Update | Delete |
|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Application Root | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Accounts Payable | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Accounts Receivable | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Ad Hoc Report Writer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Administrative Functions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Bank Reconciliation | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Bid/Quote Management | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Budget Item Detail | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Click, Drag & Drill | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Common Security Settings | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Contract Management | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Custom | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Documents Online | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Encumbrances | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Endowment Management | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Fixed Assets | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| General Ledger | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Grant Management | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Human Resources | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| IFAS 7i | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Imaging | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Insight | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Interface Module | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Nucleus | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Nucleus Data | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Nucleus Functions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Set Related Functions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Allow Entry of Set ID | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| NU - Nucleus Menu | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| PEIMS | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Payroll | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

2.1.3 Masks and Corresponding Data Set Names

The following are masks and their corresponding data set names used for security.

| Mask | Data Set Names | Valid Select Code Restrictions |
|-----------|----------------|--|
| APOHUPDV | OH-DIV-MSTR | Restrict based on AP Division codes |
| APOHBTUB | OH-DTL | Restrict based on AP Division codes |
| APIPBTUB | same | |
| APTRBTUB | same | |
| APOHIQ | same | |
| APOHRE | same | |
| | | |
| ARB TARUB | AR-TRNS-DTL | Restrict based on AR Division codes |
| ARBTCRUB | same | |
| ARIQ | same | |
| ARRE | same | |
| | | |
| CKUP | CK-CHECK-MSTR | Restrict based on Check stock ID |
| CKIQ | same | To make this happen, you must provide a NUCLEUS (form name) and CKAC CESS (field name). The question CK37 must have CKACCESS in the validation code field. |
| CKRE | same | |
| | | |
| ENUPDV | EN-DIV-MSTR | Restrict based on EN Division codes |
| ENBTUB | EN-DTL | |

| | | |
|-------------------------------------|---------------|---------------------------------------|
| Restrict based on EN Division codes | | |
| ENIQ | same | |
| ENRE | same | |
| | | |
| GL | GLG-GEN-MSTR | Restrict based on Ledger |
| GL | GLK-KEY-MSTR | Restrict based on GL Organization key |
| GL | GLK-GRP-MSTR | Restrict based on GL Org key parts |
| GL | GLO-OBJ-MSTR | Restrict based on GL Object codes |
| GL | GLO-GRP-MSTR | Restrict based on GL Obj groups |
| GL | GLB-BUDG-MSTR | Restrict based on Budget version |
| GL | GLA-ACT-MSTR | Ability to browse Actual amounts |
| GL | GLT-TRNS-DTL | GL Transaction details |
| GL | DEFAULT | |
| Default Ledger code | | |
| PEUPPE | PE-NAME-MSTR | Used to restrict by Owner ID |
| | PE-ADDR-DTL | Used to restrict by Address Code |
| | | |
| | | |
| PEUPPR | PE-PROD-MSTR | product id as the value |
| | | PE-COM-CODE |

| | | |
|-----------------------------------|----------------|--------------------------------|
| commodity code as the value | | |
| | | |
| POUPPR | PO-PV-DTL | |
| POUPPR | PRRANGE | |
| Allow access to a PR number range | | |
| POUPPR | PORANGE | |
| Allow access to a PO number range | | |
| | | |
| SIUPIN | SII-INVTRY-DTL | Restrict based on Warehouse ID |
| SIBTUB | same | |
| SIOEUB | same | |
| SIINCT | same | |

2.2 Intermediate

2.3 Advanced

Under Construction

2.4 Best Practices

Under Construction

3 Processes

Under Construction

4 Process Reference

4.1 Entry

Under Construction

4.2 Processing

4.2.1 GL functional Security

General Ledger Objects

From the General Ledger object "tree" display below, a number of process switches can be set.

Security Object Derived Execute Read Write Update Delete Extension

| Security Roles | | Derived | Execute | Read | Write | Update | Delete | Extension |
|--------------------------|------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-----------|
| Role ID: CHUCK | Role Title: CHUCK ROLE | | | | | | | |
| Security Object | | | | | | | | |
| Accounts Payable | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Accounts Receivable | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Ad Hoc Report Writer | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Administrative Functions | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Bank Reconciliation | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Bid/Quote Management | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Budget Item Detail | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Click Drag & Drill | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Common Security Settings | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| Contract Management | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Custom | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Documents Online | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| Encumbrances | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Endowment Management | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Fixed Assets | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| General Ledger | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| General Ledger Data | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| General Ledger Functions | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Budgeting | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| FTE Budgeting | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Journal Entry Functions | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| GL - General Ledger Menu | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Grant Management | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Budgeting

Budget Date Override

The Execute object check box is the only attribute that is validated. With this box checked, the operator has the authority to override the Budget Date.

Budget Transfer Limit

The budget transfer limit is set from the screen display seen below. This screen is used to set the Ledger Part/Group and Message for one or a number of ledger and Part/Group combinations.

| | Ledger | Part/Group | Block/Warn |
|---|--------|------------|------------|
| | JL | FUND | WARN |
| | GL | KEY | BLOCK |
| * | | | |

Enter the Ledger to be validated.

Enter the Part/Group to be validated.

Select the level of action indicator from the pull down.

When the proper information has been entered, double click the Save Changes icon to save the current filter record.

FTE Budgeting

Amount Override

The Execute object check box is the only attribute that is validated. With this box checked, the operator has the authority to override the Amount.

FTE Override

The Execute object check box is the only attribute that is validated. With this box checked, the operator has the authority to override the FTE.

Journal Entry Functions

Batch Proof Budget Suppress

The Execute object check box is the only attribute that is validated. With this box checked, the operator has the authority to suppress the batch proof budget checking.

Data Entry Budget Override

The Execute object check box is the only attribute that is validated. With this box checked, the operator has the authority to override the budget checking on the journal entry screen.

4.2.2 Purchasing Functional Security

Converting Purchasing Functional Security

The example shown below is an all access user. This user can print purchase orders, update and change purchase orders. Security is unrestricted.

Allow User to Approve PR's

In the example given below, approvals are controlled from the check box Allow user to approve PR's. This functionality has been replaced by the integration and use of the Workflow system.

| User | Security Code | Approval Class |
|--|--|--|
| User ID: <input type="text" value="BSI"/> | <input checked="" type="checkbox"/> Allow user to approve PRs <input checked="" type="checkbox"/> Change PO# after printing | Allow Print: <input type="text" value="Y"/> Modification Type: <input type="text" value="Y"/> Deletion Type: <input type="text" value="Y"/> Approval Type: <input type="text" value="A"/> |
| Security Codes | Approval Codes | |
| Enter four character security codes along with Read or Write access. | | Example: SEC1W |
| <input type="text" value="R"/> <input type="text" value="W"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |

This security is set up as part of the Manage Security Role function from the Security Admin/Setup Option Panel selection.

The functional security equivalent of the above setup would be a role with no where clauses in the Common Security Settings area and no database table or menu restrictions designated.

For any given role, access can be restricted in the Common Security Settings for Purchase Order Numbers and Purchasing Security Codes. That are referenced/linked throughout the Purchasing subsystem.

| Security Object | Derived | Execute | Read | Write | Update | Delete | Extension |
|-------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-----------|
| [-] Common Security Settings | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] Bank Reconciliation | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] General Ledger | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] Human Resources | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] Purchasing | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| [-] Purchase Orders | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter |
| [-] Purchasing Security Codes | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | No Filter |
| [-] Stores Inventory | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Special consideration should be given to creating where clauses. It may be necessary to include checks for blank or null values depending on the columns being interrogated in the where clause.

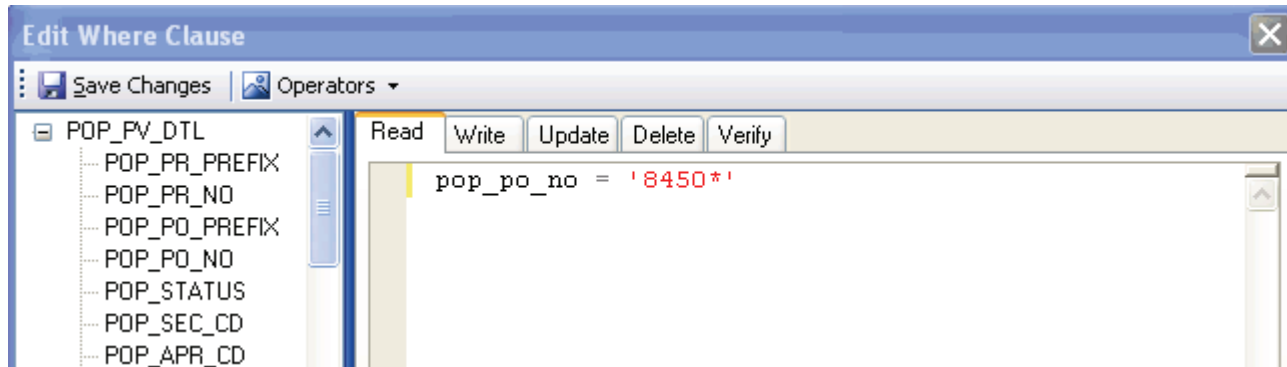
Applying where clauses to any of the attributes below will restrict access to those attributes based on the where clause. Whenever a column referenced in the pop_pv_dtl or the pos_sec_mstr where clause is being read the where clause will apply.

Common Security Settings – Purchasing

A variety of combinations of security can be developed based on where clauses written for the POP_PV_DTL and the POS_SEC_MSTR. Within each of the tables listed above, access can be granted for any combination of read, write update, delete and verify.

The most common uses of where clauses would be to limit access based on security code, being able to read and or write information containing a given security code(s) and restricting access to a specific purchase order(s) by PO number.

| Security Object | Derived | Execute | Read | Write | Update | Delete | Extension |
|---------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-----------|
| Purchasing | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Purchasing Data | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Purchasing Functions | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Assign PO number | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| Change PO number after printing | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| PO - Purchasing Menu | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | | | |



The example shown below is a restricted access user. This user can not print purchase orders, updates can be done until the PO is approved and deletions are not allowed. Security is restricted.

| User | Security Code | Approval Class |
|--|---|--|
| User ID: <input type="text" value="01PURCKH"/> | <input checked="" type="checkbox"/> Allow user to approve PRs <input type="checkbox"/> Change PO# after printing | Allow Print: <input type="text" value="N"/> Modification Type: <input type="text" value="A"/> Deletion Type: <input type="text" value="N"/> Approval Type: <input type="text" value="A"/> |
| Security Codes | | Approval Codes |
| Enter four character security codes along with Read or Write access. | | Example: SEC1W |
| <input type="text" value="OCFAW"/> | <input type="text" value="1212W"/> | <input type="text" value="Y2KPW"/> |
| <input type="text" value="ADMNW"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |

Allow Print will be controlled by the menu mask for printing PR's and PO's.

Note the execute box for PO – Print Purchase Orders is not checked the mask POPO will not be available to users assigned this role.

| Security Object | Derived | Execute |
|--------------------------------|--------------------------|--------------------------|
| [-] Purchasing | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] Purchasing Data | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] Purchasing Functions | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] PO - Purchasing Menu | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] CS - Client Specific | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] IQ - Interactive Inquiry | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] MA - PO Maintenance | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] ST - PO Status Inquiry | <input type="checkbox"/> | <input type="checkbox"/> |
| [-] PO - Print Purchase Orders | <input type="checkbox"/> | <input type="checkbox"/> |

Note the execute box for PO – Print Purchase Orders is not checked the mask POPO will not be available to users assigned this role.

Modification Type will be controlled by two elements. Type A will be controlled by the Workflow model. Approvals to be handled by workflow notification.

Modification Types P, R and Y will be controlled by Purchasing Data attributes.

Note the PO-ITEM_DTL has a value of Filtered and a check in the read column.

This setting will allow a read of the PO-ITEM_DTL table if the value of the where clause is satisfied.

| Security Object | Derived | Execute | Read | Write | Update | Delete | Extension |
|-------------------|--------------------------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-----------|
| ▶ Purchasing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ▶ Purchasing Data | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| POA_ASSOC_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| POC_CLASS_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| POE_EVENT_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| POI_ITEM_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Filtered |

Deletion Types I, N,P, R and Y will be controlled by Purchasing Data attributes.

Note the PO-ITEM_DTL has a value of Filtered and a check in the delete column.

This setting will allow a deletion of a PO-ITEM_DTL table record if the value of the where clause is satisfied.

| Security Object | Derived | Execute | Read | Write | Update | Delete | Extension |
|-------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|-----------|
| ▶ Purchasing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ▶ Purchasing Data | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| POA_ASSOC_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| POC_CLASS_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| POE_EVENT_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No Filter |
| ▶ POI_ITEM_DTL | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Filtered |

4.3 Utilities

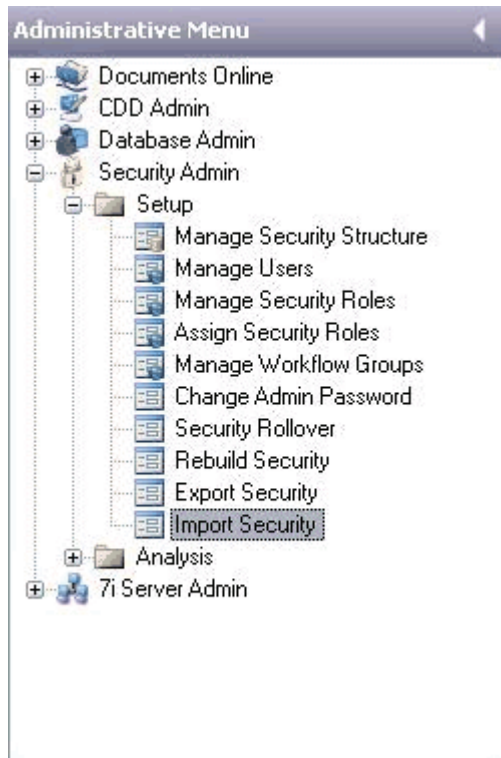
4.3.1 Security Import

Introduction

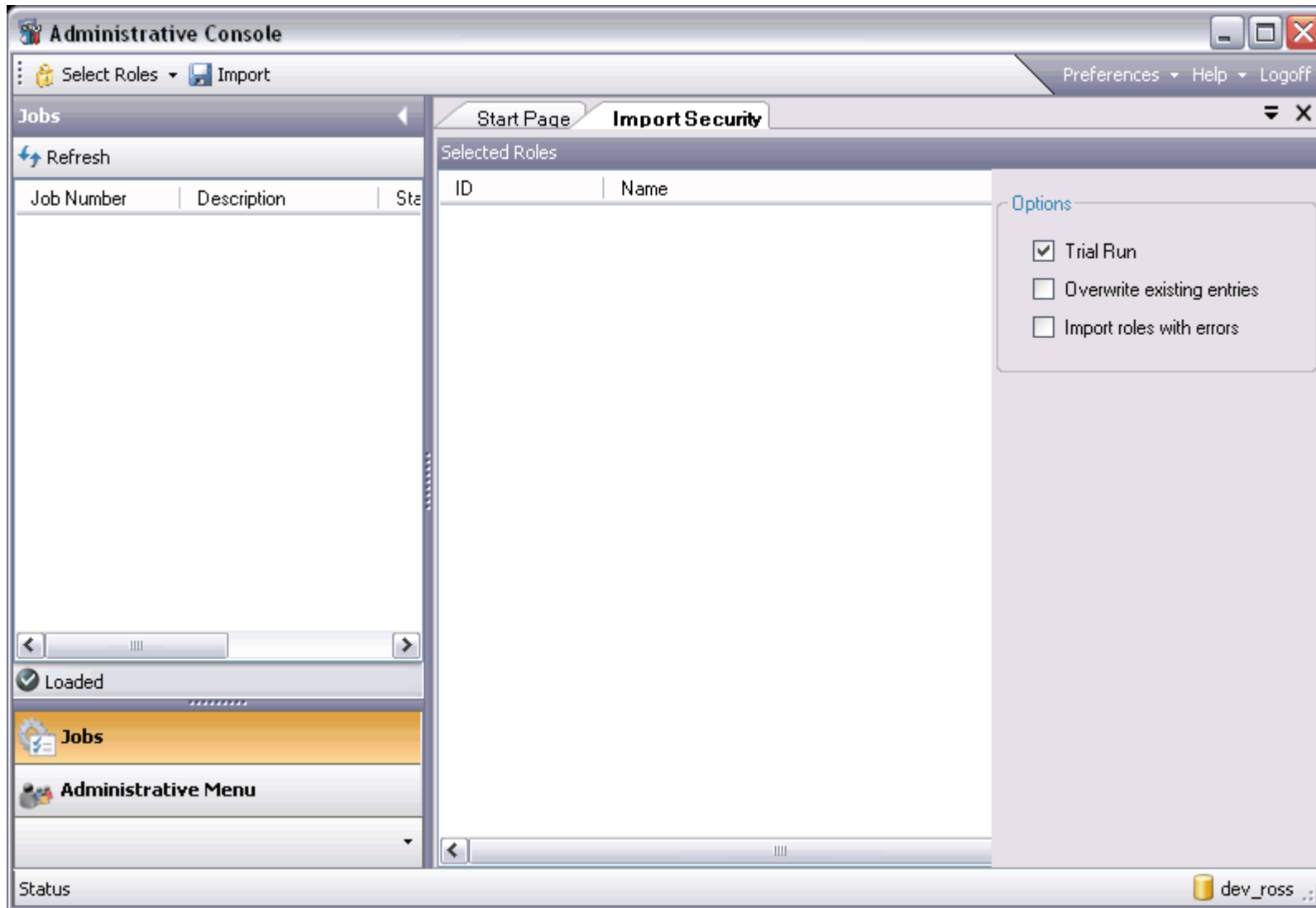
The "Import Security" plugin is used to load a file of role based security data into an account. It must be noted that an exact match is necessary on both the us_id(the users ID) and the us_no(internal ID number kept on the us_no_mstr) columns before the import of roles will take place. Any new users created in the source account and loaded into the export file will not be loaded into the target(receiving) account. A role assigned to this user only will be loading into the target account but will not be assigned to a user.

Administrative Tool

The "Import Security" screen can be found within the administrative console, in the Security Admin / Setup section. This utility will allow you to import all roles, or a selected set of roles that were previously exported. It will load the roles from an archived file (zip file) and allow you to select from that list. This process can be used to move security roles from test to production or simply to restore security roles that were previously archived.

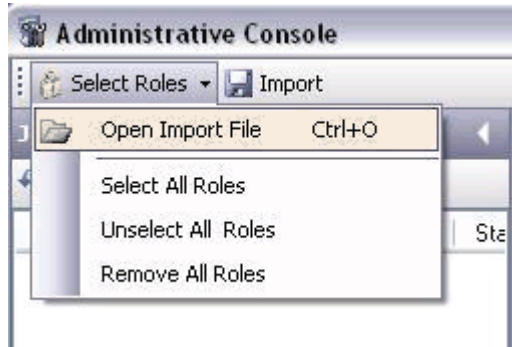


Selecting this screen by double-clicking will bring up the view shown below:

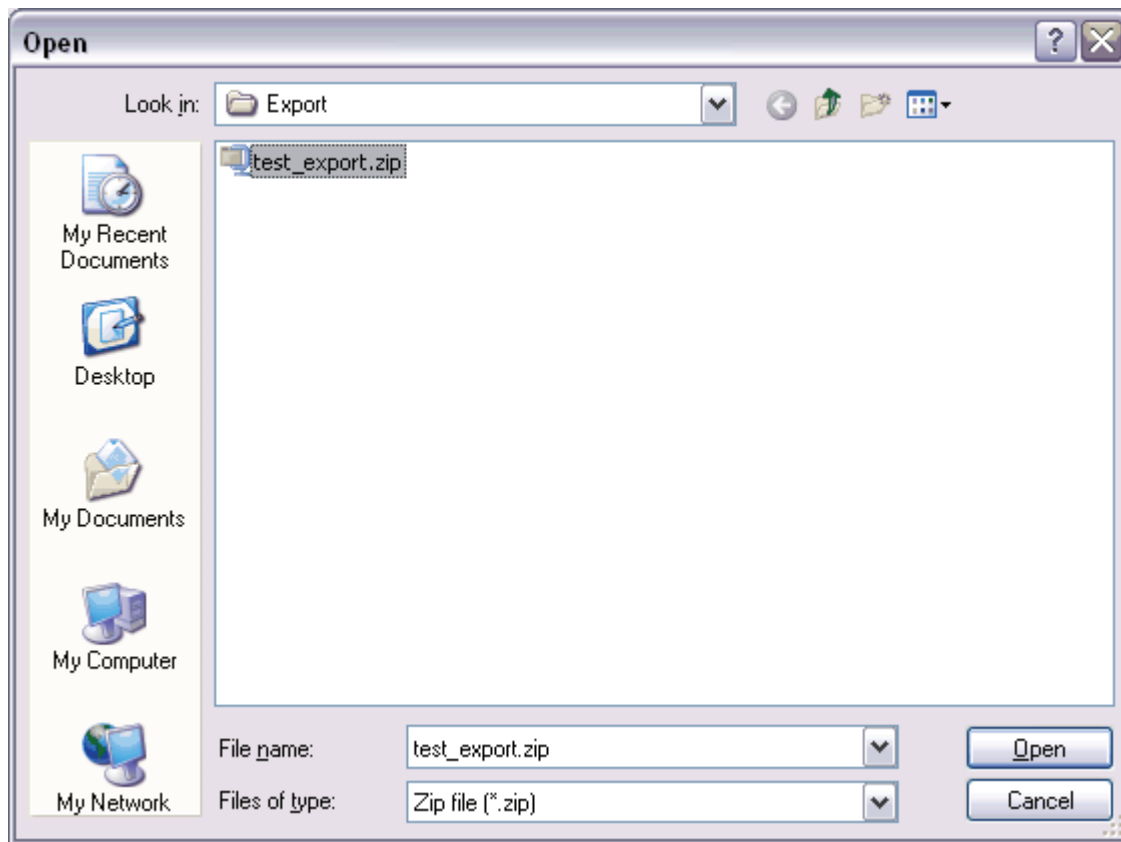


Select Roles for Import

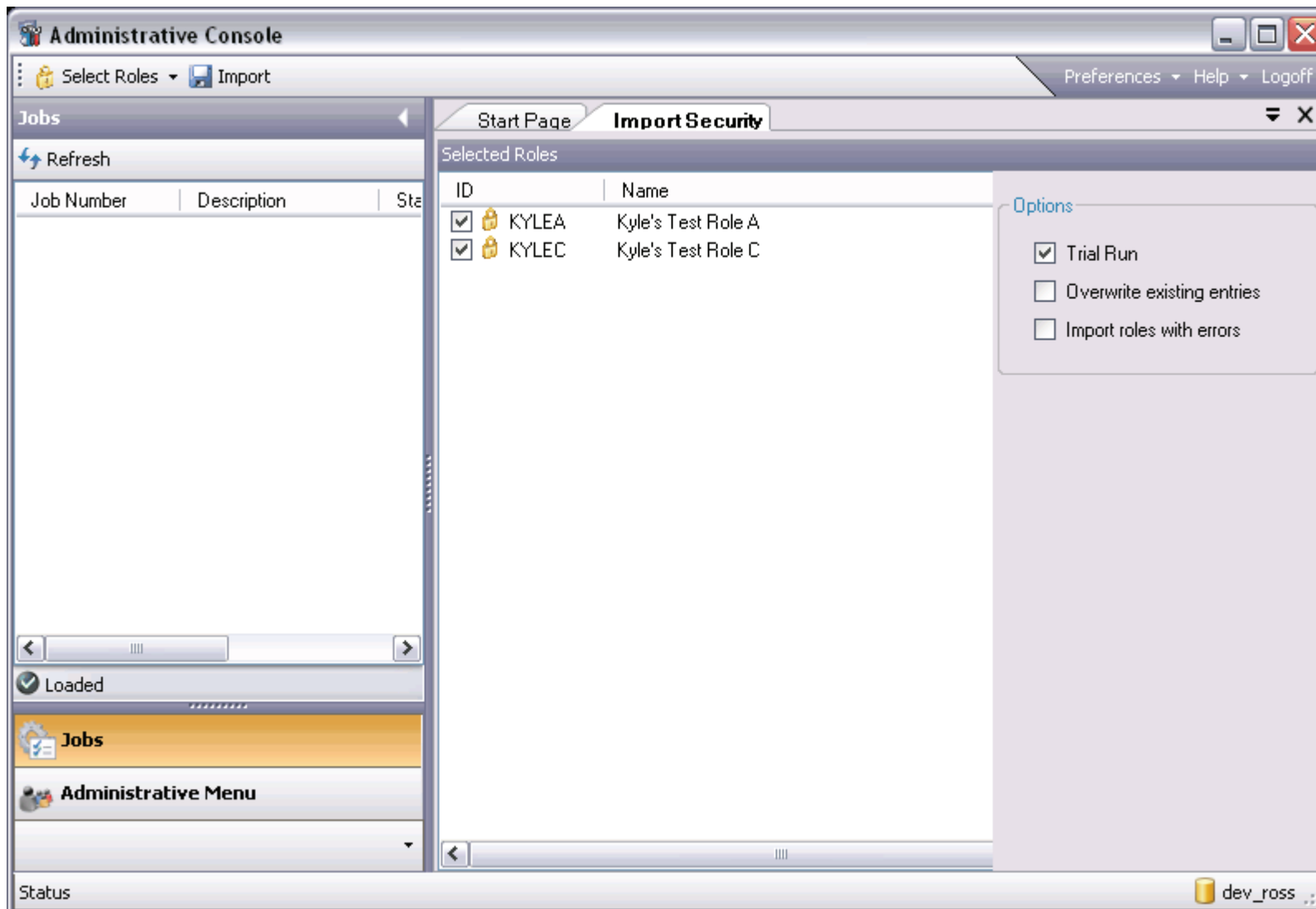
Under 'Select Roles' menu, choose 'Open Import File' to select the archive you wish to import.



An Open file dialog will display to help you locate your archive.



Once you have selected your archive and click the Open button, it will be opened (unzipped to a temp folder in the directory the archive resides) and its roles will be displayed. By default, all the roles in the archive are selected (with a check box) to indicate they will be imported.



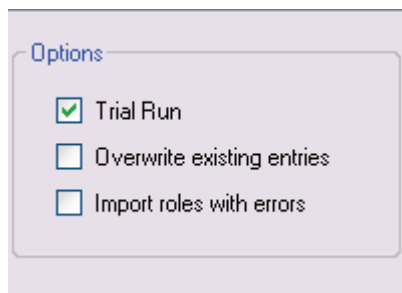
You can select which roles from that archive you wish to import. You can choose all the roles in the archive with 'Select All Roles', or use the 'Unselect All Roles' menu item to uncheck them all. The 'Remove All Roles' menu option will clear the list of roles and you will need to

choose another archive. If there is already a list of roles, and you open another archive, the role list will be just that of the new archive. Archives can only be imported one at a time.

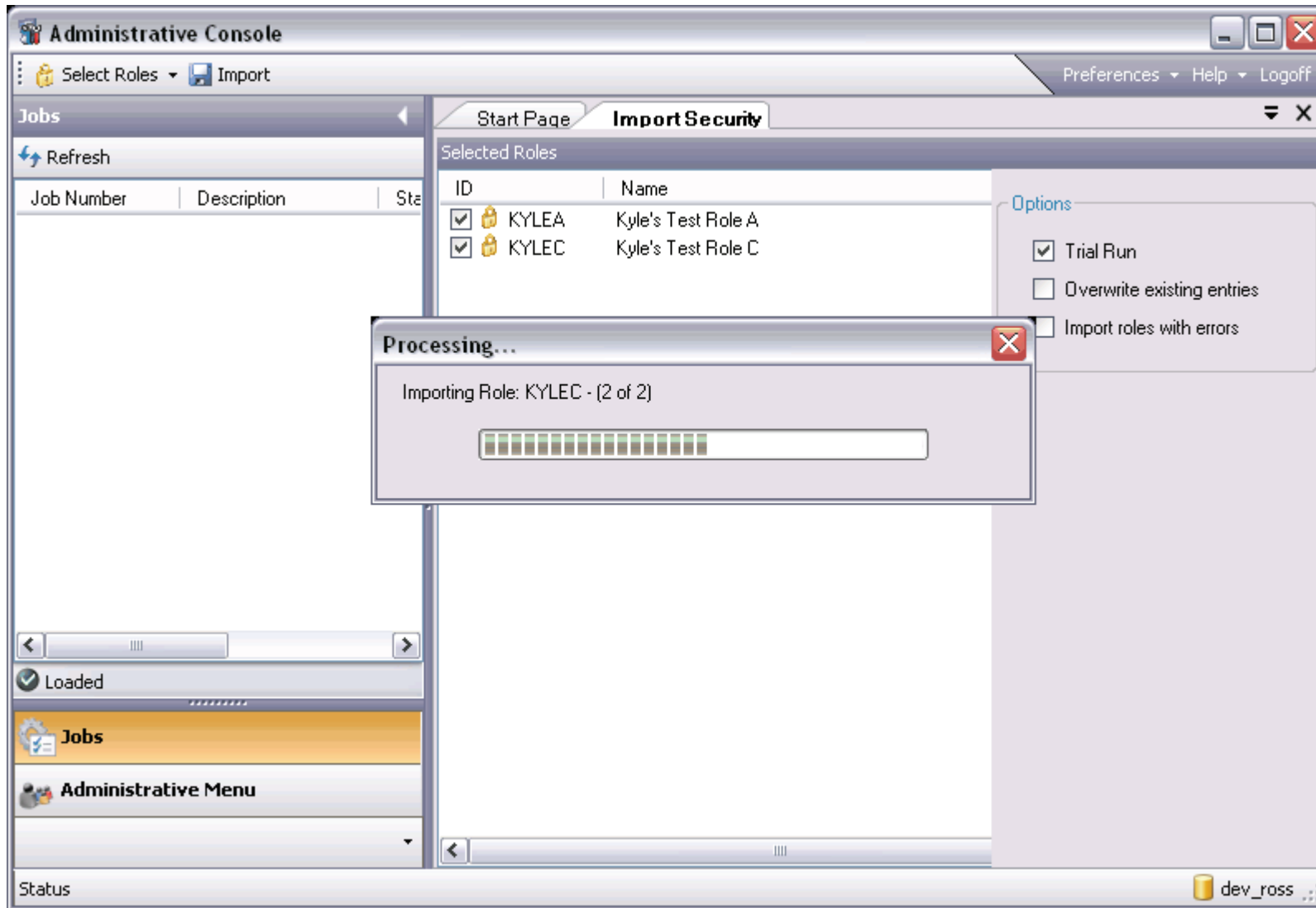
Importing Roles

There are three settings that apply to the import role process.

- 1) **Trial Run:** The process will run in report mode only, no updates will be made. This is the default mode. The job output can be viewed to check for errors or import issues that need to be resolved before the utility is run in live mode.
- 2) **Overwrite existing entries:** If this is checked, and a role already exists in the system, it will be overwritten. Leave this unchecked if you do not wish for existing roles to be updated.
- 3) **Import roles with errors:** If this is checked, and an error is encountered on the role, it will still be imported. Problems such as security objects that are in the export system do not match the import system. These errors are shown in the job output in red.



Once you have opened your archive and checked the boxes next to each role that you wish to import, simply press the 'Import' button on the top toolbar.



Once the job has completed, the Jobs panel on the left will be updated. Expand the '+' by the job number and then double click on the job out labeled 'Import Security Roles'.

The screenshot displays the Administrative Console interface. The main window is titled 'Import Security' and shows a list of 'Selected Roles' with the following data:

| ID | Name |
|---|--------------------|
| <input checked="" type="checkbox"/> KYLEA | Kyle's Test Role A |
| <input checked="" type="checkbox"/> KYLEC | Kyle's Test Role C |

On the right side, the 'Options' section contains the following settings:

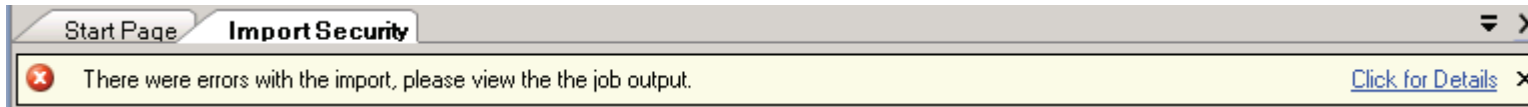
- Trial Run
- Overwrite existing entries
- Import roles with errors

In the left-hand 'Jobs' pane, a job with ID 395059 and description 'Import Security Roles (Doc ID: 13992)' is highlighted with a red oval. The job status is 'Completed'.

Sample run of job output in trial mode:

Import Errors

If the ID and US_NO are not the same in the target account as they are in the source account errors as seen below will be displayed in the PDF generated by the import process and listed on the Jobs page of the Import plugin.



Importing Role: VAN_ALL_ACC Details: 29 Assignments: 1

Creating new entry for Role for VAN_ALL_ACC

The import user id VAN_SUPER with us_no 972 for role VAN_ALL_ACC does not match the current user id 00000336 in the system, skipping.

Encountered problems importing role VAN_ALL_ACC, skipping

As a result of this error, the role and the User will not be created in the target account

The user in the source account is displayed in the following SQL of the us_usno_mstr

| us_no | us_id | us_name | us_name_u | us_desc | us_loc_cd |
|-------|-----------|----------------------------|------------------------|--------------------------------|-----------|
| 972 | VAN_SUPER | Jan Superintendent of Sch. | JAN SUPERINTENDENT ... | Superintendent Vanilla Distric | BI-TECH |

The role referred to in the above error message is found in the SQL from the us_role_dtl

| us_role_id | us_no | us_wf_enable | us_rr_pri | unique_key |
|-------------|-------|--------------|-----------|-------------------|
| VAN_ALL_ACC | 972 | Y | 0 | B1BA4738-3F45-... |

In the target account the SQL displaying us_no number 972(the source us_no) is as follows:

| us_no | us_id | us_name | us_name_u | us_desc | us_loc_cd | us_mgr_cd |
|-------|----------|-----------------|-----------------|-----------------|-----------|-----------|
| 972 | 00000336 | TESTING, TER... | TESTING, TER... | QA Engineer ... | BI-TECH | DBA |

Note that the us_id in the target account is 00000336 while in the source account the us_id was VAN_SUPER. A requirement for the import to be error free is that the us_usno and the us_id in both the source and target accounts are the same.

Import Roles with Errors

By selecting the VAN_ALL_ACC role above to be imported and allowing roles with errors to be imported, the resulting Import PDF is shown below.

Importing Role:VAN_ALL_ACC Details: 29 Assignments: 1

Creating new entry for Role for VAN_ALL_ACC

The import user id VAN_SUPER with us_no 972 for role VAN_ALL_ACC does not match the current user id 00000336 in the system, skipping.

Encountered problems importing role VAN_ALL_ACC, importing anyway.

In the above example, the role VAN_ALL_ACC is imported and the associated User VAN_SUPER is **NOT** imported due to the error listed that the US_NO and US_ID's in both accounts did not match.

Importing CDD and Documents Online Considerations

The Import utility checks the CDD folders and the Documents Online folders to verify the same structure exists in the Export and Import accounts.

The following message will be displayed in the PDF document generated by the Import process regarding CDD folders if there are no errors found:

Checking the import's Security model's CDD folders against the current model.
No problems detected for CDD folders.

If problems are detected with CDD folders, messages such as the ones below will be displayed in the PDF document

Warning! The CDD Folder 790 Standard Reports does not match up with your current Security Model, it will not be rolled.

Warning! The CDD Folder CDD IQ Standard Reports does not match up with your current Security Model, it will not be rolled.

Warning! The CDD Folder Client Reports does not match up with your current Security Model, it will not be rolled.

Warning! The CDD Folder Core Financials does not match up with your current Security Model, it will not be rolled.

The result is informational. The named CDD folders in the source account do not match the CDD folders in the target account.

Other Considerations

New Users

The intention of the security Import/Export process is to create and update security roles in an Export or test environment and apply those changes to matching users in the production or Import environment. New ID's, if created in the source(Export) account, will not be moved into or loaded in the receiving(Import) account. The result of this scenario is displayed below. The role created in the exporting account will be loaded into the importing account. The user created in the exporting will not be loaded into the importing account. In this example, the Import Roles with Errors parameter was active when the Import was run.

Importing Role:EXPORT_ROLE Details: 0 Assignments: 1

Creating new entry for Role for EXPORT_ROLE

The import user id PYACCESS with us_no 500 for role EXPORT_ROLE does not exist in the current system, skipping.

Encountered problems importing role EXPORT_ROLE, importing anyway.

Associations

Likewise, associations attached to users ID's in the exporting account will not be written to the Import account. Existing associations for a user in the import account will not be affected. That is to say the existing associations will not be lost, but new associations will not be added.

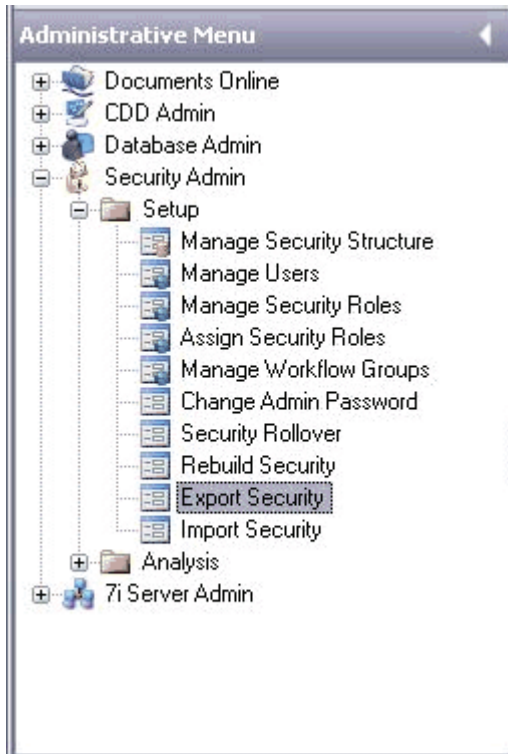
4.3.2 Security Export

Introduction

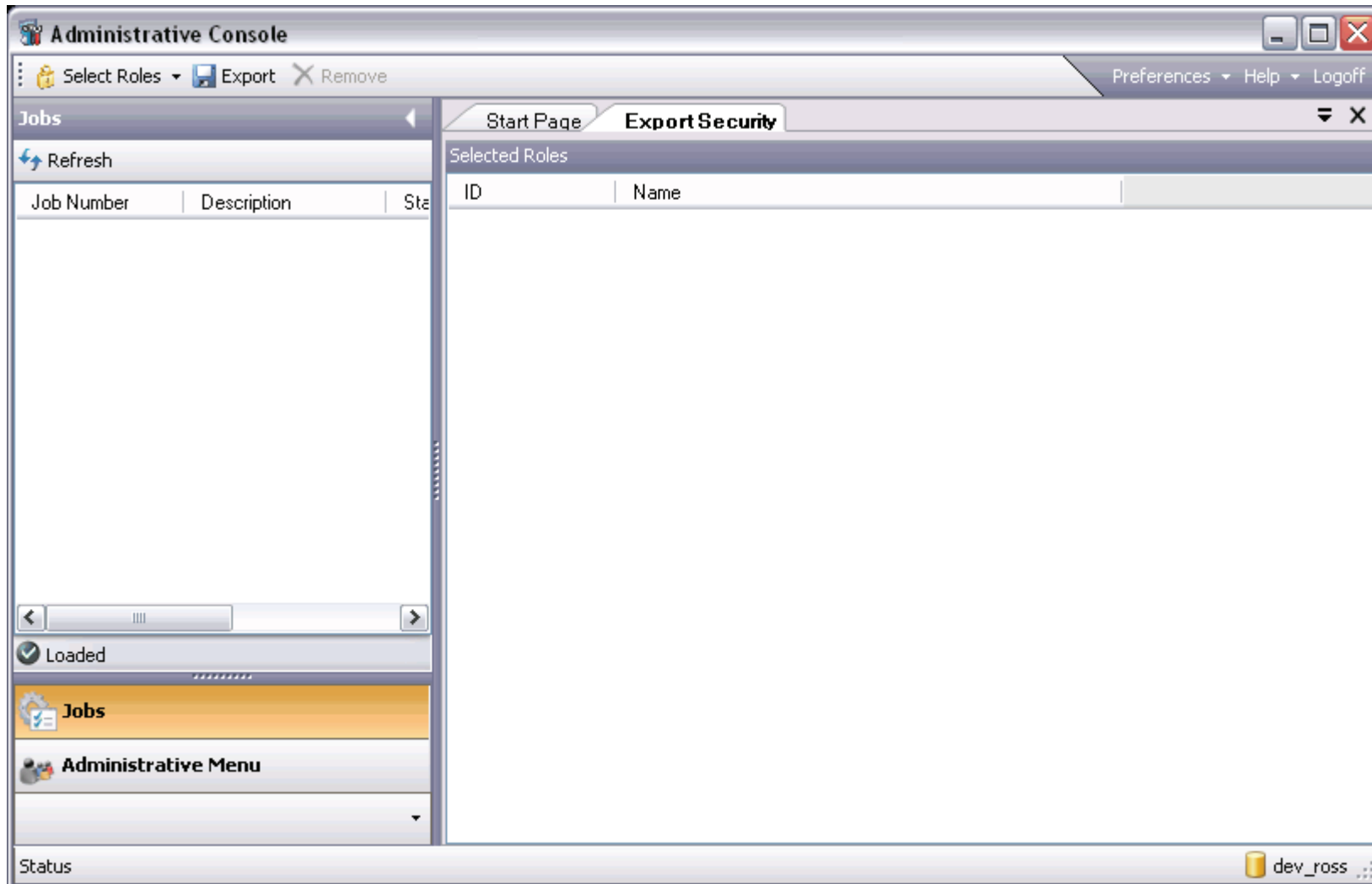
The "Export Security" plugin is used to create a file of role based security data to be moved from one account to another. This plugin is intended to move data from a mirror image account back to the originating account. That is to say the users in the source account are expected to be the same in the target account referring to the US_NO and US_ID from the US_USNO_MSTR. Use of this plugin is not intended to add new user records or update information on the user record other than assigning security roles. Its purpose is to move security role information to the import location.

Administrative Tool

The "Export Security" screen can be found within the administrative console, in the Security Admin / Setup section. This utility will allow you to export all roles, or a selected set of roles that can be moved to another account. It will unload the roles to an archived file (zip file). This process can be used to move security roles from test to production or simply to backup security roles prior to making changes.



Selecting this screen by double-clicking will bring up the view shown below:



Select Roles for Export

The 'Select Roles' menu item is used to choose the roles you wish to export. You can choose all the roles in the system with 'Select All Roles', or use the 'Find/Search' menu item to bring up the search dialog to choose individual roles:



Once the 'Find/Search' option brings up the Select Roles Dialog, enter your search criteria and click the apply button. You can then highlight the roles you wish to import and the click Ok. This process can be repeated as many times as needed to build your list of roles that is to be exported:

Select Roles

Filter

Role ID: Modified roles only

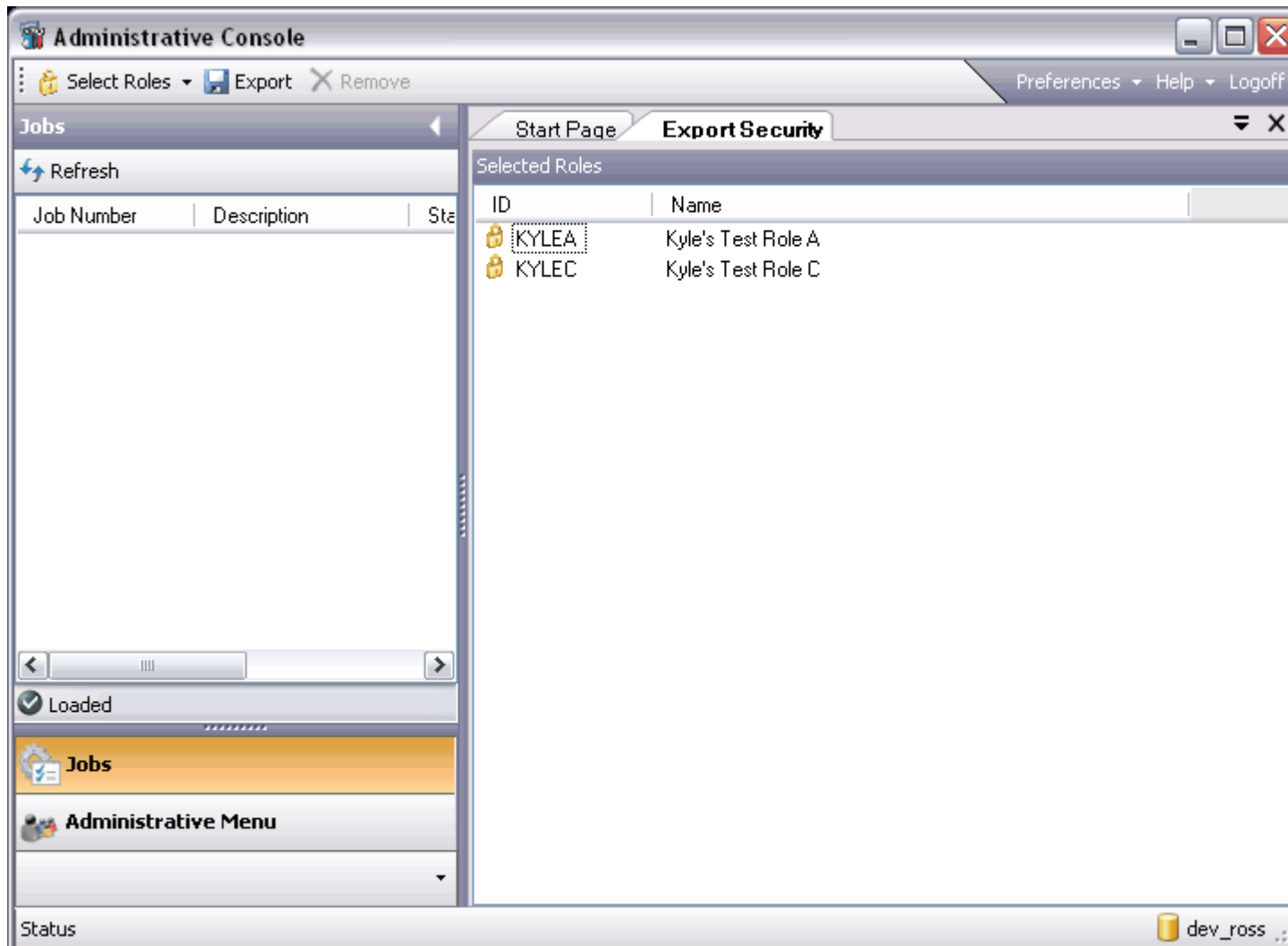
Role Title:

Use the "Apply" button to populate the list of roles based on the filter. Then select (highlight) the roles you wish to add to the list and click the "OK" button below.

Search Results

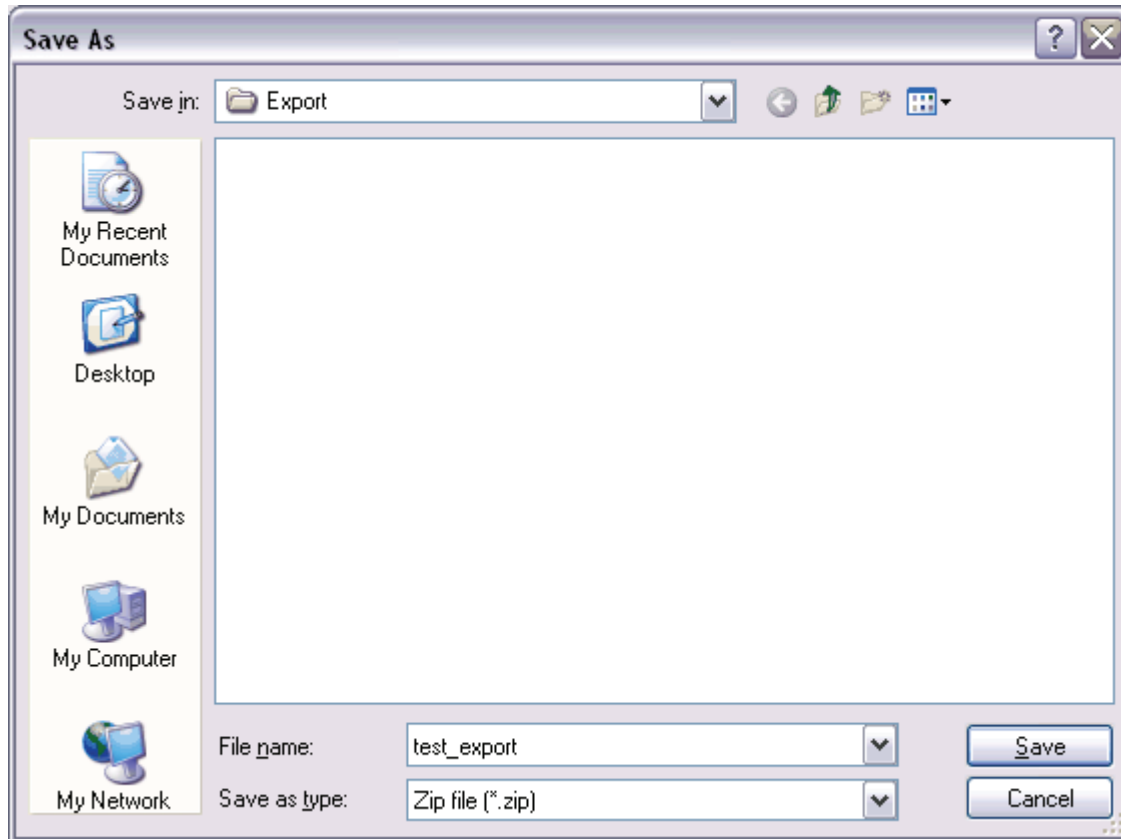
| Role ID | Role Title | Last Rebuilt |
|---------|--------------------|-----------------|
| KYLEA | Kyle's Test Role A | 12/14/2007 9:47 |
| KYLEC | Kyle's Test Role C | |

Below is a view of the screen after the roles have been added to the export list. You can remove any of the roles from the list by clicking on them to first highlight them. Then click the 'Remove' button at the top of the screen.

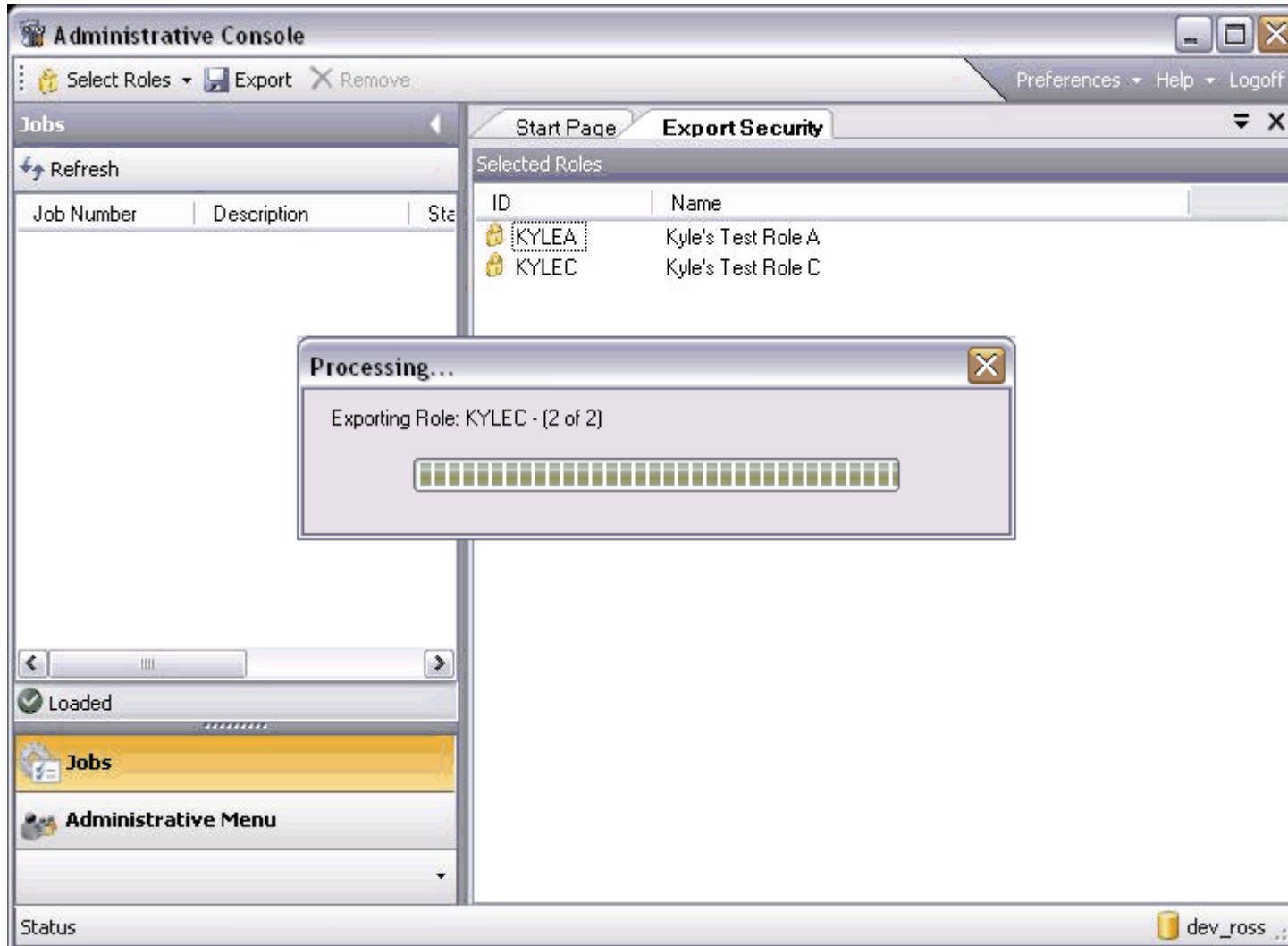


Exporting Roles

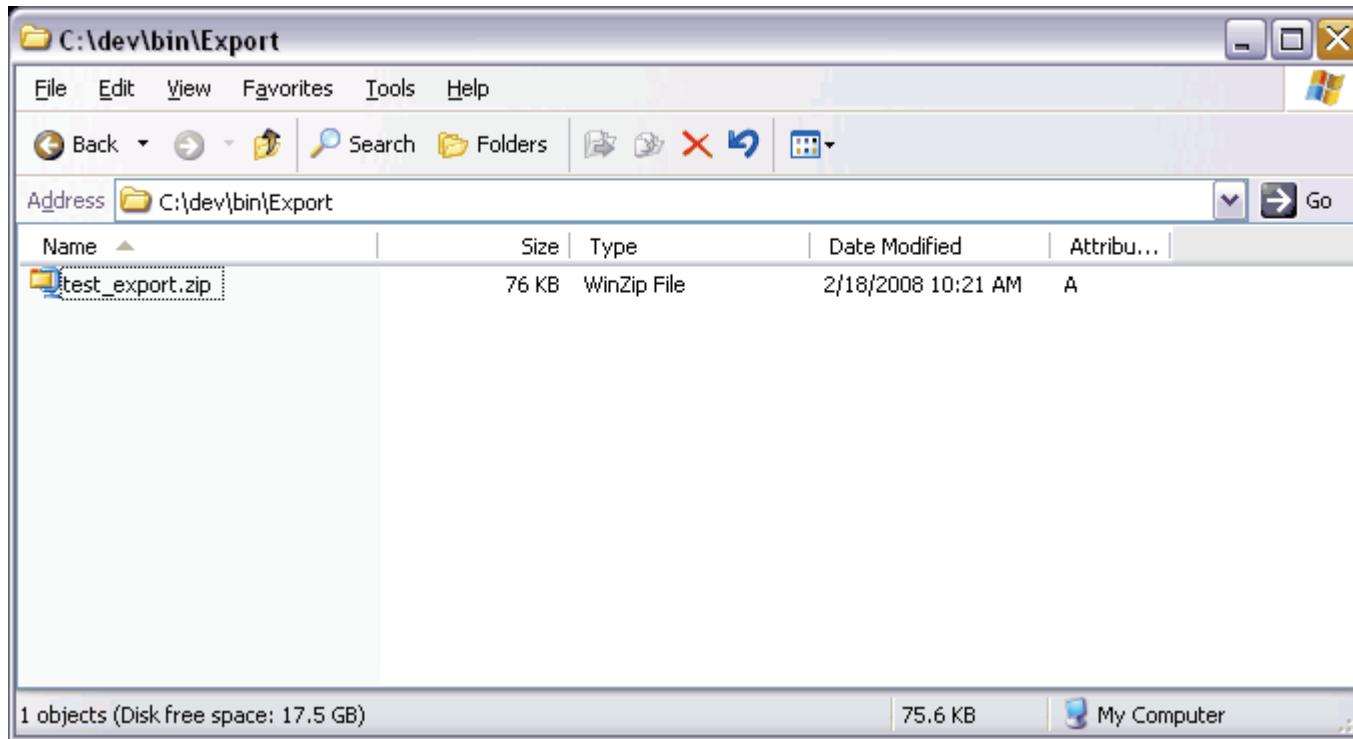
Once you have your list of roles completed, simply click on the 'Export' button in the toolbar to begin the export process. A 'Save As' file dialog will appear. Type in a file name for the archive that will be created for your export. The file extension must be .zip. It will automatically add the extension for you.



Click the Save button (see the above file dialog example) to begin the export and your archive (zip) file will be created. This file can be imported using the 'Import Security' Administrative Tool.



The resulting archive file:



4.4 Reports

Under Construction

5 Maintaining and Troubleshooting Security

5.1 Managing Security

Due to the constantly changing needs of both your organization and the software it's unlikely that your initial security setup will stand unchanged for long. It's important not to cling too tightly to your initial pass at its structure but look at it more as an ongoing process of modification and improvement.

5.2 Security Internals

Troubleshooting security problems can be made a bit easier by understanding how the Role-Based Security is handled by the application. The following is intended to provide a glimpse into the software internals of Role-Based Security.

5.2.1 Database Tables

The following tables are used in storing the Role-Based Security setup within the IFAS Database.

5.3 US_SECOBJ_MSTR

This table stores all of the Security Objects from the Security Structure. Everything that can possibly have security applied to it has an entry in this table. Initially, this table is generated from a combination of known security elements such as "CDD Reports" as well as the Masks defined in the Nucleus tables and the IFAS Tables found in the schema information. The "Manage Security Structure" web client screen allows maintenance and restoration of the contents of this table.

5.4 US_ROLE_MSTR

This table stores the master record for both Security Roles and Workflow Roles. For each role created in the system there should be one entry in this table containing both the Role ID and Role Title.

5.5 US_ROLESEC_DTL

Each Security Object (US_SECOBJ_MSTR) that has some kind of security explicitly assigned to it in a particular role gets an entry in this table. Security Objects marked as "derived" in a Security Role do not get an entry in this table.

5.6 US_ROLE_DTL

The assignment of both Security and Workflow Roles to users in the system is kept in this table.

US_ROLE_OUTPUT

Security Information used by 7i, CDD and the software running on the Application Server is generated from the above tables and stored in this table so that it can be consistently applied across applications. This table stores the individual security objects, what access the user has to and for database tables the filter information.

5.7 IFAS_DATA

This table is used by a number of IFAS modules for storing binary data. In the case of Role-Based Security the XML and XSLT representations of the Security Structure, Security Roles and User Security are generated from role information, and stored in this table.

5.7.1 Persisted Information

If the software had to read in all of the tables used to setup the Role-Based Security every time it needed to determine a user's access was it would make the software extremely slow. Therefore, to improve performance the software saves the output from those tables into a BLOB stored in the IFAS Database so that it does not have to read all of the details every time it needs security information. All of the BLOBs used by Role-Based Security are temporary storage only and if they are not found in the system they are simply regenerated by the software as-needed. The list below contains the IFAS_DATA categories used by Role-Based Security.

IFAS_DATA Blobs

| Category | Description |
|----------|--|
| XMLMODEL | A persisted XML representation of the full Security Structure. |

| | |
|------------|--|
| XSLTROLE | An XSLT representation of each Security Role in the software. These translations are applied one after another based on the user's role assignments to the XML Model enabling different aspects of the Security Structure. |
| XMLUSERSEC | A persisted version of each user's individual copy of the Security Structure after the XSLT role layers have been applied to it. |

Role Output

To help ensure that many different aspects of the system can share the same Security Information easily, a significant amount of each user's security is also saved in the US_ROLE_OUTPUT table. This way 7i, CDD and the software running on the Application Server can all share the same access information. However, it is possible for the information in this output table to become stale. One way to check for this is the User Security Audit.

To help ensure that this information is not outdated the Rebuild Security tool should be run anytime you change a user's Security Role assignments or whenever a Security Role is changed. The tool is available on both the Manage Users and Manage Security Roles web client screens and can be run for the current selection.

Technically all of the rows in this table can be regenerated at any time and no setup or configuration information is stored in them. However, the process of regenerating all Role Output for all of the users in the system can be time consuming and should not be done while the system is in use.

Server Caching

The 7i Server software maintains a number of worker threads for processing requests. One of the ways these threads have been made more efficient is by caching some of the security information so access requests can be fulfilled without going back to the database with each request.

Just as the Persisted BLOBs or Role Output tables can become outdated, so can the information on the 7i servers themselves. The "Rebuild Security" tools make an attempt at flushing the cache on the 7i Servers in the Server Farm but if you are experiencing network issues or there is a problem with the Server Group setup then it is possible that the server's cache will need to be manually flushed. The "Monitor Servers" screen available in the Admin Console provides a method of remotely connecting to a 7i server and running the "Flush Server Cache" tool against that server.

As always, in the event that you still do not believe the server's cache has been flushed properly you may find it necessary to take that server out of Network Load Balancing and restart services in order to be sure the cache has fully been cleared. Certainly this is not considered to be a normal requirement of the software.

It is important to note that flushing the server cache will have an adverse impact on the performance of the software. Obviously, if the software is not performing correctly at all this is a necessary interruption for users. Otherwise, it would be best if this could be done outside of peak usage times.

5.8 Troubleshooting Security Problems

In a perfect world once your security setup was in place it would all behave exactly as intended. Unfortunately, the reality is that from time to time you may be required to troubleshoot situations where a user's resulting security is not what was expected. In those situations, there are some tools available to help identify the cause of the problem.

First, the Role Simulator is a screen launched from the Admin Console that will help you to test the security for a given user. It allows you to view the security for a particular role combination as it would appear for Table or Menu Access. The intent is to provide a method of not only seeing what a user would currently have access to by selecting their current roles, but to estimate what the impact of additional role assignments might be without actually granting access to that in the system. The Help on that screen should provide additional details about its usage.

5.8.1 User Security Audit Report

The software has a built in method of checking on a user's security. While this report is unable to know what a user is supposed to have assigned, it does look for common security issues such as a user having no data access at all or the role output being outdated for that particular user. This report can be run from the tools panel of the "Manage Users" web client screen from the Admin Console or by running the NUUPUS mask from the 7i menu.

5.8.2 Rebuild User Isn't Working

The vast majority of security issues will probably come down to the need to "Rebuild Security". If this tool has been launched from either the Manage Users (NUUPUS) or Manage Security Roles (NUUPSR) web client screens but does not seem to be taking effect then the next step is to check the Job Manager to see if the tool is stuck waiting to be processed by Workflow. If that is the case making sure Workflow is running and that the "REBUILD_SECURITY" model is active is a good first step. Once the model is functioning again the security can be rebuilt. There is also a Rebuild Security screen that rebuilds directly, getting around any Workflow usage or issues.

5.8.3 User Lacks Expected Menu Access

Using the Role Simulator it's possible to recreate the menu access of a particular role combination. This is a good way to not only test the user's current security but to also test the impact of other roles on their menu access. If the role simulator shows their menu access to be what you expected but they are still unable to view the information in IFAS then most likely the fault lies with one of the areas where the security output is stored.

- In the Application Server components of the software, the menu access is read in from the Role Output table. If this is incorrect then the Role Output should be rebuilt using the "Rebuild Security" tool from the Manage Users web client screen. (NUUPUS)
- In 7i the Menu is actually read from the User's XML. This is stored in the IFAS_DATA table and also cached locally within the 7i worker threads. If the security looks correct on some servers but not all then most likely the IFAS_DATA information is correct and the incorrect servers simply need to be flushed. This can be done from the "Monitor Servers" screen in the Admin Console. Otherwise, once again using the "Rebuild Security" tool should help to reset the IFAS_DATA xml representations.

5.8.4 User Lacks Expected Data Access

The Role Simulator is the best way to test a user's access to a particular table. The simulator not only factors in the access to a particular table but access to the Common Security items involved as well. Once you have used the simulator to return a where clause based on role assignments and the desired access (Read, Write, Update, Delete, Execute), that where clause can then be used in an SQL query against that table to see just how many rows would be returned.

For the most part, a user's Data Access security is all read from the US_ROLE_OUTPUT table. This table not only stores whether or not they have access to a particular table but what the filter will be on that table. If this data does not match the user's actual access then a "Rebuild Security" is required to get the table rebuilt correctly.

Some Examples of Data Access Usage:

| Usage | Table Access Required |
|--------------------|-----------------------|
| 7i Screen (Browse) | Read |
| 7i Screen (Add) | Write |
| 7i Screen (Save) | Update |
| 7i Screen (Delete) | Delete |
| CDD Report | Read |

5.8.5 When and How to Use Tracing for Security

If there appear to be problems in the way the software is handling security there are some modules that can be used to track down the problems. Tracing for a particular module can be enabled using the "Configure Local Server" screen in the Admin Console on each of the 7i servers. Keep in mind that this tracing information is fairly technical.

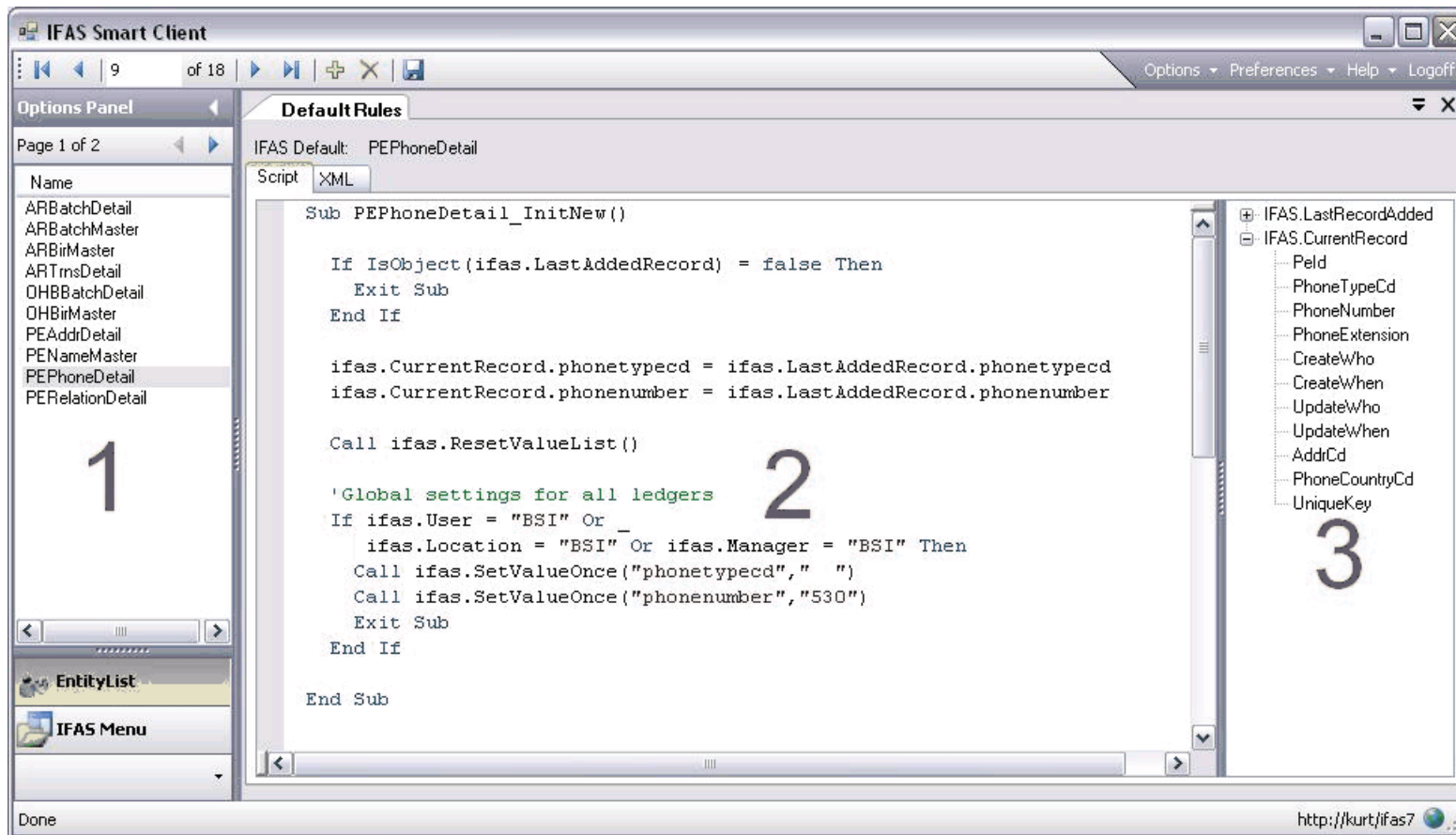
| Module | Usage |
|----------------|---|
| BT20NU | Used to generate the persisted versions of the XML Model and XSLT Roles. |
| BT30NU | Used to handle requests for User Security XML and for individual security checks on Menu Masks and Application Functionality. This module is used to cache security information within 7i. Also, this module contains the user login logic for both CDD and 7i. |
| BT30NU.Managed | This module is used to process the User Security XML into more complex information such as the specific filter on a table. |
| BT70NU | This is the Utility module used by the "Rebuild Security" tool to populate the Role Output table. This is also the module used to create the User Audit Report. |

6 Advanced/Special Configuration

6.1 Defaults Rules (NUUPDF)

The System Default Definition screen is used to define certain data elements that could default for users during data entry. Logic can be used to default in values for new records, or updated records based on what has been already entered. Each Default consists of 2 parts: VBScript and XML. The VBScript is used to create subroutines that will be called by 7i when business rules are fired on a particular BT20 (table). The XML is used to define which routine will be executed for each kind of event that can happen on a BT20.

Below is a sample screenshot of the NUUPDF screen, which runs in the IFAS Smart Client only and not as a thin 7i web screen.



The screen is made up of 3 parts:

Entity List for navigating existing defaults. This is a list of each BT20's default that has been created. New ones can be added by clicking the '+' button in the navigation bar.

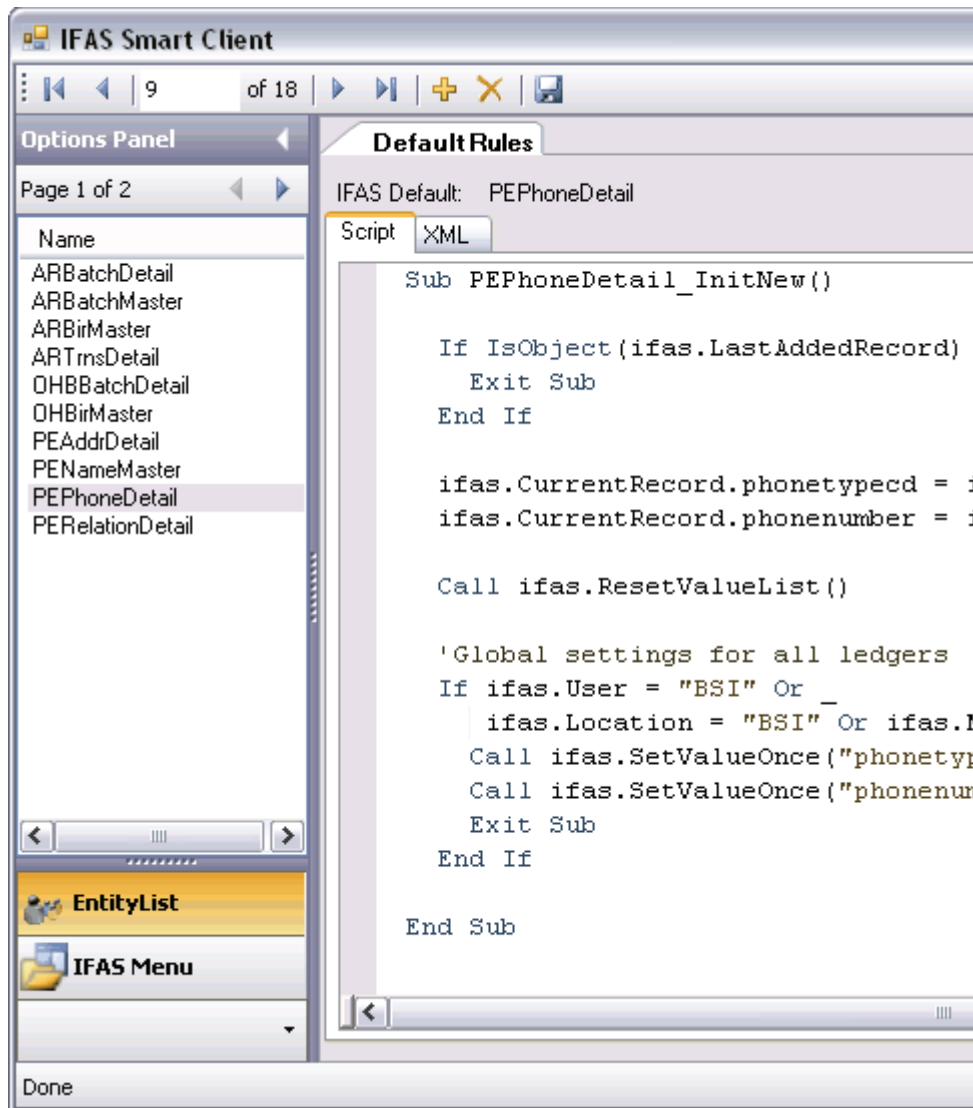
VBScript/XML Editor. This is where you can edit the default rules that will be executed for the selected BT20.

BT20 Tree View. This view will allow you to see the attributes for the selected BT20 that can be used in the VBScript/XML editors.

6.1.1 Entity List

The Entity List allows navigation of existing records and the ability to delete and create new ones. The navigation bar at the top of the screen can be used to choose records or simply click the one you wish to view in the entity list with the mouse. When you are done editing a default (or creating a new default), click the save button to save your changes to the database. Both the XML and VBScript are stored in the database in a BLOB to be accessed by the 7i server. When the save is complete, you will receive a 'Record Accepted' message. At this time, the screen has also notified the 7i server to reload the defaults it has loaded so your new changes are ready to take effect. If you (or any other users) have any open 7i screens simply exit them and open new ones for the new defaults to take effect.

CAUTION: If you delete a record, it is permanently destroyed and the defaults will have to be recreated.

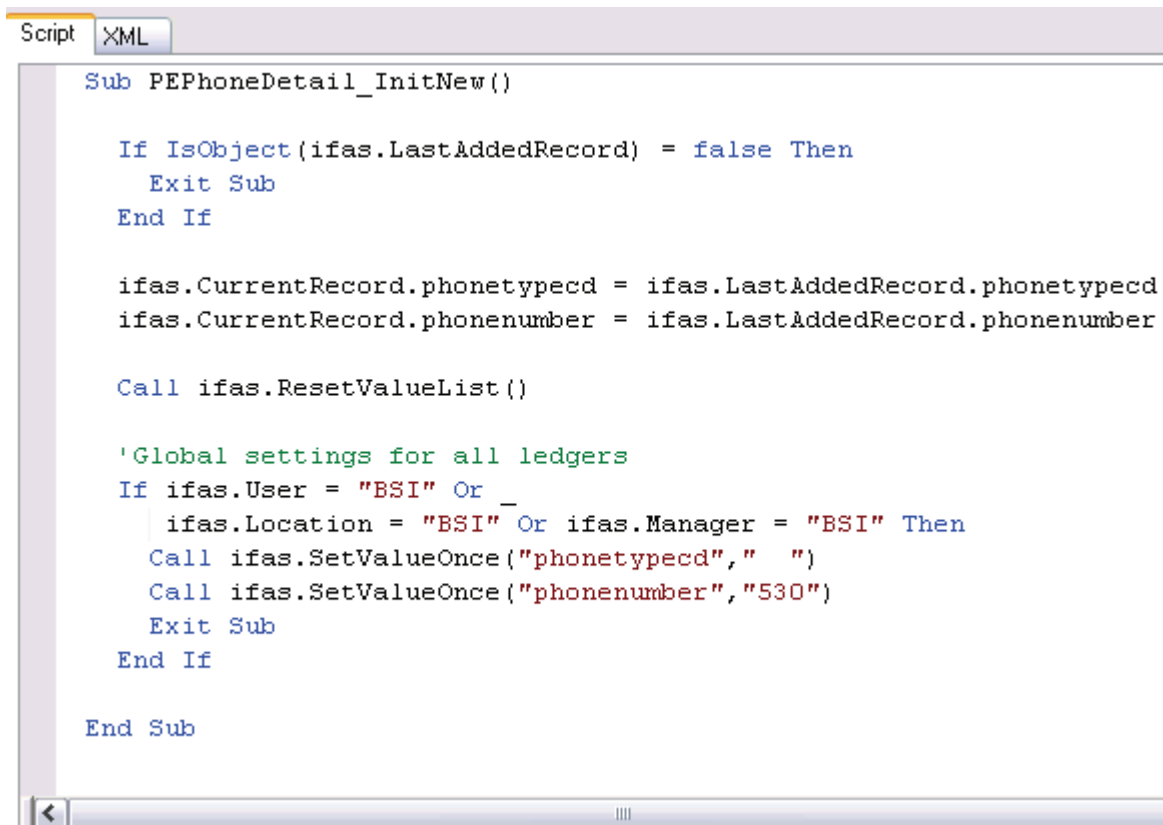


6.1.2 VBScript Editor

There are two separate tabs, one for editing the VBScript that will be run when a business rule in 7i is fired, and the other to tell 7i which routine to execute on a business rule.

The IFAS defaults are written with the VBScript language. The editor will highlight key words for you to assist in writing your defaults. Once you make a change to the script and save the record, the changes will immediately take effect in 7i. We recommend that you first test your script changes in a test account to avoid encountering problems in your live environment when developing your defaults.

Below is a sample of an InitNew routine that will be setup to be called when a new record is created when you go into add mode in a 7i screen.



```
Script XML
Sub PEPhoneDetail_InitNew()

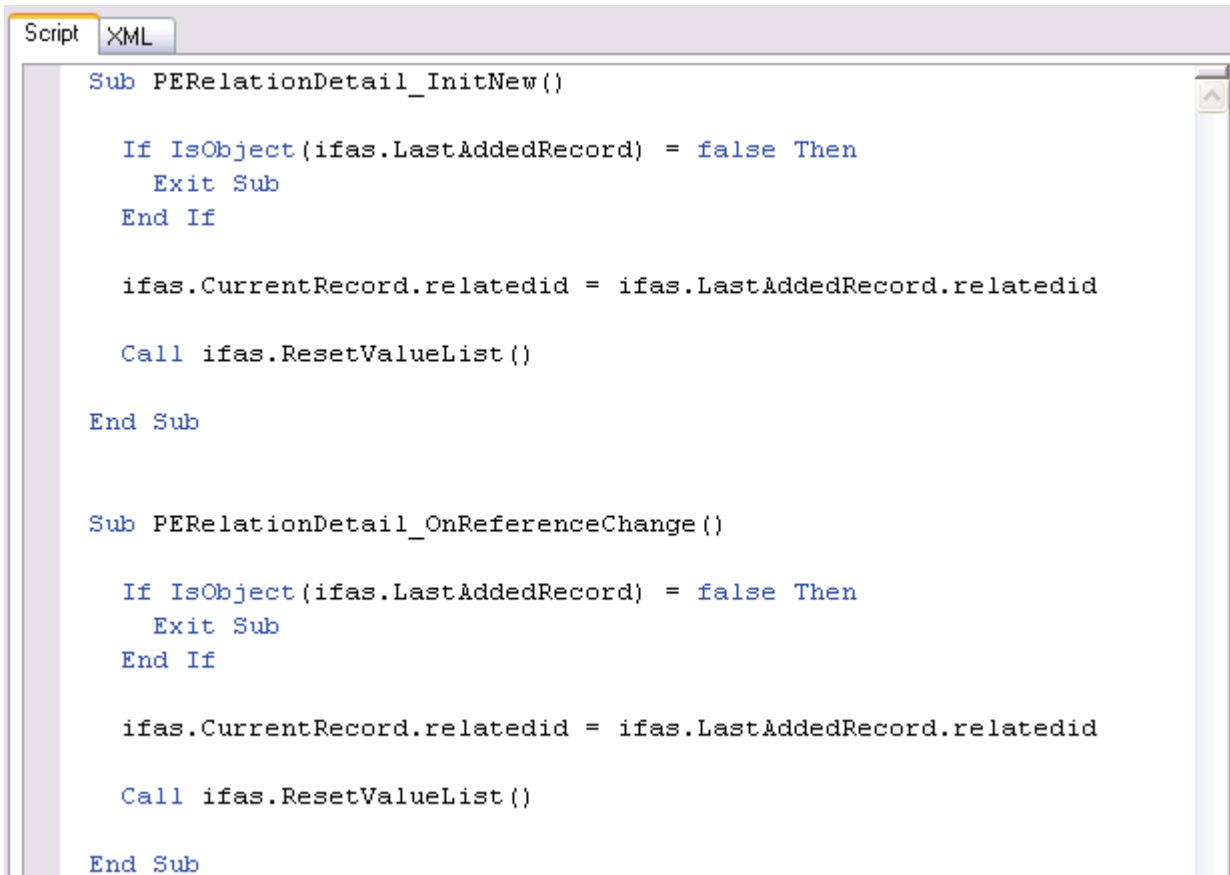
    If IsObject(ivas.LastAddedRecord) = false Then
        Exit Sub
    End If

    ivas.CurrentRecord.phonetypecd = ivas.LastAddedRecord.phonetypecd
    ivas.CurrentRecord.phonenumber = ivas.LastAddedRecord.phonenumber

    Call ivas.ResetValueList()

    'Global settings for all ledgers
    If ivas.User = "BSI" Or _
        ivas.Location = "BSI" Or ivas.Manager = "BSI" Then
        Call ivas.SetValueOnce("phonetypecd", " ")
        Call ivas.SetValueOnce("phonenumber", "530")
    End If
End Sub
```

More than one routine can be specified in the script editor. Simply separate them by a blank line:



```
Script XML
Sub PERelationDetail_InitNew()

    If IsObject(ivas.LastAddedRecord) = false Then
        Exit Sub
    End If

    ivas.CurrentRecord.relatedid = ivas.LastAddedRecord.relatedid

    Call ivas.ResetValueList()

End Sub

Sub PERelationDetail_OnReferenceChange()

    If IsObject(ivas.LastAddedRecord) = false Then
        Exit Sub
    End If

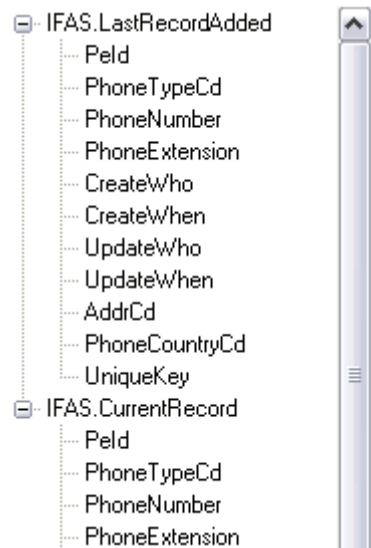
    ivas.CurrentRecord.relatedid = ivas.LastAddedRecord.relatedid

    Call ivas.ResetValueList()

End Sub
```

There are some built in variables and functions that can be used within your scripts. Below is a list:

ivas.CurrentRecord. – The data structure that holds the current record on the screen. Each column of the BT20 is defined on the data structure. See the screen shot below for an example on the PEPHONEDETAIL.



ifas.LastAddedRecord – The same data structure as above, but it contains the last record that was added through the 7i screen.

ifas.SetValueOnce(property, value) – Sets the named 'property' on the BT20 to provided 'value'. If a second call to ifas.SetValueOnce is made within the same script, it will not update the value. Calling ifas.ResetValueList() will allow additional changes to be saved.

ifas.ResetValueList() – Clears the set property list so that ifas.SetValueOnce() can set new values.

ifas.User – Retrieves the user id of the current user in the 7i screen.

ifas.UserName - Retrieves the Users' Name off of the us_usno_mstr.

ifas.UserDesc - Retrieves the Users' Description off of the us_usno_mstr.

ifas.Manager - Retrieves the Users' Manager off of the us_usno_mstr.

ifas.Location – Retrieves the Users' Location code off of the us_usno_mstr.

ifas.CurrentScreenMask - Retrieves mask of the current 7i screen.

ifas.ClientId - Retrieves the Client Id from the SYID CLIENTID common code.

ifas.GetBackgroundPart(part, ledger, key, obj) – Retrieves the key or object background part for the requested ledger.

ifas.AddMode – Returns 1 if the screen is in add mode, set to 0 otherwise.

ifas.IFASVersion – Retrieves the current IFAS Version that the 7i server is running.

ifas.CurrentLedger.GILedger – Retrieves the current user's GILedger.

ifas.CurrentLedger.JILedger – Retrieves the current user's JILedger.

ifas.TranslateText – Retrieves translated strings for known tokens such as "\CD8" will return the date in YYYYMMDD format. Below is a list of acceptable values.

Ifas.TranslateText changes all '\xxxx', where:

\Xxxx = 'G' GL side or 'J' JL side

\xXxx = 'K' key part, 'O' object part

'L' Ledger, 'B' Budget

\xxXx = 'F' Fund

'X' Function

'O' Budget officer

'D' Description (S,M,L valid)

'1-8' Array part (S,M,L valid)

'A' Account Type

'B' Budget Category

'9-0' Array part (only valid for xBXx)

'W' Working Budget (S,M,L valid)

\xxxX = 'S', 'M', 'L' short, medium, or long

\xTn = GL Account title from GEN master with

n = 'B' meaning both desc's.

Other options

\xSnn = Subsystem ID (2 char.); n=01 through 10 for

the 10 subsystems defined in GLG-GEN-MSTR

\xPnn = Period name; n=01 through 14

\xDyxxx = Symbolic Date conversion

y = Date Output Format (eg S-Z,6,8)

xxx = Symbolic Date (eg FYB,FYE,...)

\xMnn = Misc. Code desc.; n=01 through 08

Other non-ledger/ar specific

\USER = Upshifted USER ID

\CDx = Current date

Currently supported values of 'x' date format:

T = MM/DD/YYYY

8 = YYYYMMDD

M = HHMMSS (time)

Script examples

Setting a field value if a condition is true. The below script is a sample for the PEAddrDetail which is the BT20 for the Address Detail in PE. In the InitNew routine, the script sets the zip code to "95973" and the city to 'Chico' if the User's Location is set to 'CHICO'. Every time a user with this Location adds a new address record in PEUPPE, the city and zip will default in. Notice we call ifas.ResetValueList() first to ensure that our ifas.SetValueOnce() calls will take effect and save their values.

```
Script XML
Sub PEAddrDetail_InitNew()

    Call ifas.ResetValueList()

    'Default in the City and Zip for users in the home office of chico.
    If ifas.Location = "CHICO" Then
        Call ifas.SetValueOnce("City","Chico")
        Call ifas.SetValueOnce("Zip","95973")
        Exit Sub
    End If

End Sub
```

Setting a field value if a condition based on another field is true. The below is a sample for the PEAddrDetail's PreAccept routine (see the XML tab section for more information on PreAccept). Notice we call `ifas.ResetValueList()` first to ensure that our `ifas.SetValueOnce()` calls will take effect and save their values.

We then up shift the City on the current record and check to see if it equal to "CHICO". If it is, we set the zip code to 95973.

```
Script XML
Sub PEAddrDetail_OnPreAccept()

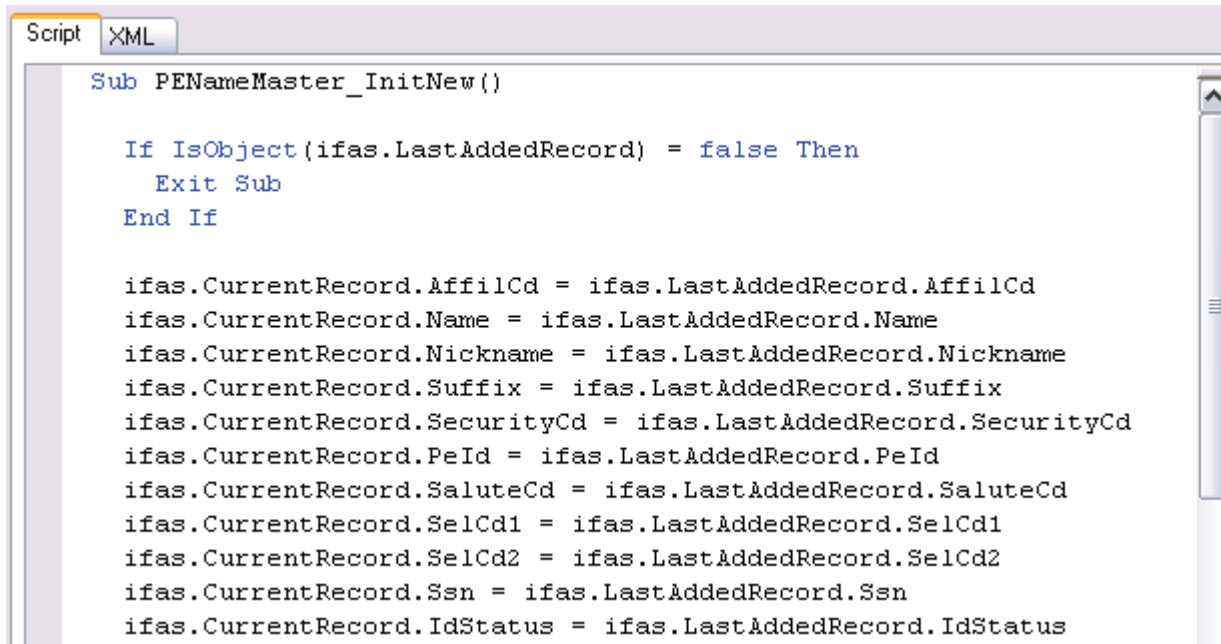
    Call ifas.ResetValueList()

    'Default in the Zip if the city is Chico
    If UCASE(ifas.CurrentRecord.City) = "CHICO" Then
        Call ifas.SetValueOnce("Zip","95973")
        Exit Sub
    End If

End Sub
```

Setting field values from the last added record. The below script is a sample for the PENameMaster which is the BT20 for the Name Master in PE. If a user is going to add more than one record in succession, we can set fields on the new record to what was previously entered to help

speed up data entry. In the InitNew routine, the script first checks to see the user has previously added a record in the screen. Then the various fields on the record are set on the CurrentRecord from the LastAddedRecord.

A screenshot of a script editor window with a tab labeled 'Script' and 'XML'. The code is as follows:

```
Sub PENAMEMASTER_InitNew()  
  
    If IsObject(ifas.LastAddedRecord) = false Then  
        Exit Sub  
    End If  
  
    ifas.CurrentRecord.AffilCd = ifas.LastAddedRecord.AffilCd  
    ifas.CurrentRecord.Name = ifas.LastAddedRecord.Name  
    ifas.CurrentRecord.Nickname = ifas.LastAddedRecord.Nickname  
    ifas.CurrentRecord.Suffix = ifas.LastAddedRecord.Suffix  
    ifas.CurrentRecord.SecurityCd = ifas.LastAddedRecord.SecurityCd  
    ifas.CurrentRecord.PeId = ifas.LastAddedRecord.PeId  
    ifas.CurrentRecord.SaluteCd = ifas.LastAddedRecord.SaluteCd  
    ifas.CurrentRecord.SelCd1 = ifas.LastAddedRecord.SelCd1  
    ifas.CurrentRecord.SelCd2 = ifas.LastAddedRecord.SelCd2  
    ifas.CurrentRecord.Ssn = ifas.LastAddedRecord.Ssn  
    ifas.CurrentRecord.IdStatus = ifas.LastAddedRecord.IdStatus
```

ADVANCED: Setting values with TranslateText. The below script is a sample for the ARBirMaster which is the BT20 for the AR Batches in ARBTARUB. This script first checks the current screen to be ARBTARUB. If it is not that screen, the script exists. At the end, it checks the GILedger code to be "GL". If it is, it sets the "RefDt" equal to the current date in "MM/DD/YYYY" format with `ifas.TranslateText("\CDT")`.

```
Script XML
Sub ARBirMaster_AR_InitNew()

    If not (CurrentScreenMask = "ARBSTARUB") Then
        Exit Sub
    End If

    If IsObject (ifas.LastAddedRecord) = false Then
        Exit Sub
    End If

    ifas.CurrentRecord.RefDt = ifas.LastAddedRecord.RefDt

    Call ifas.ResetValueList ()

    'Global settings for all users
    If ifas.CurrentLedger.GlLedger = "GL" Then
        Call ifas.SetValueOnce ("RefDt", ifas.TranslateText ("\CDT"))
    End If

End Sub
```

ADVANCED: Reading common codes. This is an example of reading a common code from a default script and setting the "Addr3" (Address line 3) field equal to the Medium description of the common code. When reading common codes, you must specify all 3 columns, the Gr, Category, and CdCode or else the common code will not be fetched. A returned result of 1 from ReadByKey on a BT20 object means we successfully read the record we were searching for.

```
Script XML
Sub PEAddrDetail_PreAccept()

    Call ifas.ResetValueList()

    dim codesReader
    set codesReader = CreateObject("BT20.CDCodesMaster")

    codesReader.Gr = "@@"
    codesReader.Category = "PECO"
    codesReader.CdCode = "US"

    Dim result
    result = codesReader.ReadByKey()
    'A result of one means we successfully read the record
    If result = 1 then
        Call ifas.SetValueOnce("Addr3", codesReader.DescM)
    End If
End Sub
```

ADVANCED: Issuing error messages. This is an example of how to issue an error message from a default. Please refer to the "7i System Documentation" on how to create your own custom errorcat with your error messages. The ErrorLevel can be set to a 1 for an error, 2 for a warning and 4 for informational. The below example is for a warning. The replaceable fields with the '~' are not implemented from default scripts at this time. You may only issue static error messages.

```
Script XML
Sub PEAddrDetail_PreAccept()

    Call ifas.ResetValueList()

    If UCASE(ifas.CurrentRecord.City) = "CHICO" then
        dim btError
        set btError = ifas.AddErrorMessage("PE", 32)

        btError.ErrorLevel = 2
    End If
```

6.1.3 XML Editor

The XML editor is used to define which VBScript business rules will be fired when certain events happen on a BT20. Below is an example of the PE Address Detail with a few routines defined. The XML needs to be defined in this specific way with the <XML> and <BUSINESSRULES> nodes. Remember, XML is case sensitive.

CAUTION: Creating an error with the XML definitions will prevent 7i from saving records for that BT20.

The image shows a screenshot of a software window titled 'Script XML'. The window contains XML code defining business rules for a BT20 object. The code is structured as follows:

```
<XML>
  <BUSINESSRULES>
    <BT20.PEAddrDetail.1>
      <INITNEW>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_InitNew" />
        </RULEOBJECT>
      </INITNEW>
      <PREACCEPT>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_PreAccept" />
        </RULEOBJECT>
      </PREACCEPT>
      <AFTERFIELD>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_OnChangePEID_GLOBAL_001">
            <BT20OBJ>
              <RETPROP>PeId</RETPROP>
            </BT20OBJ>
            <TRIGPROP>PeId</TRIGPROP>
          </METHOD>
        </RULEOBJECT>
      </AFTERFIELD>
      <CHANGEREf>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_OnReferenceChange" />
        </RULEOBJECT>
      </CHANGEREf>
    </BT20.PEAddrDetail.1>
  </BUSINESSRULES>
</XML>
```

XML Syntax

Below is sample XML with all of the available nodes defined that can be set to perform different actions on the BT20.

```
<XML>
  <SORTS></SORTS>
  <FILTERS></FILTERS>
  <COLDATA></COLDATA>
  <BUSINESSRULES>
    <BT20. PEAddrDetail.1>

    <INITNEW></INITNEW>
    <TAGNAME></TAGNAME>
    <AFTERFIELD></AFTERFIELD>
    <PREACCEPT></PREACCEPT>
    <PREINSERT></PREINSERT>
    <POSTINSERT></POSTINSERT>
    <PREUPDATE></PREUPDATE>
    <POSTUPDATE></POSTUPDATE>
    <PREDELETE></PREDELETE>
    <SetControlProperties></SetControlProperties>
    <TOOLS></TOOLS>
    </BT20. PEAddrDetail.1>
  </BUSINESSRULES>
</XML>
```

Node definitions

```
<XML>
```

Required starting XML tag.

```
<SORTS>
```

Used to make client defined sorts. Beware that adding a sort on columns that are not indexed will cause a big hit in performance.

Here is a sample to sort the addresses by City:

```
<SORTS>
  <INDEX desc="By City">
```

```

    <PROP>City</PROP>
</INDEX>
</SORTS>
<FILTERS>
    Not for use at this time.
<COLDATA>

```

Used to specify which columns are required. Below is a sample to make City required on the PEAddrDetail:

```

<COLDATA>
    <City><REQUIRED/></City>
</COLDATA>
<BUSINESSRULES>

```

Used for executing business rules on a BT20 object when certain events occur on that object. The node directly under <BUSINESSRULES> must be the exact BT20 name that the default has been created for with a .1 on the end of it.

Example for PEAddrDetail:

```

<BUSINESSRULES>
    <BT20. PEAddrDetail.1>

```

...

```

</BT20. PEAddrDetail.1>
</BUSINESSRULES>

```

Each triggering event node will have a similar format with a RULEOBJECT node and a METHOD NODE. In all cases, the SCRIPTLOCATION attribute will need to be set to "Defaults". The SCRIPTNAME attribute will need to be the same name as the Default. The METHOD node's ID attribute will be the name of routine you will create in the VBScript for this default.

Example for PEAddrDetail:

```

<RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_InitNew" />
</RULEOBJECT>

```

Each node under the BT20 node is a triggering event on the BT20, which can fire a business rule from the VBScript that is created in the other VBScript tab. Below is each type of triggering event and when it gets fired.

<INITNEW>

This will execute a VBScript routine when in add mode and a new record is initialized. The default 7i INITNEW will fire first, then, your custom one defined here can set any changes after.

```
<INITNEW>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_InitNew" />
  </RULEOBJECT>
</INITNEW>
```

<TAGNAME>

This will execute a VBScript routine when the tags on the screen are created. You can set the tag names from your script. A screencompile will need to be performed for the tags on the screen to take effect. They are created by the screencompile routine. Note: If 7i is already setting the tagname, setting it in your own custom script will not override it. Setting it should only be done on additional fields you have added.

```
<TAGNAME>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_TagName" />
  </RULEOBJECT>
</TAGNAME>
```

Script sample:

```
Sub PEAddrDetail_TagName ()

    Call ifas.CurrentRecord.SetTag("ZIP", "Zip Code")

End Sub
```

<AFTERFIELD>

This will execute a VBScript routine when a field on the screen is left and its value has changed. For example, changing a field from '1' to '2' and tabbing out of that field will trigger this event.

There is additional triggering information on this event. The <BT20OBJ> node is to signify that we are going to pass the BT20 object (CurrentRecord) into the routine. The <RETPROP> node below it specifies that we are going to allow changes to the PeId field and return its changes. The <TRIGPROP> node is used to specify which field will trigger this Afterfield method.

```
<AFTERFIELD>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_OnChangePEID_GLOBAL_001">
      <BT20OBJ>
        <RETPROP>PeId</RETPROP>
      </BT20OBJ>
      <TRIGPROP>PeId</TRIGPROP>
    </METHOD>
  </RULEOBJECT>
</AFTERFIELD>
```

Script sample:

```
Sub PEAddrDetail_OnChangePEID_GLOBAL_001(Obj)

  Call ifas.ResetValueList()

  'Global settings for all ledgers
  Call ifas.SetValueOnce("Addr1","123 Main St.")

End Sub
```

<PREACCEPT>

This will execute a VBScript routine when a record has been submitted by the screen to be inserted or updated into the database. It will trigger this business rule prior to the update/insert taking place. If the business rule has an error, the update/insert will not happen.

```
<PREACCEPT>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_PreAccept" />
  </RULEOBJECT>
</PREACCEPT>
```

<PREINSERT>

The same as PREACCEPT, except it is for inserted records only and not updated records.

<POSTINSERT>

The same as PREINSERT, except it is triggered after a record has been inserted instead of before.

<PREUPDATE>

The same as PREINSERT, except it is for updated records only and not inserted records.

<POSTUPDATE>

The same as POSTINSERT, except it is triggered after a record has been updated instead of before.

<PREDELETE>

The same as PREACCEPT, except it is triggered after a record has been deleted instead of before.

<SetControlProperties>

Fields on the screen can be given default behavior by setting them in the SetControlProperties node. A screencompile and reopening the screen must be performed for the VBScript changes to take effect.

```
<SetControlProperties>  
  <Url ScreenMask="PEUPPE" Qbe="1" Add="1" Update="1" Init="1"  
    PreserveCase="false" Length="2" Enabled="false"/>  
</SetControlProperties>
```

Below is a list of what can be set on each field. The PENAMEMaster's Url field was used as an example.

ScreenMask – Set this equal to the mask that you want this control property to apply to. Leaving it off applies to all masks.

Qbe – Set this equal to "1" to disable the control in Find mode.

Add – Set this equal to "1" to disable the control in Add mode.

Update – Set this equal to "1" to disable the control in Update mode.

Init – Set this equal to "1" to disable the control in when a record is first initialized.

Length – Set this to limit the length of accepted data in the control.

Enabled – Set this equal to "false" to allow setting the update/add/init/qbe attributes to "1" to disable.

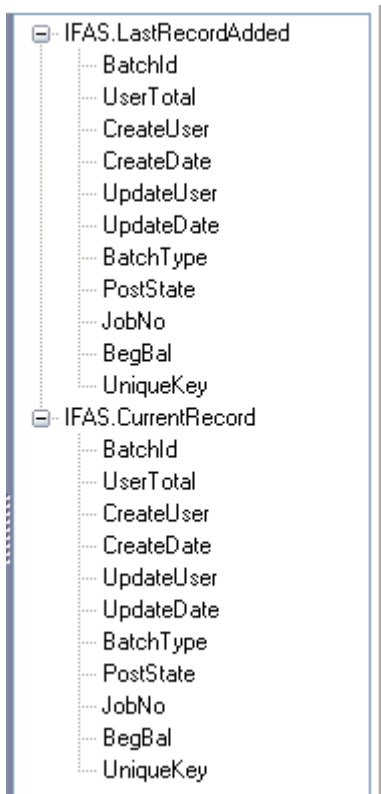
PreserveCase – Set to false to upshift the control's values.

<TOOLS>

Not for use at this time.

6.1.4 BT20 Tree View

The tree view displays BT20 information that can be used to help you write your VBScripts and XML definitions for defaults. Any of the BT20 columns can be dragged and dropped into the editor with the mouse. Double clicking an item will insert it where the cursor is positioned.



6.2 Security Recovery

In the event that something catastrophic happens and you no longer have an administrative login to the system it is possible to restore enough access to get things working again. In extreme cases, these recovery steps will require significant access to the system and are not considered to be a normal part of the application's functionality.

There are 2 main types of security data: (1) data that is generated or persisted from the security definitions, and (2) the actual security definitions. Recovery of this data depends on which of these two types of data were lost or corrupted.

The data that is generated and persisted is generally what IFAS looks at to determine access at runtime, and this data can be regenerated from the definitions using the Rebuild Security tools discussed above. The data in this category includes the US_ROLE_OUTPUT table, and the rows in IFAS_DATA where the CATEGORY column is equal to "XMLMODEL", "XMLUSERSEC", or "XSLTROLE". This data is recreated or repaired when security is rebuilt.

The more troublesome issue is when the security definitions themselves become damaged. The actual security definitions are stored in the following tables: US_SECOBJ_MSTR, US_ROLE_MSTR, US_ROLE_DTL, and US_ROLESEC_DTL. These are discussed above in the Security Internals section. Depending on what data in these tables is lost or corrupted, there are several recovery paths.

In the event that the Security Structure (US_SECOBJ_MSTR) has become damaged, you may be able to recover the missing pieces by using the Manage Security Structure screen in the Admin Console to rebuild the security. This will analyze your current setup looking for gaps in the expected structure and will proceed to fill in any missing security elements. In most cases this should be a harmless recovery step. There are several parts of the Security Structure that are client defined, and cannot be recovered. These include CDD folders and anything client-defined under the Custom folder of the Security Structure. For this reason it may be necessary to manually recreate CDD folders in the Folder Manager, and to recreate custom menus and tables.

Assuming the Security Structure (US_SECOBJ_MSTR) is intact, the next question is what state Security Roles are in. It could just be that the persisted versions of the security roles have become damaged somehow and simply need to be refreshed. In that case, deleting the persisted XML from the IFAS_DATA table (see below for more details) and removing the roles from the US_ROLE_OUTPUT should allow you to rebuild one or all of the users on the system based on the actual security setup.

If one or all of the remaining security definition tables (US_ROLE_MSTR, US_ROLE_DTL, and/or US_ROLESEC_DTL) are missing data or somehow corrupted, it may become necessary to force an administrative user into the system to get the software working again. Because the Admin Console only reads the User Security XML it's possible to create a single Role and Role Assignment that will let you back into the software.

Emergency Restoration Steps:

Step 1: Insert one row into the US_ROLE_MSTR with a role_id such as "_FULL" to use for recovery purposes, as follows:

```
INSERT INTO US_ROLE_MSTR (US_ROLE_ID,US_ROLE_TITLE,US_WF_ENABLE)
VALUES ('_FULL','Full Access for Recovery','N')
```

Step 2: Insert one row into the US_ROLESEC_DTL for your new role with a US_SO_ID of "APPLICATION" as follows:

```
INSERT INTO US_ROLESEC_DTL (US_ROLE_ID,US_SO_ID,US_SO_ACCESS_R, US_SO_ACCESS_W,
US_SO_ACCESS_U, US_SO_ACCESS_D, US_SO_ACCESS_X)
VALUES ('_FULL','APPLICATION',1,1,1,1,1)
```

If this SQL gives you a constraint violation, it is likely because your US_SECOBJ_MSTR is missing data. You may need to

use the following SQL to fix US_SECOBJ_MSTR before the previous SQL will work:

```
INSERT INTO US_SECOBJ_MSTR(US_SO_ID,US_SO_DESC,US_SO_PARENT_ID)
VALUES('APPLICATION', 'Application Root','XML')
```

Step 3: Select an existing IFAS user and insert one row into the US_ROLE_DTL for that user with your admin role. If your user name is SBI, for example, first ascertain that users US_NO (user number) with the following statement:

```
SELECT US_NO FROM US_USNO_MSTR WHERE US_ID = 'SBI'
```

If this US_NO comes back as 693, create the US_ROLE_DTL entry as follows:

```
INSERT INTO US_ROLE_DTL (US_NO,US_ROLE_ID) VALUES (693,'_FULL')
```

Step 4: Delete the rows from IFAS_DATA where the category is "XMLMODEL", "XSLTROLE" or "XMLUSERSEC".

```
DELETE FROM IFAS_DATA WHERE CATEGORY IN ('XMLMODEL', 'XSLTROLE', 'XMLUSERSEC')
```

Step 5: Log into the Admin Console and use the "Rebuild Security" screen to rebuild the security of your admin user. Depending on how much was lost, the Console may inform you to rebuild the security structure and restart before continuing.

At this point your admin user should have full access to the software and you can begin setting up the necessary security for other users.

6.3 Rebuild Security

Over the course of many years IFAS has grown to have many different and complex security needs. Because of this there are very few centralized paths or standard ways of requesting security. IFAS Menu security is requested while displaying the 7i menu. IFAS Data security is requested while fetching the data from the database. CDD Functional Security is fetched while building the application menus. However, the desire of our users was to merge all of these needs into one model so that they could centralize the process of setting up security.

To help accommodate this design objective there is a very deliberate separation from the way the Security is setup and configured and the way it is referenced by the software. This was done so that the setup itself could be stored in a way that is as flexible as possible while the software could utilize it in a way that does not create a significant performance problem.

Rebuild Security Steps:

Step 1: The prior Security Structure, Security Role and User Security XML/XSLT BLOB rows are removed from the IFAS_DATA table. How many depends on whether you are rebuilding a single user or all the users assigned to a Security Role.

Step 2: New versions of those XML/XSLT BLOBs are generated based on the current security setup. This is a complex process that starts by converting the Security Structure to XML and the Security Roles to XSLT layers. Then, by applying the layers assigned to each user to the Security Structure a resulting user's security is generated.

Step 3: For each user the User Security XML document is used to determine what changes are necessary in the US_ROLE_OUTPUT table. If the table is empty the software begins inserting all of the user's security into the table. If the table is fully accurate based on the User Security XML no changes are made.

Step 4: A broadcast message is sent out to all of the servers in the Server Group informing them that they should flush their local cache of security information. This may take a few minutes to complete depending on the amount of server activity. Also, this may cause some latency for those users until 7i can get the newer version of the security information cached again.

6.4 Forgot Login Page

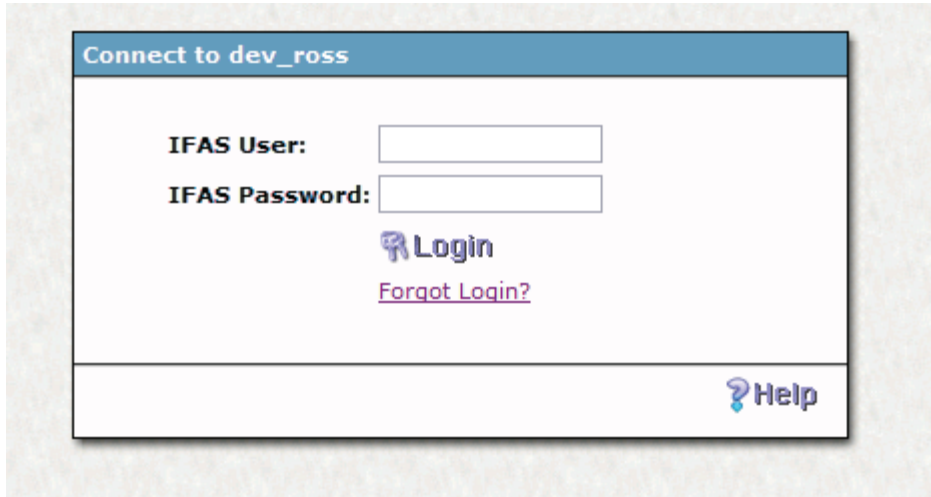
6.4.1 Configuration

The "Settings" tab within the Dashboard Customize section is used to control which portals allow the Forgot Login feature. You can select each one individually. This setting only controls if the link to the Forgot Login page will appear. A workflow model controls which associations allow the user's password to be changed and emailed. This workflow model can be found on the install DVD under /WFModels/Nucleus/FORGOTLOGIN.wdl. If you have been previously using this functionality for Applicant Online or Bid Online, you will need to load this new WF model for that functionality to continue to work.

The 7i / Finance option applies to both the main 7i login page and the dashboard users that have the 'FINANCE' association. Timecard Online and Employee online are both combined into one setting. Applicant Online, Professional Development and Bid Online are always on by default.

6.4.2 Login Page

The login page now has a Forgot Login link:

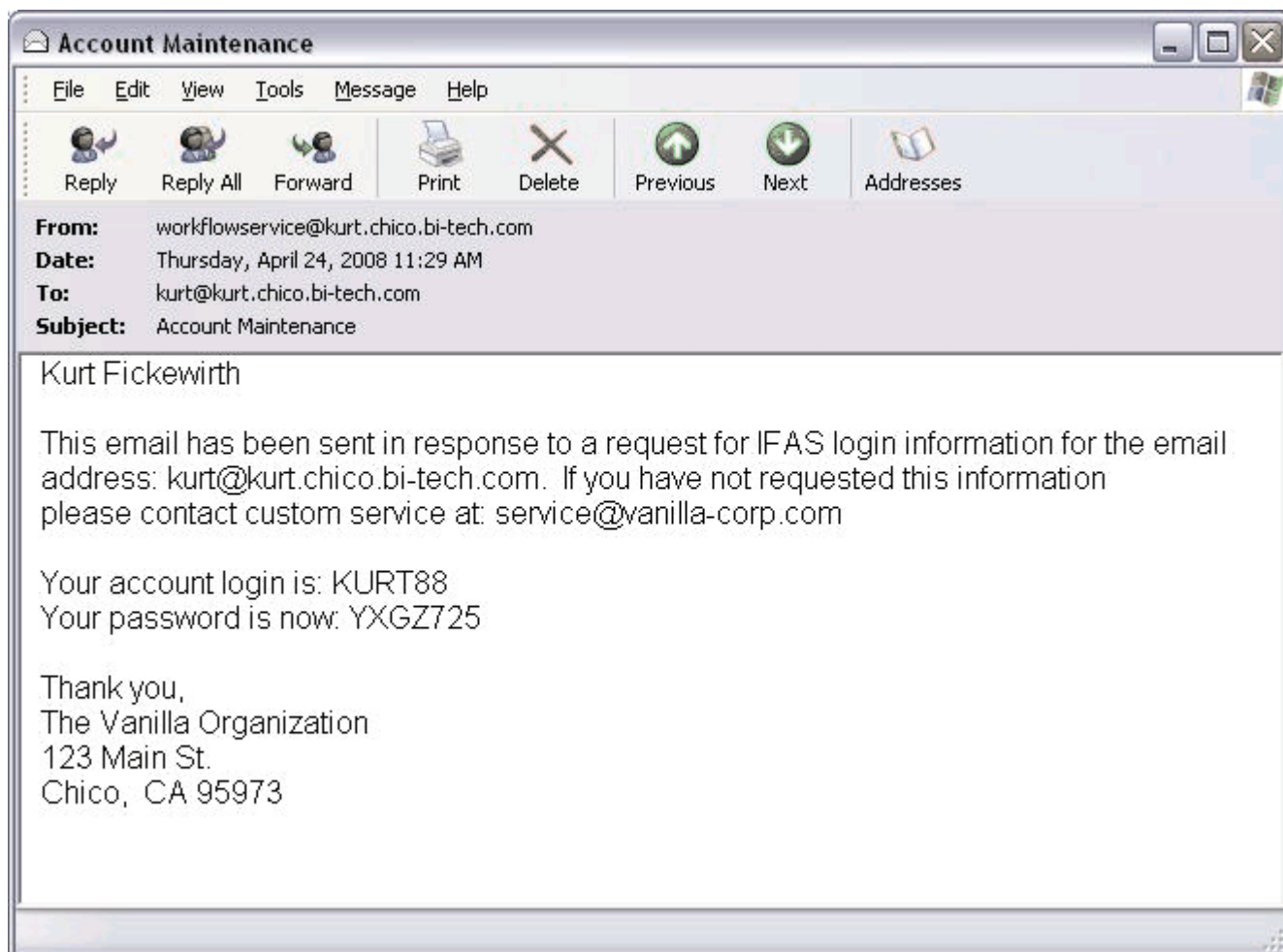


The screenshot shows a login form with a blue header bar containing the text "Connect to dev_ross". Below the header, there are two input fields: "IFAS User:" followed by a text box, and "IFAS Password:" followed by a text box. Below the password field, there is a "Login" button with a key icon and a "Forgot Login?" link in purple text. At the bottom right of the form, there is a "Help" button with a question mark icon.

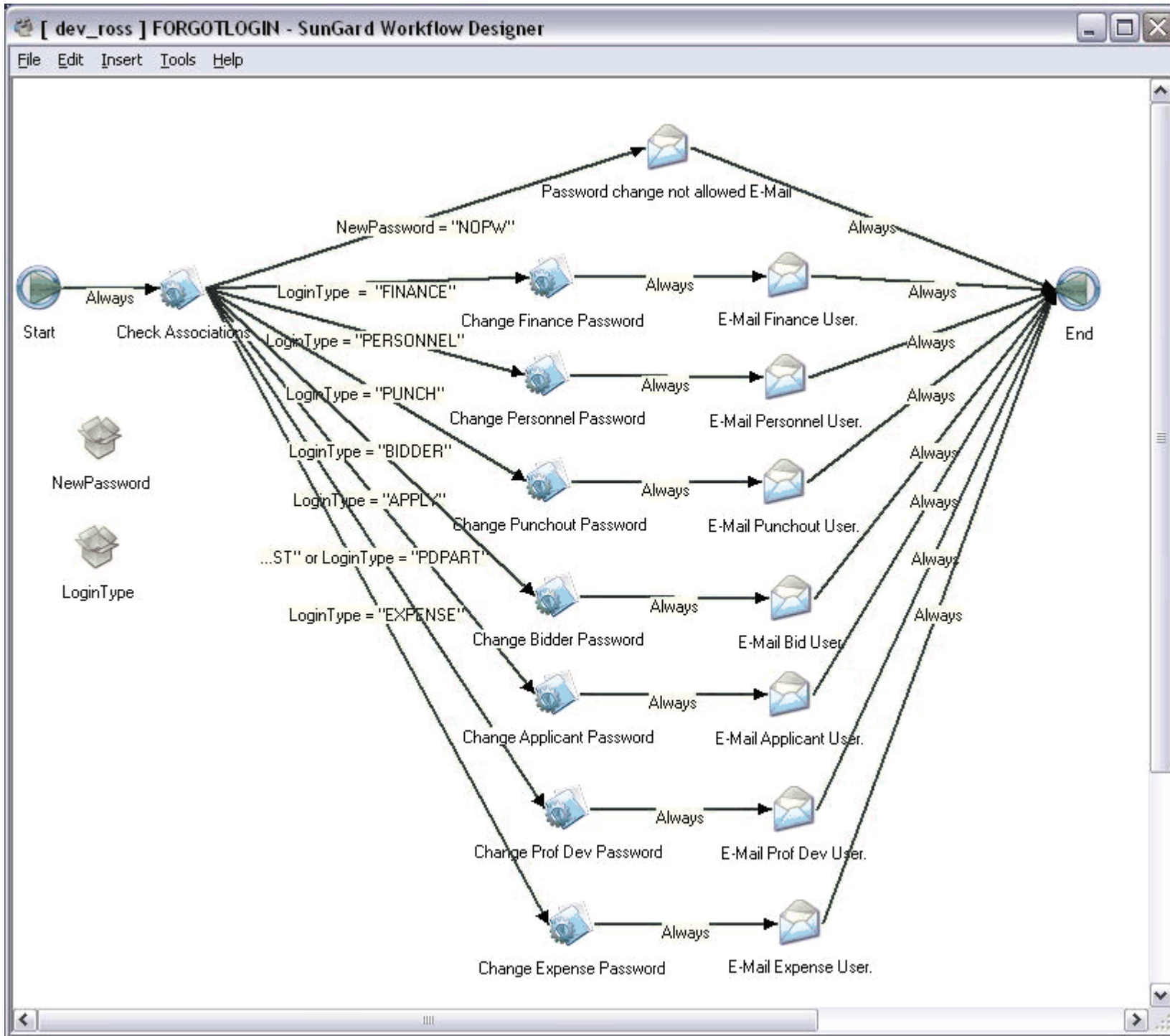
The user will be redirect to this page to allow them to fill in their email address. Their email address will be used to perform a reverse lookup up to obtain their login and password.

6.4.3 Workflow Email

An email will be sent to the user. Below is the standard one that comes with the default workflow model.



A new random password will be generated and emailed to the user. If would like to not reset the password, simply alter the Workflow model to just email it instead of resetting it. You can also force the user to change their password at their next login by setting the 'Status' to 'P' on the USUsnoMaster record from the Change Password VBScript in the Workflow model (see the Workflow guide on how to program in VBScript). There is a path in the model to not allow password changes if 'NOPW' is in use. You can modify this email activity to CC your system admin if a user attempts to change their password and cannot. Below is the sample WF model that allows different password rest schemes and emails for each portal. They are all the same in this model, it is just a starting point.



7 Implementation

7.1 Dependencies

7.1.1 Comparing 7.9 Security and its Predecessors

With the 7.9 release SunGard Bi-Tech has centralized several different types of security into one model. This includes security on Menu Masks, Data, CDD Folders and Application Functionality. This includes not only the Nucleus Security (NUUPUS) but the TRIAD Security (NUC), and the subsystem specific for example Purchasing (POUPUS). A somewhat abstract model was required to meet the needs of all the different types of security. This approach is also designed to be flexible enough to accommodate the diverse organizational structures of our clients as well as the continuing growth of the software.

IFAS Menu Masks

Access to the IFAS Menu Masks is no longer controlled by the Nucleus Job Running security as it was prior to the 7.9 release. Now each subsystem's menu structure is available in the Security Structure. Just as access to a menu mask could be granted by either granting that full mask or part of the mask, the Role-Based Security allows access to a particular IFAS Menu option to be granted individually or by granting execute access to a security object above that menu option.

Application Functionality

The software has been using Role-Based security for CDD, Documents Online and other applications for several versions prior to the 7.9 release. For those applications the security is essentially the same. The change in this release is that this concept is now extended beyond those applications to all of the subsystems in IFAS. Prior releases may have used miscellaneous masks or special coded values in a Database Access class to control them. More detailed usage can be found in the individual user guides for each subsystem.

IFAS Data Access

Prior to 7.9 user access to the data was controlled either by the Nucleus Database Access classes or the TRIAD cluster security. With 7.9 those have been condensed into the Security Structure and access granted through Security Roles.

The Nucleus Database Access classes provided the ability to control access to a single table or a group of tables based on which database schema they resided in. (Example: GLK_KEY_MSTR resides in the GLDB) However, the type of security that could be applied was limited to how the select codes were interpreted for each specific table. TRIAD Cluster Security had to be specified on a cluster by cluster basis or by

use of a wildcard, but also provided not only the ability to grant access to a table but also specify an SQL Where Clause to be used to secure the data.

The Role-Based security is intended to provide the strengths of both of its predecessors. Access can be granted to entire subsystem of tables by using the subsystem's data node. Alternatively, access can be granted on a table by table basis and a where clause used to provide more specific filtering.

Common Security

One of the changes to Role-Based Security that is being introduced with the 7.9 release is the concept of Common Security objects. Just as CDD Folders and CDD Functionality work together to provide access to reporting, access to a particular IFAS Table and its related Common Security objects work together to provide access to data.

| IFAS Table | Common Linkages |
|--------------|--|
| GLK_KEY_MSTR | <ul style="list-style-type: none"> • "Ledger Security" by ledger • "Account Key Security" by key |
| GLT_TRNS_DTL | <ul style="list-style-type: none"> • "Ledger Security" by ledger • "Account Key Security" by key • "Object Code Security" by object |
| HR_EMPMSTR | <ul style="list-style-type: none"> • "Employee Definition" by ID |
| HR_EMPPAY | <ul style="list-style-type: none"> • "Employee Definition" by ID |

Security Rollover

For many sites upgrading to the 7.9 release may involve running the Security Rollover in the Admin Console. However, the Exclusive security prior to Role-Based software didn't easily map into the new Inclusive model in a way that allowed for an accurate match for all the different variations of security. Therefore, the only way to create a rollover that matched the old security was to put all of a user's security into one Security Role and assign that role to each user. While this should be sufficient for the initial setup of IFAS 7.9 it is not intended to be a long term solution.

It would be in the best interests of sites to plan on reevaluating their security needs once the initial upgrade to 7.9 is complete. All of the rollover roles will be named with the "_R" suffix added to each Role ID. As time goes by and security can be fully thought out and applied to the organization, it is the intent that these rollover roles will be unassigned and removed from the system completely.

7.2 Templates

Under Construction

7.3 Agendas

7.3.1 Security Agenda

1. Admin Console
2. Role-based Security
3. Managing the Security Structure
4. Managing Security Roles

Data Tables

Functions

Menus

5. Assigning Security Roles
6. Reset/Resync Security

8 Module Integration

Under Construction

9 FAQ

Under Construction