

Detail Design Specification (A007)	Project Aspire
---	-----------------------

3.0 Appendix B – Project Aspire Enhancement Functional Design

ADML ID	45
ADML Description	Department Row-level Security
ADML Tech #	45

3.1 Background

3.1.1 Functional Requirement

Users must be restricted to making transactions only against the orgs for which they are authorized.

3.1.2 Delivered Functionality

PeopleSoft comes with delivered row-level security that restricts access to business units, SetIDs, projects, and other functional data. These delivered restrictions can be implemented on a user and/or permission list basis.

3.1.3 Gap Description

Row-level security is required to restrict user access at the org level. Org level security is a custom feature in addition to the delivered access restrictions on business unit, SetIDs and project.

3.2 Description of New Functionality

To facilitate the custom org row-level security (ORLS) a new Organization Security table will be created. The new table will link; a) authorized orgs to b) tree nodes to c) a permission list. Trees will be used to consolidate the authorized orgs' row information into nodes. One or more tree nodes will associate with a permission list.

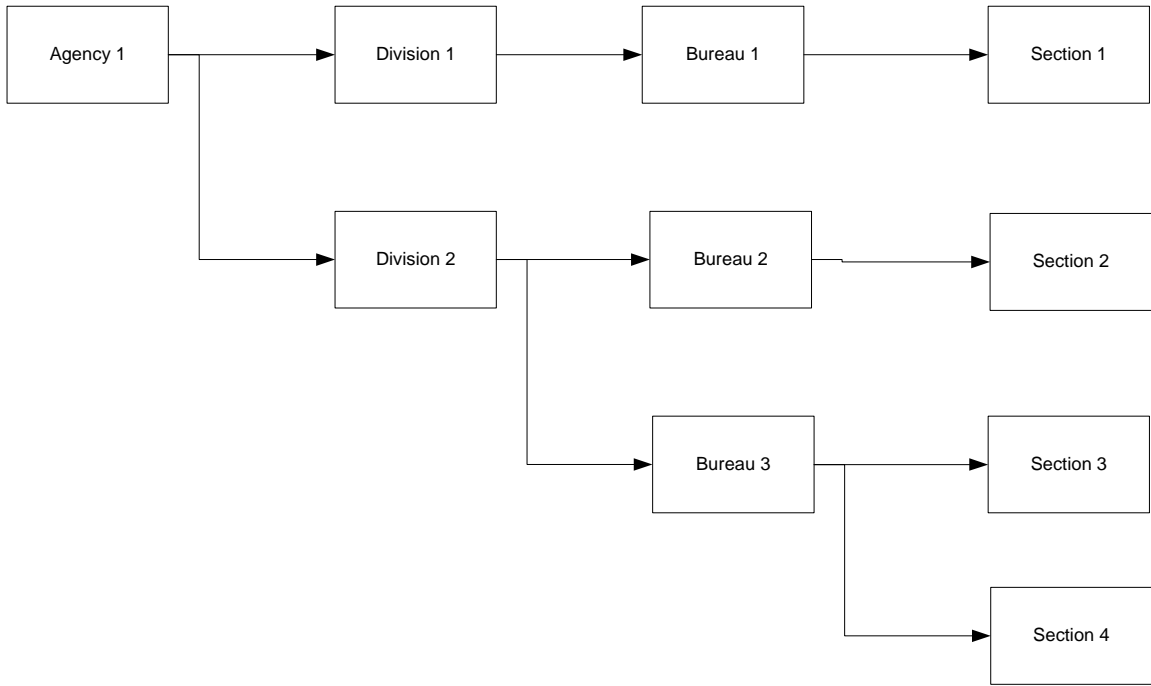


Next, a new SQL view restricting org access will be tied to the user.. The authorized orgs are linked to the permission lists linked to the user id. User ids are carried with the user

to the application component currently engaged. This user id based view will return a list of orgs and descriptions available to that user.

Finally, the new SQL view restricting org access will be attached to all pages with transaction distribution line level entry activity (Requisitions, Purchase Orders, Vouchers, AR Accounting Entries, Journals). The information will present itself as a lookup view. When the user clicks the 'lookup/spyglass' button on the org field to get a list of valid org values, the underlying view returns a list of values authorized for that user. If the user keys an org value in directly, the system will validate that org against the view and will return an error if the org is not in the user's list.

An example of these elements follows:



Custom Org Security Table

Primary Permission List	Effective Date	Tree Name	Tree Node
PermList1	01/01/1901	Agy1_Sec_Tree	Section 1
PermList1	01/01/1901	Agy1_Sec_Tree	Bureau 2
PermList2	01/01/1901	Agy1_Sec_Tree	Division 2

User 1 = PermList1

User2 = PermList2

Result:

When User1 gets a list of valid org values, that user will get Section 1 and Section 2.

When User2 gets a list of valid org values, that user will get Section 2, Section 3 and Section 4.

3.3 Navigation path

Set up Financials/SCM -> Common Definitions -> Security -> Org Security.

3.4 Set Up/Control Data

Each agency will need an org tree (node-based) that describes the security access. When a user is given access to a node on this tree, that user has access to all detail orgs that roll up to that node. A user can have more than one node.

3.5 Application Changes (e.g., Pages, Components, Menus, Records, App Engines, SQRs, etc.)

A new record is needed to establish the linkage between the user and the tree nodes. The structure of the record is:

Permission List

Effective Date

Tree Name

Tree Node

All fields on this record are keys.

A new component is required to enable updates to the user/tree node data.

A new view is required to select all orgs from the tree nodes associated with the user.

Each of the CFxx_AN_SBR subrecords must be altered to change the org prompt view to the new view.

3.6 Unit Test Considerations

A user's list of valid values contains only orgs related to their tree nodes.

An unauthorized org cannot be entered or saved on any transaction.

If there are multiple effective dated records, the view returns the correct list of orgs given the accounting date.

No orgs are returned if there are no user/tree node records in the table.

3.7 Miscellaneous

NA.

3.8 Assumptions

Agencies will maintain the security tree.