

ASSETS AND PURCHASING MANAGEMENT TOOL

FOR

IT GOVERNANCE TEAM

INSEAD



The Business School  
for the World®

## **Table of Contents**

❖ Document Information and Distribution List.....	3
❖ Technology and Compatibility Details .....	4
❖ Components.....	5
➤ Controllers.....	5
➤ Models .....	5
❖ Functions .....	6
➤ Controllers.....	6
✓ Adminman.....	6
✓ Alerts .....	9
✓ Assetman.....	10
✓ Config.....	13
✓ Login.....	14
✓ Purchasing .....	14
✓ Reports.....	17
✓ Requests.....	17
✓ Util.....	18
➤ Custom Tag.....	18
❖ Database .....	19
➤ Tables .....	19
➤ Packages .....	21
❖ Appendix.....	22
❖ Term Definitions .....	22

## Document Information and Distribution List

### Version History

<b><u>Version#</u></b>	<b><u>Date</u></b>	<b><u>Revised by</u></b>	<b><u>Reason for change</u></b>
1.0	12 <sup>th</sup> December,2011	NagaRaju BHANOORI	Initiation

The document gives the clear developer details of Asset and Purchase Management Tool.

### Distribution List

<b><u>Name</u></b>	<b><u>Title</u></b>	<b><u>Date</u></b>
Jae CHAE	Governance Manager	12 <sup>th</sup> December,2011
Sanjeev Krishna MANDALAPU	Software/System Analyst - FEAD	12 <sup>th</sup> December,2011
Virnali SY	Purchasing Officer	12 <sup>th</sup> December,2011
Ghislaine LABRO	IT Budget and Control Manager	12 <sup>th</sup> December,2011
David GABAY	Purchasing Manager	12 <sup>th</sup> December,2011

## Technology and Compatibility Details

Tool Compatible for

**Webserver:** ColdFusion 8 and Above.

**Database:** Oracle 10 and Above.

**Browsers:** All

**Technologies/Tools used:** HTML, CSS, JavaScript, JQuery, Adobe Live Cycle Designer

**Architecture:** The Architecture used in developing this Tool mimics “MVC” Architecture. MVC Architecture is built using **Model**, **View** and **Controller**.

### 1. Model

The **model** manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller). In event-driven systems, the model notifies observers (usually views) when the information changes so that they can react.

In the case of our tool, Models are Data Base Tables

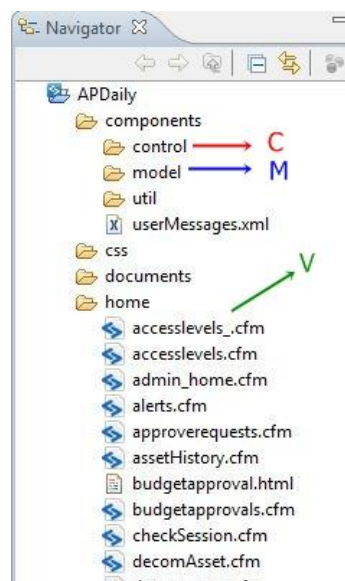
### 2. View

The **view** renders the model into a form suitable for interaction, typically a user interface element. In the case of our tool, Views are CFML pages the user sees.

### 3. Controller

The **controller** receives user input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and a view to perform actions based on that input. In the case of our tool, Controllers are ColdFusion functions which interact with View and Models.

The following snapshot explains the structure in our Tool.



## Components

### Controllers

The various controllers used in this tool are

a) Adminman

This controller handles all requests from Administrator Section and thereby interacts with corresponding Models as per the request.

b) Alerts

This controller is responsible for all the Alerts in the home screen.

c) Asset man

This controller is responsible for all requests related to Asset Management.

d) Config

This Controller is responsible for display of error Messages of user.

e) Login

This Controller handles the Login part i.e., Authentication and creating Session.

f) Purchasing

This controller is responsible for all requests related to Purchase Management.

g) Reports

This Controller is responsible for generating Reports in Report Section.

h) Requests

This Controller is responsible for handling All Purchase Requests and their Approvals.

### Models

The various models reflect the corresponding Tables in the Database. They are

- i. Asset\_Permissions
- ii. Budget\_roles
- iii. Currency
- iv. Departments
- v. Feedback
- vi. PrdUsrHistory
- vii. Products
- viii. Purchase\_Items\_type
- ix. Purchase\_permissions
- x. Purchase\_status

- xi. PurchaseItems\_Models
- xii. PurchaseRequests
- xiii. Purchases\_Items
- xiv. Purchases
- xv. Request\_Items
- xvi. Softwares
- xvii. StatusLog
- xviii. Users
- xix. Utils
- xx. Vendors

## Functions

This section gives a clear description of all functions used in all controllers and where are they called from.

## Controllers

### i. Adminman

Functions in this controller are as follows:

#### ➤ AssignAccess

Calls:

- ✓ AuthenticateUser function from “Users” model.
- ✓ AssignAccess function from “Users” model.

Called from:

- ✓ Called from “Accesslevels.cfm” for assigning role.

#### ➤ ExpireAction\_user

Calls:

- ✓ UpdateExpireAction function from “Users” model for updating Active status in database.

Called from:

- ✓ Called from “ExpUser.cfm” for disabling a user .[ “ExpUser.cfm” opens in a window which is clicked in grid of “Users.cfm”]

#### ➤ Expire\_user

Demands a confirmation for Disabling/Enabling the User.

Called from:

- ✓ Called from “ExpUser.cfm” for disabling a user.[ “ExpUser.cfm” opens in a window which is clicked in grid of “users.cfm”]

#### ➤ ExpireAction\_vendor

Calls:

- ✓ UpdateExpireAction function from “Vendors” model for updating Active status in database.

Called from:

- ✓ Called from “ExpVendor.cfm” for disabling a vendor .[ “ExpVendor.cfm” opens in a window which is clicked in grid of “Seeallvendors.cfm”]

➤ **Expire\_vendor**

Demands a confirmation for Disabling/Enabling the Vendor.

Called from:

- ✓ Called from "ExpVendor.cfm" for disabling a user.[ "ExpVendor.cfm" opens in a window which is clicked in grid of "Seeallvendors.cfm"]

➤ **Autosuggestusers**

Calls:

- ✓ Autosuggest function from "Users" model for getting autosuggest from database.

Called from:

- ✓ Called from "Accesslevels.cfm" for getting an autosuggest.

➤ **POAutoSuggest**

Calls:

- ✓ Autosuggest function from "Purchases" model for getting autosuggest from database.

Called from:

- ✓ Called from "Feedback.cfm" for getting an autosuggest.

➤ **Feedback**

Calls:

- ✓ Insertfeedbackfunction from "Feedback" model for inserting feedback.
- ✓ GetpurchId function from "Purchases" model for getting INS format of Purchaseid.

Called from:

- ✓ Called from "Feedback.cfm" for getting an autosuggest.

➤ **SeeAllPermissionsForGrid**

Calls:

- ✓ ListPermissions function from "Users" model.

Called from:

- ✓ Called from "SeeAllAccess.cfm" for getting an autosuggest.

➤ **ListDepartmentsforGrid**

Calls:

- ✓ Listdepartments function from "departments" model.

Called from:

- ✓ Called from "seeAlldepartments.cfm".

➤ **editForGrid\_Department**

Calls:

- ✓ updatedepartment function from "departments" model.

Called from:

- ✓ Called from "seeAlldepartments.cfm".

➤ **ViewModelsForGrid**

Calls:

- ✓ ListModelsForGrid function from "PurchasesItems\_Models" model.

Called from:

- ✓ Called from "seeAllModels.cfm".

➤ **EditModelsForGrid**

Calls:

- ✓ UpdateItemModel function from "PurchasesItems\_Models" model.

Called from:

- ✓ Called from “seeAllModels.cfm”.

➤ **FeedbackResults**

Calls:

- ✓ ListResponses function from “Feedback” model.

Called from:

- ✓ Called from “seeAllResponses.cfm”.

➤ **ViewPurchItemTypesForGrid**

Calls:

- ✓ ListPurchtypesfunction from “Purchase\_Items\_type” model.

Called from:

- ✓ Called from “seeAllTypes.cfm”.

➤ **EditForGridPurchaseTypes**

Calls:

- ✓ UpdatePurchaseType from “Purchase\_Items\_type” model.

Called from:

- ✓ Called from “seeAllTypes.cfm”.

➤ **ResultForGrid\_Vendors**

Calls:

- ✓ ListvendorsForGrid function from “Vendors” model.

Called from:

- ✓ Called from “seeAllVendors.cfm”.

➤ **EditForGrid\_Vendors**

Calls:

- ✓ EditForGrid\_Vendors function from “Vendors” model.

Called from:

- ✓ Called from “seeAllVendors.cfm”.

➤ **ListAccessRoles**

Calls:

- ✓ ListPermissions function from “Asset\_Permissions” model.
- ✓ ListPermissions function from “Purchase\_permissions” model.

Called from:

- ✓ Called from “Users.cfm”.

➤ **ListDepartments**

Calls:

- ✓ ListDepartmentsForListing function from “Departments” model.

Called from:

- ✓ Called from “Users.cfm”.

➤ **ResultForGridforSearch**

Calls:

- ✓ SearchUsers function from “Users” model.

Called from:

- ✓ Called from “Users.cfm”.

➤ **EditForGrid\_User**

Calls:

- ✓ Updateuser function from “Users” model.
- ✓ GetDeptID function from “Users” model.



Called from:

- ✓ Called from "Users.cfm".

➤ **VendorRegistration**

Calls:

- ✓ Insertvendor function from "Vendors" model.

Called from:

- ✓ Called from "Vendors.cfm".

➤ **Listusers**

Calls:

- ✓ Listusers function from "Users" model.

Called from:

- ✓ Called from "Purchase\_Home.cfm".

➤ **Listvendors**

Calls:

- ✓ Listvendors function from "Vendors" model.

Called from:

- ✓ Called from "Purchase\_Home.cfm".

➤ **ListBudgetApprovers**

Calls:

- ✓ Showbudgetmgrs function from "Budget\_roles" model.

Called from:

- ✓ Called from "Purchase\_Home.cfm".

➤ **TypesAdd**

Calls:

- ✓ Inserttype function from "Purchase\_Items\_type" model.

Called from:

- ✓ Called from "PurchaseType.cfm".

➤ **AddModel**

Calls:

- ✓ InsertModel function from "PurchaseItems\_Models" model.

Called from:

- ✓ Called from "PurchaseType.cfm".

➤ **ListPurchaseItemTypes**

Calls:

- ✓ Listitems function from "Purchases\_items\_type" model.

Called from:

- ✓ Called from "ItemModel.cfm".

➤ **AddDepartment**

Calls:

- ✓ Insertdepartment function from "Departments" model.

Called from:

- ✓ Called from "departments.cfm".

**ii. Alerts**

➤ **DelivDate**

Calls:

- ✓ DeliveryDateAlert function from "Purchases\_Items" model.

Called from:

- ✓ Called from “index.cfm”.

➤ **MissedDeliveries**

Calls:

- ✓ MissedDeliveries function from “Purchases\_Items” model.

Called from:

- ✓ Called from “index.cfm”.

➤ **AssetRegistrationMismatch**

Calls:

- ✓ AssetsNotRegistered function from “Purchases” model.

Called from:

- ✓ Called from “index.cfm”.

➤ **AssetToPurchaseCountMismatch**

Calls:

- ✓ ProductAssetsPurchasesItemsCountMismatch function from “Purchases” model.

Called from:

- ✓ Called from “index.cfm”.

➤ **swLicenseTerm**

Calls:

- ✓ SwLicenseTerm function from “Softwares” model.

Called from:

- ✓ Called from “index.cfm”.

➤ **NewRequestsAlert**

Calls:

- ✓ NewRequestsAlert function from “PurchaseRequests” model.

Called from:

- ✓ Called from “index.cfm”.

**iii. Assetman**

Functions in this controller are as follows:

➤ **AssetHistory**

Calls:

- ✓ AstHistory function from “Products” model for getting history from database.
- ✓ GetpurchId function from “Purchases” model for getting INS format Purchase order number.

Called from:

- ✓ Called from “AssetHistory.cfm” for listing history of an Asset.[ “AssetHistory.cfm” opens in a window which is clicked in “EditAsset.cfm”]

➤ **Decom**

Displays a confirmation for Disabling/Enabling the Product with reason

Called from:

- ✓ Called from “DecomAsset.cfm” for decommissioning a Product.[ “DecomAsset.cfm” opens in a window which is clicked in grid of “EditAsset.cfm”]

➤ **DecomAction**

Calls:

- ✓ UpdatedecomAction functions from “Products” model for updating decommission status in database.
- ✓ UpdateEnddateDecom function from “PrdUsrHistory” model for updating the enddate of mapping asset.

Called from:

- ✓ Called from “DecomAsset.cfm” for listing history of an Asset.[ “DecomAsset.cfm” opens in a window which is clicked in grid of “EditAsset.cfm”]

➤ **ResultForGrid**

Calls:

- ✓ ListAll function from “Products” model for getting search results from database.
- ✓ ListSoftwaresfunction from “Softwares” model for getting search results from database.

Called from:

- ✓ Called from “EditAsset.cfm” for showing search results in Grid.
- ✓ Called from “SearchAsset.cfm”.

➤ **EditForGrid\_Product**

Calls:

- ✓ Updateproduct function from “Products” model for updating product details to database.
- ✓ Listvendors function from “Vendors” model for getting vendorid from database.
- ✓ AuthenticateUser function from “Users” model for getting userid from database.
- ✓ Listmap\_assets function from “PrdUsrHistory” model for getting mapping details from database.
- ✓ UpdatemapEnddate function from “PrdUsrHistory” model for updating enddate mapping details to database.
- ✓ Insertprd\_hist function from “PrdUsrHistory” model for inserting a new mapping to database.

Called from:

- ✓ Called from “EditAsset.cfm” for editing search results of Products in Grid.

➤ **EditForGrid\_Software**

Calls:

- ✓ Updatesoftware function from “Softwares” model for updating software details to database.
- ✓ Listvendors function from “Vendors” model for getting vendorid from database.

Called from:

- ✓ Called from “EditAsset.cfm” for editing search results of Softwares in Grid.

➤ **Expire**

Demands a confirmation for Disabling/Enabling Software.

Called from:

- ✓ Called from “ExpSoftware.cfm” for listing disabling/enabling a software.[ “ExpSoftware.cfm” opens in a window which is clicked in grid of “EditAsset.cfm”].

➤ **ExpireAction**

Calls:

- ✓ UpdateExpireAction function from “Softwares” model for updating expiry status in database.

Called from:

- ✓ Called from “ExpSoftware.cfm” for listing history of an Asset.[ “ExpSoftware.cfm” opens in a window which is clicked in grid of “EditAsset.cfm”]

➤ **Registration**

Calls:

- ✓ Listvendors function from “Vendors” model for getting vendorid.
- ✓ GetpurchId function from “Purchases” model.
- ✓ Insertproduct function from “Products” model for inserting a new Product.
- ✓ AuthenticateUser function from “Users” model.
- ✓ Insertprd\_hist function from “PrdUsrHistory” model.
- ✓ Insertsoftware function from “Softwares” model.

Called from:

- ✓ Called from “Newasset.cfm” for registering an Asset (Product or Software).

➤ **ListTypes**

Calls:

- ✓ Listitems function from “Purchases\_items\_type” model.

Called from:

- ✓ Called from “Newasset.cfm” for binding Asset Types.

➤ **ListModels**

Calls:

- ✓ Listmodels function from “PurchaseItems\_Models” model.

Called from:

- ✓ Called from “Newasset.cfm” for binding Item Models.

➤ **AutosuggestVendors**

Calls:

- ✓ Autosuggest function from “Vendors” model.

Called from:

- ✓ Called from “Newasset.cfm” for autosuggest of Vendors.

➤ **POAutosuggest**

Calls:

- ✓ Autosuggest function from “Purchases” model.

Called from:

- ✓ Called from “Newasset.cfm” for autosuggest of PurchaseOrderNumber.

➤ **AutosuggestUsers**

Calls:

- ✓ Autosuggest function from “Users” model.

Called from:

- ✓ Called from “Newasset.cfm” for autosuggest of Users.
- ✓ RegForProduct function from “Purchasing” controller.
- **SoftwareRegForWindow**
  - Calls:
    - ✓ Insertsoftware function from “Softwares” model.
  - Called from:
    - ✓ Called from “PurchRegister.cfm”. [ “PurchRegister.cfm” opens in a window which is clicked in grid of “PurchaseActions.cfm”]
- **PurchaseItemsBlkUpload**
  - Calls:
    - ✓ \$readCSV function from “Util” controller.
    - ✓ ListitemsforId function from “Purchases\_items\_type” model.
    - ✓ ListmodelsforId function from “PurchaseItems\_Models” model.
    - ✓ Registration’ function from “Assetman” controller.
  - Called from:
    - ✓ Called from “PurchRegister.cfm”. [ “PurchRegister.cfm” opens in a window which is clicked in grid of “PurchaseActions.cfm”]
- **ProductRegForWindow**
  - Calls:
    - ✓ Insertproduct function from “Products” model.
    - ✓ AuthenticateUser function from “Users” model.
    - ✓ Insertprd\_hist function from “PrdUsrHistory” model.
  - Called from:
    - ✓ Called from “PurchRegister.cfm”. [ “PurchRegister.cfm” opens in a window which is clicked in grid of “PurchaseActions.cfm”]
- **resultForReport**
  - Calls:
    - ✓ ListAll function from “Products” model.
    - ✓ Autovendor function from “Vendors” model.
    - ✓ GetpurchId function from “Purchases” model.
    - ✓ UserProductIdForSearch function from “Users” model.
    - ✓ ListSoftwares function from “Softwares” model.
  - Called from:
    - ✓ Called from “Reports.cfm”.
- **SeeAllApprovalsForGrid**
  - Calls:
    - ✓ SeeAllApprovalsForGrid function from “Purchases” model.
  - Called from:
    - ✓ Called from “seeAllApprovals.cfm”.

#### **iv. Config**

Functions in this controller are as follows:

- **getMessageForCode**
  - Calls:
  - Called from:
    - ✓ Called from “ValidateUser” function from “Login” Controller.
    - ✓ Called from “Login.cfm” for displaying relevant Error Messages.

**v. Login**

Functions in this controller are as follows:

➤ **ValidateUser**

Calls:

- ✓ CheckLDAP function from “Login” Controller.
- ✓ CheckDB function from “Login” Controller.
- ✓ GetMessageForCode function from “Config” Controller.
- ✓ \$starterPack function from “Util” Controller.

Called from:

- ✓ Called from “Login.cfm” for Authentication and Session creation.

➤ **checkLDAP**

Calls:

- ✓ \$ldapAuthenticate function from “Util” Controller.

Called from:

- ✓ Called from “ValidateUser” function from “Login” Controller.

➤ **CheckDB**

Calls:

- ✓ AuthenticateUserForLogin function from “Users” Model.

Called from:

- ✓ Called from “ValidateUser” function from “Login” Controller.

**vi. Purchasing**

Functions in this controller are as follows:

➤ **ListPO**

Calls:

- ✓ ListpurchasesForBudgetApproval function from “Purchases” model for getting pending PO’s from database.

Called from:

- ✓ Called from “BudgetApprovals.cfm” for listing pending PO’s.

➤ **genpurchID**

Calls:

- ✓ AuthenticateUser function from “Users” model for getting userid from database.
- ✓ Insertpurchases function from “Purchases” model.
- ✓ ListitemsifApproved function from “PurchaseRequests\_Items” model.
- ✓ Listvendors function from “Vendors” model for getting vendorid.
- ✓ Insertpurchases\_items function from “Purchases\_Items” model.
- ✓ Updatereq function from “PurchaseRequests” model.
- ✓ GetpurchId function from “Purchases” model.
- ✓ GetEmail function “Users” model.

Called from:

- ✓ Called from “Requests” Controller.

➤ **ListItemsid**

Calls:

- ✓ ListIdPurchaseItems function from “Purchases\_items\_type” model.
- ✓ GetpurchId function from “Purchases” model.

Called from:

- ✓ Called from "Newasset.cfm" listing Item numbers of a Purchase Order.

➤ **POdisplay**

Displays PO details for Budget Manager.

Calls:

- ✓ Listpurchases\_itemsforBudgetman function from "Purchases\_Items" model.

Called from:

- ✓ Called from "PO.cfm" for listing details of a PO for a budget controller. [ "PO.cfm" opens in a window which is clicked in grid of "BudgetApprovals.cfm"]

➤ **UpdateBudgetmandecision**

Calls:

- ✓ UpdateBudgetmandecision function from "Purchases" model .

Called from:

- ✓ Called from "PO.cfm" for updating Budget Managers Decision.

➤ **POAutosuggest**

Calls:

- ✓ Autosuggest function from "Purchases" model.

Called from:

- ✓ Called from "PurchaseActions.cfm" for autosuggest of PurchaseOrderNumber.
- ✓ Called from "RegisterAsset.cfm".

➤ **UpdatePurchase**

Calls:

- ✓ Updatepurchases function from "Purchases" model.
- ✓ Insertlog function from "StatusLog" model.
- ✓ Updatepurchases\_items function from "Purchases\_Items" model.

Called from:

- ✓ Called from "PurchaseActions.cfm" for updating a PO.

➤ **listitemsForReg**

Calls:

- ✓ Insertproduct function from "Products" model.
- ✓ AuthenticateUser function from "Users" model.
- ✓ Insertprd\_hist function from "PrdUsrHistory" model.

Called from:

- ✓ Called from "PurchRegister.cfm". [ "PurchRegister.cfm" opens in a window which is clicked in grid of "PurchaseActions.cfm"]

➤ **regForProduct**

Calls:

- ✓ AutosuggestUsers function from "Assetman" controller.
- ✓ ListitemsforId function from "Purchases\_Items" model.

Called from:

- ✓ Called from "PurchRegister.cfm". ["PurchRegister.cfm" opens in a window which is clicked in grid of "PurchaseActions.cfm"].
- **regForSoftware**  
Calls:
  - ✓ ListitemsforId function from "Purchases\_Items" model.  
Called from:
  - ✓ Called from "PurchRegister.cfm". ["PurchRegister.cfm" opens in a window which is clicked in grid of "PurchaseActions.cfm"].
- **assetRegistrationMismatch**  
Calls:
  - ✓ AssetsNotRegistered function from "Purchases" model.  
Called from:
  - ✓ Called from "RegisterAsset.cfm"
- **GetpurchId**  
Calls:
  - ✓ GetpurchId function from "Purchases" model.  
Called from:
  - ✓ Called from "RegisterAsset.cfm".
  - ✓ Called from "purchActions\_form.cfm".
- **AutosuggestVendors**  
Calls:
  - ✓ Autosuggest function from "Vendors" model.  
Called from:
  - ✓ Called from "purchActions\_form.cfm".
- **ListpurchasesforDisplay**  
Calls:
  - ✓ ListpurchasesforDisplay function from "Purchases" model.  
Called from:
  - ✓ Called from "purchActions\_form.cfm".
- **Liststatus**  
Calls:
  - ✓ Liststatus function from "Purchase\_status" model.  
Called from:
  - ✓ Called from "purchActions\_form.cfm".
- **Listusernames**  
Calls:
  - ✓ Listusernames function from "Users" model.  
Called from:
  - ✓ Called from "purchActions\_form.cfm".
- **Listcurrency**  
Calls:
  - ✓ Listcurrency function from "Currency" model.  
Called from:
  - ✓ Called from "purchActions\_form.cfm".
- **ListitemsForDisplay**  
Calls:



- ✓ ListitemsForDisplay function from “Purchases\_Items” model.  
Called from:
  - ✓ Called from “purchActions\_form.cfm”.
- **DataForPrintOrder**  
Calls:
  - ✓ DataForPrintOrder function from “Purchases\_Items” model.  
Called from:
  - ✓ Called from “printorder.cfm”.
- **PriceDataForPrintOrder**  
Calls:
  - ✓ PriceDataForPrintOrder function from “Purchases\_Items” model.  
Called from:
  - ✓ Called from “printorder.cfm”.
- **ItemsForPrintOrder**  
Calls:
  - ✓ ItemsForPrintOrder function from “Purchases\_Items” model.  
Called from:
  - ✓ Called from “printorder.cfm”.

## **vii. Reports**

## **viii. Requests**

Functions in this controller are as follows:

- **Listreq**  
Calls:
  - ✓ Listreq function from “Purchase Requests” model for displaying all Requests.  
Called from:
  - ✓ Called from “ApproveRequests.cfm” for listing all requests.
- **Itemsdisplay**  
Calls:
  - ✓ Itemsdisplay function from “PurchaseRequests\_Items” model.
  - ✓ ListreqForwindow function from “PurchaseRequests” model.
  - ✓ Showbudgetmgrs function from “Budget\_roles” model.
  - ✓ Autosuggest function from “Vendors” model.  
Called from:
  - ✓ Called from “Item.cfm” for listing items in Purchase Request.[  
“Item.cfm” opens in a window which is clicked from  
“ApproveRequests.cfm”]
- **ProcessReq**  
Calls:
  - ✓ UpdateItem function from “PurchaseRequests\_Items” model.
  - ✓ Updatereq function from “PurchaseRequests” model.
  - ✓ GenpurchID function from “Purchasing” controller.  
Called from:

- ✓ Called from “Item.cfm” for listing items in Purchase Request.[  
“Item.cfm” opens in a window which is clicked from  
“ApproveRequests.cfm”].
- **GenReq**  
Calls:
  - ✓ Insertreq function from “PurchaseRequests” model.
  - ✓ Insertitem function from “PurchaseRequests\_Items” model.Called from:
  - ✓ Called from “PurchaseRequest.cfm”.
- **ListAllrequestsForGrid**  
Calls:
  - ✓ ListreqForGrid function from “PurchaseRequests” model.Called from:
  - ✓ Called from “seeAllReq.cfm”.

#### **ix. Util**

- **CreateDirectory**  
Called from:
  - ✓ Called from “purchActions\_form.cfm”.
- **Listfiles**  
Called from:
  - ✓ Called from “purchActions\_form.cfm”.
- **\$readCSV**  
Called from:
  - ✓ Called from “PurchaseItemsBlkUpload” function of “Assetman” controller.
- **\$starterPack**  
Called from:
  - ✓ Called from ValidateUser function of “Login” Controller.
- **\$ldapAuthenticate**  
Called from:
  - ✓ Called from checkLDAP function from “Login” controller.
- **\$ldapFind**  
Called from:
  - ✓ Called from “Users.cfm”.
- **\$replacePlaceholders**  
Called from:
  - ✓ Called from “Purchase\_Home.cfm”.

#### **Custom Tag**

Custom tags are user-defined tags which can be made specific to the requirements of the developer. The custom tags used as part of this Tool are as follows:

- **Cf\_CheckSession**  
This tag checks for the Session Expiry and also relevant access required for a particular page.

## Database

### Tables

The Database tables used in this tool and their description are as follows:

Name	Description	Notes
ITGOV_ASSET_PERMISSIONS	Permissions for Asset Management	Assetaccess_id is the primary key which follows the sequence "ITGOV_ASSETACCESSID_SEQ" on a trigger "ITGOV_ASSETACCESSID"
ITGOV_CURRENCY	Different Types of currency	Currency_id is the primary key which follows the sequence "CURRENCYID_SEQ" on a trigger "CURRENCYID_TRG"
ITGOV_DEPARTMENTS	Maintains All departments	Depart_id is the primary key which follows the sequence "ITGOV_DEPARTID_SEQ" on a trigger "ITGOV_DEPARTID"
ITGOV_FEEDBACK	Maintains All Feedback from users	Feedback_id is the primary key which follows the sequence "ITGOV_FEEDBACKID_SEQ" on a trigger "ITGOV_FEEDBACKID"
ITGOV_PRDUSRHISTORY	Maintains All Product and User mapping	Asset_id is the primary key which follows the sequence "ITGOV_ASSETID_SEQ" on a trigger "ASSETID"
ITGOV_PRODUCTS	All Registered Products reside here	Product_id is the primary key which follows the sequence "PRODUCTID_SEQ" on a trigger "PRODUCTID"
ITGOV_PURCHASE_ITEMS_TYPE	Type of Purchase	Purchasetype_id is the primary key which follows the sequence "PURCHASETYPEID_SEQ" on a trigger "ITGOV_PURCHASETYPEID"
ITGOV_PURCHASE_PERMISSIONS	Permissions for Purchase Management	Assetaccess_id is the primary key which follows the sequence "ITGOV_PURCHASEACCESSID_SEQ" on a trigger "ITGOV_PURCHASEACCESSID"
ITGOV_PURCHASE_STATUS	All different status during Purchasing cycle	Status_id is the primary key which follows the sequence "ITGOV_STATUSID_SEQ" on a trigger "ITGOV_STATUSID"
ITGOV_PURCHASEITEMS_MODEL	Different Item Models for each Purchase Type	Model_id is the primary key which follows the sequence "ITGOV_MODELID_SEQ" on a trigger

		"ITGOV_MODELID_TRG"
ITGOV_PURCHASEREQUESTS	All Purchase Requests reach here	Request_id is the primary key which follows the sequence "ITGOV_REQUESTID " on a trigger "ITGOV_REQUESTID_TRG"
ITGOV_PURCHASES	Maintains all Purchases	Purchase_id is the primary key which follows the sequence "ITGOV_PURCHASEID_SEQ " on a trigger "ITGOV_PURCHASEID "
ITGOV_PURCHASES_ITEMS	Maintains All Items in Purchases	Purchases_items_id is the primary key which follows the sequence "ITGOV_PURCHASEITEMSID_SEQ" on a trigger "ITGOV_PURCHASEITEMSID"
ITGOV_REPORT_A_PROBLEM	Any reports on problems reside here	Report_id is the primary key which follows the sequence "ITGOV_REPORTID_SEQ" on a trigger "ITGOV_REPORTID"
ITGOV_REQUEST_ITEMS	All Items in Purchase Requests reside here	Item_id is the primary key which follows the sequence "ITGOV_ITEMID_SEQ" on a trigger "ITGOV_ITEMID_TRG"
ITGOV_SOFTWARES	All Registered Softwares reside here	Software_id is the primary key which follows the sequence "ITGOV_SOFTWAREID_SEQ" on a trigger "ITGOV_SOFTWAREID_TRG"
ITGOV_STATUSLOG	Log of All changes in the Status of Purchasing Cycle	Log_id is the primary key which follows the sequence "ITGOV_LOGID_SEQ" on a trigger "ITGOV_LOGID"
ITGOV_USERS	Maintains All Users	Us_id is the primary key which follows the sequence "ITGOV_USID_SEQ" on a trigger "ITGOV_USID"
ITGOV_VENDORS	Maintains All Vendors	Vendor_id is the primary key which follows the sequence "ITGOV_VENDORID_SEQ" on a trigger "ITGOV_VENDORID"

## **Packages**

The Database packages used in this tool are as follows:

### **ITGOV\_PACK**

The following are the functions and their functionality included in the above package.

- Insert\_Product  
Procedure which inserts a new record into ITGOV\_PRODUCTS table.
- Insert\_Software  
Procedure which inserts a new record into ITGOV\_SOFTWARES table.
- Insert\_Request  
Procedure which inserts a new record into ITGOV\_PURCHASEREQUESTS table.
- Insert\_Item  
Procedure which inserts a new record into ITGOV\_REQUEST\_ITEMS table.
- Insert\_Purchase  
Procedure which inserts a new record into ITGOV\_PURCHASES table.
- Insert\_Purchases\_Item  
Procedure which inserts a new record into ITGOV\_PURCHASES\_ITEMS table.
- Concatenate\_List  
Function which concatenates the input into a comma separated list

## Appendix

### Term Definitions

- HTML – HTML stands for **Hyper Text Mark Up Language**. HTML elements are the basic building-blocks of webpages.
- CSS – CSS stands for **Cascading Style Sheets**. It's most common application is to style web pages written in HTML and XHTML.
- CFML – CFML stands for **ColdFusion Markup Language**. Multiple commercial and open source implementations of CFML engines are available, including Adobe ColdFusion, New Atlanta Blue Dragon, Railo. We at INSEAD use Adobe Cold Fusion.
- JQuery - **jQuery** is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML.
- Adobe Live Cycle Designer - Its software used to create robust and intuitive templates for creating interactive forms and personalized documents.