



Building COINS Desktops

Prepared by:

Documentation Team

Prepared for:

Learning Resources

Module:

Business Intelligence

Date:

29 September 2016

Document Ref:

LMDSY0063

Version:

11.05

Construction Industry Solutions Ltd.
11 St. Laurence Way
SL1 2EA



Contents

1 Building COINS Desktops - Overview	1
2 Desktop Structure	2
2.1 Creating the Functions (Manually)	3
2.1.1 Desktop Functions	3
2.1.2 Desktop Tab Functions	3
2.2 Add the Desktop to a user profile	4
2.3 Desktop Section Functions	6
2.4 Uploading Desktop Images	8
2.5 Tile Rows and Columns	10
2.5.1 Desktop - Tile Rows	11
2.5.2 Desktop - Tile Columns	12
2.6 Add the Desktop to a user profile	13
2.7 Alternative Desktops	15
2.7.1 User Maintenance	15
2.8 Desktop Import/Export	17
2.8.1 User Maintenance	17
2.8.2 Export	17
2.8.3 Import	19
3 Tiles	21
3.1 Smart Tiles	21
3.2 Info Tiles	24
3.3 Using Calculation Programs to prepare data for the Desktop	28
3.4 Desktop Tile Values	31
3.4.1 Examples of Desktop Tile Values	31
3.5 Tile Options	34
3.5.1 NamedFilter Example	36
3.5.2 InitContainer Example	39
3.6 Tables on Tiles	41
3.6.1 Tables : Example 1 - Direct Query against the database	42
3.6.2 Example 2 - Using a Data Set	44
3.7 Charts on Tiles	46
3.7.1 Dataset driven Charts	46



Copyright 2016 Construction Industry Solutions Ltd.. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Construction Industry Solutions Ltd..

Construction Industry Solutions Ltd.
11 St. Laurence Way
SL1 2EA

THIS USER GUIDE WAS CREATED USING MADCAP FLARE.



1 Building COINS Desktops - Overview

This guide is intended for anyone with OA Designer skills to create and maintain COINS Desktops.

This guide is not intended for end-users for which Desktop Wizards are available to build desktops from pre-defined components. End User documentation is available from the OA Navigation section of the COINS Learning Resources on the COINS Client Area:
www.coins-global.com)

The guide will work through the creation of an example Desktop. It covers the creation of the Desktop structure and explores the configuration of the various Tile types that are available.



2 Desktop Structure

To configure a user's desktop, set up a new three-level menu function.

- The first level is the desktop itself. This is the function specified on the user record
- The second level must contain one or more menus which correspond to the tabs on the desktop.
- Each of these "tab" menus must contain one or more menu functions with a new type: Section. These correspond to the collapsible sections on the desktop. If a tab has only one section, the section is displayed without a title.
- The menu items on the section menu correspond to the tiles. Desktop Menu Item Maintenance allows you to configure which function the tile runs, how the function opens, and what the tile looks like.

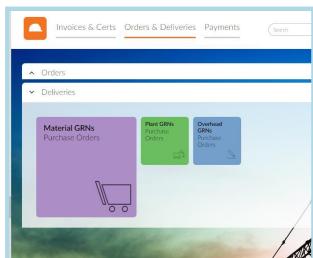


If the user configures their own desktop, COINS automatically creates functions with the following naming convention:

Tabs - the function code of the desktop menu + T + n (where n is the number of the tab).

Sections - the function code of the tab + S + n (where n is the number of the section on the tab)

Desktop Menu for User JSITH



Function	Explanation
JSITH	Desktop Menu
JSITH1	First Tab
JSITH2	Second Tab
JSITH2S1	First Section on Second Tab
JSITH2S1	Second Section on Second Tab
%WPOxxxxxx	Functions on Section
%WPLxxxxxx	
%WSCxxxxxx	
%WJCxxxxxx	

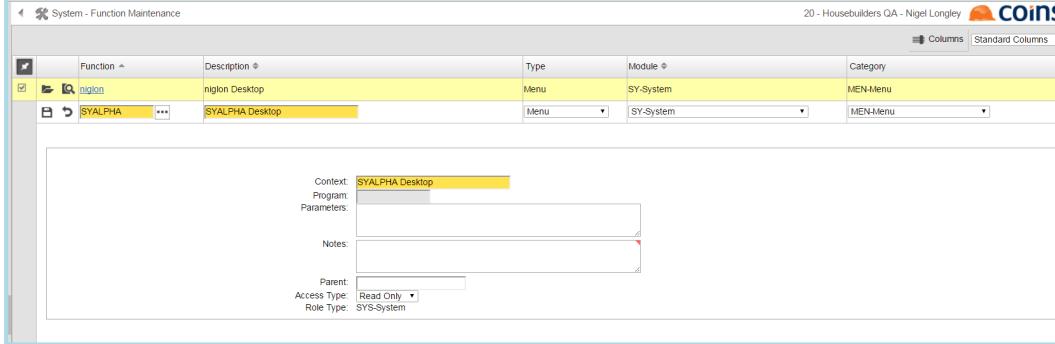
2.1 Creating the Functions (Manually)

A desktop wizard is available to automate much of this work, however to ensure a good understanding of how the desktop works, we will build it manually so that we can explore each of the components.

For our example, we are going to create a Desktop called SYALPHA with two Tabs and two sections on each tab

2.1.1 Desktop Functions

First we need to create the Top Level Desktop Function.



Function	Description	Type	Module	Category
nigel	nigel Desktop	Menu	SY-System	MEN-Menu
SYALPHA	SYALPHA Desktop	Menu	SY-System	MEN-Menu

Context: SYALPHA Desktop
Program:
Parameters:
Notes:
Parent:
Access Type: Read Only
Role Type: SYS-System

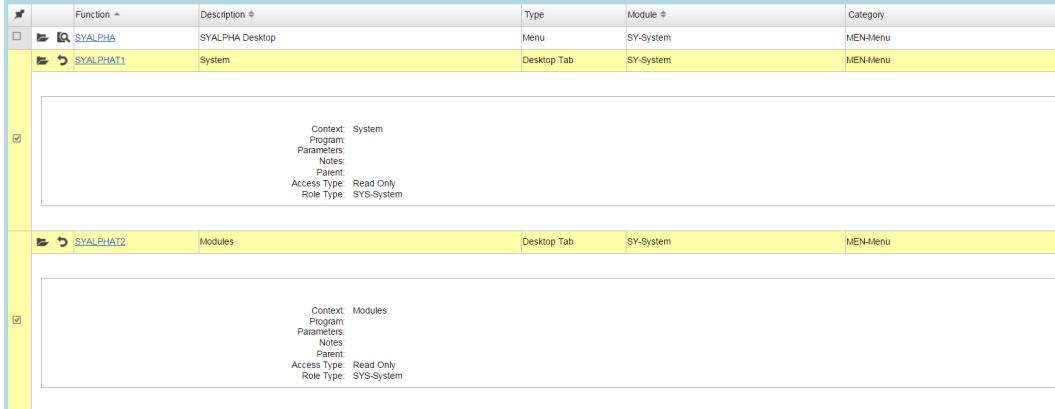


Ensure that the module is SY-System and the Category is MEN-Menu.

Click Save

2.1.2 Desktop Tab Functions

- Next we can create the two Tabs. We will have two Tabs, one called System and the other called Modules
- Copy the Top Level Function and add T1 to the function name for the first tab, and add T2 to the name of the second function for the second tab.
- Change the Type to Desktop Tab
- Change the Description and Context fields to System and Modules respectively.



Function	Description	Type	Module	Category
SYALPHA	SYALPHA Desktop	Menu	SY-System	MEN-Menu
SYALPHAT1	System	Desktop Tab	SY-System	MEN-Menu
SYALPHAT2	Modules	Desktop Tab	SY-System	MEN-Menu

Context: System
Program:
Parameters:
Notes:
Parent:
Access Type: Read Only
Role Type: SYS-System

Context: Modules
Program:
Parameters:
Notes:
Parent:
Access Type: Read Only
Role Type: SYS-System



Now we need to add our tabs to the top level.

- Click on the hyper link for function SYALPHA and click on the menu items tab.
- Add the functions SYALPHAT1 and SYALPHAT2 to the list.

	Function	Description	Context	Type	Module	Category
<input type="checkbox"/>	SYALPHAT1	System		Desktop Tab	SY-System	Menu
<input type="checkbox"/>	SYALPHAT2	Modules		Desktop Tab	SY-System	Menu

We can now test the initial design

2.2 Add the Desktop to a user profile

The desktop needs to be configured for each user that will use it. Different users can use the same desktop, or each user can have their own desktop. Configuring the desktop involves setting up a menu and specifying the menu on the user record.

In System > User Maintenance> Users select a user (or create a new account for test purposes) and select the User Preferences tab.



Field	Description
Desktop	<p>Allows the system administrator to specify the menu function that displays the desktop for this user. If the menu function does not already exist, COINS creates a new function; this in effect creates an empty Desktop, which either the system administrator or the user can configure.</p> <p>For our example, enter SYALPHA as the desktop name.</p> <p>If this is left blank, the user has no Desktop set up, and the Set Homepage button is available to allow the user to select a single function as their homepage.</p>
Desktop Image	<p>Allows the system administrator to specify the background image for this user's desktop. This is a drop-down list of the available image files. Standard images are available, and administrators can upload additional ones (see later).</p> <p>Choose an existing image for your desktop.</p> <p>Login as your test user. The desktop should be displayed on login.</p>



If a Desktop has been set up for a user, but they have a user home page set, when they log in they will see the home page (but will still be able to access the Desktop). If you clear the home page from their user record, they will see the Desktop when they log in.



2.3 Desktop Section Functions

Currently we only have the desktop and two tabs. Now we can look at creating the sections.

In Function Maintenance, create two new functions SYAPLHAT1S1 and SYAPLHAT1S2

For both, set the Type to Desktop Section

For SYALPHAT1S1 the description and Context should be Utilities

For SYALPHAT1S2 the description and Context should be Maintenance

	Function	Description	Type	Module	Category
	SYALPHA	SYALPHA Desktop	Menu	SY-System	MEN-Menu
	SYALPHAT1	System	Desktop Tab	SY-System	MEN-Menu
	SYALPHAT1S1	Utilities	Desktop Section	SY-System	MEN-Menu
	SYALPHAT1S2	Maintenance	Desktop Section	SY-System	MEN-Menu

Context: Maintenance Program: Parameters: Notes: Parent: Access Type: Read Only Role Type: SYS-System

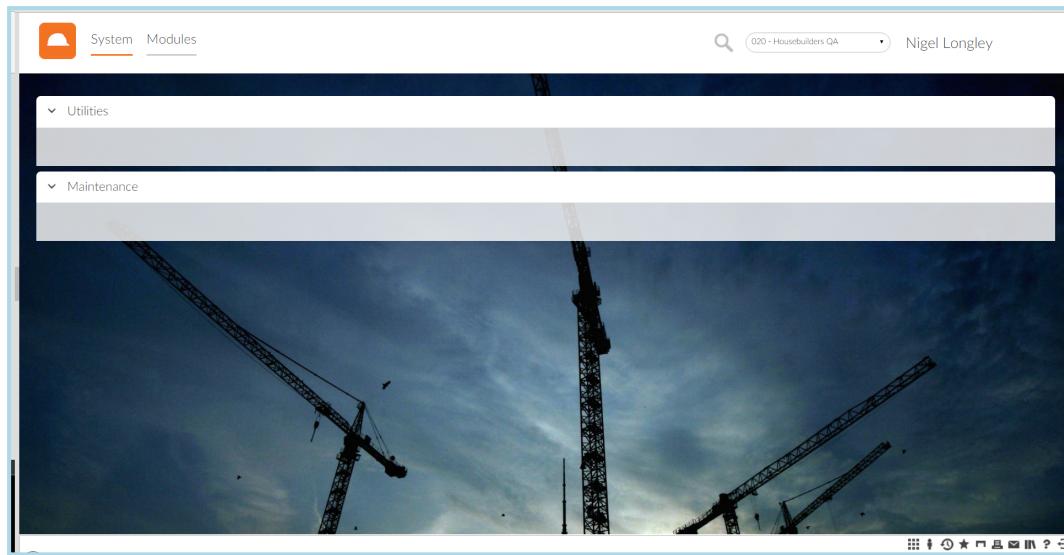
Context: Modules Program: Parameters: Notes: Parent: Access Type: Read Only Role Type: SYS-System

Add these two functions to the Menu Items Tab of SYALPHAT1

Function: SYALPHAT1 System							Link	Options
	Function	Description	Context	Type	Module	Category		
	SYALPHAT1S1	Utilities		Desktop Section	SY-System	Menu		
	SYALPHAT1S2	Maintenance		Desktop Section	SY-System	Menu		

(Navigation icons, Add, Copy, Insert, Up, Down, Open, Delete, Copy dropdown, Advanced search, Open button)

Test the design for your test user.

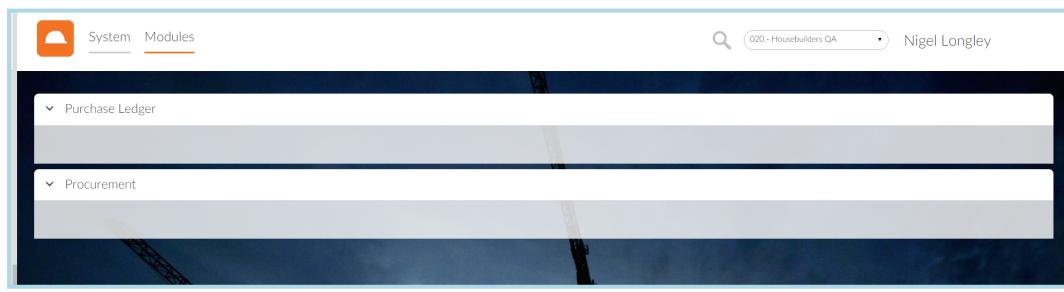
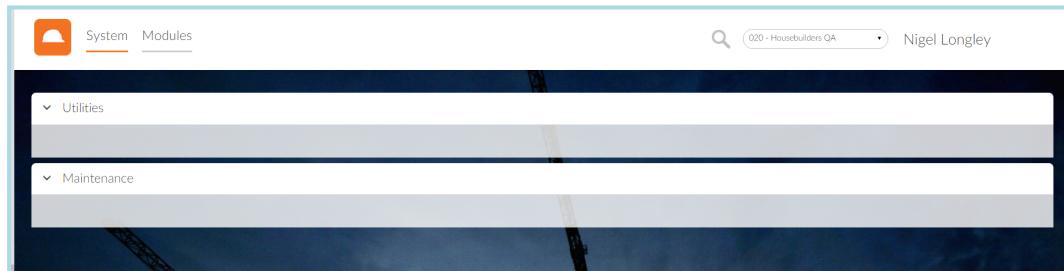


Repeat the steps above to create sections for the Second Tab (SYALPHAT2S1 and SYALPHAS2T2)

For SYALPHAT2S1 the description and Context should be Purchase Ledger

For SYALPHAS2T2 the description and Context should be Procurement

Test your design



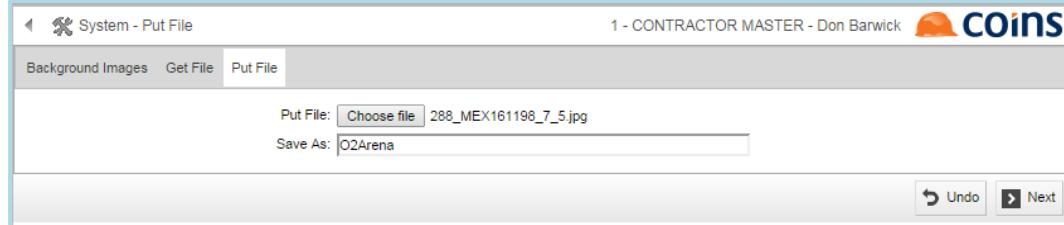


2.4 Uploading Desktop Images

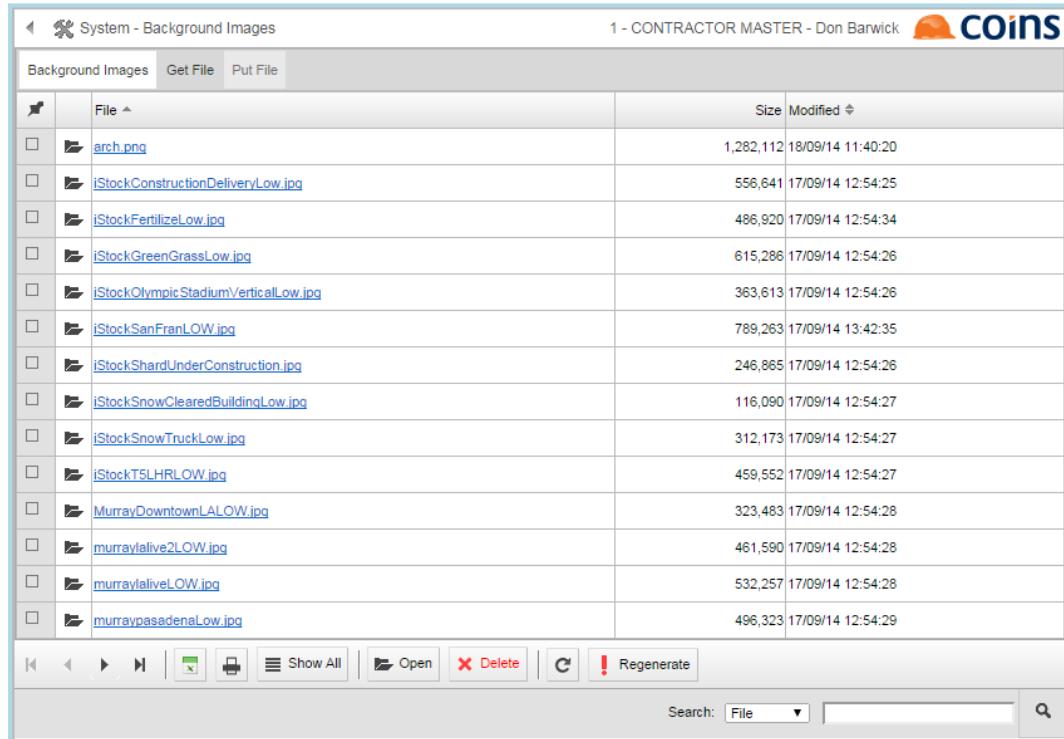
Two new functions allow the system administrator to upload images used on the desktop.

- System > System Setup > Background Images allows you to load images that can be used as the desktop background. Individual users can then select one of these to use on their desktop.

Background Images - Put File Tab



The file name you give to the file is used in the Background Image selection list in User Maintenance. The images are saved in the \$BASE/custimage/bg/ directory. Some standard images are provided in the \$BASE/images/bg/ directory; you can copy these or upload them using Background Images.



- System > System Setup > User Images allows you to load the images that are used as the account picture on each user's desktop. The image files must be JPG format files, and should be approximately 60px square. (You can use larger images but the browser will resize them, which does not always produce good results.) You must name the files

<userid>.jpg (for example, for a user whose user ID is JSMITH, the file would be jsmith.jpg. The images are saved in the \$BASE/custimage/sysuser/ directory.

A screenshot of a web-based application window titled "System - Put File". The window shows a file upload interface with a "Choose file" button containing the text "nigel.jpg". Below it is a "Save As:" field with the value "nigcop.jpg". At the bottom right are "Undo" and "Next" buttons. The top right corner of the window displays the COINS logo. The top bar also shows the title "1 - CONTRACTOR MASTER - Don Barwick".

If there is no file for a user, the account image will be the image from the user's COINS HR record if one exists, otherwise it will be a silhouette image.

Account Image Silhouette





2.5 Tile Rows and Columns

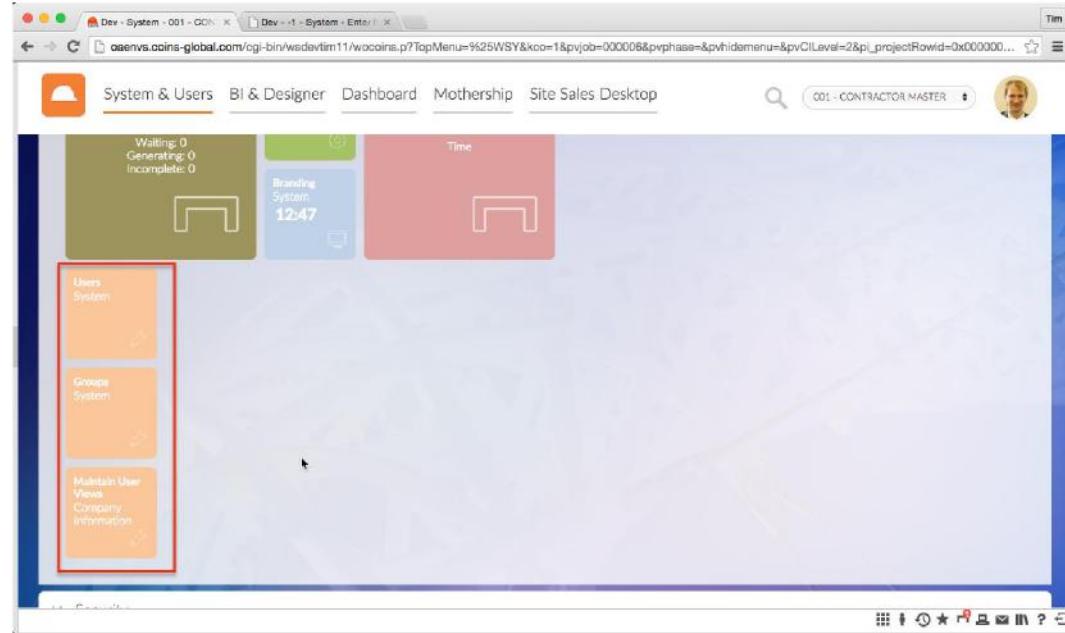
On tile maintenance it is now possible to group tiles into rows or columns, by assigning codes to tiles.

#	Function	Description	Title	Override	Subtitle	Size	Open In	Type	Module	Category	Row	Column
1		Report Status	Scheduler			Normal	Frame	Container	SY-System	Workbench		
2		Current Users				Small	Frame	Function	SY-System	Administration	A	
3		Module (Licence) (Enquiry)	Branding			Small	Frame	Function	SY-System	Enquiry	A	
4		Report Status	Scheduled Reports	<input checked="" type="checkbox"/>		Normal	Main Area	Container	SY-System	Workbench		
5		User Maintenance				Small	Frame	Function	SY-System	Maintenance	A	
6		Group Maintenance				Small	Frame	Function	SY-System	Maintenance	A	
7		Maintain User Views				Small	Frame	Function	CI-Company Information	Maintenance	B	

The Row code allows row breaks to be inserted. If a row code changes from one item to the next then a new row is started. (This can already be achieved by inserting a blank function to cause a new row to start but this is introduced to provide equivalent of the column code; see below.)



2.5.1 Desktop - Tile Rows



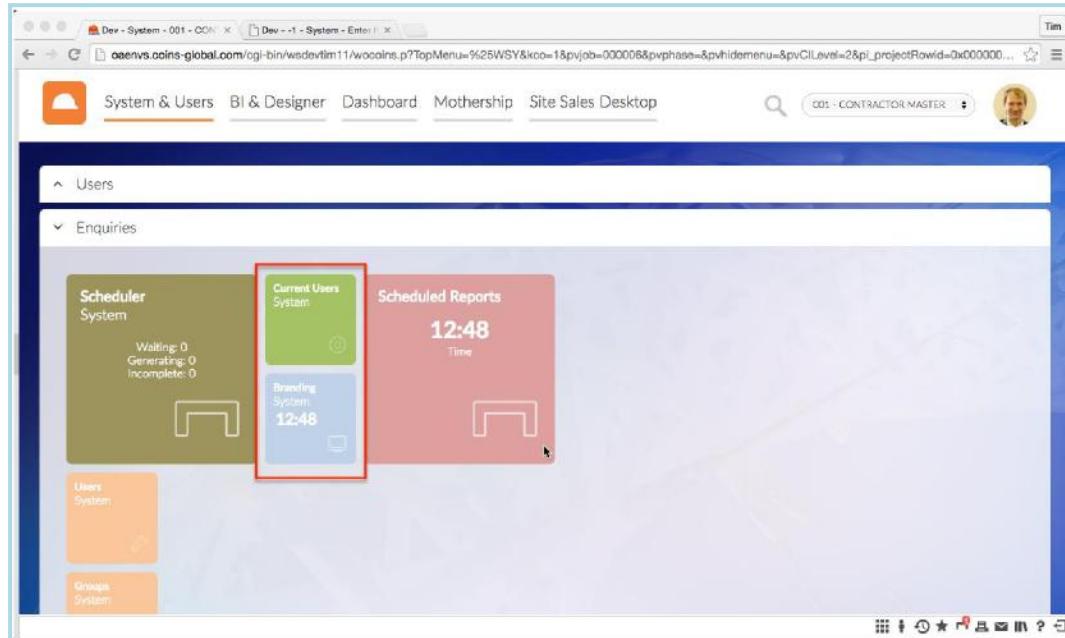
In the example above a new row will start for User Maintenance (A), Group Maintenance (Blank) and Maintain User Views (B).

It is also possible to group tiles in to columns.

Where a set of tiles shares a common Column code then they are set out in a single column.



2.5.2 Desktop - Tile Columns



In the example above, Current Users and Module Licence Enquiry share the same column code, "A", so are grouped in a single column on the desktop. Columns are only grouped for non-blank column codes.



2.6 Add the Desktop to a user profile

The desktop needs to be configured for each user that will use it. Different users can use the same desktop, or each user can have their own desktop. Configuring the desktop involves setting up a menu and specifying the menu on the user record.

In System > User Maintenance > Users select a user (or create a new account for test purposes) and select the User Preferences tab.

Field	Description
Desktop	<p>Allows the system administrator to specify the menu function that displays the desktop for this user. If the menu function does not already exist, COINS creates a new function; this in effect creates an empty Desktop, which either the system administrator or the user can configure.</p> <p>For our example, enter SYALPHA as the desktop name.</p> <p>If this is left blank, the user has no Desktop set up, and the Set Homepage button is available to allow the user to select a single function as their homepage.</p>
Desktop Image	<p>Allows the system administrator to specify the background image for this user's desktop. This is a drop-down list of the available image files. Standard images are available, and administrators can upload additional ones (see later).</p> <p>Choose an existing image for your desktop.</p> <p>Login as your test user. The desktop should be displayed on login</p>



If a Desktop has been set up for a user, but they have a user home page set, when they log in they will see the home page (but will still be able to access the Desktop). If you clear the home page from their user record, they will see the Desktop when they log in.



2.7 Alternative Desktops

You can setup more than one desktop (which you can think of as roles of the user or personas of the user) the user then can select from the available desktops switching their role/persona. The tabs, sections and items on the desktop refresh based on the latest selected role/persona.

2.7.1 User Maintenance

A new option is introduced on the user maintenance page

The screenshot shows the COINS User Maintenance page. On the left is a navigation tree with various system modules like Home Page, System Administration, My Files, and User Maintenance. In the main area, there's a form for 'User: TIMARM' with fields for Name, Last Login, and several configuration options under 'Recall' and 'Report' sections. A red box highlights the 'Alternative Desktops' field, which contains 'TIMARM,ALESMY'. Other fields shown include 'Current Desktop: TIMARM', 'Desktop Image: coins_arch.png', and 'Main Menu: %WMAIN'. At the bottom, there are RSS feed checkboxes and a toolbar with Audit, Undo, Save, and other icons.

The alternative desktops is a list of desktops that will already exist that the user can switch to. The currently selected desktop is shown in the existing desktop field. This is the one that the user will see if they login (or refresh) the COINS page.

The alternative desktops are shown at the top left of the desktop above the tabs.

2 Desktop Structure



A screenshot of a web browser window showing the COINS desktop interface. The title bar says "Dev - System - 001 - CON". The address bar shows "oaenvs.coins-global.com/cgi-bin/wsdevtim11/wocoins.p?kco=1&pvjob=000006&pvphase=&pvhidemenu=&pvCILevel=2&rtbtask=016781&pvFrame=W%2C1333%2C705&To...". The top navigation bar includes tabs for "System & Users", "BI & Designer", "Dashboard", "Mothership", and "Test". A search bar and user profile "Tim Armitage" are on the right. The main content area has a blue header "Users" with tiles for "System Administration", "Users System", "My Groups System", "User Report System", "User Report System 46363" (highlighted in orange), "TIM System", and "Roles System". Below this are sections for "Enquiries" and "Security".

As soon as you select an alternative (examples are similar but different tabs, sections and tiles) your user record is changed so that this becomes your default desktop and the page is refreshed showing you the new desktop.

A screenshot of a web browser window showing the COINS desktop interface. The title bar says "Dev - System - 001 - CON". The address bar shows "oaenvs.coins-global.com/cgi-bin/wsdevtim11/wocoins.p?kco=1&pvjob=000006&pvphase=&pvhidemenu=&pvCILevel=2&rtbtask=016781&TopMenu=%25WSY&pi_projectRow...". The top navigation bar includes tabs for "System & Users", "BI & Designer", "RTB", "Sandpit", and "Empty". A search bar and user profile "Tim Armitage" are on the right. The main content area has a blue header "Users" with sections for "Enquiries" and "Security".



2.8 Desktop Import/Export

Desktops can be maintained in COINS and are retained as functions, menus and menu items. It is possible to copy the functions, menus and menu items using standard function export and import features. From 11.04 new functionality makes the export and import of desktops between users and/or between environments easier.

2.8.1 User Maintenance

Two new buttons have been added to user maintenance. A desktop export and a desktop import button alongside the Desktop field on the summary page.

These can be used to export the user's desktop data (menus and menu items) to a file and then to upload this data either to another user or to another environment.

2.8.2 Export

Pressing the export button causes a download of the menu and menu item data for the current user.

This finds the desktop menu (in this case TIMARM) and exports it together with all the submenus (sections and tabs) and the items on those menus. Only menus (and contents) of user specific menus are exported i.e. beginning TIMARM. This is to prevent any standard menus being copied (and overwritten). It is expected that standard menus (tabs/sections) would be copied using existing standard techniques. The menus containing the reference to the standard menus is exported.



The screenshot shows the COINS desktop application interface. The main window title is "System - User Details" for user "TIMARM". The left sidebar contains a navigation tree with categories like System Administration, My Files, User Maintenance, Function Security, and System Setup. The main content area displays various configuration settings and links. At the bottom left, there is a file list with "desktop.txt" highlighted by a red box.

The export file contains the original desktop code (TIMARM) and an export of the functions, menus and items that make it up.

```

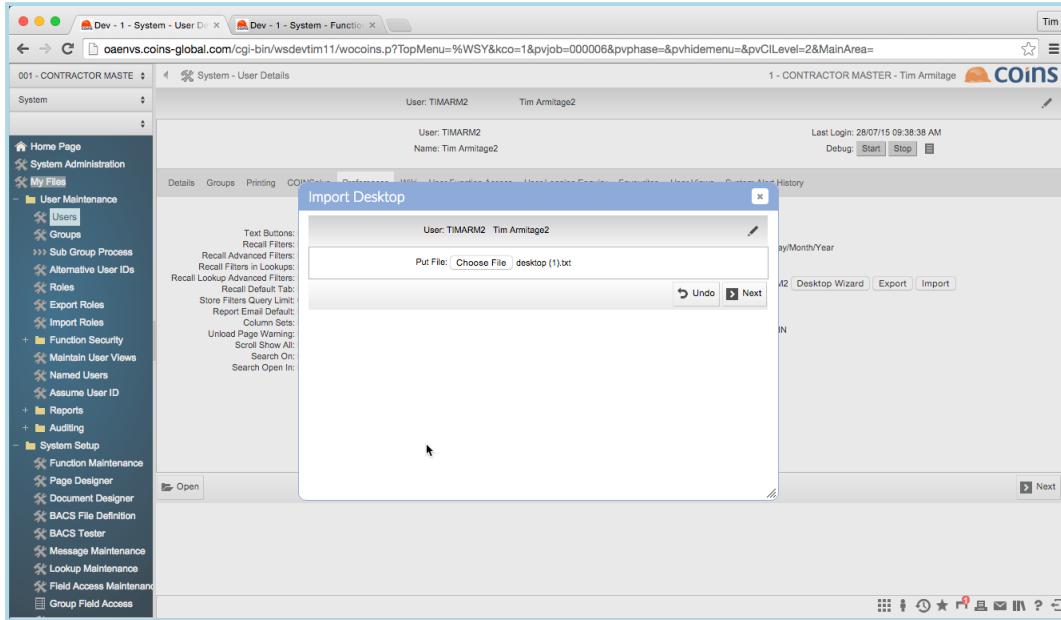
"TIMARM"
"TIMARM" "TIMARM Home Page" "" "" "" 0 "SY" "Menu" "TIMARM" "TIMARM" "MEN" "TIMARM Home
Page" yes "" "SYS" 20140709.63729 "TIMARM" "11.02" no "" "" ""
"2014-07-09T17:42:090x0000000001d31884" "" "" ""
"TIMARMT1" "System & Users" "" "" "" 0 "R" "SY" "Menu" "TIMARMT1" "" "MEN" "System &
Users" yes "" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""
"2014-12-18T08:29:510x000000000215ea05" "" "" ""
"TIMARMT1S1" "Users" "" "" "" 0 "R" "SY" "Section" "TIMARMT1S1" "" "MEN" "Users" yes ""
"SYS" 20141218.30591 "TIMARM" "11.03" no "" "" "" "2014-12-18T08:29:510x000000000215ea08"
"" "" ""
"TIMARMT1S2" "Enquiries" "" "" "" 0 "R" "SY" "Section" "TIMARMT1S2" "" "MEN" "Enquiries"
yes "" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""
"2014-12-18T08:29:510x000000000215eb00" "" "" ""
"TIMARMT1S3" "Security" "" "" "" 0 "R" "SY" "Section" "TIMARMT1S3" "" "MEN" "Security" yes
"" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""
"2014-12-18T08:29:510x000000000215eb06" "" "" ""
"TIMARMT2" "BI & Designer" "" "" "" 0 "R" "SY" "Menu" "TIMARMT2" "" "MEN" "BI & Designer"
yes "" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""
"2014-12-18T08:29:510x000000000215eb0a" "" "" ""
"TIMARMT2S1" "Maintenance" "" "" "" 0 "R" "SY" "Section" "TIMARMT2S1" "" "MEN"
"Maintenance" yes "" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""
"2014-12-18T08:29:510x000000000215eb0d" "" "" ""
"TIMARMT2S2" "Utilities" "" "" "" 0 "R" "SY" "Section" "TIMARMT2S2" "" "MEN" "Utilities" yes
"" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""
"2014-12-18T08:29:510x000000000215eb15" "" "" ""
"TIMARMT3" "Dashboard" "" "" "" 0 "R" "SY" "Menu" "TIMARMT3" "" "MEN" "Dashboard" yes ""
"SYS" 20141218.30591 "TIMARM" "11.03" no "" "" "" "2014-12-18T08:29:510x000000000215eb81"
"" "" ""
"TIMARMT3S1" "Activity" "" "" "" 0 "R" "SY" "Section" "TIMARMT3S1" "" "MEN" "Activity" yes
"" "SYS" 20141218.30591 "TIMARM" "11.03" no "" "" ""

```



2.8.3 Import

Pressing the import button brings up an upload dialog.



Selecting the export file previous downloaded will cause the desktop that was exported to be imported as the desktop for the selected user. All the menus and functions that are extensions of the desktop menu will be renamed for the new desktop menu.

e.g. TIMARMT1 would be renamed TIMARM2T1

TIMARMT1S1 would be renamed TIMARM2T1S1

This applies to the menus and the items (sub menus) on them.

Having imported the TIMARM desktop the TIMARM2 desktop looks the same



This screenshot shows the COINS desktop environment. The top navigation bar includes tabs for 'System & Users', 'BI & Designer', 'Dashboard', 'Mothership', and 'Site Sales Desktop'. A search bar and user profile for 'Tim Armitage2' are also present. The main content area displays a hierarchical menu structure under 'System & Users'.

This is copy of the original and TIMARM2* functions, tabs and sections now exist.

Function	Description	Type	Module	Category	Locked By
TIMARM2	TIMARM2 Desktop	Menu	SY-System	MEN-Menu	
TIMARM2T1	System & Users	Menu	SY-System	MEN-Menu	
TIMARM2T1S1	Users	Section	SY-System	MEN-Menu	
TIMARM2T1S2	Enquiries	Section	SY-System	MEN-Menu	
TIMARM2T1S3	Function Maintenance - Function Details	Section	SY-System	MEN-Menu	
TIMARM2T2	BI & Designer	Menu	SY-System	MEN-Menu	
TIMARM2T2S1	Maintenance	Section	SY-System	MEN-Menu	
TIMARM2T2S2	Utilities	Section	SY-System	MEN-Menu	
TIMARM2T3	Dashboard	Menu	SY-System	MEN-Menu	
TIMARM2T3S1	Activity	Section	SY-System	MEN-Menu	
TIMARM2T3S2	Reports	Section	SY-System	MEN-Menu	
TIMARM2T4	Mothership	Menu	SY-System	MEN-Menu	
TIMARM2T4S1	Dashboard	Section	SY-System	MEN-Menu	
TIMARM2T4S2	Other	Section	SY-System	MEN-Menu	

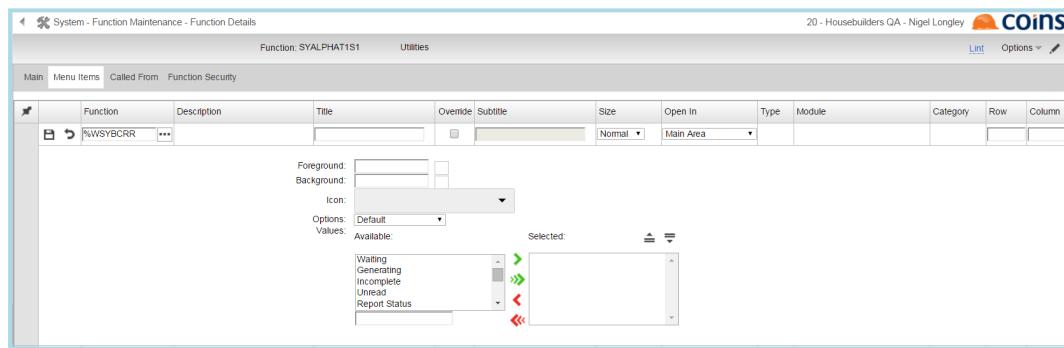
3 Tiles

3.1 Smart Tiles

Now that the basic structure of the desktop is in place we can start to add tiles to the sections. First of all we will add a smart tile to a standard coins function.

Smart Tiles show summary information about a function (such as financial values or the number of items requiring attention). The values on these tiles are kept up to date and refreshed automatically.

On the Desktop Section Function SYALPHAST1, open the Menu Items Tab and add the function %WSYBCRR (Report Status Workbench).



This additional fields allow you to configure the tile on the section.

For each tile, you can specify the following:

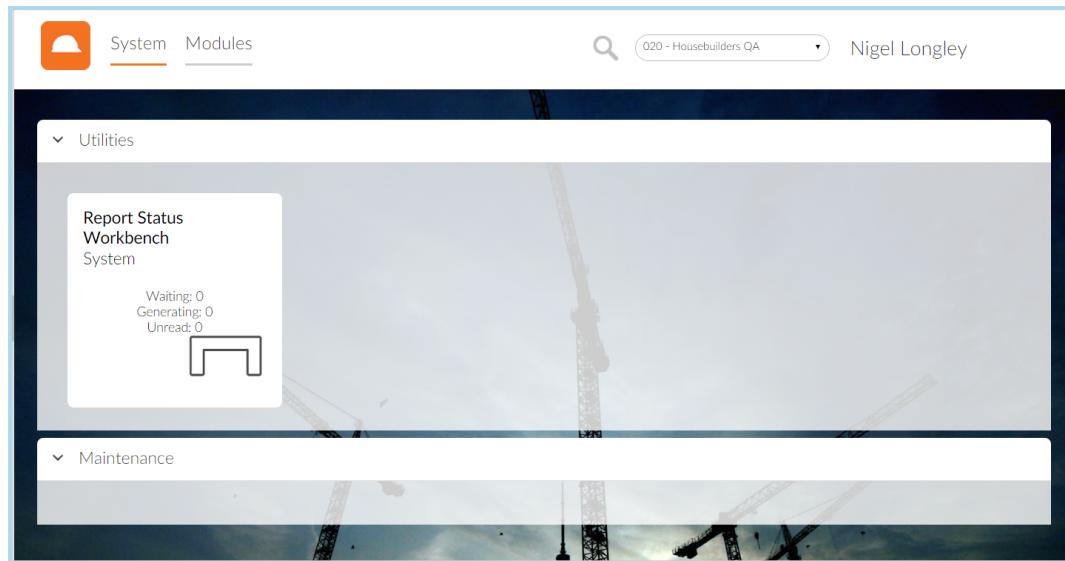
Field	Description
Function	The COINS function that the tile runs. You can leave this blank (that is, enter a line without a function) to force the following tiles to display on a new line.
Title	The description to display on the tile. If blank, the tile shows the standard context description.
Override	Whether to override the subtitle of the tile. If this is not ticked, the module the function belongs to is used as the subtitle.
Subtitle	The subtitle to display on the tile. If blank (and if Override is ticked), the subtitle is blank.
Size	The size of the tile (small, wide or normal). 'Wide' is intended for charts. 'Small' shows a compact tile; if the tile is configured to show data, labels are omitted.
Open In	How to run the function: in the main area (replacing the Desktop), in a new window (or tab, depending on the behaviour of your browser), or in a frame on top of the Desktop.



Field	Description																																	
Foreground	<p>The colour of the text on the tile. You can use the colour picker to choose a colour, or you can enter a hexadecimal colour code directly.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Foreground: <input style="width: 100px; border: none; border-bottom: 1px solid #ccc; height: 15px; vertical-align: middle;" type="text" value="#244061"/> #244061</p> <p>Background: <input style="width: 100px; border: none; border-bottom: 1px solid #ccc; height: 15px; vertical-align: middle;" type="text" value=""/> </p> <p>Size: <input style="width: 50px; border: none; border-bottom: 1px solid #ccc; height: 15px; vertical-align: middle;" type="text" value="14"/> 14</p> <p>Values: <input style="width: 100px; border: none; border-bottom: 1px solid #ccc; height: 15px; vertical-align: middle;" type="text" value=""/> </p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <p>Theme Colours</p> <table border="1" style="margin-bottom: 5px; border-collapse: collapse;"> <tr><td style="width: 15px; height: 15px; background-color: black;"></td><td style="width: 15px; height: 15px; background-color: white;"></td><td style="width: 15px; height: 15px; background-color: #003366;"></td><td style="width: 15px; height: 15px; background-color: #3366CC;"></td><td style="width: 15px; height: 15px; background-color: #CC6633;"></td><td style="width: 15px; height: 15px; background-color: #669999;"></td><td style="width: 15px; height: 15px; background-color: #99CC66;"></td><td style="width: 15px; height: 15px; background-color: #CC9933;"></td></tr> <tr><td style="width: 15px; height: 15px; background-color: #666666;"></td><td style="width: 15px; height: 15px; background-color: #CCCCCC;"></td><td style="width: 15px; height: 15px; background-color: #FFCCBC;"></td><td style="width: 15px; height: 15px; background-color: #FF9966;"></td><td style="width: 15px; height: 15px; background-color: #FFB74D;"></td><td style="width: 15px; height: 15px; background-color: #FFCCBC;"></td><td style="width: 15px; height: 15px; background-color: #FF9966;"></td><td style="width: 15px; height: 15px; background-color: #FFB74D;"></td></tr> </table> <p>Standard Colours</p> <table border="1" style="border-collapse: collapse;"> <tr><td style="width: 15px; height: 15px; background-color: #333333;"></td><td style="width: 15px; height: 15px; background-color: #666666;"></td><td style="width: 15px; height: 15px; background-color: #999999;"></td><td style="width: 15px; height: 15px; background-color: #CC9999;"></td><td style="width: 15px; height: 15px; background-color: #CC6666;"></td><td style="width: 15px; height: 15px; background-color: #CC3333;"></td><td style="width: 15px; height: 15px; background-color: #993366;"></td><td style="width: 15px; height: 15px; background-color: #666699;"></td><td style="width: 15px; height: 15px; background-color: #336699;"></td><td style="width: 15px; height: 15px; background-color: #3366CC;"></td><td style="width: 15px; height: 15px; background-color: #6699CC;"></td><td style="width: 15px; height: 15px; background-color: #99CC99;"></td><td style="width: 15px; height: 15px; background-color: #CC9966;"></td><td style="width: 15px; height: 15px; background-color: #CC6633;"></td><td style="width: 15px; height: 15px; background-color: #993333;"></td><td style="width: 15px; height: 15px; background-color: #666666;"></td><td style="width: 15px; height: 15px; background-color: #333333;"></td></tr> </table> <p>More Colours History</p> <p style="margin-top: 5px;"> #244061 #953734 </p> </div> </div>																																	
	If this is blank, the text is black.																																	
Background	The colour of the tile background. As with the foreground, you can use the colour picker or enter a hexadecimal code. If this is blank, the tile background is white.																																	
Icon	The icon to display on the tile. If this is blank, the default icon, which is based on the category of the function, is shown on the tile.																																	
Options																																		
Values	If the function has been written to allow data to be displayed on the tile, this allows you to select which data to show.																																	

For our first tile, enter the title of Report Status Workbench, Open In Frame and select the values for Waiting, Generating and Unread.

Save the entry and test the Desktop.



Where values are displayed, such as our tile above, the polling of the Tile Values (refresh) is controlled by the same parameter that controls the User Info Messages on Report Status etc and this param is SY/USERINFO

3.2 Info Tiles

In the previous example, we used Tiles to link to functions that allowed us to work with data in COINS.

Another type of Tile, called an Info Tile, can be used to simply display information. This is essentially a detail page presented as a tile.



It is important to note that Info Tiles do not automatically refresh - the data displayed will remain static until a manual refresh of the desktop is done. For this reason, they may not be suitable on a dashboard screen where dynamic data is required.

Create a new function called SYAPLHAINF1. For this to work correctly, it is IMPORTANT that the Category used is INF - Information Tile and the program is set as wou005.

The description for now can be simply Information.

	Function ▾	Description ▾	Type	Module ▾	Category	Program	Access Type
□	SYALPHA	SYALPHA Desktop	Menu	SY-System	MEN-Menu		Read Only
□	SYALPHAT1	System	Desktop Tab	SY-System	MEN-Menu		Read Only
□	SYALPHAT1S1	Utilities	Desktop Section	SY-System	MEN-Menu		Read Only
□	SYALPHAT1S2	Maintenance	Desktop Section	SY-System	MEN-Menu		Read Only
□	SYALPHAT2	Modules	Desktop Tab	SY-System	MEN-Menu		Read Only
□	SYALPHAT2S1	Purchase Ledger	Desktop Section	SY-System	MEN-Menu		Read Only
□	SYALPHAT2S2	Procurement	Desktop Section	SY-System	MEN-Menu		Read Only
□	SYAPLHAINF1	Information	Function	SY-System	INF-Information Tile	wou005	Read Only

Context: Information
Program:
Parameters:

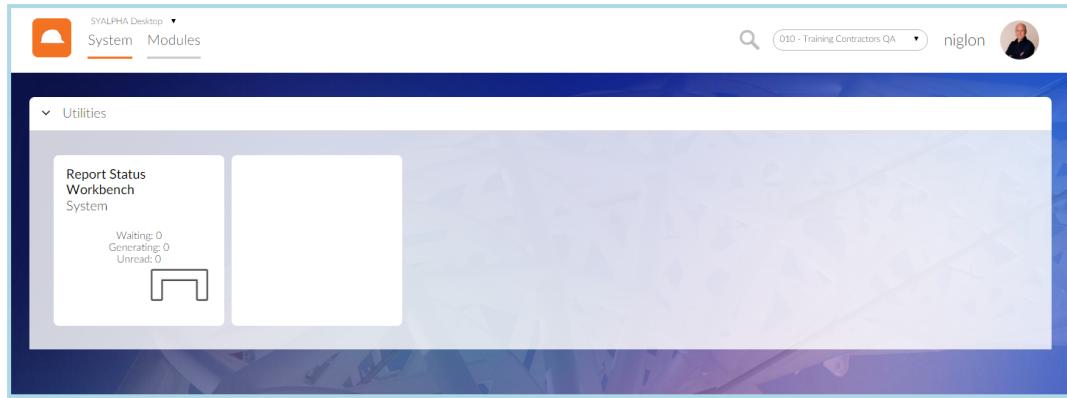
Notes:

Parent:
Access Type:
Role Type:

Save this new function, and then add it to the Menu Tab SYALPHAT1S1, setting the size to Normal.

	Function	Description	Title	Override	Subtile	Size	Open In	Type	Module	Category	Row	Column
□	%WSYBCRR	Report Status	Report Status Workbench	<input type="checkbox"/>		Normal	Main Area	Container	SY-System	Workbench		
□	SYAPLHAINF1	Information		<input type="checkbox"/>		Normal	Frame	Function	SY-System	Information Tile		

Refresh the Desktop to see the result - you should now have an empty Info Tile displayed on the Utilities section.



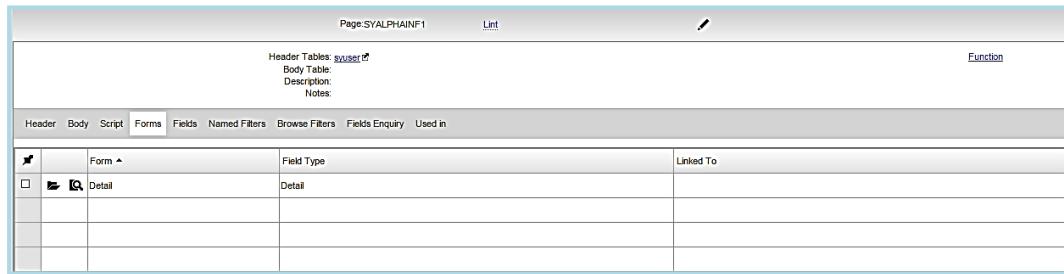
Now we need to define the Detail Page that will populate the tile.

By default, when a user logs into COINS, tables such as co_config, sysuser and menuparm are already available, so for our first Info Tile we will build a simple tile that displays the details of the logged in user.

In Page Designer, create a new Page Section also called SYALPHAINF1.

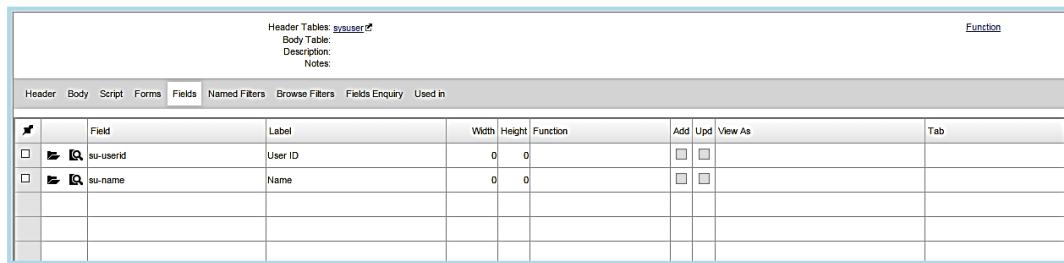
Set the Header Table as sysuser.

Click Save and add a Detail Form to the Page.



On the Fields Tab, set the Form filter to Detail and apply the filter.

Now on the Details Form add the fields su-userid and su-name



Again, refresh the Desktop to see the changes.



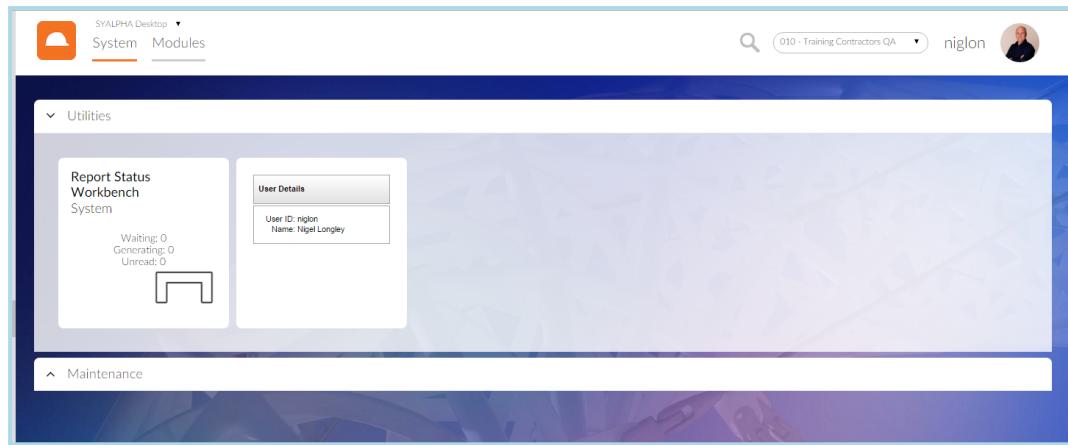
All of the functionality of a standard Detail Page are available to be used on Info Tiles.

To demonstrate this we will add a layout to the data.

Create a new field on the Detail Form call LAY1, with the Label of User Details and set View AS to Layout. In the Class, enter LAYA.

Save the new field and then for su-userid and su-name, set the Layout field to LAY1.

Refresh the Desktop to see the changes.



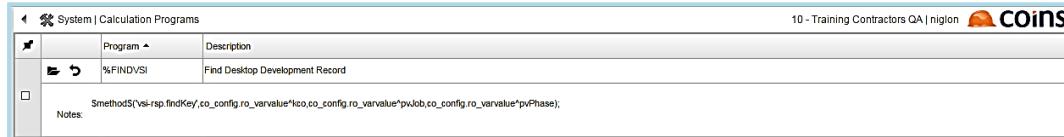
3.3 Using Calculation Programs to prepare data for the Desktop

Our Info Tile was able to access information from co_config, sysuser and menuparm without any additional work on our part.

It is possible to set up access to other tables before the desktop loads using Calculation Programs, this will then allow the tiles access to that information..

Calculation programs are essentially pre-written standard OA Calculation syntax that can be called on reports and browse screens. They can be as complex as required.

For example, for House Sales there is a standard Calculation program called %FINDVSI



The screenshot shows a table with one row for the %FINDVSI program. The columns are 'Program' and 'Description'. The 'Program' column contains '%FINDVSI' with a small icon next to it. The 'Description' column contains 'Find Desktop Development Record'. Below the table, a note says: 'Notes: \$method\$('vsi-rsp.findKey',co_config.ro_varvalue^kco,co_config.ro_varvalue^pvJob,co_config.ro_varvalue^pvPhase);'.

This uses the following method to access vsi-rsp with a given kco, Development and Phase code to return the RowID of that Development.

```
$method$('vsi-rsp.findKey',co_config.ro_varvalue^kco,co_config.ro_varvalue^pvJob,co_config.ro_varvalue^pvPhase);
```

Note that the values are stored as kco (company number retrieved from the URL), pvJob and pvPhase. The pv prefix indicates that these are permanent variables and once stored are retained in the URL until removed or their value changes. Setting the pv values allows a specific development to be selected and any relevant Info Tiles on the next Desktop build will then show information for that development.

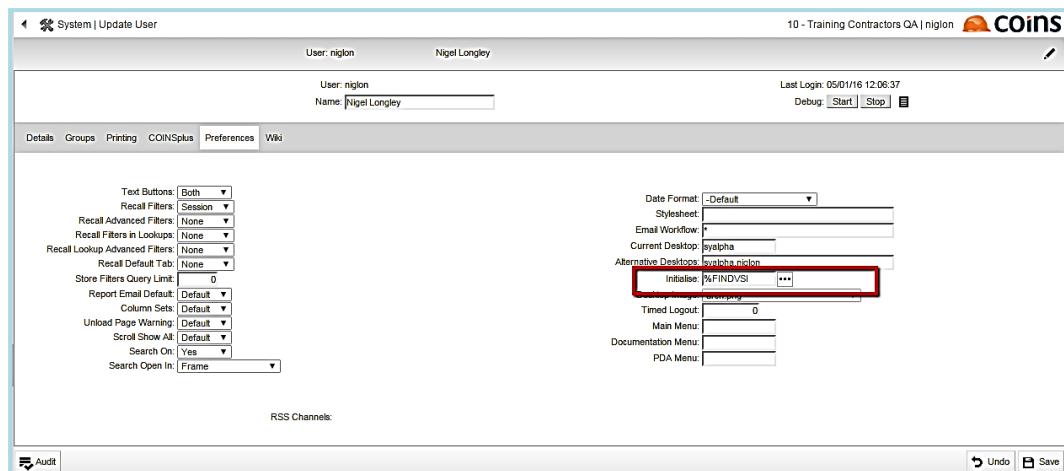
Currently (Jan 2016) there are only four pv values currently available - JC job (pvJob) and phase (pvPhase) and CI (pvView) and SRM (pvSibling).

To see how this can work, first we need to add this program to our user record.

In User Maintenance, find your userid and open the record. Select the Preferences Tab.

In the Initialise field, enter %FINDVSI

3 Tiles



The screenshot shows the 'System | Update User' page for a user named 'nigion'. The 'Preferences' tab is selected. In the 'Initialise' field, the value '%FINDVSI' is highlighted with a red box.

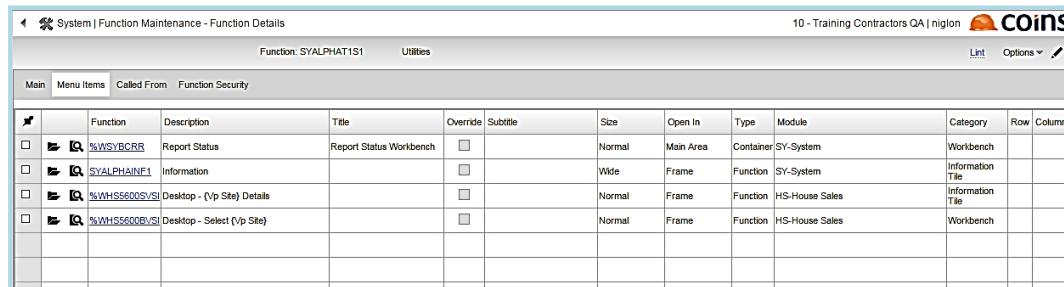
Save the record.

For demonstration purposes we are going to borrow some functionality from the Standard House Sales Dashboard

On the Utilities section of your Desktop add the two following Coins Standard Function as separate entries.

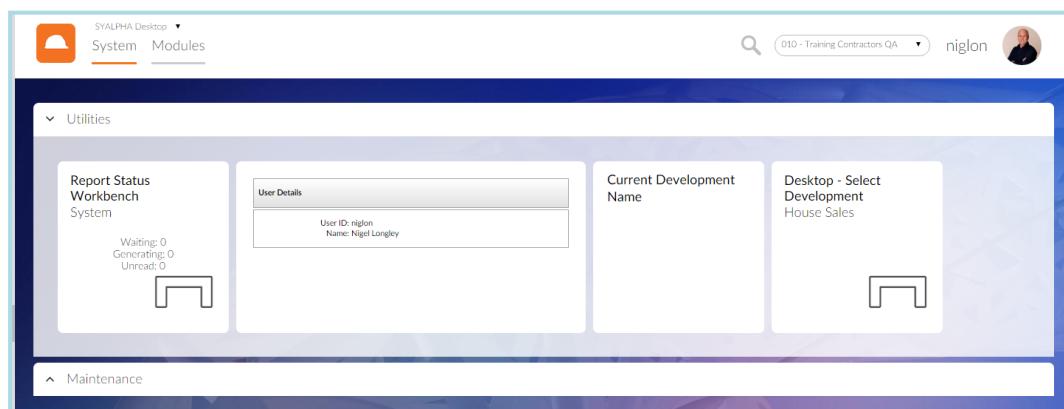
%WHS5600SVSI

%WHS5600BVS1



The screenshot shows the 'Function Maintenance - Function Details' screen. The 'Utilities' tab is selected. A table lists several functions, including %WHS5600SVSI and %WHS5600BVS1, which are the two standard functions being added.

Now refresh the Desktop.



The screenshot shows the COINS desktop interface. The 'Utilities' section is open, displaying three tiles:

- Report Status Workbench**: Shows system statistics: Waiting: 0, Generating: 0, Unread: 0.
- User Details**: Displays the user information for 'User ID: nigion' and 'Name: Nigel Longley'.
- Current Development Name**: Shows the development name 'House Sales'.



On first run, the current development Tile will be blank as it does not know what information we want. This is not good practice as in a more complex desktop, tiles that cannot retrieve their information may cause the desktop to fail, so your desktop would normally be configured to populate with something on first load

Click the Select Development Tile, this runs a function that allows a development to be selected, runs the calculation program and rebuilds the desktop. Choose a development (this guide assumes you are using a COINS training environment configured for House Sales)

The Info Tile should now change to reflect the chosen development. In the URL note that the pv Values are shown. Any Info Tile can use those values to reflect information relating to the selected Development. These pv Values will be retained in the URL until another development is selected.



3.4 Desktop Tile Values

Certain Tile Values are hard-coded against a function - such as the ones we selected for our first Tile. Typically these will be for values that are standard for any COINS user, such as the number of report generating, the number of Uncommitted Orders, the number of users on the system etc.

COINS functions that have been set up to display tile values have an information code specified in the function parameters (for example: &info=cosval.reports on the Report Status Workbench function).

	Function ▾	Description ▾	Type	Module ▾	Category
	%WWSYBORG	Report Status	Container	SY-System	WBN-Workbench
Context: Report Status Program: wou005 Parameters: Info=cosval.reports Notes: Parent: Access Type: Role Type: SYS-System					

For specific user requirements, it is possible to define other values which can be used on tiles.

You can either set up additional calculations that are available to any function that uses the same information code, or set up calculations that will be available on any tile by leaving the information code blank.



Before writing a new Desktop Tile Value to address a specific client's needs, consideration should be given as to whether Development should hard code the calculation against the function or if a bespoke Tile value is needed.

There may be performance implications - see notes (in red) below

3.4.1 Examples of Desktop Tile Values

Desktop Tile Values allows you to set up calculations to show information on tiles on the COINS Desktop.

Accessing a Table Field

System - Desktop Tile Values				20 - Housebuilders QA - Nigel Longley	COINS
	Info Code ▾	Info ID	Description		
	FUNCTION %WHSBVWB	%AVAIL	Available Plots		
Configuration Tables: Calculation: <pre>1 iAvailPlots = vp_site.RO_hds_avail;</pre>					

Accessing a Data Mart Value



System - Desktop Tile Values				20 - Housebuilders QA - Nigel Longley	
	Info Code ▲	Info ID	Description		
	FUNCTION.%WHS5604BVWB	%COMPTODAY	Exchanges Expected Today		
Configuration Tables: co_config					
Calculation:					
<pre> 1 \$param = '%WHS5604BVWB LAST KCOJOBETD diva_count ' + vp_site.kco + ' ' + vp_site.job_num + 2 iCompdue = method('coc-rsp.getR0_coc_dmvalue',param); </pre>					

Using a Method to Write a Query to Retrieve Information

System - Desktop Tile Values				20 - Housebuilders QA - Nigel Longley	
	Info Code ▲	Info ID	Description		
		PO_OSORDERS	Open Orders		
Configuration Tables: co_config					
Calculation:					
<pre> 1 method('syuquery.aggregate','for each po_hdr where kco = "' + co_config.kco + " and po_hdr.item_type='MATERIAL' and po_hdr.poh_accno = "' + co_config.R0_diaryUser + "' AND poh_chno = '" + kco') 2 method('syuquery.getDagg',1); 3 method('syuquery.getIagg'); </pre>					

Field	Description
Info Code	<p>The information code to identify which function(s) to make this value available on.</p> <p>COINS functions that have been set up to display tile values already have an information code specified in the function parameters (for example: &info= cosval.batches). If a user-defined value has an information code that matches the code on the function (for example, cosval.batches), the user-defined value will be available for users to show on the tile, in addition to the standard values.</p> <p>The information code can also be FUNCTION.<MainArea> where FUNCTION indicates that this value is for a single function only and <MainArea> is the function code.</p> <p>If the information code is blank, the user-defined value will be available for users to show on any tile.</p>
Info ID	The information value ID. The combination of info code and info ID must be unique.
Description	A description of the user variable value. This is used as the label for this value on the tile.
Configuration Tables	<p>The configuration tables that should be positioned before the calculation is evaluated.</p> <p>co_config, sysparam and menuparm are available by default, pointing at the correct records.</p>



Calculation	<p>The calculation to evaluate for the value shown on the tile.</p> <p>The calculation can be any standard OA Designer Calculation syntax as in the examples above.</p> <p>Note that the values for all tiles will be updated with a frequency that is determined by the SY/USERINFO parameter. Therefore, although the calculations can be quite complex (using Method Queries for example), if they take too long to run this could have an adverse effect on performance.</p>
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



3.5 Tile Options

Desktop Tile Options allows you to set up alternative behaviour for the links on COINS Desktop tiles (for example, to open a specific tab or a named filter). These are then available as options that the user can select when configuring the tile.

In effect, this allows you to have the same function, multiple times, on a Desktop with each tile displaying different information.

Function	Option	Description
%WHSB/WB	%AVAIL	Available Plots
%WPO101ABPHL	%UNCOMMITTED	Uncommitted Orders
%WPO101BBPHL	%UNCOMMITTED	Uncommitted Orders
%WSYBCRR	%WSY2110BCRR	Scheduler Tab
%WSYBCRR	%WSY2130BCRR	Report Status Tab

Function	Option	Description
%WHSB/WB	%AVAIL	Available Plots
%WPO101ABPHL	%UNCOMMITTED	Uncommitted Orders
%WPO101ATPHL		
%WPO101ATPHL		
%WPO101BBPHL	%UNCOMMITTED	Uncommitted Orders
%WSYBCRR	%WSY2110BCRR	Scheduler Tab
%WSYBCRR	%WSY2130BCRR	Report Status Tab

Field	Description
Function	The function to which the tile option should be made available.
Option	The ID of the tile option.
Description	A description of the tile option. This is used as the label for this option in the Options drop-down list on the tile maintenance screen.



Parameters	<p>The parameters that should be passed to the link when the tile is selected.</p> <p>The parameters are a string of values that are set in the URL when the tile is selected. This may typically be :</p> <ul style="list-style-type: none"> initTab to set a tab, initContainer to set a container tab or NamedFilter to set the named filter on that function to be used (if available). This is a comma separated list of the values to be used. queryFilter to determine the filter type to use (simple or advanced) queryColumnFilter to set the column that you are filtering on <p>The parameters may be used either alone or in combination with ‘&’ as the separator (no spaces).</p> <p>E.g. &queryFilterType=simple&NamedFilter=UnCommittedOrdersOnly ,MyOrders</p>
Security Function	<p>An optional security function associated with the option.</p> <p>Typically this would be used for tabs where the user will only be able to access a tab if the function is available to them and the tile value should therefore also only be available based on this function. If access is not available to the user then the option will be suppressed.</p> <p>If blank then the option will be available to all users.</p>
Build Condition	An optional build condition that determines whether the option is available.



3.5.1 NamedFilter Example

The NamedFilter parameter allows a Tile Option to access any existing Named Filters on a function and to pre-populate (and apply) them with values before building the browse.

NamedFilter affects the URL, so must always be preceded by "&" when used so that it correctly becomes a parameter in the URL.

Desktop Tile Options

%WPO101ATPHL is the Function for the Material Orders Workbench. This function has a two sets of pre-defined Named Filters for the Types of Order (Committed, Uncommitted etc.) and The Buyers (All or My Orders)

Page %WPO101ATPHL					Function
Header Tables: po_config, po_config Body Tables: po_hdr Description: Notes: Material Orders (Live) Column Fields: 2 stdlExport - from v11.03 the system uses the last entry of duplicate fields instead of the first one as it is in v11.02 and earlier versions.					
Header	Body	Script	Forms	Fields	
				Named Filters	
<input type="checkbox"/>		All Orders		po_hdrtive	
<input type="checkbox"/>		Exclude Cancelled Orders		po_hdr	
<input type="checkbox"/>		Incomplete Orders & VO's Only		po_hdrtive	
<input type="checkbox"/>		Incomplete Original Orders Only		po_hdrtive	
<input type="checkbox"/>		Un-Released Orders Only		po_hdr	po-fsp.UsingPOWorkflow
<input type="checkbox"/>		Released Orders Only		po_hdr	po-fsp.UsingPOWorkflow
<input type="checkbox"/>		Committed Orders Only		po_hdrtive	
<input type="checkbox"/>		Uncommitted Orders Only		po_hdrtive	
<input type="checkbox"/>		Uncommitted Orders & VOS Only		po_hdrtive	
<input type="checkbox"/>		Foreign Currency Orders Only		po_hdrtive	
<input type="checkbox"/>		Cancelled Orders Only		po_hdr	
<input type="checkbox"/>		Head Office Orders Only		po_hdrtive	
<input type="checkbox"/>		Site Orders Only		po_hdrtive	
<input type="checkbox"/>		Orders Requiring Call-Offs		po_hdrtive	po-fsp.buildSysCallOffs
<input type="checkbox"/>		Reserved Orders Only		po_hdrtive	
<input type="checkbox"/>		All Templates		po_hdrtive	
<input type="checkbox"/>		Unassigned Templates		po_hdrtive	
<input type="checkbox"/>		Replan (PO Variation) Orders		po_hdr	
<input type="checkbox"/>		All Buyers		po_hdrtive	
<input type="checkbox"/>		My Orders		po_hdrtive	<input checked="" type="checkbox"/>

On my desktop I want to have three tiles, one for All Orders, one for My Committed Orders and one for my uncommitted orders. Running the function alone should give me All orders by default (subject to my filter recall settings). So I only need to set up two Tile Value Options against %WPO101ATPHL.

System Desktop Tile Options				10 - Training Contractors QA nglon
	Function	Option	Description	
<input checked="" type="checkbox"/>	%WPO101ATPHL	Committed_Mine	My Committed Orders	
<input checked="" type="checkbox"/>	%WPO101ATPHL	UnCommitted_Mine	My UnCommitted Orders	
<input checked="" type="checkbox"/>	%WPO101ATPHL	UnCommitted_Mine	My UnCommitted Orders	

For the first, the parameter will be:

&NamedFilter=CommittedOrdersOnly,MyOrders

This sets the First Named Filter to Committed Orders Only and the second filter to My Orders.

3 Tiles

The second function entry has a similar parameter, but sets the first filter to UnCommitted Orders.

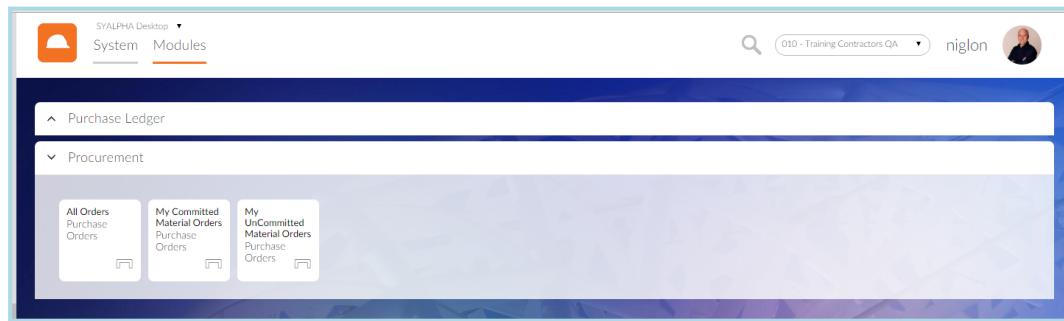
Desktop

On our Example Desktop %WPO101ATPHL has three separate Tiles defined.

- For the first entry, the Title is set to All Orders and the Option is left as default.
- For the second Entry the Title is set to My Committed Orders and the Option is set to My Committed Orders.
- For the third Entry the Title is My Uncommitted Orders, the Option is set to My Uncommitted orders

System Function Maintenance - Function Details											
Function: SYALPHAT2S2 Procurement											
Main	Menu Items	Called From	Function Security								
				Title	Override	Subtitle	Size	Open In	Type	Module	Category
				All Orders			Small	Frame	Tab	PO-Purchase Orders	Workbench
				My Committed Material Orders			Small	Frame	Tab	PO-Purchase Orders	Workbench
				My UnCommitted Material Orders			Small	Frame	Tab	PO-Purchase Orders	Workbench

The refreshed Desktop now looks like this:



Clicking the first tile we get all orders and all buyers



All Orders												
<input type="checkbox"/> Material Orders <input type="checkbox"/> By Contract												
	PO Number ▲	Chg No	Supplier ▲	Account ▲	Original Order Date ▲	Order Date	Contract ▲	Status	Com	Buyer	Type	Amount Cur
<input type="checkbox"/>	9777		Prater Roofing	PRA001	04/05/10	04/05/10	FM57	COM	<input type="checkbox"/>	000500	N	9,546.00 EUR
<input type="checkbox"/>	6668		Edmundson Electrical Ltd	EDM001	04/05/10	04/05/10	FM57	COM	<input type="checkbox"/>	000500	N	0.00 GBP
<input type="checkbox"/>	6688		Prater Roofing	PRA001	04/05/10	04/05/10	FM57	COM	<input type="checkbox"/>	000500	N	9,546.00 EUR
<input type="checkbox"/>	Cement		Edmundson Electrical Ltd	EDM001	14/04/10	14/04/10	FM57		<input type="checkbox"/>	ewahug	T	0.00 GBP
<input type="checkbox"/>	cement_2		Edmundson Electrical Ltd	EDM001	16/04/10	16/04/10			<input type="checkbox"/>	ewahug	T	0.00 GBP
<input type="checkbox"/>	HM-10000/0001		Fast Delivery Specialist	FDS001	20/05/11	20/05/11	10000		<input type="checkbox"/>	harmas	N	28,043.00 GBP

The second tile will automatically give us Committed orders, my orders

My Committed Material Orders												
<input type="checkbox"/> Material Orders <input type="checkbox"/> By Contract												
	PO Number ▲	Chg No	Supplier ▲	Account ▲	Original Order Date ▲	Order Date	Contract ▲	Status	Com	Buyer	Type	Amount Cur
<input type="checkbox"/>	HM-10000/0004		Abbey Glass	ABB001	30/09/11	30/09/11	10000	COM	<input type="checkbox"/>	nigton	N	1,000.00 GBP
<input type="checkbox"/>	HM-10000/0005		Abbey Glass	ABB001	30/09/11	30/09/11	10000	COM	<input type="checkbox"/>	nigton	N	1,000.00 GBP
<input type="checkbox"/>	HM-10000/0007		Abbey Glass	ABB001	01/01/11	01/01/11	10000	COM	<input type="checkbox"/>	nigton	N	1,500.00 GBP
<input type="checkbox"/>	HM-10000/0011		Jewson Limited	JEW001	24/06/15	24/06/15	10000	COM	<input type="checkbox"/>	nigton	N	1,250.00 GBP
<input type="checkbox"/>	HM-10000/0012		Abbey Glass	ABB001	06/07/15	06/07/15	10000	COM	<input type="checkbox"/>	nigton	N	1,500.00 GBP
<input type="checkbox"/>	HM-21100/0002		AG Lawrence (Plastering) Ltd	AGLA005	16/05/12	16/05/12	21100	COM	<input type="checkbox"/>	nigton	N	405.00 GBP

The final tile will automatically display Uncommitted Orders, My Orders

My UnCommitted Material Orders												
<input type="checkbox"/> Material Orders <input type="checkbox"/> By Contract												
	PO Number ▲	Chg No	Supplier ▲	Account ▲	Original Order Date ▲	Order Date	Contract ▲	Status	Com	Buyer	Type	Amount Cur
<input type="checkbox"/>	HM-10000/0008		Abbey Glass	ABB001	01/01/11	01/01/11	10000		<input type="checkbox"/>	nigton	N	1,000.00 GBP
<input type="checkbox"/>	HM-10000/0009		Abbey Glass	ABB001	01/01/11	01/01/11	10000		<input type="checkbox"/>	nigton	N	30,000.00 GBP
<input type="checkbox"/>	HM-10000/0010		Jewson Limited	JEW001	01/06/15	01/06/15	10000	REL	<input type="checkbox"/>	nigton	N	1,250.00 GBP



3.5.2 InitContainer Example

On Container Functions with multiple tabs (For example the Report Status Workbench) the parameter InitContainer allows the tile to preselect the tab to display when the tile is clicked.

Desktop Tile Options

The Activity Workbench (%WHS1150SHSAI) is a Container Function

Filter	Function ▲	Description ▾	Type	Module ▾	Category	Program	Access Type
<input type="checkbox"/>	  %WHS1128CHST	Journal Lookup	Container	HS-House Sales	ENQ-Enquiry	wou005	Read Only
<input checked="" type="checkbox"/>	  %WHS1150SHSA	Activity Workbench	Container	SY-System	WBN-Workbench	wou005.p	Read Only

It has the following tabs:

System Function Maintenance - Function Details						20 - Housebuilders QA nigion	 COINS
Function: %WHS1150SHSA			Activity Workbench			Lint	Options
Main	Menu Items	Called From	Function Security				
	Function	Description	Context	Type	Module	Category	
<input type="checkbox"/>	 %WHS1153BHSA	Todays Actions		Function	SY-System	Workbench	
<input type="checkbox"/>	 %WHS1154BHSA	My New Actions		Function	SY-System	Workbench	
<input type="checkbox"/>	 %WHS1151BHSA	Appointments		Function	SY-System	Workbench	
<input type="checkbox"/>	 %WHS1152BHSA	Task		Function	SY-System	Workbench	
<input type="checkbox"/>	 %WHS1150BHSA	All Actions		Function	SY-System	Workbench	
<input type="checkbox"/>	 %WHS1160BHSA	My (Diary)		Function	SY-System	Maintenance	
<input type="checkbox"/>	 %WHS1161BHSA	Group View		Function	SY-System	Maintenance	
<input type="checkbox"/>	 %WHS1150BSUR	Another User View		Function	SY-System	Maintenance	

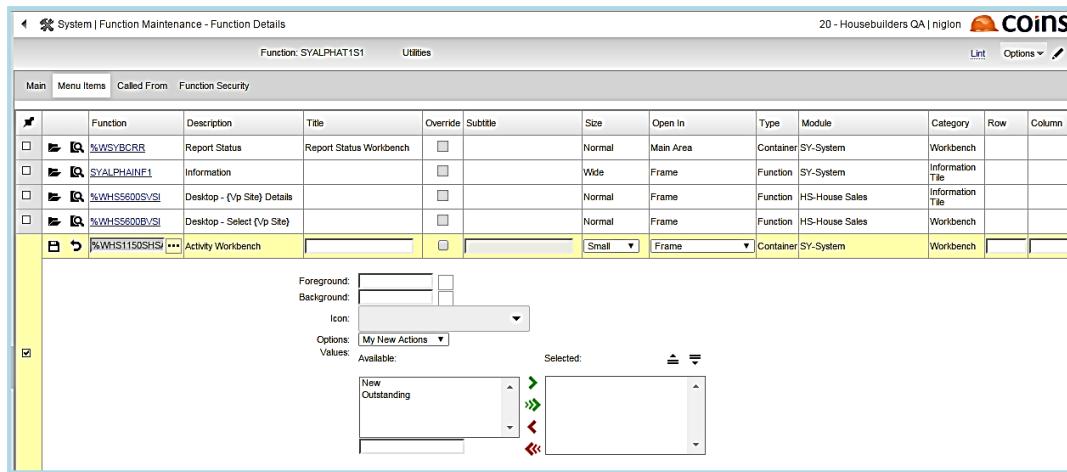
To create a Tile that will take us directly to the My New Actions Tab we can set up a Desktop Tile Value as below:

System Desktop Tile Options				20 - Housebuilders QA nigon	 COINS
	Function ▲	Option	Description		
<input checked="" type="checkbox"/>	%WHS1150SHSA	%WHS1154BHS/	My New Actions		
	Parameters:	initContainer=%WHS1154BHSA			
	Security Function:	%WHS1150SHSA	<input type="button" value="..."/>		
	Build Condition:				

Desktop

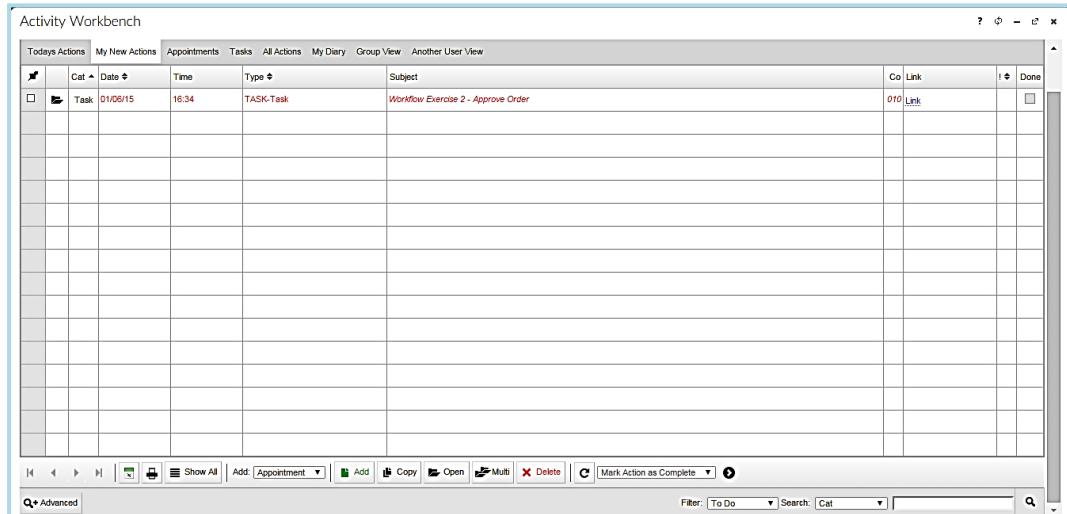
If we add the Activity Workbench function to our desktop, the Option field will now display the options of Default (use the standard recall setting on my user preferences) or My New Actions.

3 Tiles



Select My New Actions, save and then refresh the desktop.

Now when the Activity Workbench Tile is clicked, it will always open on the My New Actions Tab





3.6 Tables on Tiles

Tables are scrollable tables of information that can be used anywhere, but here we concentrate on adding them to a Tile.

In order to put a table on a tile, you need to build a browse with fields on it. This can be directly against a database table, if you already have data or you can build a dataset to create the data required before the chart is produced.

Tables offer the benefit of a pre-built search field, allowing any information in the tile to be located, and (since the data is sent client side) the ability to include RO_Fields on the table which will be sortable - something that previously required a dataset.



3.6.1 Tables : Example 1 - Direct Query against the database

In this example, we will use the table ap_vendbal to display a table of supplier period balances

Create the Function

The function is a standard function with the exception that the Category must be TBL - Table. Be careful not to select Category TAB - DB Table by mistake

	Function ▾	Description ▾	Type	Module ▾	Category	Program	Access Type
	SYALPHAAPAY	Supplier Balances	Function	SY-System	TBL-Table	wou005	Read Only

Context: Supplier Balances
Program: wou005
Parameters:
Notes:
Parent: SYALPHAAPAY
Access Type: Read Only
Role Type:

Create the Page

The Page should use the table ap_vendbal as its body table and with an appropriate body query to retrieve the records to be displayed. For Table Tiles DO NOT specify a number of rows.

When creating a table tile, give consideration to the number of rows that will be returned. Since the data is all sent client side there will be a performance issue if a tile returns too much information across multiple desktops. A maximum of 1000 records should be considered best practice.

The screenshot shows the 'Page Designer' interface for creating a page named 'SYALPHAAPAY'. In the 'Header Tables' section, 'ap_vendbal' is selected as the body table. The 'Body' tab is active, showing the following configuration:

- Body Program:** ap_vendbal
- Body Rows:** 1 FOR EACH ap_vendbal WHERE kco = {kco} AND avb_year = 2015,
2 EACH ap_vendor OF ap_vendbal
- Query Condition:** (empty)
- Condition:** (empty)
- Ignore Security:** (checkbox)

On our body fields, we need to specify just the fields we are displaying.

For each value, you must specify an ID (this can be any unique reference for each value)



Page Designer

Page SYALPHAAPAY

Field	Label	Width	Height	Function	Add	Upd	View As	Tab	Layout	Append	Hidden
avm_num	A/C No	5	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
avm_name	Supplier	5	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
avb_pbal_1	January	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
avb_pbal_2	February	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
avb_pbal_3	March	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
avb_pbal_4	April	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
avb_pbal_5	May	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
ap_vndbal.R0_avb_totalB	Balance YTD	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>

Form: Body View: Grouped

Desktop

Adding the Function to a desktop will produce the following output (Tile has been set to Full as its size):

SYALPHA Desktop

System Modules

PINN 15,281.89
PINV 265,892.42

Showing 1 to 3 of 3 entries

Supplier Balances System

A/C No	Supplier	January	February	March	April	May	Balance YTD
AB8001	Abbey Glass	4,222.89	4,222.89	4,222.89	4,222.89	4,222.89	4,222.89
BET001	Better Bricks	43,183.80	43,183.80	43,183.80	43,183.80	43,183.80	43,183.80
BRK005	W A Brick & Stone Ltd	2,400.00	2,400.00	2,400.00	2,400.00	2,400.00	2,400.00
CRY001	Crystal Cleaning	270.00	270.00	270.00	270.00	270.00	270.00
JEW001	Jewson Limited	14,359.00	14,359.00	14,359.00	14,359.00	14,359.00	14,359.00

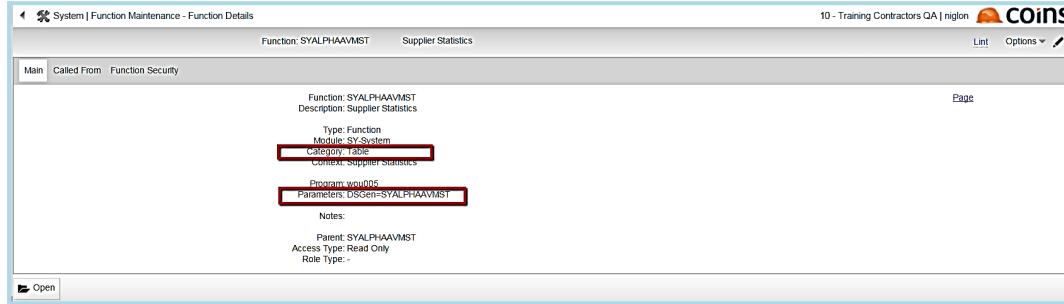
Showing 1 to 5 of 5 entries

Procurement

3.6.2 Example 2 - Using a Data Set

In this (and the next example - charts - we will use a dataset to build our data)

Create the Function

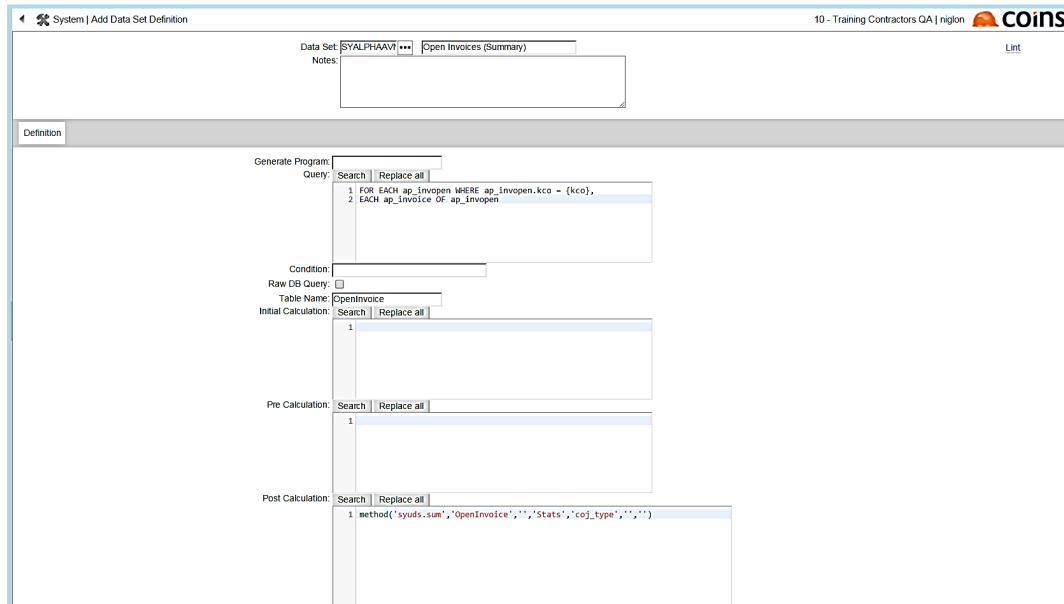


The function is a standard function with the exception that the Category must be TBL - Table. Be careful not to select Category TAB - DB Table by mistake

In the parameters field the parameter DSgen= specifies the name of the dataset to be built before the chart is generated.

Create the Dataset

Our example dataset will go through all the open invoices for the currently selected company and will then create a summary of the data to be displayed.



In this example, the query will return a large number of invoices, which we could display, but in this instance the Post Calculation is specified to use the Method syuds.sum to aggregate all the returned records by Transaction type so that we have one total value for Gross for each transaction type.

Syuds.sum requires the current temp table (OpenInvoice), a WHERE condition (if required), the new temp table to be created (Stats), the field(s) to aggregate the data by (coj_type). The remaining to parameter are left blank in this example.

Syuds. Calculation methods are covered in detail in the Guide LMDSY0036 - How to Create and Use Datasets available from the Learning Resources section of the COINS Client Area on our website (www.coins-global.com) under Business Intelligence and Analytics/OA Designer.

The data set has the following fields defined against it.

System | Data Set Summary

Data Set: SYALPHAAVMST Open Invoices (Summary)
Notes:

10 - Training Contractors QA | nigion COINS Lint

Definition		Fields					
	Field	Key	Extent	Label	Data Type	Format	Source
<input type="checkbox"/>	avm_num	<input type="checkbox"/>		Supplier	Character	x(8)	ap_invoice.avm_num
<input type="checkbox"/>	ain_inv	<input type="checkbox"/>		Invoice	Character	x(8)	ap_invoice.ain_inv
<input type="checkbox"/>	coj_type	<input type="checkbox"/>		Type	Character	x(8)	ap_invoice.coj_type
<input type="checkbox"/>	dGross	<input type="checkbox"/>		Gross	Decimal	>>>>>9.99	ap_invoice.ain_amount

Advanced

Desktop

Adding the Function to a desktop will produce the following output:



SYALPHA Desktop ▾
System Modules

 010 - Training Contractors QA ▾


3.7 Charts on Tiles

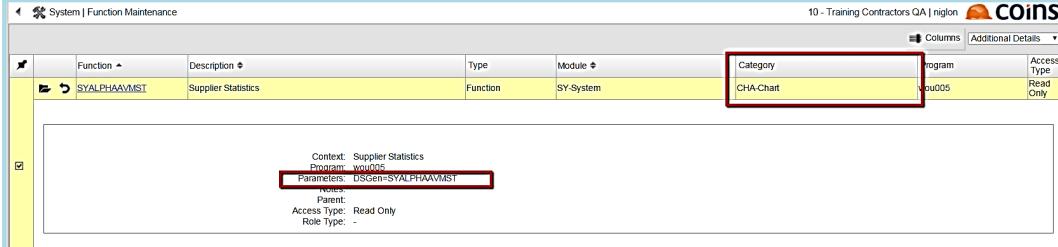
In order to put a chart on a tile, you need to have a set of records with an X coordinate value and a Y coordinate value that you intend to plot.

To achieve this, you need to build a browse with fields on it. This can be directly against a database table, if you already have data to plot or you can build a dataset to create the data required before the chart is produced.

3.7.1 Dataset driven Charts

In this example, we will create a dataset to generate Supplier Statistics add the chart to a new tile.

Create the Function



Function	Description	Type	Module	Category	Program	Access Type
SYALPHAAMST	Supplier Statistics	Function	SY-System	CHA-Chart	you005	Read Only

Context: Supplier Statistics
 Program: you005
 Parameters: DSgen=SYALPHAAMST

Notes:
 Parent:
 Access Type: Read Only
 Role Type: -

The function is a standard function with the exception that the Category must be Chart.

In the parameters field the parameter DSgen= specifies the name of the dataset to be built before the chart is generated.

Create the Dataset

Our example dataset will go through all the open invoices for the currently selected company and will then create a summary of the data to be displayed.



In this example, the query will return a large number of invoices, which we could plot, but the graph would be very confusing. So in this instance the Post Calculation is specified to use the Method syuds.sum to aggregate all the returned records by Transaction type so that we have one total value for Gross for each transaction type.

Syuds.sum requires the current temp table (OpenInvoice), a WHERE condition (if required), the new temp table to be created (Stats), the field(s) to aggregate the data by (coj_type). The remaining to parameter are left blank in this example.

Syuds. Calculation methods are covered in detail in the Guide LMDSY0036 - How to Create and Use Datasets available from the Learning Resources section of the COINS Client Area on our website (www.coins-global.com) under Business Intelligence and Analytics/OA Designer.

The data set has the following fields defined against it.

	Field	Key	Extent	Label	Data Type	Format	Source
<input type="checkbox"/>	avm_num	<input type="checkbox"/>	Supplier	Character	x(8)		ap_invoice.avm_num
<input type="checkbox"/>	ain_inv	<input type="checkbox"/>	Invoice	Character	x(8)		ap_invoice.ain_inv
<input type="checkbox"/>	coj_type	<input type="checkbox"/>	Type	Character	x(8)		ap_invoice.coj_type
<input type="checkbox"/>	dGross	<input type="checkbox"/>	Gross	Decimal	>>>,>>>,>>9.99		ap_invoice.ain_amount

Create the Page

The Page should use the temp table ttStats as its query.



Header Tables:
Body Table: ttStats
Description:
Notes:

Header Body Script Forms Fields Named Filters Browse Filters Fields Enquiry Used In

Body Program:
Body Rows:
Body Query: 1 FOR EACH ttStats

Query Condition: Ignore Security:

Sort Limit: 0
Body Title:
Body Sum:
Aligns:
Frozen Columns: 0
Dialogue Width: 0
Ajax Update:
Matrix Columns:
Fixed Columns:
Auto Add:
Single Add:
Sequenced Update:
Graph Formatting:
Mandatory Columns:
Essential Fields:

Condition:

On our body fields, we need to specify just the fields we are displaying.

For each value, you must specify an ID (this can be any unique reference for each value)

	Field	Label	Width	Height	Function	Add	Upd	View As	Tab	Layout	Append	Hidden
<input type="checkbox"/>	ttStats.coj_type	Type	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ttStats.dGross	Gross	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
<input type="button" value="H"/> <input type="button" value="I"/> <input type="button" value="J"/> <input type="button" value="K"/> <input type="button" value="L"/> Show All <input type="button" value="Σ"/> <input type="button" value="Add"/> <input type="button" value="Copy"/> <input type="button" value="Insert"/> <input type="button" value="Up"/> <input type="button" value="Down"/> <input type="button" value="Open"/> <input type="button" value="Multi"/> <input type="button" value="Bulk:"/> <input type="button" value="Delete"/> <input type="button" value="Detail"/> <input type="button" value="C"/> Transfer Fields <input type="button" value=""/>												

Create the Page

The Page should use the temp table ttStats as its query.



The screenshot shows a page configuration screen in the COINS system. The 'Body' tab is active. The main content area contains the following code:

```
1 FOR EACH ttStats
```

Below the code, there are several configuration sections:

- Query Condition:** Contains fields for Sort Limit (0), Body Title, Body Sum, Active, Frozen Columns (0), Dialogue Width (0), Ajax Update (unchecked), Matrix Columns, Fixed Columns, Auto Add (unchecked), Single Add (unchecked), Sequenced Update (unchecked), Graph Formatting (with a legend example), Mandatory Columns, and Essential Fields.
- Condition:** A dropdown menu.

The Gauge formatting is used in a chart page to set (via JavaScript) various graph axis and legend formatting.

Full details of the available parameters may be found at:

<http://c3js.org/reference.html>

The commands used in this example are:

`legend: {position:value}`

Currently values bottom, right and inset are supported.

`axis: {x: {type:value}}`

Currently values timeseries, category and indexed are supported.

`axis: {y: {label:text,position}}`

Text values will display the text as the axis label. Null will suppress any title on the specified axis

Position values (not used in our example) are:

If it's the horizontal axis:

inner-right [default], inner-center, inner-left, outer-right, outer-center, outer-left

If it's the vertical axis:

inner-top [default], inner-middle, inner-bottom, outer-top, outer-middle, outer-bottom

`axis: {y: {tick:{count:value}}}`

A value entered here will set the number of y axis ticks to show

`axis: {y: {tick:{format:d3.format.(specifier)}}}`

Returns a new format function with the given string specifier. (Equivalent to `locale.numberFormat` for the default U.S. English locale.) A format function takes a number as the only argument, and returns a string representing the formatted number. The general form of a specifier is:

`[[fill][align][sign][symbol][0][width][,][.precision][type]]`

The fill can be any character other than "{" or "}". The presence of a fill character is signaled by the character following it, which must be one of the align options.

The align can be:

- ("<") Forces the field to be left-aligned within the available space.
- (">") Forces the field to be right-aligned within the available space. (This is the default).
- ("^") Forces the field to be centered within the available space.

The sign can be:

- plus ("+") - a sign should be used for both positive and negative numbers.
- minus ("-") - a sign should be used only for negative numbers. (This is the default.)
- space (" ") - a leading space should be used on positive numbers, and a minus sign on negative numbers.

The symbol can be:

- currency ("\$") - a currency symbol should be prefixed (or suffixed) per the locale.
- base ("#") - for binary, octal, or hexadecimal output, prefix by "0b", "0o", or "0x", respectively.

The "0" option enables zero-padding.

The width defines the minimum field width. If not specified, then the width will be determined by the content.

The comma (",") option enables the use of a comma for a thousands separator.

The precision indicates how many digits should be displayed after the decimal point for a value formatted with types "f" and "%", or before and after the decimal point for a value formatted with types "g", "r" and "p".

The available type values are:

- exponent ("e") - use Number.toExponential.
- general ("g") - use Number.toPrecision.
- fixed ("f") - use Number.toFixed.
- integer ("d") - use Number.toString, but ignore any non-integer values.
- rounded ("r") - round to precision significant digits, padding with zeroes where necessary in similar fashion to fixed ("f"). If no precision is specified, falls back to general notation.
- percentage ("%") - like fixed, but multiply by 100 and suffix with "%".
- rounded percentage ("p") - like rounded, but multiply by 100 and suffix with "%".
- binary ("b") - outputs the number in base 2.



octal ("o") - outputs the number in base 8.

hexadecimal ("x") - outputs the number in base 16, using lower-case letters for the digits above 9.

hexadecimal ("X") - outputs the number in base 16, using upper-case letters for the digits above 9.

character ("c") - converts the integer to the corresponding unicode character before printing.

SI-prefix ("s") - like rounded, but with a unit suffixed such as "9.5M" for mega, or "1.00µ" for micro.

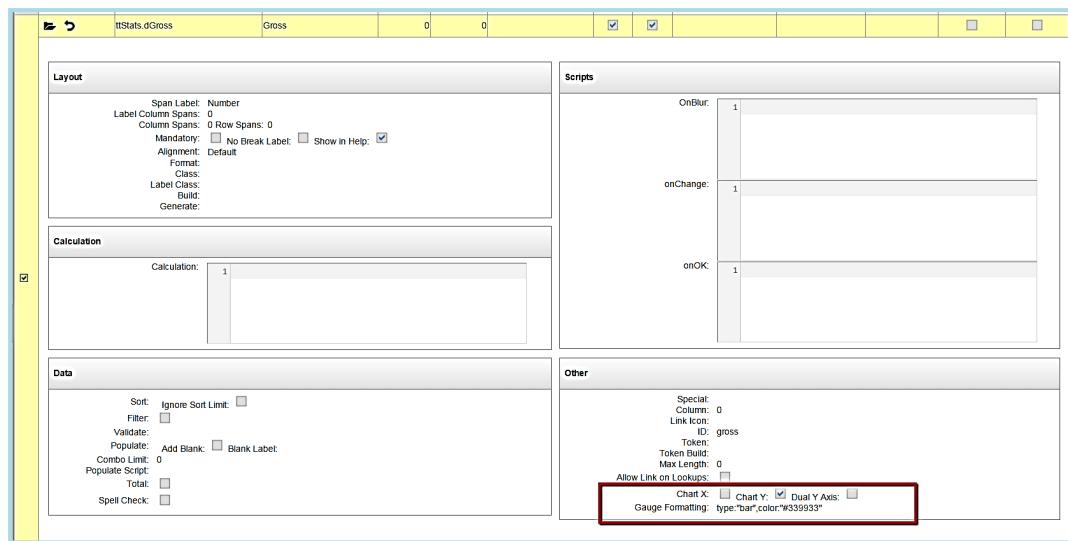
The type "n" is also supported as shorthand for ",g".

On our body fields, we need to specify just the fields we are plotting, and indicate which is the X value and which is the Y value.

For each value, also specify an ID (this can be any unique reference for each value)

	Field	Label	Width	Height	Function	Add	Upd	View As	Tab	Layout	Append	Hidden																																																																																																								
<input checked="" type="checkbox"/>	ttStats.coj_type	Type	12	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>																																																																																																								
<table border="1"> <tr> <td colspan="4">Layout</td> <td colspan="9">Scripts</td> </tr> <tr> <td colspan="4"> Span Label: Label Column Spans: 0 Column Spans: 0 Row Spans: 0 Mandatory: <input type="checkbox"/> No Break Label: <input type="checkbox"/> Show in Help: <input checked="" type="checkbox"/> Alignment: Default Format: Class: Label Class: Build: Generate: </td> <td colspan="9"> OnBlur: 1 onChange: 1 onOK: 1 </td> </tr> <tr> <td colspan="13"> Calculation </td> </tr> <tr> <td colspan="13"> Calculation: 1 </td> </tr> <tr> <td colspan="13"> Data </td> </tr> <tr> <td colspan="13"> Sort: Ignore Sort Limit: <input type="checkbox"/> Filter: <input type="checkbox"/> Value: Populate: Add Blank: <input type="checkbox"/> Blank Label: Combo Limit: 0 Populate Script: Total: <input type="checkbox"/> Spell Check: <input type="checkbox"/> </td> </tr> <tr> <td colspan="13"> Other </td> </tr> <tr> <td colspan="13"> Special Column: 0 Link Icon: 10 Token: 0 Token Build: Max Length: 0 Allow Link on Lookups: <input type="checkbox"/> Chart X: <input checked="" type="checkbox"/> Chart Y: <input type="checkbox"/> Dual Y Axis: <input type="checkbox"/> Gauge Formatting: </td> </tr> </table>													Layout				Scripts									Span Label: Label Column Spans: 0 Column Spans: 0 Row Spans: 0 Mandatory: <input type="checkbox"/> No Break Label: <input type="checkbox"/> Show in Help: <input checked="" type="checkbox"/> Alignment: Default Format: Class: Label Class: Build: Generate:				OnBlur: 1 onChange: 1 onOK: 1									Calculation													Calculation: 1													Data													Sort: Ignore Sort Limit: <input type="checkbox"/> Filter: <input type="checkbox"/> Value: Populate: Add Blank: <input type="checkbox"/> Blank Label: Combo Limit: 0 Populate Script: Total: <input type="checkbox"/> Spell Check: <input type="checkbox"/>													Other													Special Column: 0 Link Icon: 10 Token: 0 Token Build: Max Length: 0 Allow Link on Lookups: <input type="checkbox"/> Chart X: <input checked="" type="checkbox"/> Chart Y: <input type="checkbox"/> Dual Y Axis: <input type="checkbox"/> Gauge Formatting:												
Layout				Scripts																																																																																																																
Span Label: Label Column Spans: 0 Column Spans: 0 Row Spans: 0 Mandatory: <input type="checkbox"/> No Break Label: <input type="checkbox"/> Show in Help: <input checked="" type="checkbox"/> Alignment: Default Format: Class: Label Class: Build: Generate:				OnBlur: 1 onChange: 1 onOK: 1																																																																																																																
Calculation																																																																																																																				
Calculation: 1																																																																																																																				
Data																																																																																																																				
Sort: Ignore Sort Limit: <input type="checkbox"/> Filter: <input type="checkbox"/> Value: Populate: Add Blank: <input type="checkbox"/> Blank Label: Combo Limit: 0 Populate Script: Total: <input type="checkbox"/> Spell Check: <input type="checkbox"/>																																																																																																																				
Other																																																																																																																				
Special Column: 0 Link Icon: 10 Token: 0 Token Build: Max Length: 0 Allow Link on Lookups: <input type="checkbox"/> Chart X: <input checked="" type="checkbox"/> Chart Y: <input type="checkbox"/> Dual Y Axis: <input type="checkbox"/> Gauge Formatting:																																																																																																																				

3 Tiles



In the Gauge Formatting field on the Y axis value, the value type:"bar",color:"#339933" sets the graph type as a Bar Chart and assigns a colour hex code to the chart. It may be good practice to assign a colour as there is a possibility of a white tile displaying a white graph - which would appear to the user as an empty chart.

Desktop

Adding the Function to a desktop will produce the following output:

