DesignBais

Top 10 Developer Features

Version 8.5.1.1

Copyright © 2021
DesignBais International
DesignBais Pty Ltd

Table of Contents

Overview 3
Run a Form 4
Code Editor 4
Access to Code Editor from Forms Designer 6
Create New Subroutine & Insert Basic Code Segment 7
Find String 9
Find String - Conjunct Search Option 9
Find String – Replacement String Option 11
Find String - Clean Up a Basic Library or Form File 13
Compare 14
Form Compare 15
DBDS Log 16
View Como 17
Add Button 18
View Backup 21
Forms Designer – Recent button 24
Forms Designer – Form Section control 25
Forms Designer – Form Fields enquiry 26
Copy Form 28
Field Properties - Advanced Copy 29
Field Properties - Refine by Field Name 29
Dialog Box alternative 30
Menu Definitions – Finding a menu 31
Menu Definitions – Maintaining a menu called from a menu 32
Menu Definitions – Menu List lookup 33
Active Users 34
Form Help 35
Review File Pointers 36
User Maintenance 37
Checklists 39
Entity Security 45
Global Parameters 46

Overview

DesignBAIS is a toolset for building full cross-browser and Responsive Design application software. The user interface is intuitive and easy to use. Links to the underlying database are transparent such that developers do not require intimate knowledge of the particular database on which the software development is undertaken.

DesignBAIS knows about Tables or Files which have Fields. These are defined within DesignBAIS. They are then available to the developer to use in Forms, Reports and Selection Processes.

DesignBAIS also incorporates a host of other features that allow sophisticated software to be developed and deployed quickly and easily. These features include Styles (or Classes) and Style Groups, Menus, Users and User Groups, Business Rules, Glossary with multi-language support, Form and Field context sensitive help, Entity Security definitions, Field Lockdown, End of Period Processing, Full Word Indexing, Express Reporting and a comprehensive range of Upgrade and Migration tools.

This document highlights the features in DesignBAIS Release 7 and above that make application software development easier and quicker as compared to DesignBAIS Version 6. The *Top 10* part of this document's name is a marketing phrase – there are many more than 10 features that deserve highlighting.

Functions such as the Forms Designer do not feature highly in this *Top 10* document. Clearly this is not because these functions are not feature-rich. Rather it is because the standard, and excellent, DesignBAIS tools are well known to software developers who have been using DesignBAIS for many years. This document is designed to highlight some newer features that add even more ease to the development process.

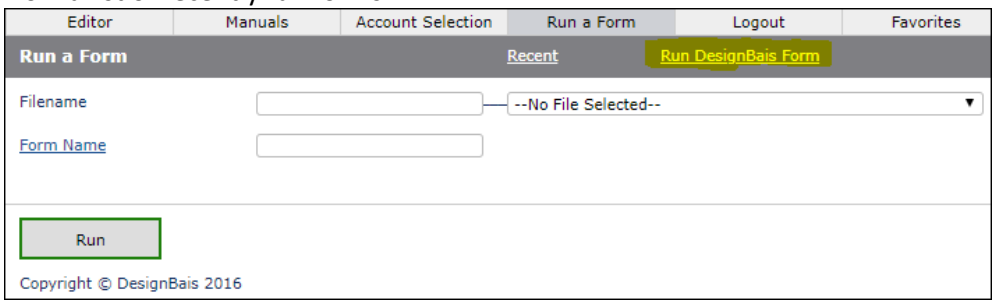
If you are new to DesignBAIS this summary should be read in conjunction with the DesignBAIS Reference Manual which provides a comprehensive reference to all the features of DesignBAIS.

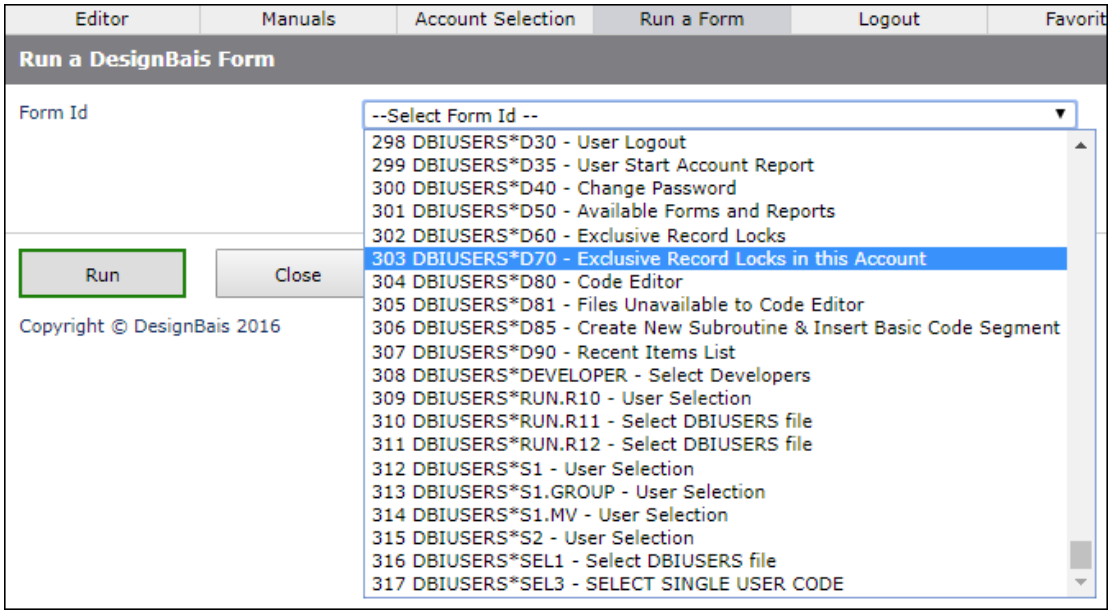
Happy Developing!

(DesignBAIS welcomes your feedback or questions. Contact us via email at *support@bais.con.au*)

Run a Form

There is a new option to *Run DesignBAIS Form*. This is useful for developers wanting to run a form located in DBISYSFORMS and which may not be on any menu. The *Recent* option allows you to select from a list of recently run forms.



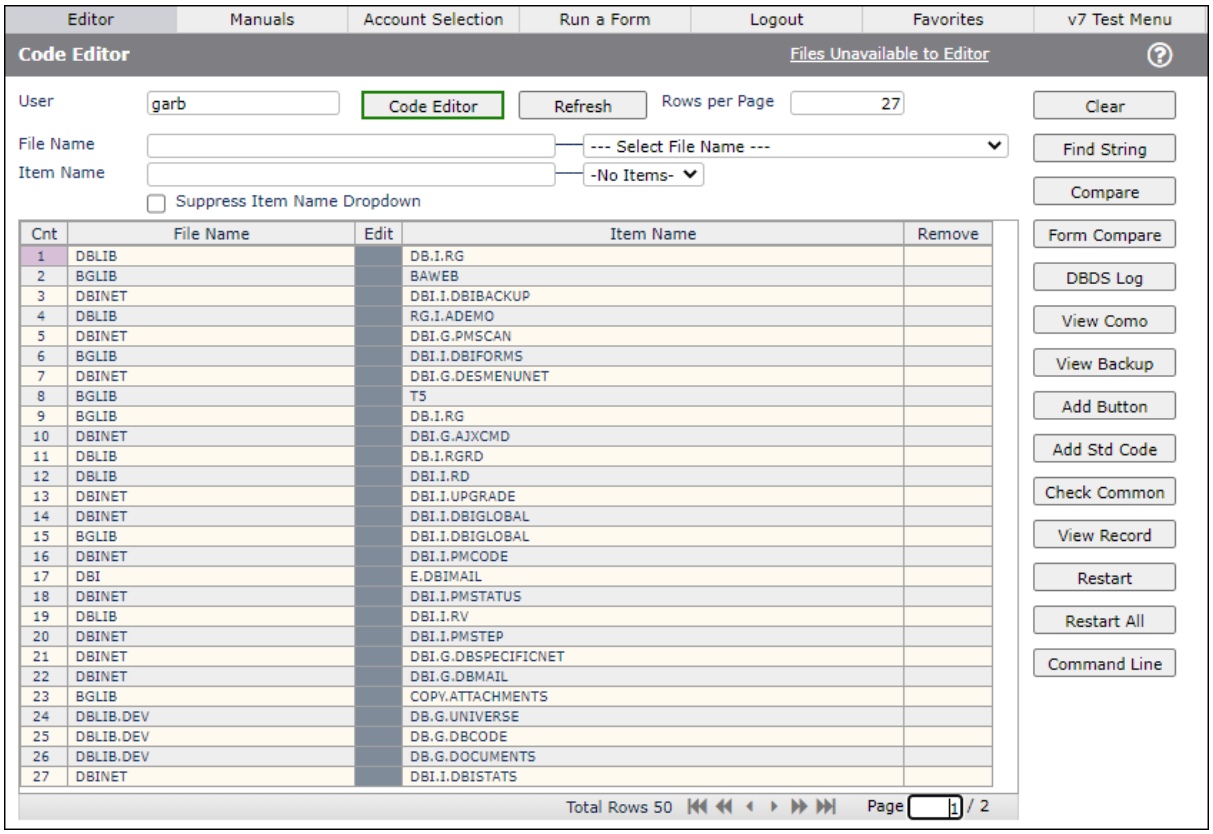


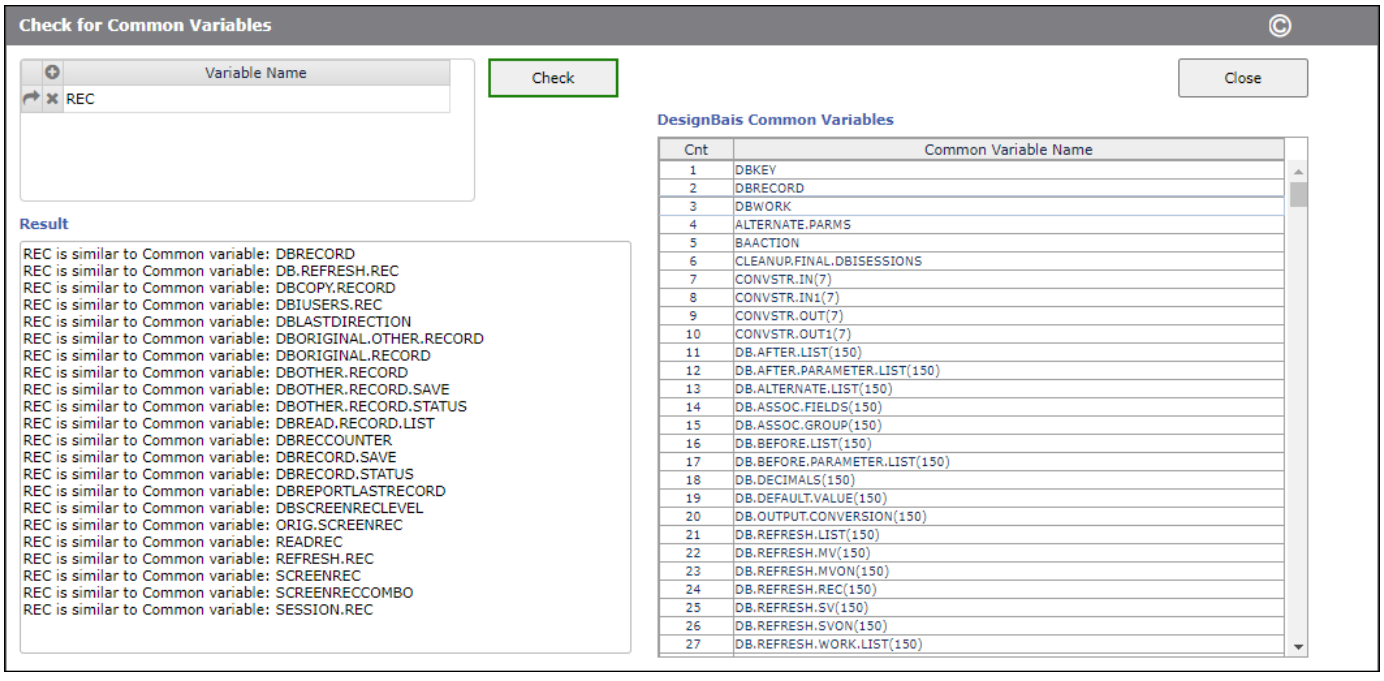
Code Editor

Click the *Editor* Button on the top menu of the Development Tools form. The *Code Editor* form provides access to a range of functions designed to make applications development quicker and easier.

- **Code Editor** - DesignBAIS retains a list of the last 50 records accessed using the Code Editor. Simply click the *Edit* column in the row displaying the required item to access recently edited items.
- **Find String** – locate any string in any file
- **Compare** – compares items in the same or different files and highlights differences
- **Form Compare** – a tool to highlight differences between two similar forms
- **DBDS Log** – retains and displays a trail of all DBDS output
- **View Como** – displays the significant content of your como (on databases that support como)
- **View Backup** – retains a copy of specified file records with the ability to compare to the current version
- **Add Button** –quickly add the *Display Common* and *Form Help* buttons to any form

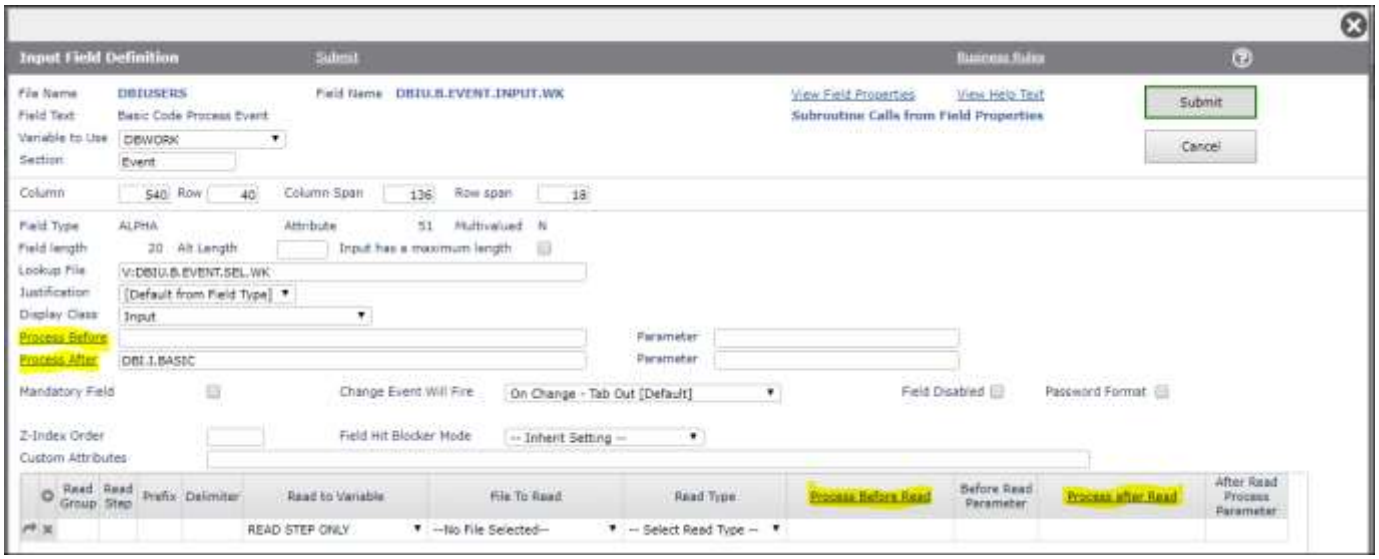
- **Subroutine** – create a new subroutine and add skeleton basic code segments to basic subroutines
- **Check Common** – check if a variable name is defined as a DesignBAIS Common variable
- **View Record** – useful to view records with associated multivalues
- **Restart / Restart All** – recycle the website when required to pick up new object code
- **Command Line** – useful to run simple TCL commands when a telnet session is unavailable



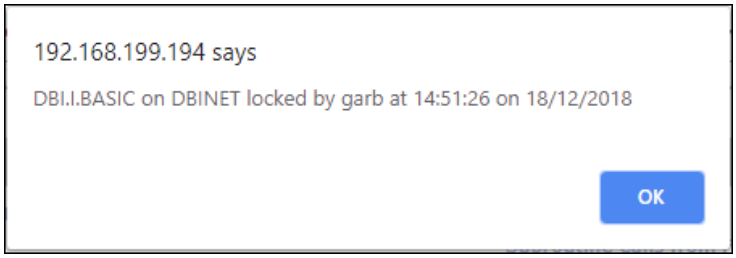


Access to Code Editor from Forms Designer

Process slots in Forms Designer have a hyperlink tag to allow direct access to the Code Editor.

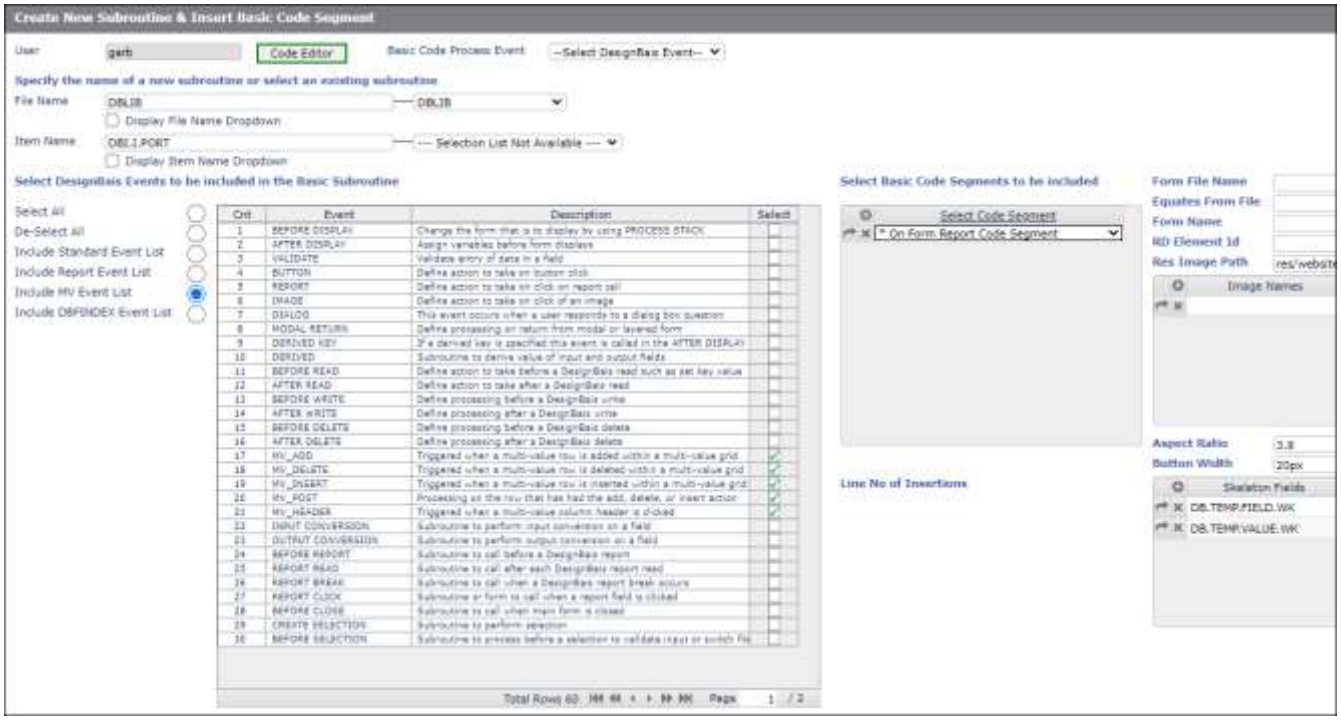


Click the *Process After* tag, see above image, to access the displayed basic subroutine in code editor. If the routine is locked then a message displays, as shown below, rather than attempting to open Code Editor.



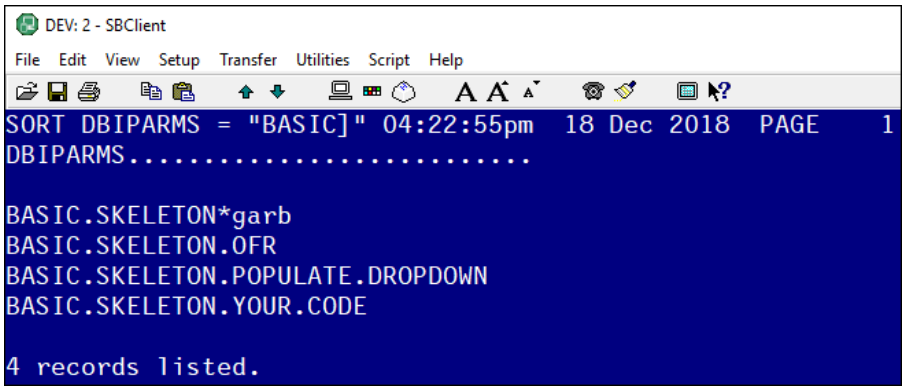
Create New Subroutine & Insert Basic Code Segment

Clicking a process link where the associated process field is null displays the following form. This allows you to create a new subroutine or to add basic code segments to an existing subroutine.

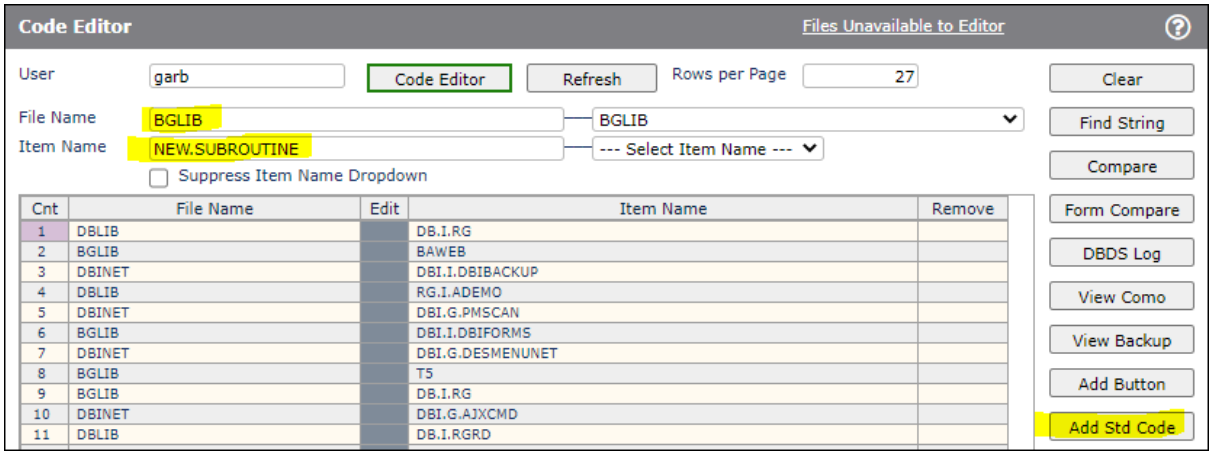


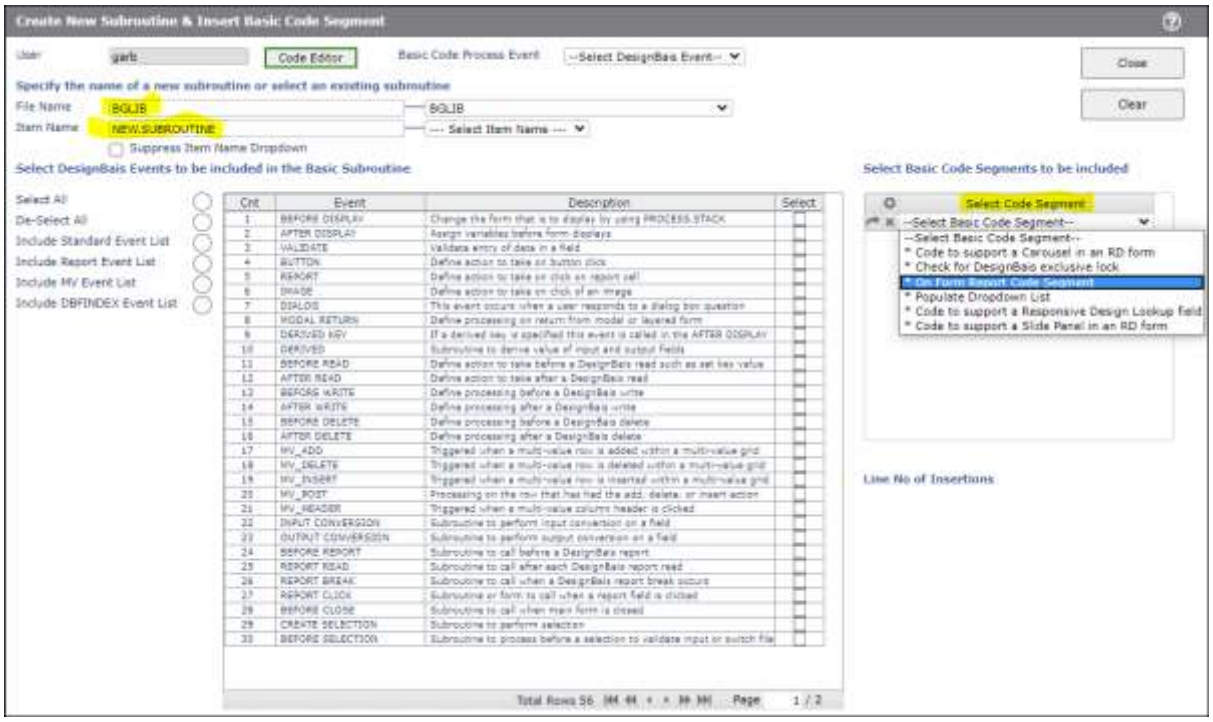
Select any number of DesignBAIS events from the list, and select from the Basic Code Segments list if required, then click the *Code Editor* Button. If the subroutine already exists then the selected code segments are added to it. If the routine is new then it is created from skeleton code and the selected code segments are added.

You can create your own basic code skeleton code. If you want a customised skeleton to be used when creating new subroutines then place your code into DBIPARMS with a key of "BASIC.SKELETON*_weblogon_" where *weblogon* is your DesignBAIS user id. If you want other often-used basic code segments then add these to DBIPARMS with a key of "BASIC.SKELETON.*anyname*". Make line 1 of these segments a meaningful comment line as this will display as the description in the dropdown list.



The Create New Subroutine & Insert Basic Code Segment form can also be launched from the Code Editor form. Enter (or select) the File and Item names then click the *Add Std Code* button.


Code Editor
Files Unavailable to Editor
User
garb
Code Editor
Refresh
Rows per Page
27
File Name
BGLIB
Item Name
NEW.SUBROUTINE
Suppress Item Name Dropdown
--- Select Item Name ---
Clear
Find String
Compare
Form Compare
DBDS Log
View Como
View Backup
Add Button
Add Std Code
Cnt
File Name
Edit
Item Name
Remove
1 DBLIB DB.I.RG
2 BGLIB BAWEB
3 DBINET DBI.I.DBIBACKUP
4 DBLIB RG.I.ADEMO
5 DBINET DBI.G.PMSCAN
6 BGLIB DBI.I.DBIFORMS
7 DBINET DBI.G.DESMENUNET
8 BGLIB T5
9 BGLIB DB.I.RG
10 DBINET DBI.G.AJXCMD
11 DBLIB DB.I.RGRD

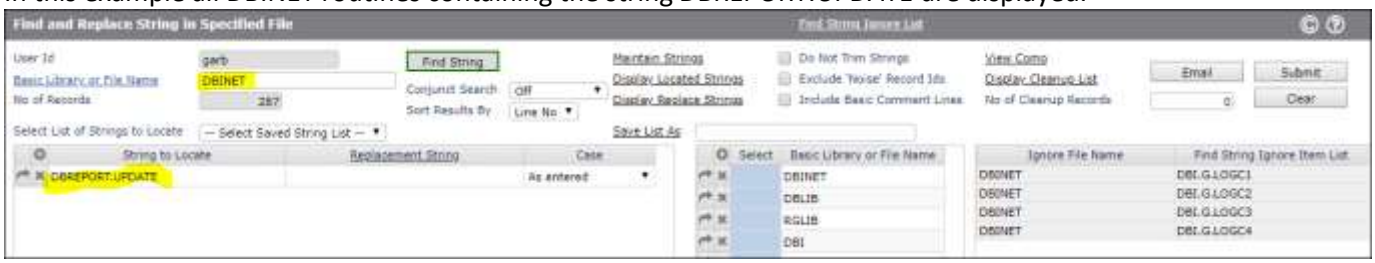

Create New Subroutine & Insert Basic Code Segment
Code Editor
Basic Code Process Event
--Select DesignBais Event--
Close
Clear
Specify the name of a new subroutine or select an existing subroutine
File Name
BGLIB
Item Name
NEW.SUBROUTINE
Suppress Item Name Dropdown
Select DesignBais Events to be included in the Basic Subroutine
Select All
De-Select All
Include Standard Event List
Include Report Event List
Include MV Event List
Include DBFINDEX Event List
Cnt
Event
Description
Select
Select Basic Code Segments to be included
Select Code Segment
--Select Basic Code Segment--
--Select Basic Code Segment--
* Code to support a Carousel in an RD form
* Check for DesignBais exclusive lock
* On Form Report Code Segment
* Populate Dropdown List
* Code to support a Responsive Design Lookup field
* Code to support a Slide Panel in an RD form
Line No of Insertions

Find String

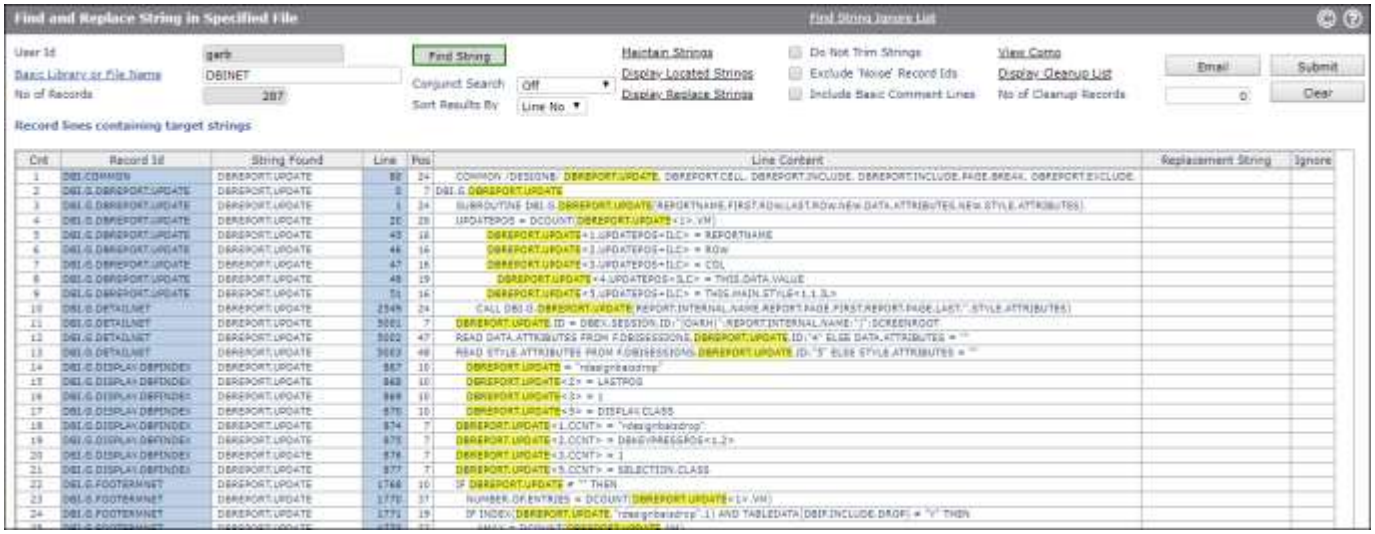
Find String provides the ability to search for multiple strings within (almost) any file. The results are displayed in an On Form report. Clicking on any line of the results takes you to the record containing the line, and the record is displayed in the appropriate maintenance form (such as DBIUSERS, DBISELECT, DBISTYLE, DBICHK) or in the code editor for records that do not relate to a specific DesignBAIS function.

If a *Replacement String* is entered then all occurrences of the target string can be replaced. See below.

In this example all DBINET routines containing the string *DBREPORT.UPDATE* are displayed.



Click the *Record Id* column to view the selected item in the Code Editor or in the DesignBAIS tool associated with the File Name being searched. If you are finding a string in DBIFORMS then clicking the *Record Id* will open Forms Designer.



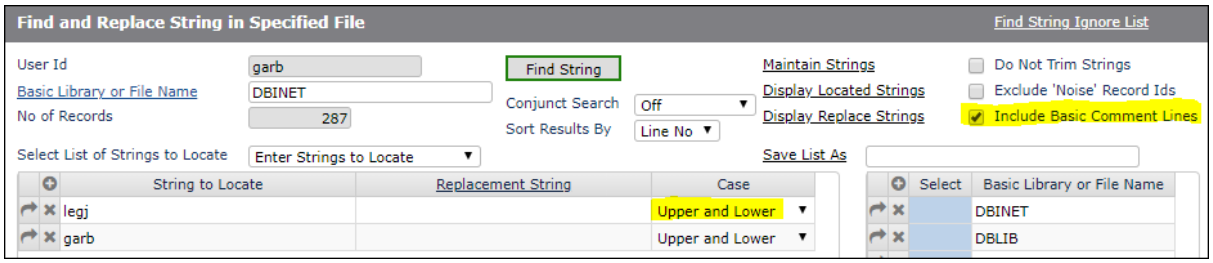
Find String - Conjunct Search Option

This allows you to find a target string within the result list from the previous find.

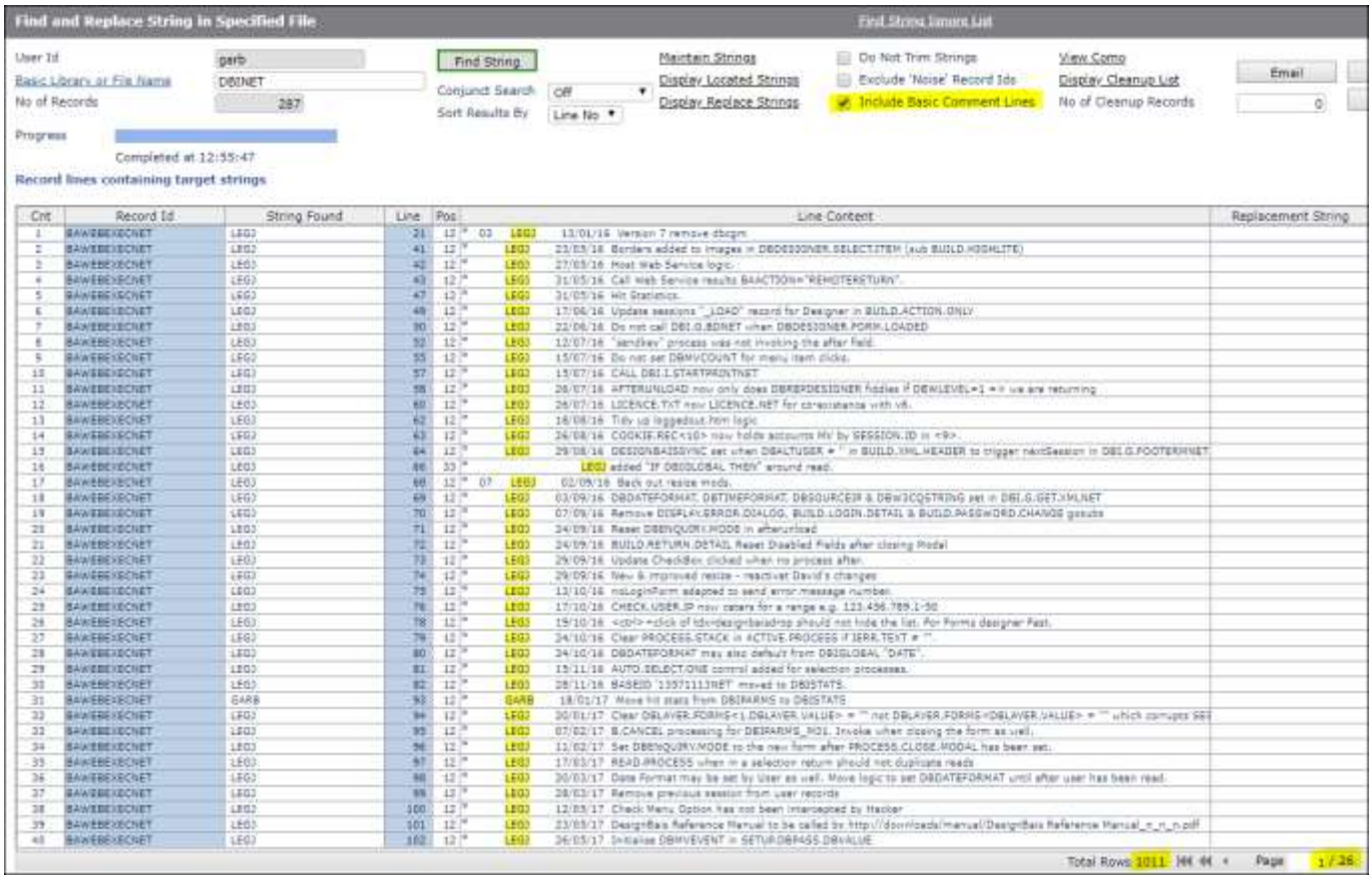
Use this feature to find a subset of items which match several criteria. For example you may want to find all changes to subroutines made by either particular developers, for example LEGJ or GARB, in 2019.

In this case the file to search is DBINET. Enter the search strings and select *Case* to be "*Upper and Lower*" to pick up both upper case and lower case versions of the search strings. Also tick the *Include*

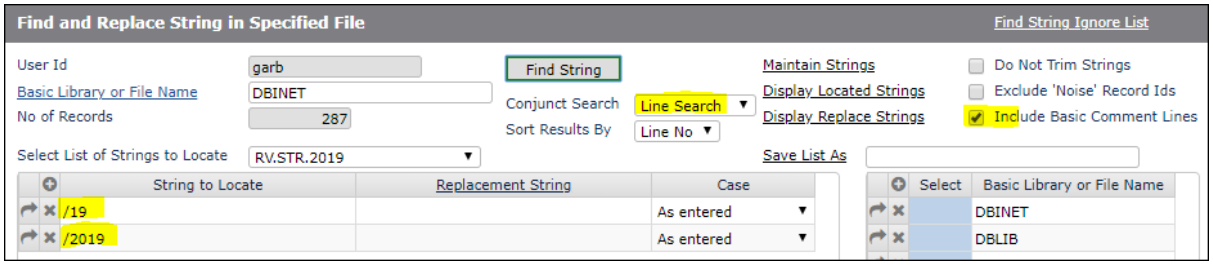
Basic Comment Lines since most occurrences of the strings you are searching for are going to be comment lines.



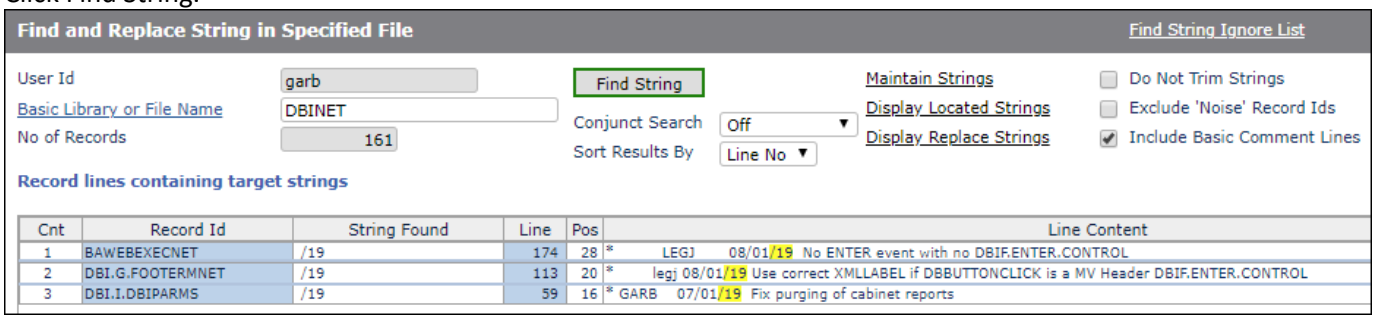
The search returns 1011 lines. This is too large a result to check manually so use the *Conjunct Search* option.



Select *Line Search* in the *Conjunct Search* dropdown. Enter the strings to search for in each line of the result set from the previous search.

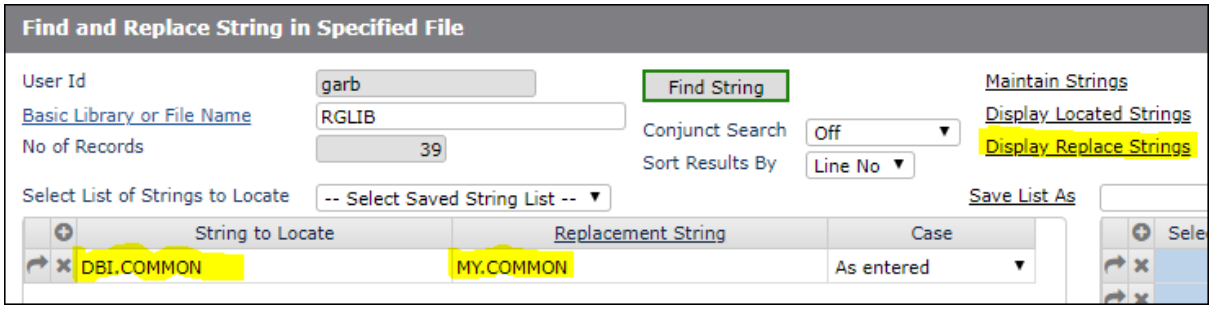


Click Find String.

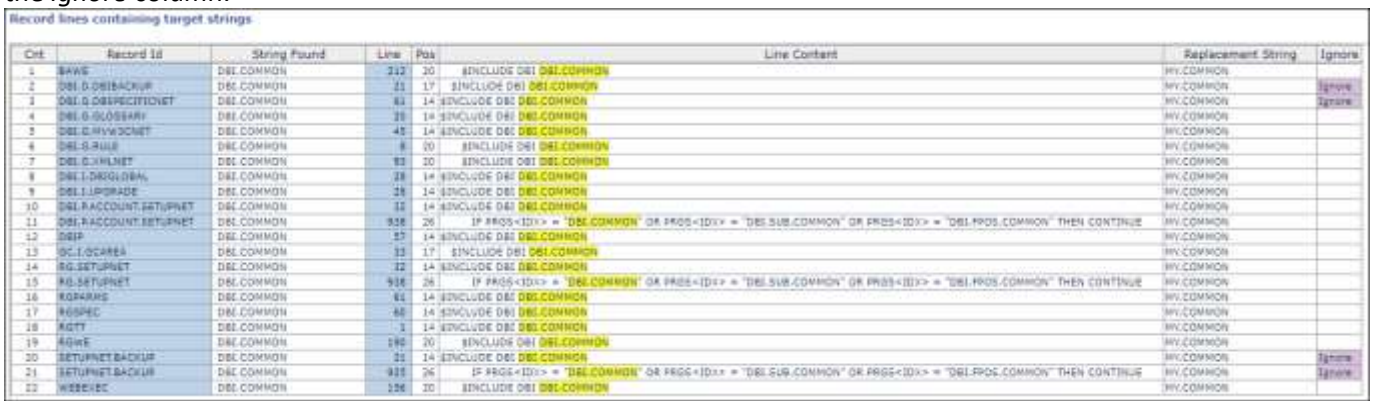


The result set is the 3 routines that have been modified in 2019.

Find String – Replacement String Option



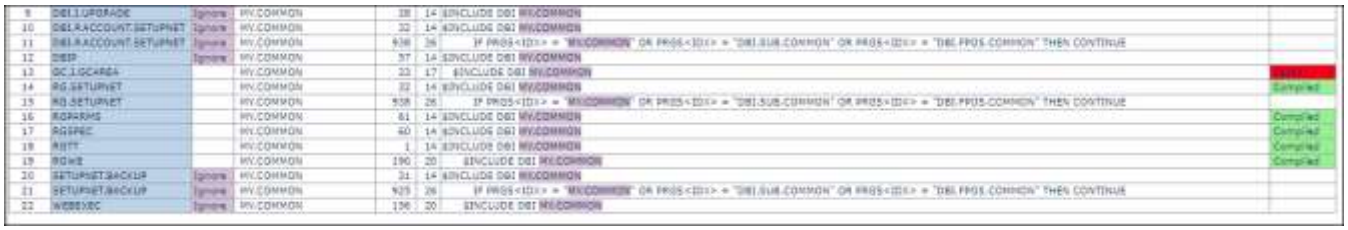
You can use the *Replacement String* option to replace a target string. In this example all occurrences of *DBI.COMMON* are to be replaced with *MY.COMMON*. Individual rows can be Ignored by clicking the *Ignore* column.



Click *Display Replace Strings* to view the target lines with the replacement string in place.

Click the *Replace* button to confirm the replacement. All records are backed up prior to being updated. You can click the *Use DBIBACKUP* checkbox in order to save the pre-change image in the DBIBACKUP file. This makes the backups immediately available in the DesignBAIS *Backup* function, which provides a roll-back feature.

If you are dealing with a program library then click *Compile* to compile all changed routines. Compilation results are displayed in the final column. If a routine fails compilation it can be edited by clicking the *Record Id* column to open the code editor. After correcting the any errors the *Compile* button can be used again to do the compiles.



Use the *View Como* option to view the results of the compilations.

View Como

My Como: garb

Como Line Count: 1 | Como.1 Line Count: 200 | Refresh | Clear Como | My Locks

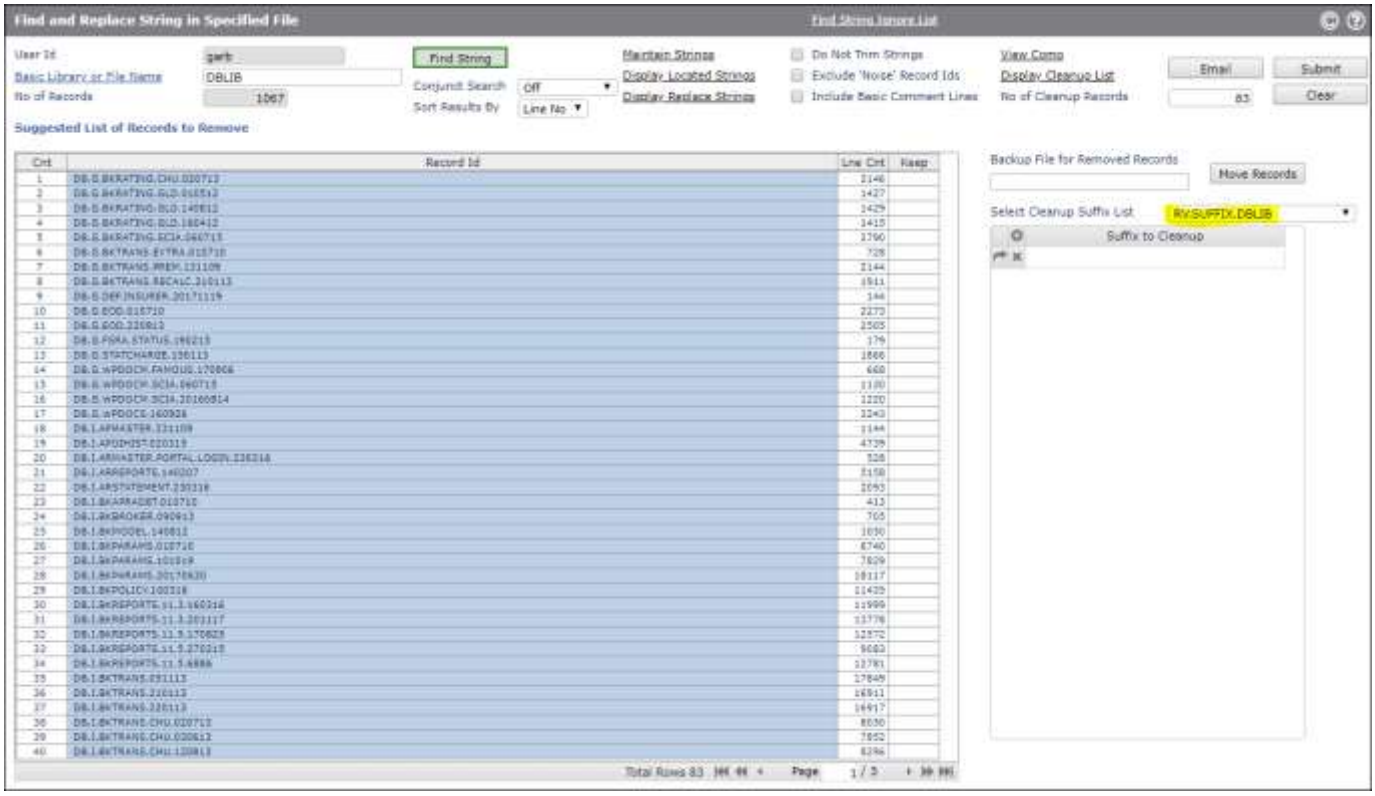
Cnt	Como Line #	Como Line Content
32	32	WARNING: Variable \'GCPE.CANC.DATE\' never assigned a value.
33	33	WARNING: Variable \'GCPE.BEST.CONTACT.NO\' never assigned a value.
34	34	WARNING: Variable \'GCPE.RETURN.FORM.RECEIVED\' never assigned a value.
35	35	WARNING: Variable \'GCPE.LICENCE.NUMBER\' never assigned a value.
36	36	WARNING: Variable \'LIC.FIRST.NAME\' never assigned a value.
37	37	WARNING: Variable \'LIC.OTHER.NAMES\' never assigned a value.
38	38	WARNING: Variable \'LIC.SURNAME\' never assigned a value.
39	39	WARNING: Variable \'LIC.MOBILE\' never assigned a value.
40	40	WARNING: Variable \'GCPE.ISSUE.DATE\' never assigned a value.
41	41	WARNING: Variable \'GCPE.VEHICLE.MAKE\' never assigned a value.
42	42	WARNING: Variable \'GCPE.VEHICLE.MODEL\' never assigned a value.
43	43	WARNING: Variable \'GCPE.VEHICLE.REGISTRATION\' never assigned a value.
44	44	WARNING: Variable \'GCPE.VEHICLE.COLOUR\' never assigned a value.
45	45	WARNING: Variable \'LIC.GENERAL.OR.RESTRICTED\' never assigned a value.
46	46	WARNING: Variable \'GCPE.DOG.BREED\' never assigned a value.
47	47	WARNING: Variable \'GCPE.GROUP.ID\' never assigned a value.
48	48	WARNING: Variable \'GCPE.AREA.IDENTIFICATION\' never assigned a value.
49	49	WARNING: Variable \'GCA.FULL.NAME\' never assigned a value.
50	50	WARNING: Variable \'AMP.MAPTYPE.A\' never assigned a value.
51	52	9 Errors detected, No Object Code Produced.
52	54	Compilation Complete.
53	55	"GC.I.GCAREA" cataloged.
54	56	Compiling: Source = \'RGLIB/RG.SETUPNET\', Object = \'RGLIB.O/RG.SETUPNET\'
55	57	**********
56	59	Compilation Complete.
57	60	"RG.SETUPNET" cataloged.
58	61	Compiling: Source = \'RGLIB/RGPARMS\', Object = \'RGLIB.O/RGPARMS\'
59	62	**********
60	64	Compilation Complete.
61	65	"RGPARMS" cataloged.
62	66	Compiling: Source = \'RGLIB/RGSPEC\', Object = \'RGLIB.O/RGSPEC\'
63	67	**********
64	69	Compilation Complete.
65	70	"RGSPEC" cataloged.
66	71	Compiling: Source = \'RGLIB/RGTT\', Object = \'RGLIB.O/RGTT\'
67	72	**********
68	74	Compilation Complete.
69	75	"RGTT" cataloged.
70	76	Compiling: Source = \'RGLIB/RGWE\', Object = \'RGLIB.O/RGWE\'
71	77	**********
72	79	Compilation Complete.
73	80	"RGWE" cataloged.
74	81	RGLIB - 1 record(s) failed to compile.
75	201	COMO garb established 08:30:31 26 JUL 2019

Find String - Clean Up a Basic Library or Form File

It is often the case that backup copies of programs or forms are saved in the production library.

The *Find String* function automatically creates a list of items that are likely to be backup copies, based on the structure of the record id. In addition you can create a list of suffix names, such as *"weblogon name"* and record ids that have any of these suffixes will be added to the cleanup list.

In the following example all DBLIB items with a suffix in the form of a date have been added to the cleanup list. Any items that are not to be cleaned up can be retained by clicking the *Keep* column.

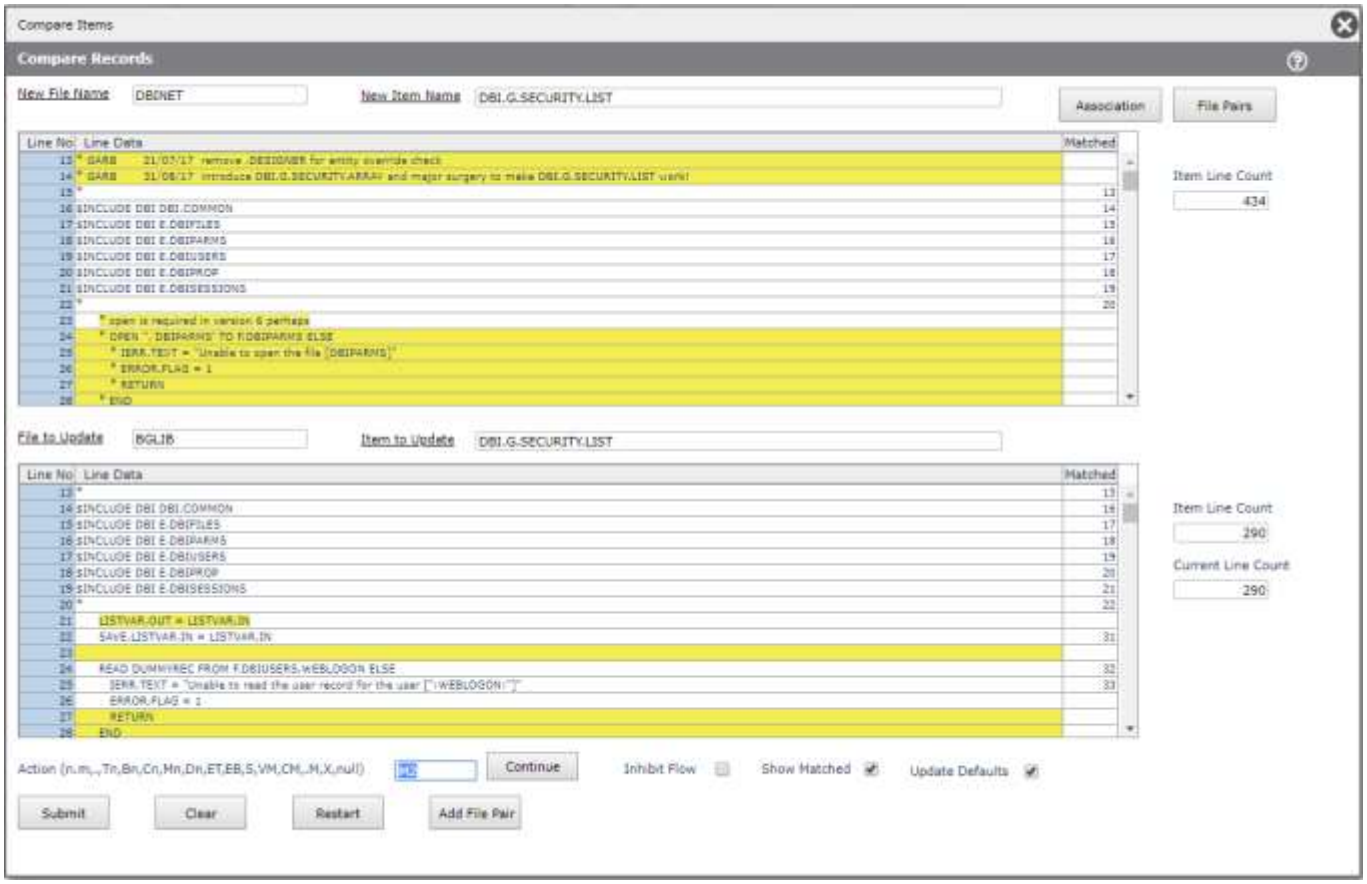


You can also enter and save a list of suffixes. Any record ids that end with a suffix from this list will be flagged for cleanup. The list shown at right is saved in DBIPARMS with an id of RV.SUFFIX.DBLIB.

You can then specify a *Backup File for Removed Records* and click the *Move Records* button. This will move all records in the cleanup list, where *Keep* is not flagged, from your production file to the backup file. If the backup file is set to *DBIBACKUP* then cleaned up records will be accessible via the *Backup* function.

Compare

The DesignBAIS Compare function allows you to compare 2 items from a file or from 2 different files.



Differences are highlighted.

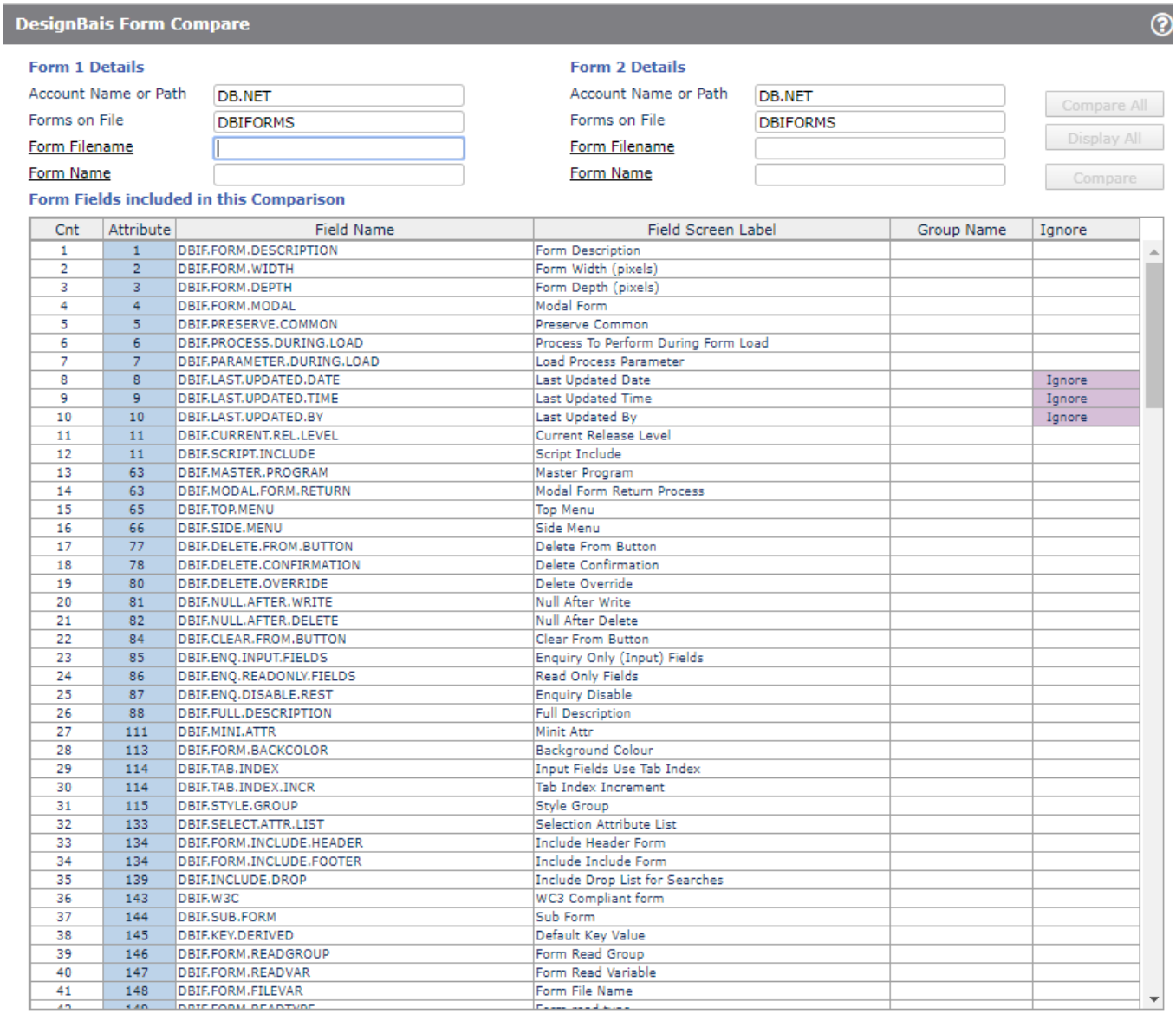
The *Update Defaults* checkbox is a toggle to control the default action that takes place when the *Continue* button is clicked.

With *Update Defaults* off the *Continue* button can be used to progressively move through all the differences without updating the record displayed in the bottom of the form. The display of each record is halted as each difference is encountered.

With *Update Defaults* on the *Action* will be automatically set to achieve the merging of differences into the bottom record, the *Item to Update*. In the example above *Action* is set to "*M2*" which is the command to merge, into the bottom record, the 2 highlighted lines at the top of the top record.

Form Compare

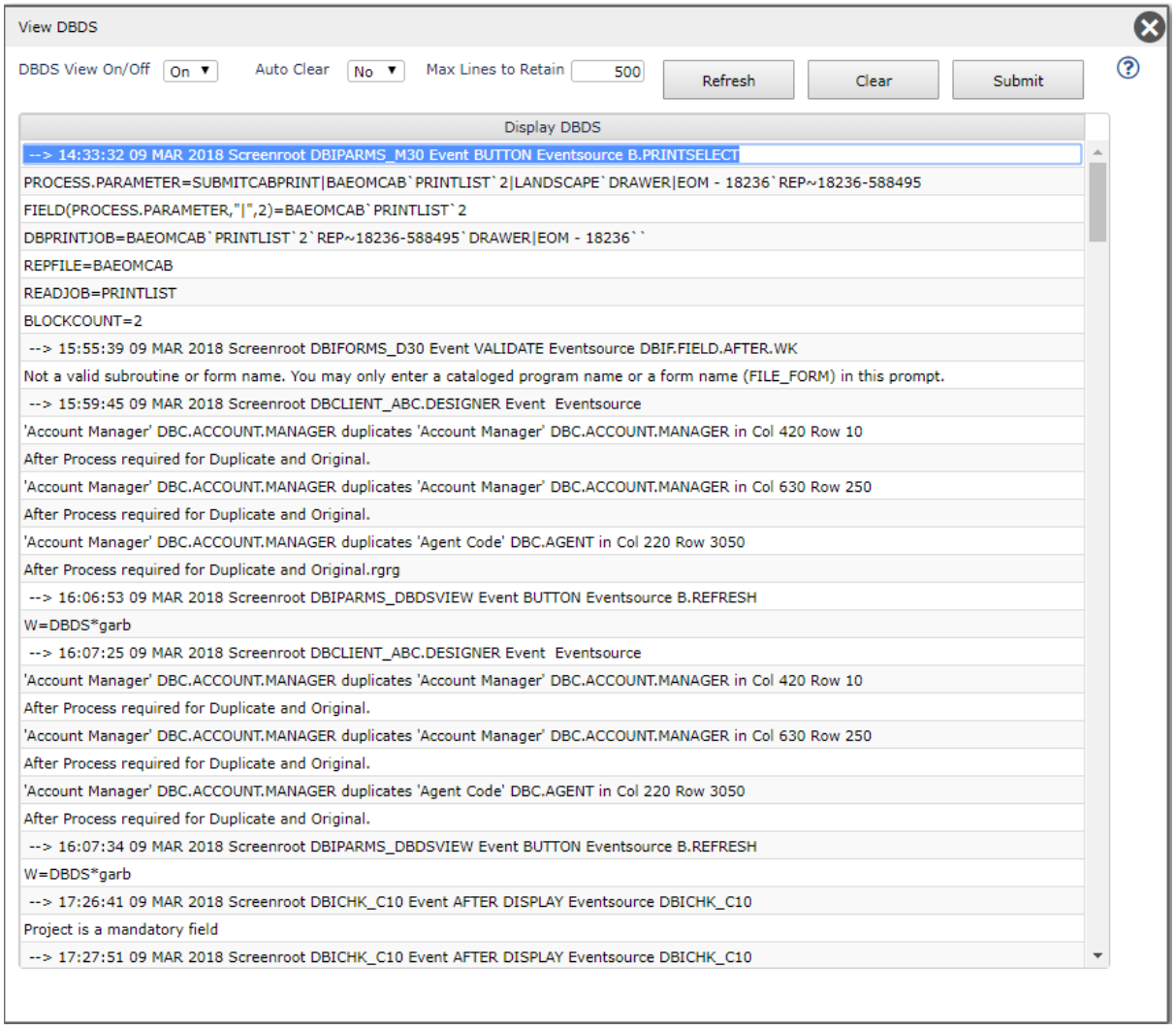
Compares two forms. This function performs best when there are only slight differences between the forms. This tool is designed to quickly highlight those differences.



Cnt	Attribute	Field Name	Field Screen Label	Group Name	Ignore
1	1	DBIF.FORM.DESCRIPTION	Form Description		
2	2	DBIF.FORM.WIDTH	Form Width (pixels)		
3	3	DBIF.FORM.DEPTH	Form Depth (pixels)		
4	4	DBIF.FORM.MODAL	Modal Form		
5	5	DBIF.PRESERVE.COMMON	Preserve Common		
6	6	DBIF.PROCESS.DURING.LOAD	Process To Perform During Form Load		
7	7	DBIF.PARAMETER.DURING.LOAD	Load Process Parameter		
8	8	DBIF.LAST.UPDATED.DATE	Last Updated Date		Ignore
9	9	DBIF.LAST.UPDATED.TIME	Last Updated Time		Ignore
10	10	DBIF.LAST.UPDATED.BY	Last Updated By		Ignore
11	11	DBIF.CURRENT.REL.LEVEL	Current Release Level		
12	11	DBIF.SCRIPT.INCLUDE	Script Include		
13	63	DBIF.MASTER.PROGRAM	Master Program		
14	63	DBIF.MODAL.FORM.RETURN	Modal Form Return Process		
15	65	DBIF.TOP.MENU	Top Menu		
16	66	DBIF.SIDE.MENU	Side Menu		
17	77	DBIF.DELETE.FROM.BUTTON	Delete From Button		
18	78	DBIF.DELETE.CONFIRMATION	Delete Confirmation		
19	80	DBIF.DELETE.OVERRIDE	Delete Override		
20	81	DBIF.NULL.AFTER.WRITE	Null After Write		
21	82	DBIF.NULL.AFTER.DELETE	Null After Delete		
22	84	DBIF.CLEAR.FROM.BUTTON	Clear From Button		
23	85	DBIF.ENQ.INPUT.FIELDS	Enquiry Only (Input) Fields		
24	86	DBIF.ENQ.READONLY.FIELDS	Read Only Fields		
25	87	DBIF.ENQ.DISABLE.REST	Enquiry Disable		
26	88	DBIF.FULL.DESCRIPTION	Full Description		
27	111	DBIF.MINI.ATTR	Minit Attr		
28	113	DBIF.FORM.BACKCOLOR	Background Colour		
29	114	DBIF.TAB.INDEX	Input Fields Use Tab Index		
30	114	DBIF.TAB.INDEX.INCR	Tab Index Increment		
31	115	DBIF.STYLE.GROUP	Style Group		
32	133	DBIF.SELECT.ATTR.LIST	Selection Attribute List		
33	134	DBIF.FORM.INCLUDE.HEADER	Include Header Form		
34	134	DBIF.FORM.INCLUDE.FOOTER	Include Include Form		
35	139	DBIF.INCLUDE.DROP	Include Drop List for Searches		
36	143	DBIF.W3C	WC3 Compliant form		
37	144	DBIF.SUB.FORM	Sub Form		
38	145	DBIF.KEY.DERIVED	Default Key Value		
39	146	DBIF.FORM.READGROUP	Form Read Group		
40	147	DBIF.FORM.READVAR	Form Read Variable		
41	148	DBIF.FORM.FILEVAR	Form File Name		

DBDS Log

DBDS output can now be retained in a log for later review. This is very useful when you need to display large amounts of data and overcomes the problem of the DBDS output being truncated in the Windows alert display.

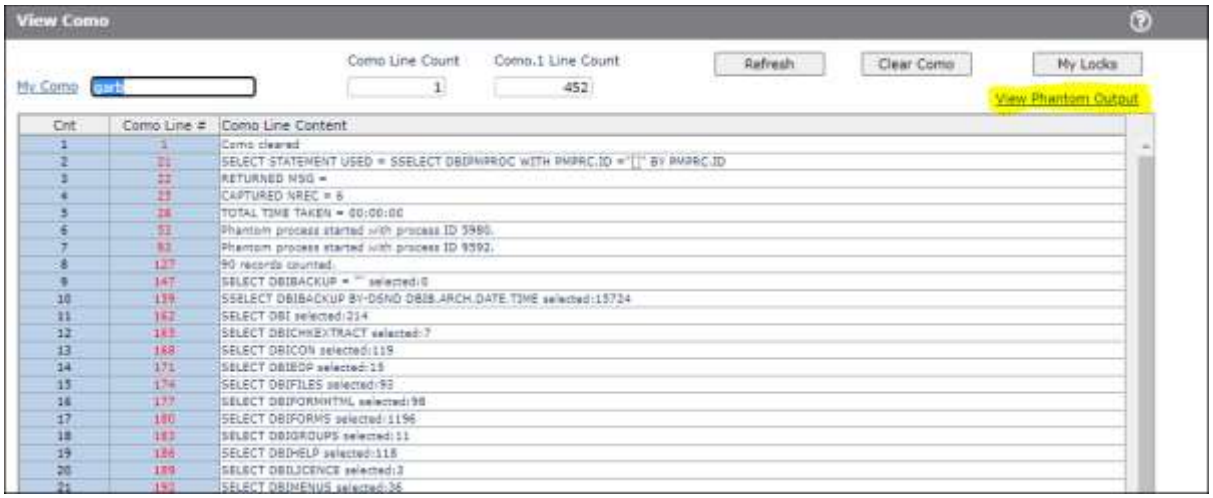


View Como

DesignBAIS now provides direct access to your Como record. Only significant lines are displayed. This feature is only available on databases that support Como or its equivalent.

It is now easy to quickly check for things like variable unassigned errors.

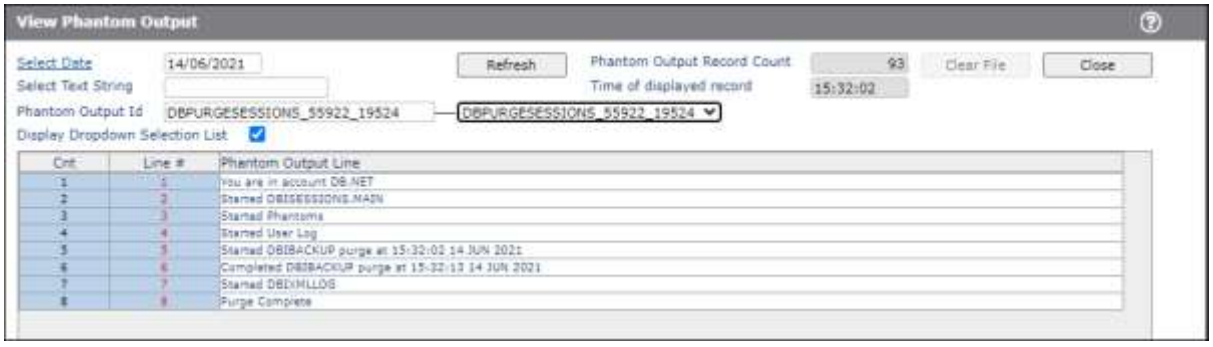
If a form aborts then first view the Como record to check for invalid reads or writes. If the Como is clear then run *dbi.run.last* in a terminal session which may reveal the cause of the abort.



It is also possible to view the Como of another developer by entering the *weblogon* in the first field. Click *My Como* to revert to the display of your Como.

The *View Como* form can also be accessed directly from the *Find String* and the *Active Users* forms.

The *View Phantom Output* provides quick access to view records in the &PH& file (on Universe) or equivalent file on other databases.

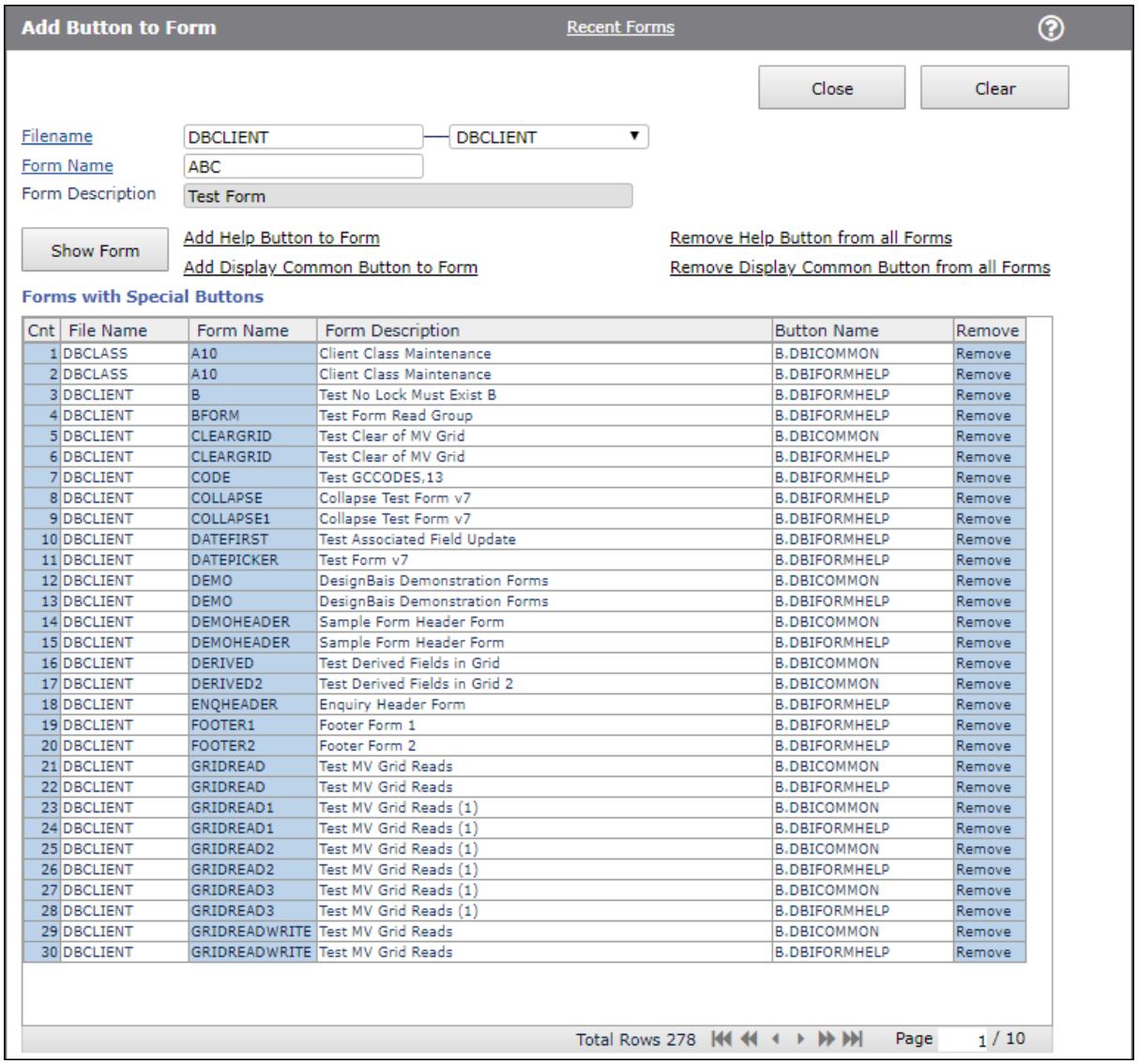


Add Button

There are several buttons that are recognized by DesignBAIS. The *Display Common* button can be added to any form in order to assist with debugging your application. The *Form Help* button when added to a form can be clicked to invoke form help.

Use the *Add Button* button on the *Code Editor* form to display the *Add Button to Form* form. This displays a list of all forms that currently have the standard buttons in place. These can be removed from individual forms, or from all forms in one go.

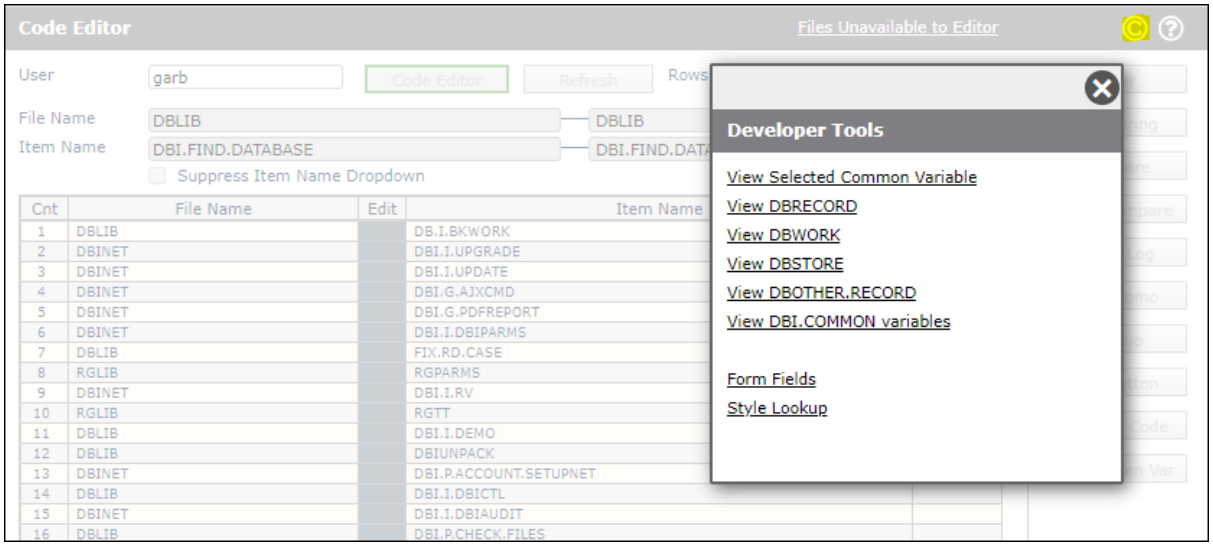
Retrieve any form by entering the *Filename* and *Form Name* and click the *Add Help* or *Add Display Common* buttons to add the buttons to a form.



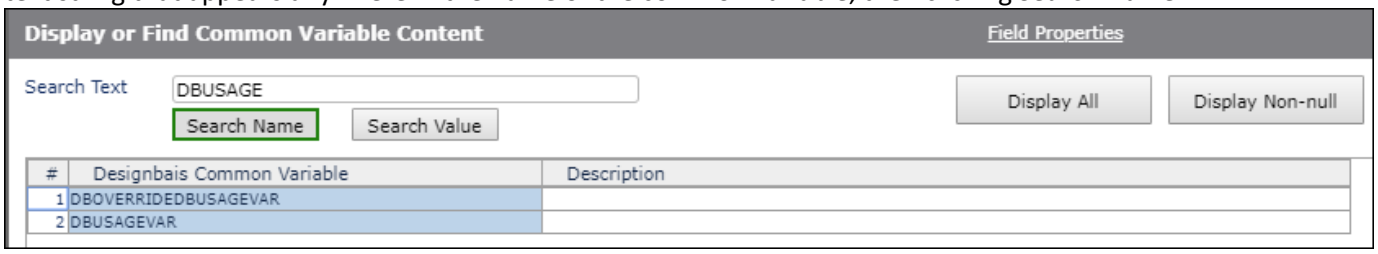
Cnt	File Name	Form Name	Form Description	Button Name	Remove
1	DBCLASS	A10	Client Class Maintenance	B.DBICOMMON	Remove
2	DBCLASS	A10	Client Class Maintenance	B.DBIFORMHELP	Remove
3	DBCLIENT	B	Test No Lock Must Exist B	B.DBIFORMHELP	Remove
4	DBCLIENT	BFORM	Test Form Read Group	B.DBIFORMHELP	Remove
5	DBCLIENT	CLEARGRID	Test Clear of MV Grid	B.DBICOMMON	Remove
6	DBCLIENT	CLEARGRID	Test Clear of MV Grid	B.DBIFORMHELP	Remove
7	DBCLIENT	CODE	Test GCCODES,13	B.DBIFORMHELP	Remove
8	DBCLIENT	COLLAPSE	Collapse Test Form v7	B.DBIFORMHELP	Remove
9	DBCLIENT	COLLAPSE1	Collapse Test Form v7	B.DBIFORMHELP	Remove
10	DBCLIENT	DATEFIRST	Test Associated Field Update	B.DBIFORMHELP	Remove
11	DBCLIENT	DATEPICKER	Test Form v7	B.DBIFORMHELP	Remove
12	DBCLIENT	DEMO	DesignBais Demonstration Forms	B.DBICOMMON	Remove
13	DBCLIENT	DEMO	DesignBais Demonstration Forms	B.DBIFORMHELP	Remove
14	DBCLIENT	DEMOHEADER	Sample Form Header Form	B.DBICOMMON	Remove
15	DBCLIENT	DEMOHEADER	Sample Form Header Form	B.DBIFORMHELP	Remove
16	DBCLIENT	DERIVED	Test Derived Fields in Grid	B.DBICOMMON	Remove
17	DBCLIENT	DERIVED2	Test Derived Fields in Grid 2	B.DBICOMMON	Remove
18	DBCLIENT	ENQHEADER	Enquiry Header Form	B.DBIFORMHELP	Remove
19	DBCLIENT	FOOTER1	Footer Form 1	B.DBIFORMHELP	Remove
20	DBCLIENT	FOOTER2	Footer Form 2	B.DBIFORMHELP	Remove
21	DBCLIENT	GRIDREAD	Test MV Grid Reads	B.DBICOMMON	Remove
22	DBCLIENT	GRIDREAD	Test MV Grid Reads	B.DBIFORMHELP	Remove
23	DBCLIENT	GRIDREAD1	Test MV Grid Reads (1)	B.DBICOMMON	Remove
24	DBCLIENT	GRIDREAD1	Test MV Grid Reads (1)	B.DBIFORMHELP	Remove
25	DBCLIENT	GRIDREAD2	Test MV Grid Reads (1)	B.DBICOMMON	Remove
26	DBCLIENT	GRIDREAD2	Test MV Grid Reads (1)	B.DBIFORMHELP	Remove
27	DBCLIENT	GRIDREAD3	Test MV Grid Reads (1)	B.DBICOMMON	Remove
28	DBCLIENT	GRIDREAD3	Test MV Grid Reads (1)	B.DBIFORMHELP	Remove
29	DBCLIENT	GRIDREADWRITE	Test MV Grid Reads	B.DBICOMMON	Remove
30	DBCLIENT	GRIDREADWRITE	Test MV Grid Reads	B.DBIFORMHELP	Remove

The Code Editor form is shown below with the *Form Help* and *Display Common* buttons.
Clicking the *Display Common* button displays a form that allows you to view the current contents of DesignBAIS Common Variables as they are at the point that the button is clicked.

You can review the contents of DBRECORD, DBOTHER.RECORD, DBWORK, DBSTORE and all other common variables using the *View DBI.COMMON variables* button.



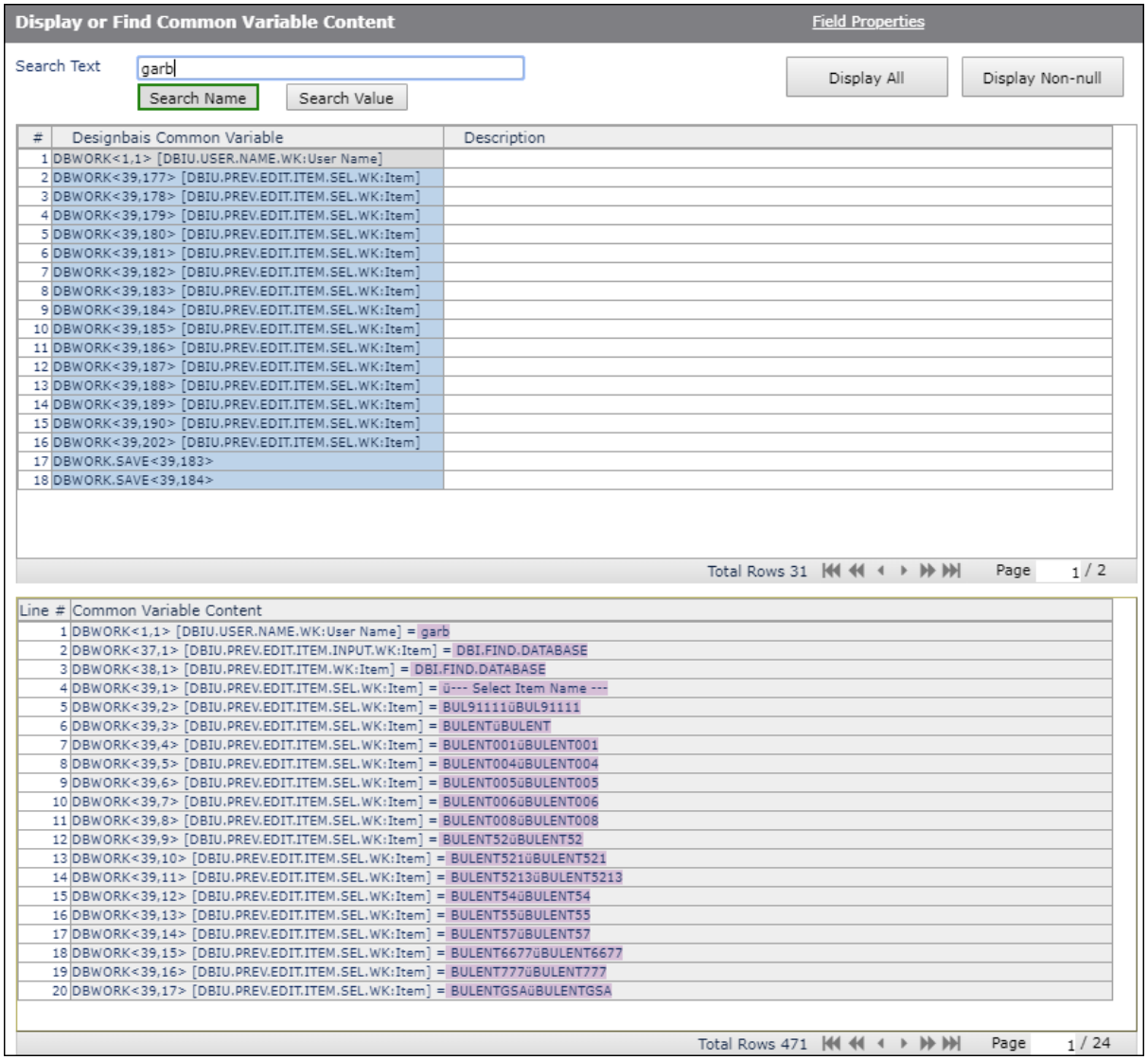
The *View Selected Common Variable* button allows you to view an common variable by entering a text string that appears anywhere in the name of the common variable, then clicking *Search Name*.



Click the blue-highlighted variable name to see the contents of that variable.

Alternatively enter a Search Text string and click *Search Value*. A list of all common variables that contain the string in their current value will be displayed, and you can click the blue-highlighted column to display the full value of a common variable from the list.

Use the *Display Non-null* button to limit the list of Common Variables displayed to only those that have a value.

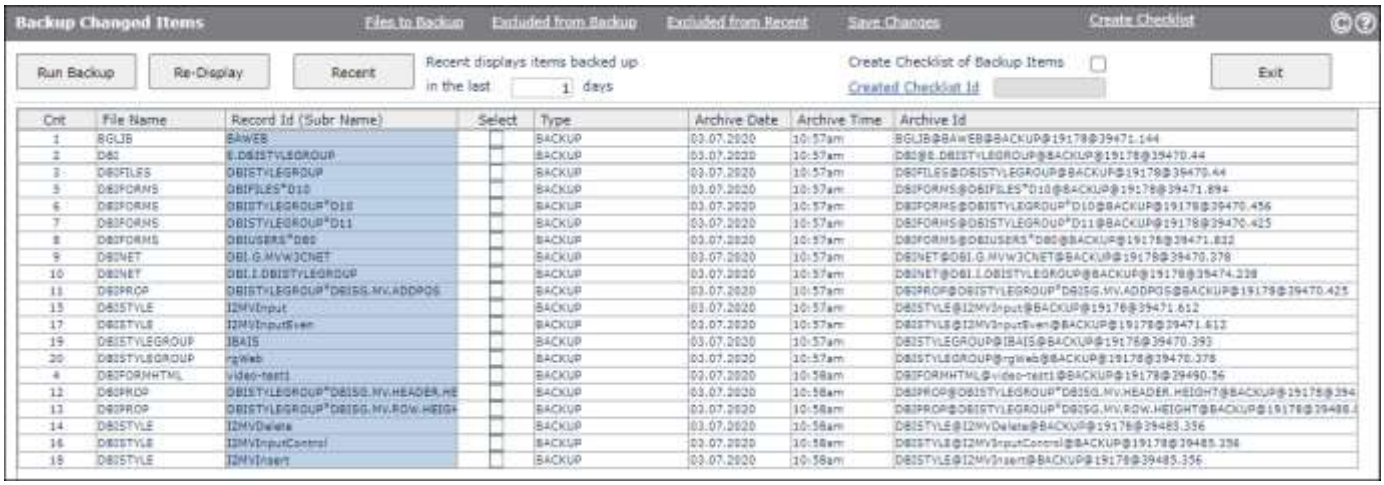

Display or Find Common Variable Content
Field Properties
Search Text
garb
Search Name
Search Value
Display All
Display Non-null
Designbais Common Variable Description
1 DBWORK<1,1> [DBIU.USER.NAME.WK:User Name]
2 DBWORK<39,177> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
3 DBWORK<39,178> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
4 DBWORK<39,179> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
5 DBWORK<39,180> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
6 DBWORK<39,181> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
7 DBWORK<39,182> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
8 DBWORK<39,183> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
9 DBWORK<39,184> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
10 DBWORK<39,185> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
11 DBWORK<39,186> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
12 DBWORK<39,187> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
13 DBWORK<39,188> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
14 DBWORK<39,189> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
15 DBWORK<39,190> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
16 DBWORK<39,202> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item]
17 DBWORK.SAVE<39,183>
18 DBWORK.SAVE<39,184>
Total Rows 31
Page 1 / 2
Line # Common Variable Content
1 DBWORK<1,1> [DBIU.USER.NAME.WK:User Name] = garb
2 DBWORK<37,1> [DBIU.PREV.EDIT.ITEM.INPUT.WK:Item] = DBI.FIND.DATABASE
3 DBWORK<38,1> [DBIU.PREV.EDIT.ITEM.WK:Item] = DBI.FIND.DATABASE
4 DBWORK<39,1> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = ü--- Select Item Name ---
5 DBWORK<39,2> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BUL91111üBUL91111
6 DBWORK<39,3> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENTüBULENT
7 DBWORK<39,4> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT001üBULENT001
8 DBWORK<39,5> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT004üBULENT004
9 DBWORK<39,6> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT005üBULENT005
10 DBWORK<39,7> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT006üBULENT006
11 DBWORK<39,8> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT008üBULENT008
12 DBWORK<39,9> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT52üBULENT52
13 DBWORK<39,10> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT521üBULENT521
14 DBWORK<39,11> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT5213üBULENT5213
15 DBWORK<39,12> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT54üBULENT54
16 DBWORK<39,13> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT55üBULENT55
17 DBWORK<39,14> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT57üBULENT57
18 DBWORK<39,15> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT6677üBULENT6677
19 DBWORK<39,16> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENT777üBULENT777
20 DBWORK<39,17> [DBIU.PREV.EDIT.ITEM.SEL.WK:Item] = BULENTGSAüBULENTGSA
Total Rows 471
Page 1 / 24

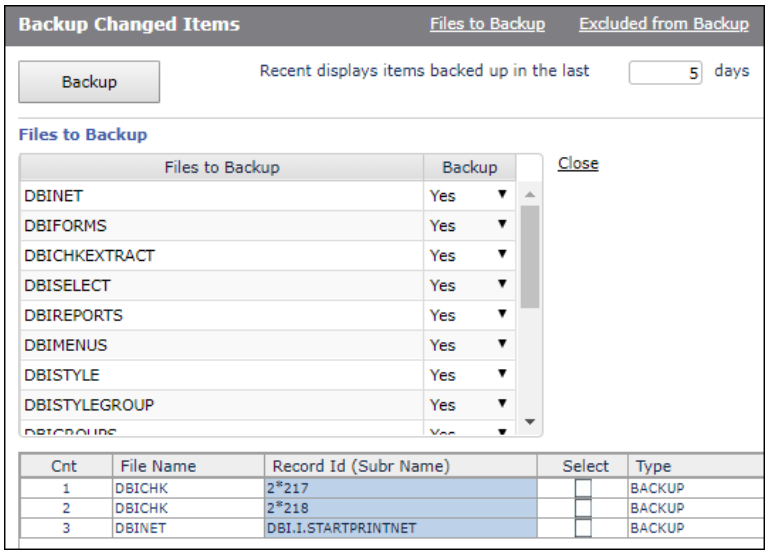
View Backup

The Backup function is designed for developers. It provides a method of retaining multiple versions of software elements, for example basic programs or forms. Changed items are written to the file DBIBACKUP. This file can be either a folder (Universe Type 1 or 19) or a large hashed file with reasonable separation since it will store potentially large basic subroutines.

There are two aspects to this tool. The backup function and the version review function.

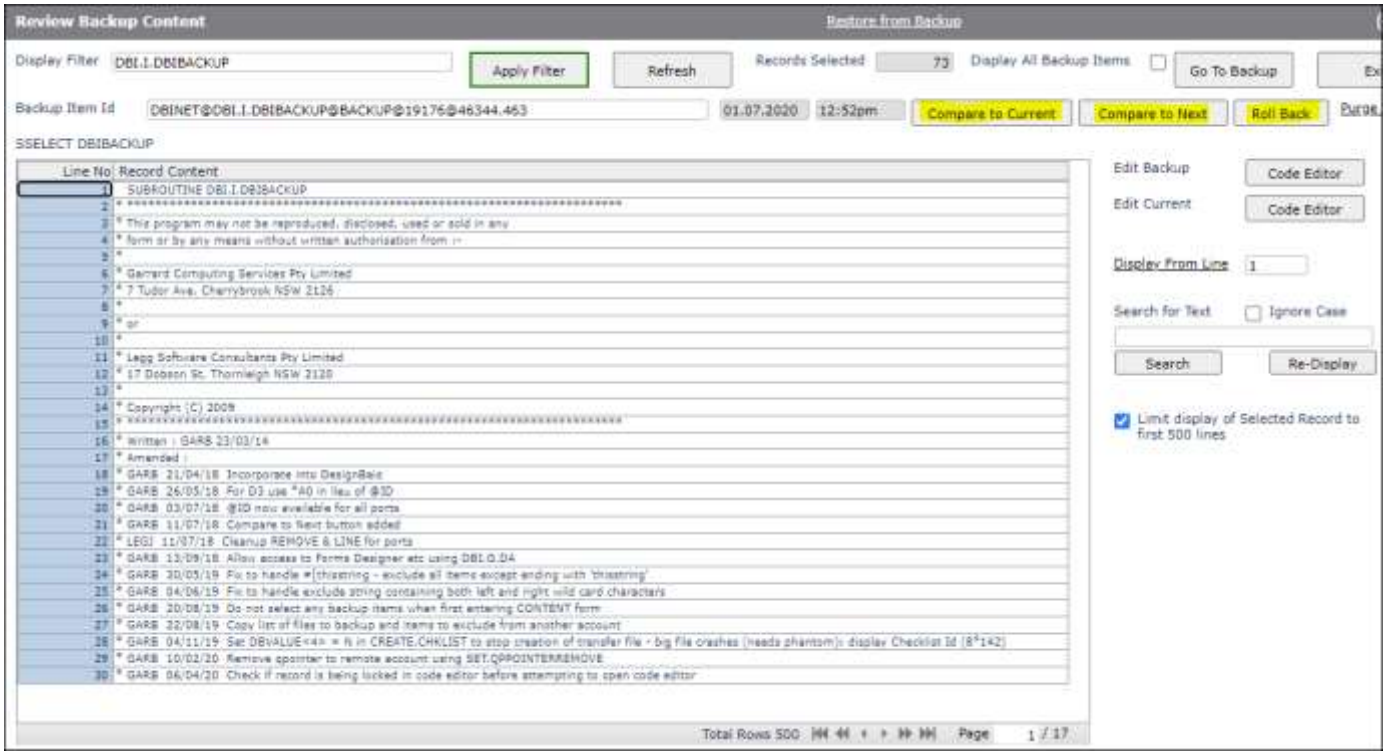
Click the *Run Backup* button to initiate a backup of all files listed in the *Files to Backup* list. This list is maintained in System Parameters or from this form. Any items in the files in the list that differ from the most recent version in the DBIBACKUP file will be backed up again.



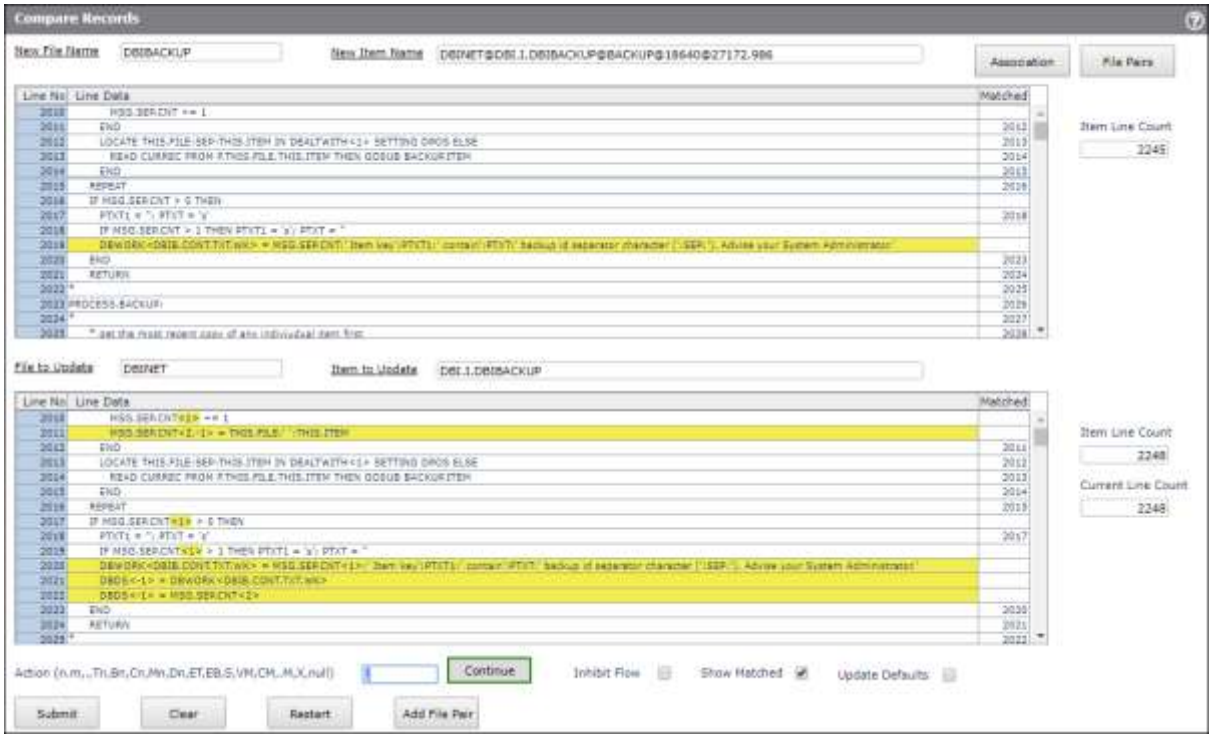


The *Review Backup Content* form is the form that displays first when *View Backup* is clicked. This form is used to view and access items in the backup file. Most often you will enter a string in the *Display* Filter field, say the name of a subroutine, then tab or click *Apply Filter*.

The list of items matching this filter will display. Click the *Record Id* in the selected row of the third column to access the backed up item.

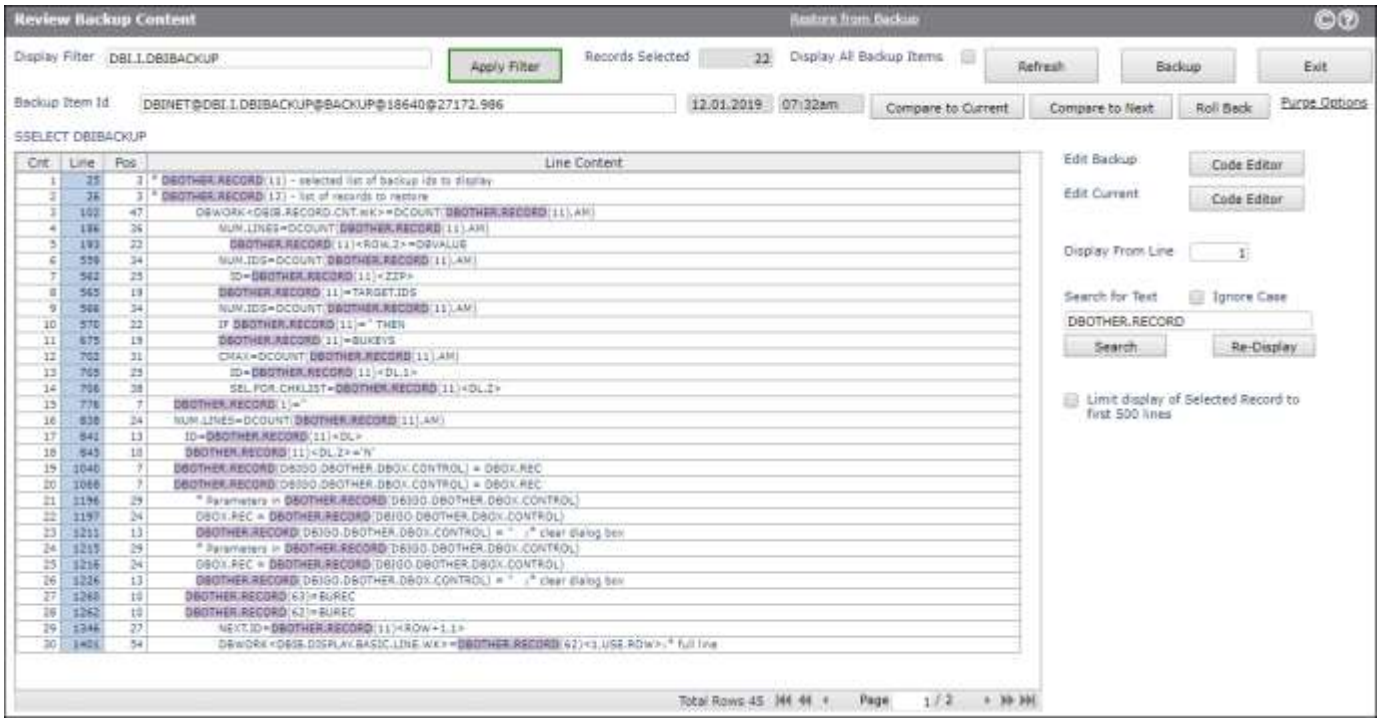


Use the *Compare to Current* or *Compare to Next* to invoke a line by line compare of the backup vs the current version, or the backup vs the next most recent backup.



In this way you can review what changes have occurred over the life of the selected item.

Use the *Search for Text* field to display only those lines that contain a particular string. Sometimes it is useful to be able to view selected lines blocked together rather than clicking through a large routine using a *Find Next* function in Code Editor.

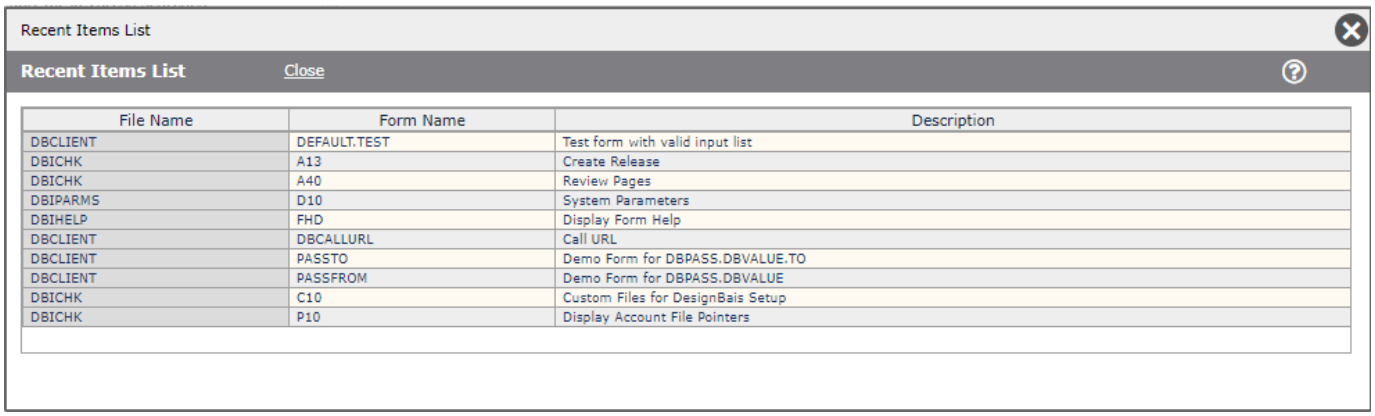


Use the *Roll Back* button to replace the current version with a selected version from backup.

Forms Designer – Recent button

Forms Designer, *Report Designer*, *Selection Process* and *Run a Form* all now have a ***Recent*** button to display a list of the 10 most recently accessed items.

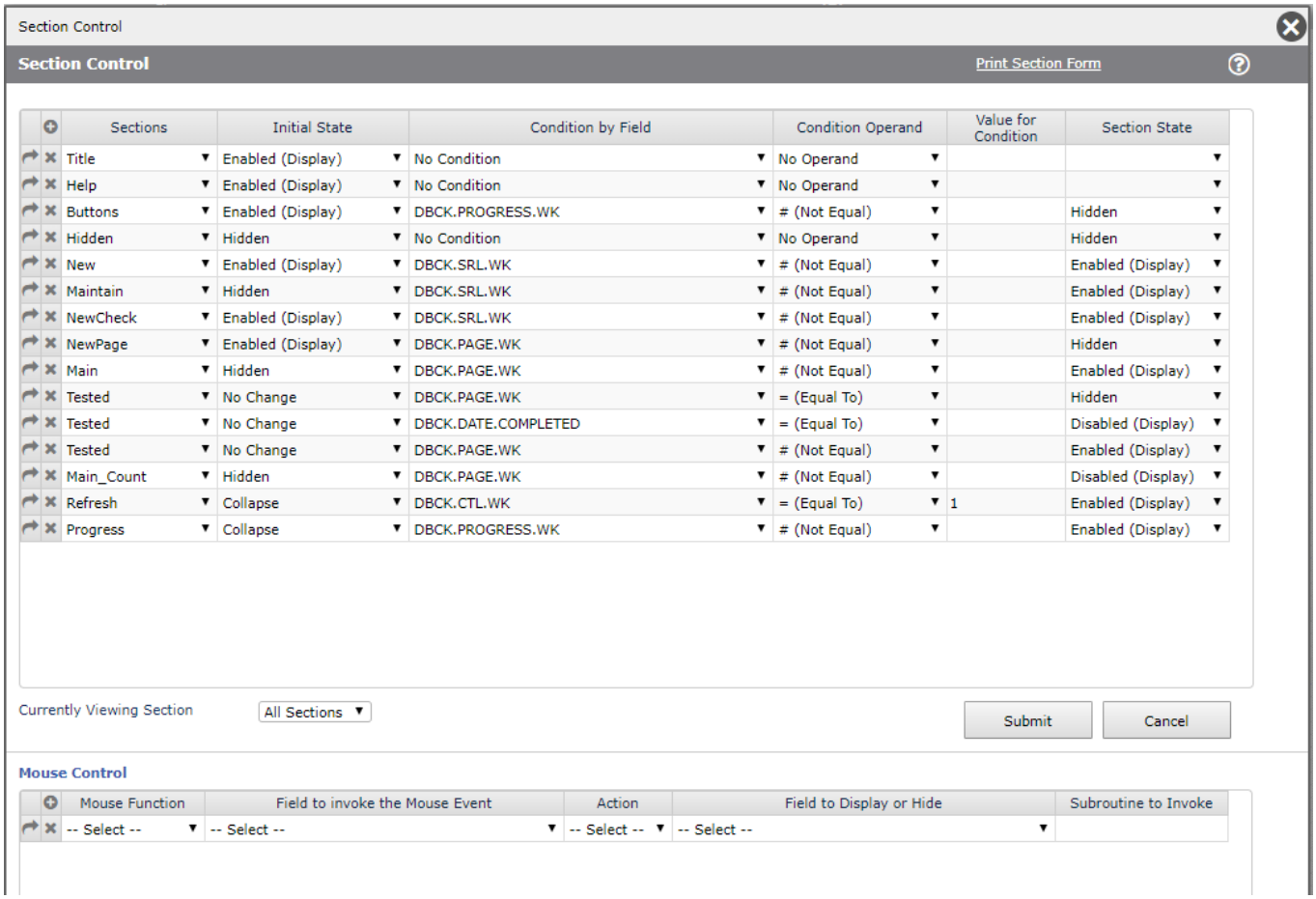
Use the *Recent* button to avoid having to repeatedly type in the file and form name.



File Name	Form Name	Description
DBCLIENT	DEFAULT.TEST	Test form with valid input list
DBICHK	A13	Create Release
DBICHK	A40	Review Pages
DBIPARMS	D10	System Parameters
DBIHELP	FHD	Display Form Help
DBCLIENT	DBCALLURL	Call URL
DBCLIENT	PASSTO	Demo Form for DBPASS.DBVALUE.TO
DBCLIENT	PASSFROM	Demo Form for DBPASS.DBVALUE
DBICHK	C10	Custom Files for DesignBais Setup
DBICHK	P10	Display Account File Pointers

Forms Designer – Form Section control

Forms Designer now controls the Sections within a form. If all fields for a section are removed from a form then the section is removed from the list. Sections are sorted based on column and row. Warnings can be displayed for overlapping sections that potentially interfere with collapsing.

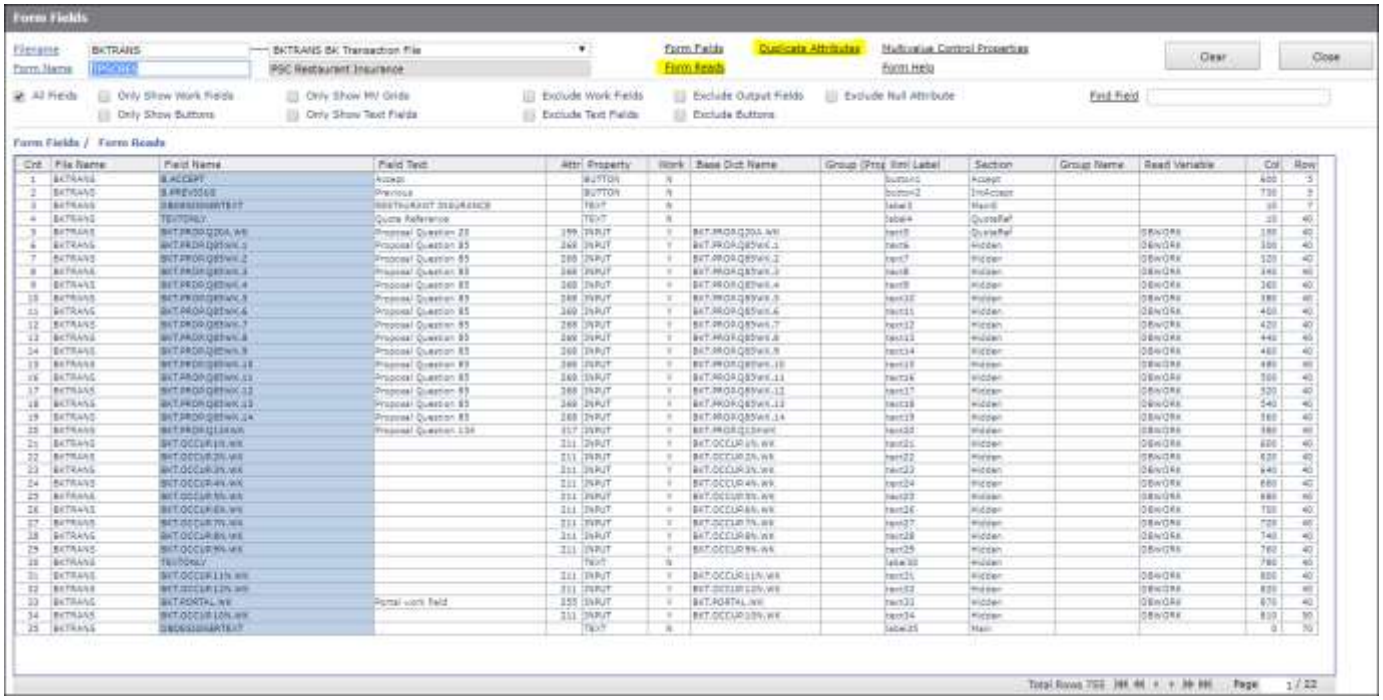


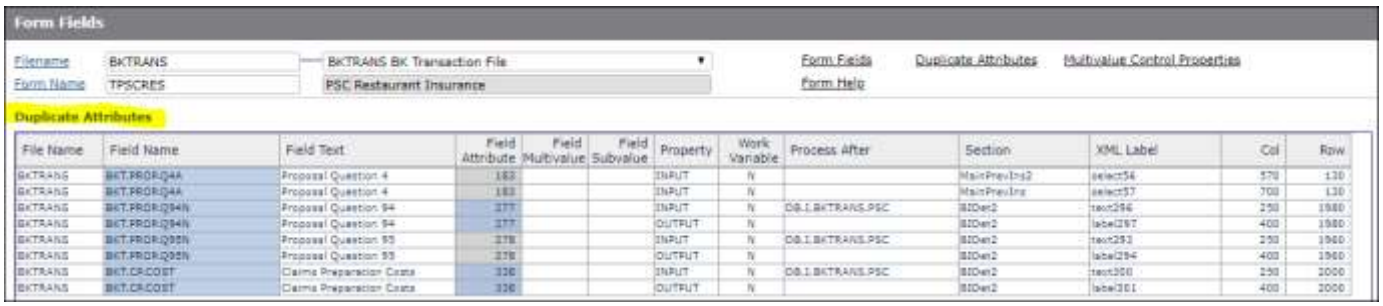
Forms Designer – Form Fields enquiry

Accessed from the *Help Definition* option on the Tools Menu.

This form allows you to analyse large forms in order to reveal duplicate attribute and DBWORK usage.

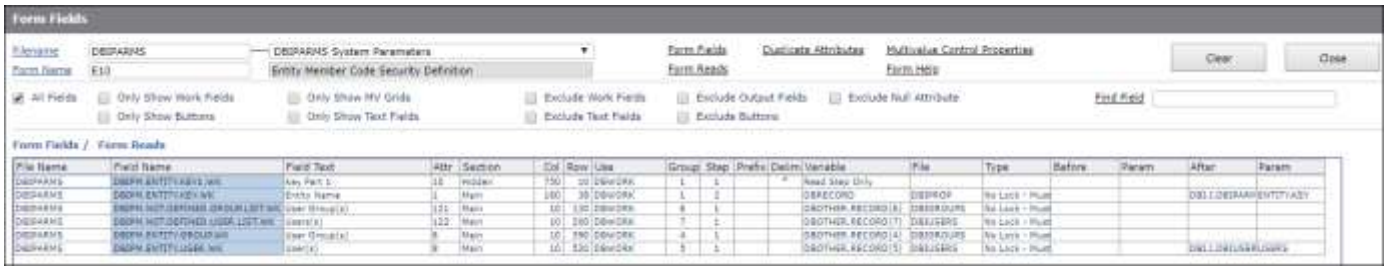
All columns can be sorted to allow you, for example, to find where on a form a particular field is located, or if the field is in fact on the form.





The *Form Reads* option displays all reads on the form as shown in the next example.

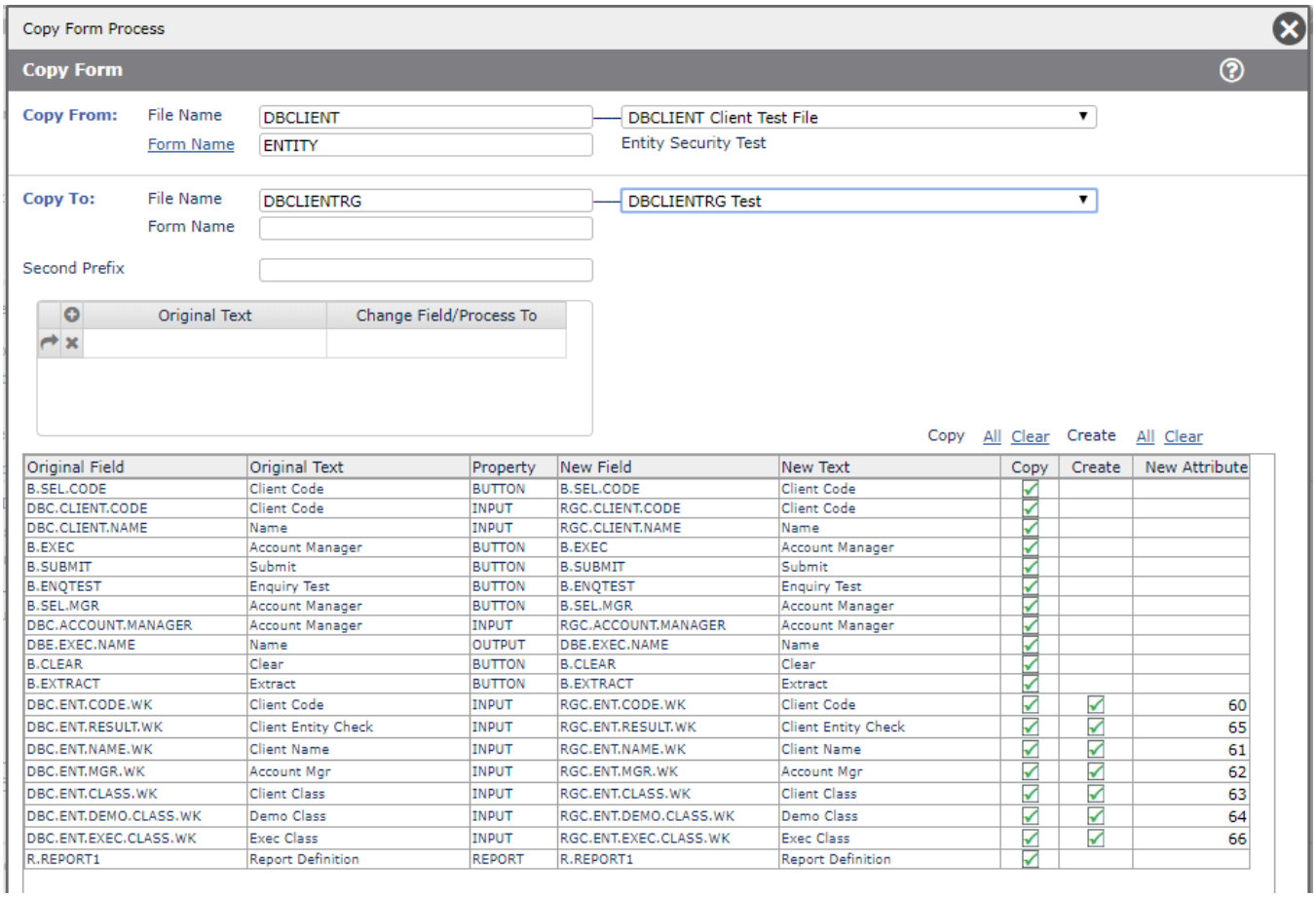
Form Fields
Filename
DBPARMS
DBPARMS System Parameters
Form Name
E10
Entity Member Code Security Definition
Form Fields
Duplicate Attributes
Multivalue Control Properties
Form Reads
Form Help
Clear
Close
All Fields
Only Show Work Fields
Only Show Buttons
Only Show MV Grids
Only Show Text Fields
Exclude Work Fields
Exclude Text Fields
Exclude Output Fields
Exclude Buttons
Exclude Null Attribute
Find Field
Form Fields / Form Reads


Form Fields
Filename
DBPARMS
DBPARMS System Parameters
Form Name
E10
Entity Member Code Security Definition
Form Fields
Duplicate Attributes
Multivalue Control Properties
Form Reads
Form Help
Clear
Close
All Fields
Only Show Work Fields
Only Show Buttons
Only Show MV Grids
Only Show Text Fields
Exclude Work Fields
Exclude Text Fields
Exclude Output Fields
Exclude Buttons
Exclude Null Attribute
Find Field
Form Fields / Form Reads

Copy Form

Copy Form has been enhanced to provide much more functionality and flexibility.

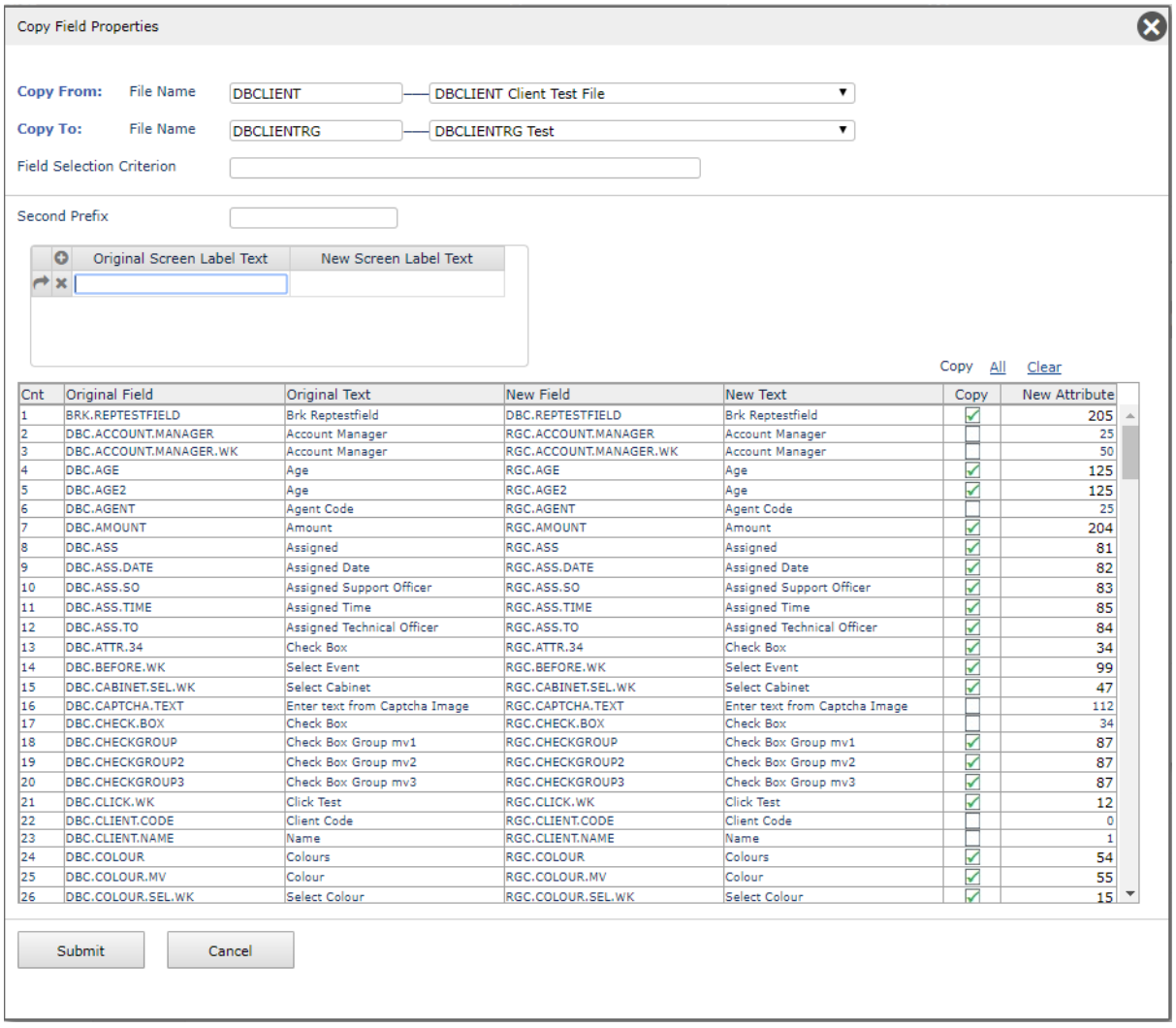
- allows selected fields to be copied, rather than all fields on the source form
- if copying to a different file then the field equates prefix is changed to that of the new file
- screen label text and attribute number can be amended



Original Field	Original Text	Property	New Field	New Text	Copy	Create	New Attribute
B.SEL.CODE	Client Code	BUTTON	B.SEL.CODE	Client Code	✓		
DBC.CLIENT.CODE	Client Code	INPUT	RGC.CLIENT.CODE	Client Code	✓		
DBC.CLIENT.NAME	Name	INPUT	RGC.CLIENT.NAME	Name	✓		
B.EXEC	Account Manager	BUTTON	B.EXEC	Account Manager	✓		
B.SUBMIT	Submit	BUTTON	B.SUBMIT	Submit	✓		
B.ENQTEST	Enquiry Test	BUTTON	B.ENQTEST	Enquiry Test	✓		
B.SEL.MGR	Account Manager	BUTTON	B.SEL.MGR	Account Manager	✓		
DBC.ACCOUNT.MANAGER	Account Manager	INPUT	RGC.ACCOUNT.MANAGER	Account Manager	✓		
DBE.EXEC.NAME	Name	OUTPUT	DBE.EXEC.NAME	Name	✓		
B.CLEAR	Clear	BUTTON	B.CLEAR	Clear	✓		
B.EXTRACT	Extract	BUTTON	B.EXTRACT	Extract	✓		
DBC.ENT.CODE.WK	Client Code	INPUT	RGC.ENT.CODE.WK	Client Code	✓	✓	60
DBC.ENT.RESULT.WK	Client Entity Check	INPUT	RGC.ENT.RESULT.WK	Client Entity Check	✓	✓	65
DBC.ENT.NAME.WK	Client Name	INPUT	RGC.ENT.NAME.WK	Client Name	✓	✓	61
DBC.ENT.MGR.WK	Account Mgr	INPUT	RGC.ENT.MGR.WK	Account Mgr	✓	✓	62
DBC.ENT.CLASS.WK	Client Class	INPUT	RGC.ENT.CLASS.WK	Client Class	✓	✓	63
DBC.ENT.DEMO.CLASS.WK	Demo Class	INPUT	RGC.ENT.DEMO.CLASS.WK	Demo Class	✓	✓	64
DBC.ENT.EXEC.CLASS.WK	Exec Class	INPUT	RGC.ENT.EXEC.CLASS.WK	Exec Class	✓	✓	66
R.REPORT1	Report Definition	REPORT	R.REPORT1	Report Definition	✓		

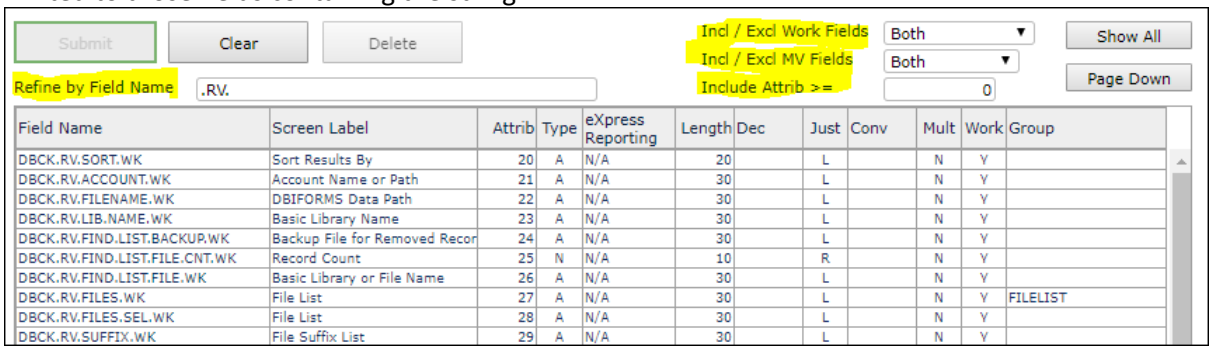
Field Properties - Advanced Copy

You can now selectively copy properties to a new file. The field equates prefix is changed to that of the new file. Screen label text and attribute can also be changed. To access click *Copy Field* then click *Advanced Copy* on the form header bar.



Field Properties - Refine by Field Name

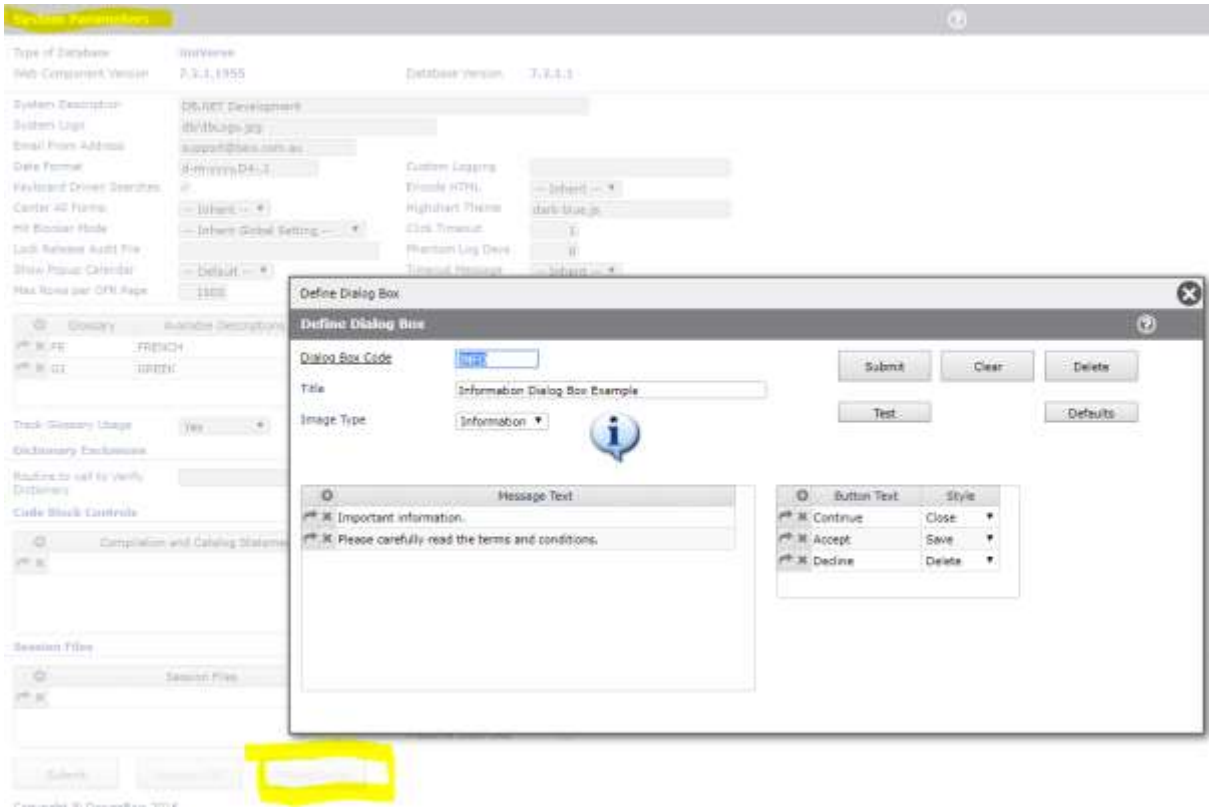
Use the *Refine by Field Name* to filter the display of field properties. In this example the display is limited to those fields containing the string *".RV."*.



You can also filter by Work Field, Multivalue field or attribute number.

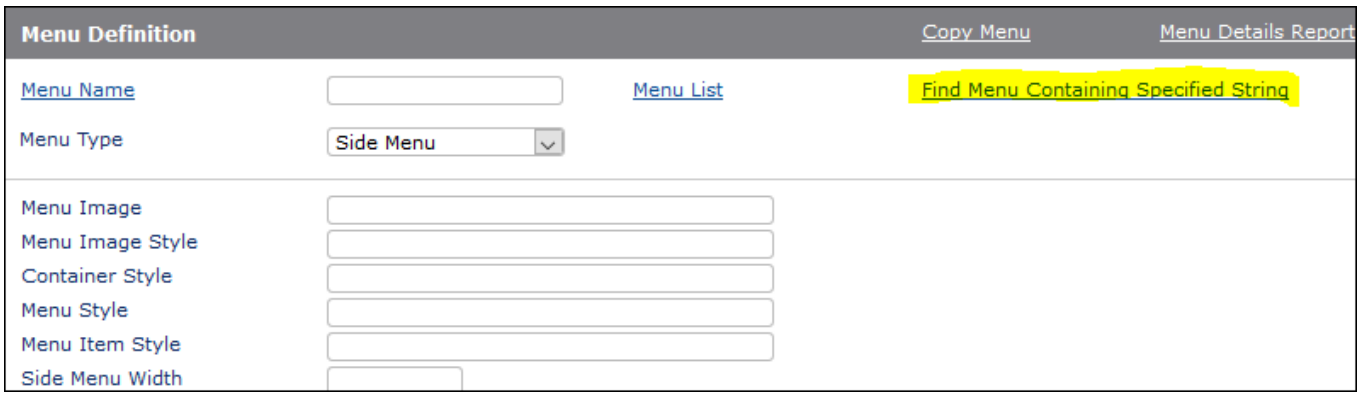
Dialog Box alternative

There is an alternative Dialog Box function allowing up to 5 buttons on the dialog box form. There is the provision to set up standard dialog boxes (see System Parameters *Dialog Setup* button)

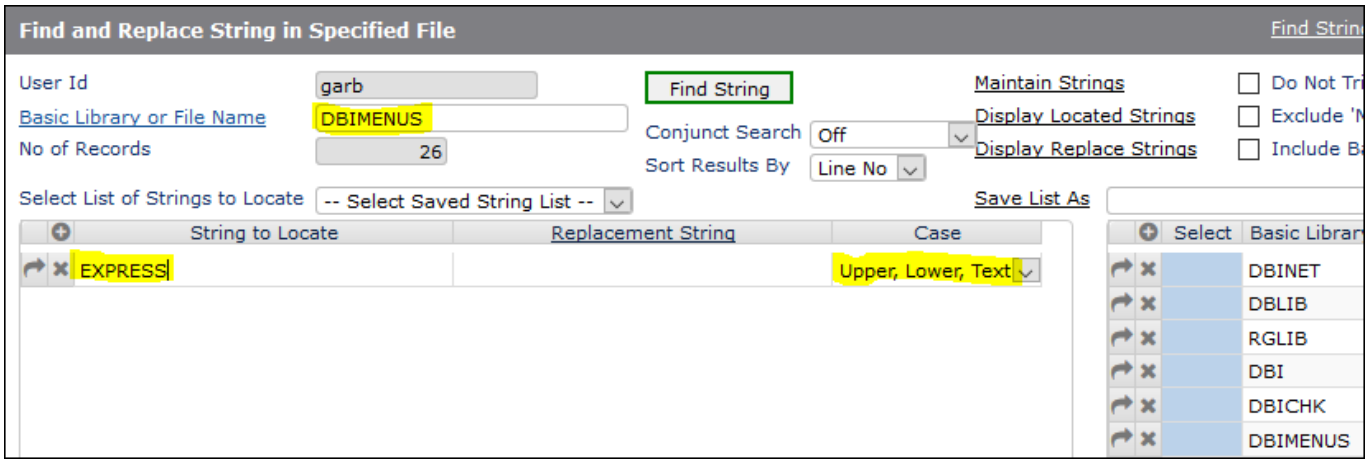


Menu Definitions – Finding a menu

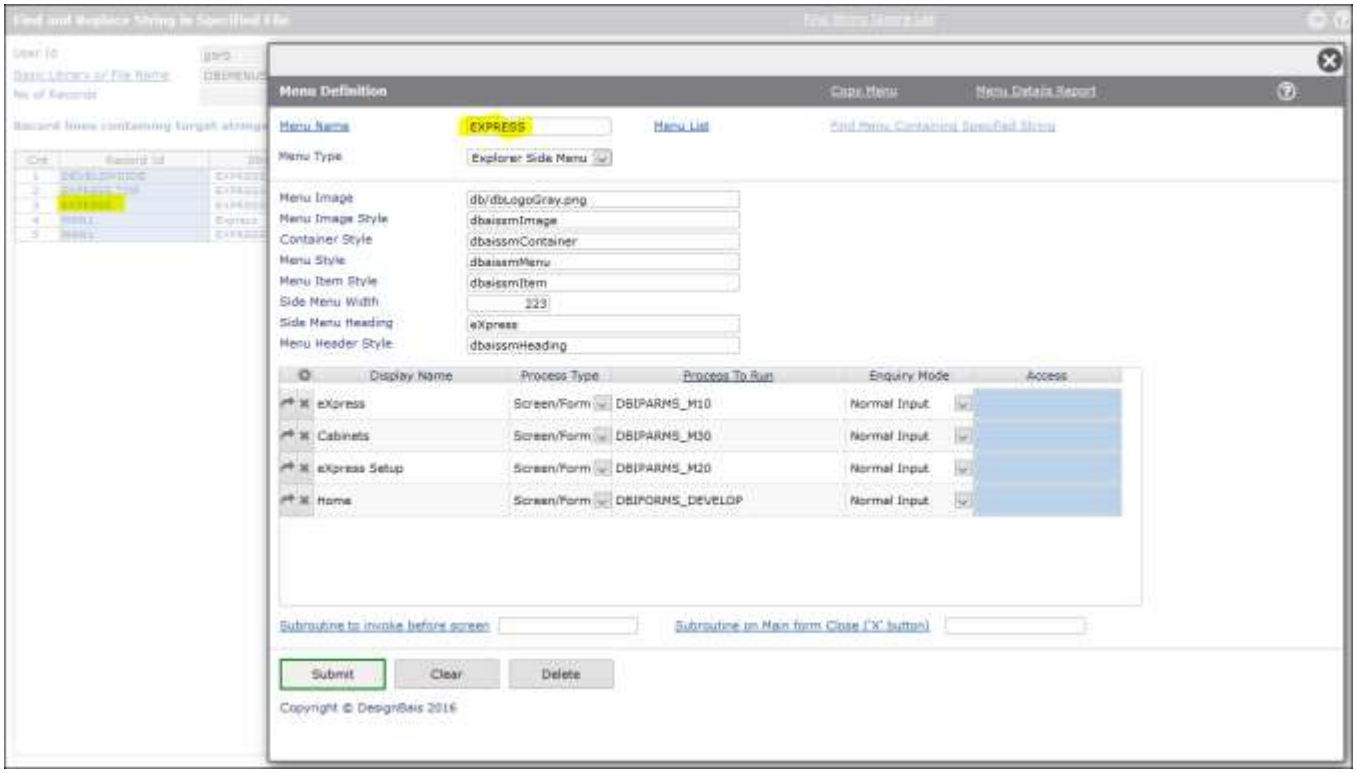
A common problem is to know which menu contains the option that you wish to amend, or to know which menu it is that you want to add an option to. Click on the *Menu Definitions* option on the main tools menu then click *Find Menu Containing Specified String*.



The *Find and Replace String in Specified File* will open with the file *DBIMENUS* selected. Enter the string that you want to search for. In the example below the string to locate is "*EXPRESS*". You may use the *Case* dropdown to look for upper, lower and text case if required. Then click *Find String* or press the Enter key.

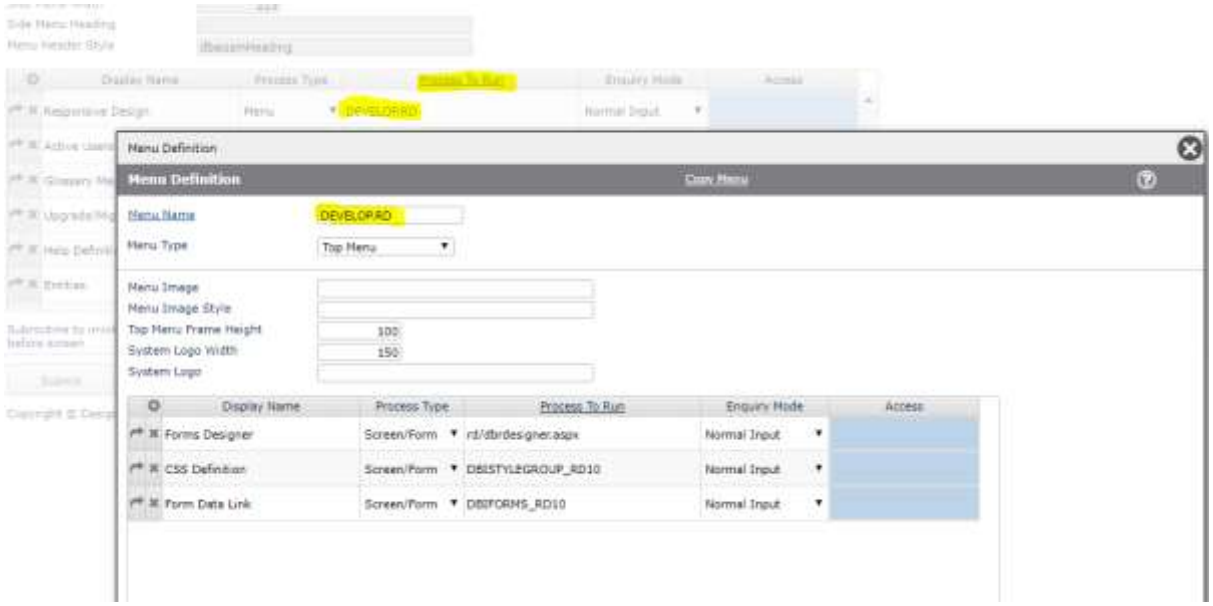


The menus in which the string is found are displayed. Click the row in the *Record Id* column to call up that menu in the Menu Definitions maintenance tool.



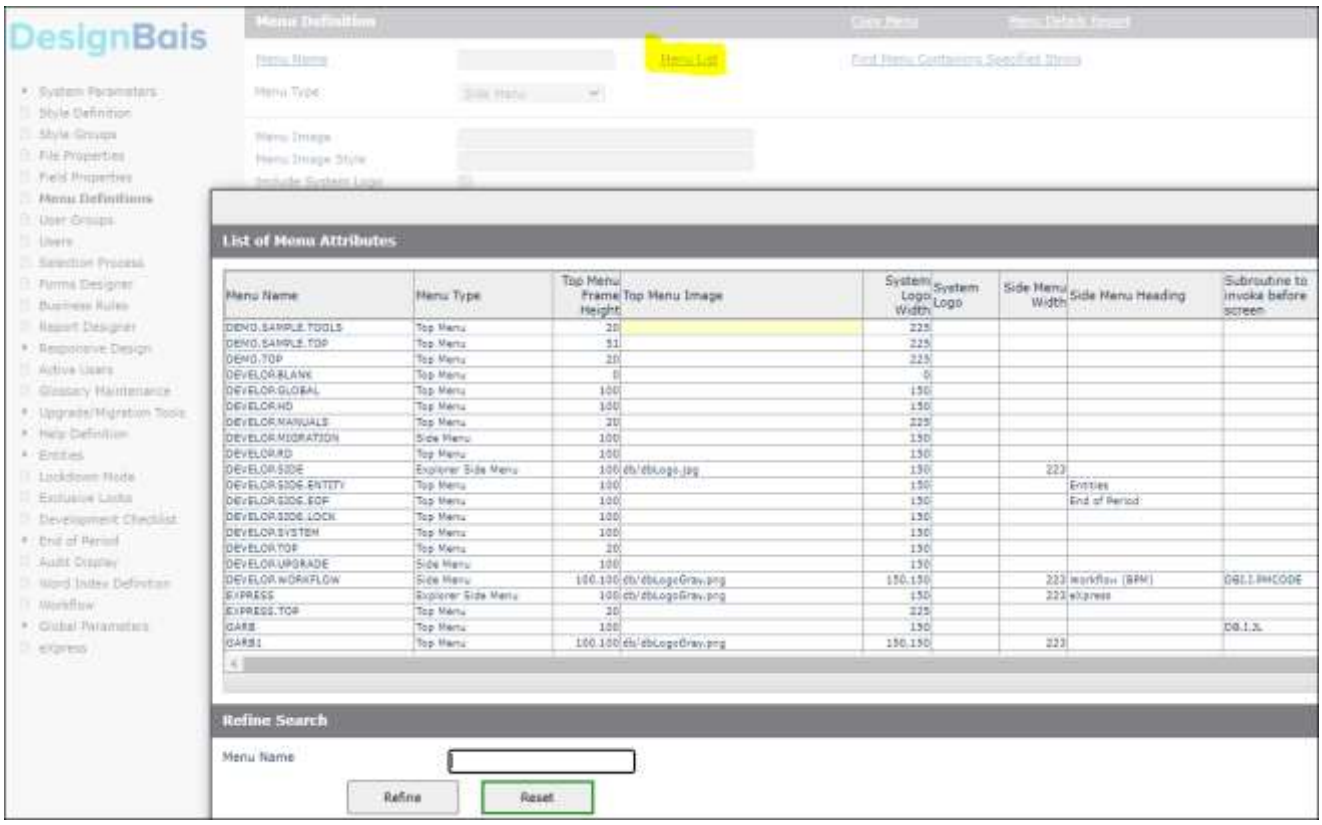
Menu Definitions – Maintaining a menu called from a menu

You can now directly access a menu that is called from the menu that you are maintaining by placing focus on the row of the grid and clicking *Process to Run*.



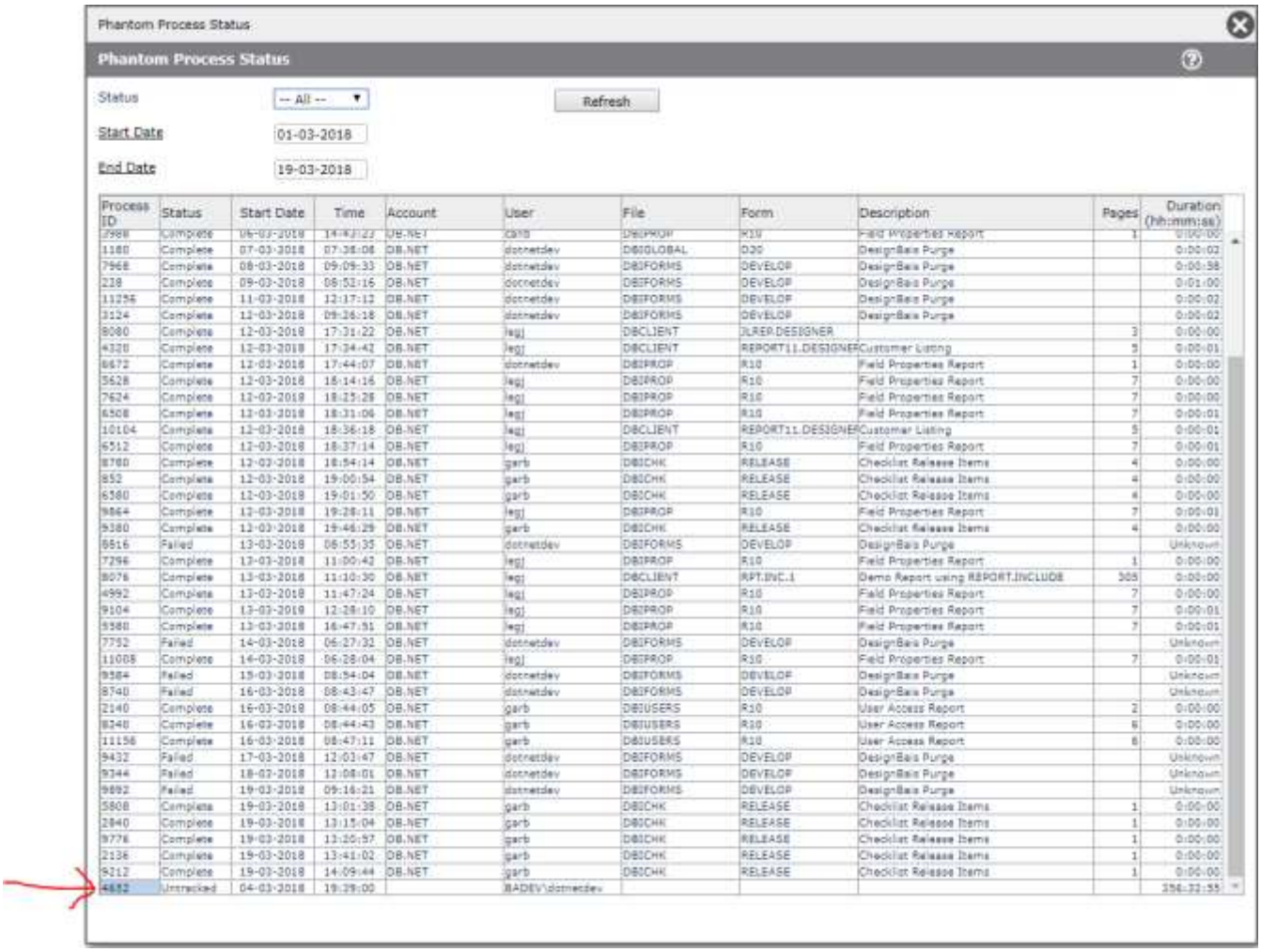
Menu Definitions – Menu List lookup

Using the *Menu List* search button, rather than the traditional *Menu Name* lookup, may provide an easier to read list of available menus. Select the menu that you wish to view or maintain from the list in column 2 of the display.



Active Users

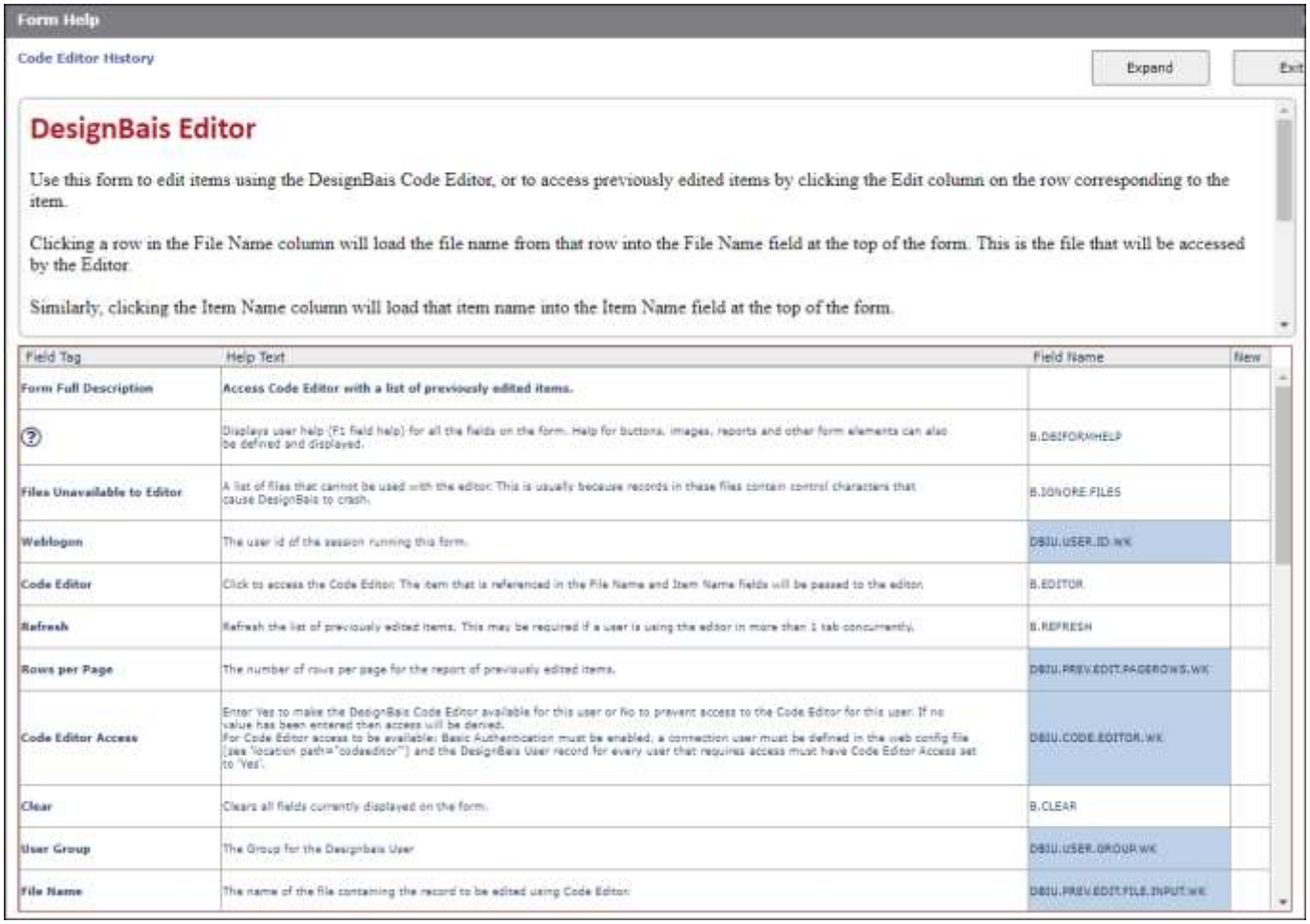
There are buttons to display *Phantom Status*, *Hit Status* and *Como*. Phantom processes can be killed with a button click (see red arrow below pointing to the active (blue highlight) process.



Form Help

New Form Help function allows F1 help text for all fields on a form to be viewed and amended. Users can access the help form for the form they are in by clicking the help button. Developers can directly access the field properties by clicking on the *Field Name* column.

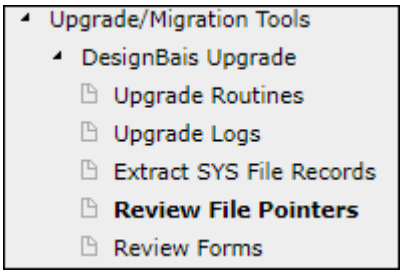
- Use the *Help Definition → Form Help* menu option to enter and maintain form help.
- Use the *Add Button* function on the *Editor* form to add the form help button to any form.
- The DesignBAIS F1 Help function now displays field help from the form help record if it exists. This provides the ability to have different field help for the same field used on different forms.
- F1 help can now be attached to buttons.



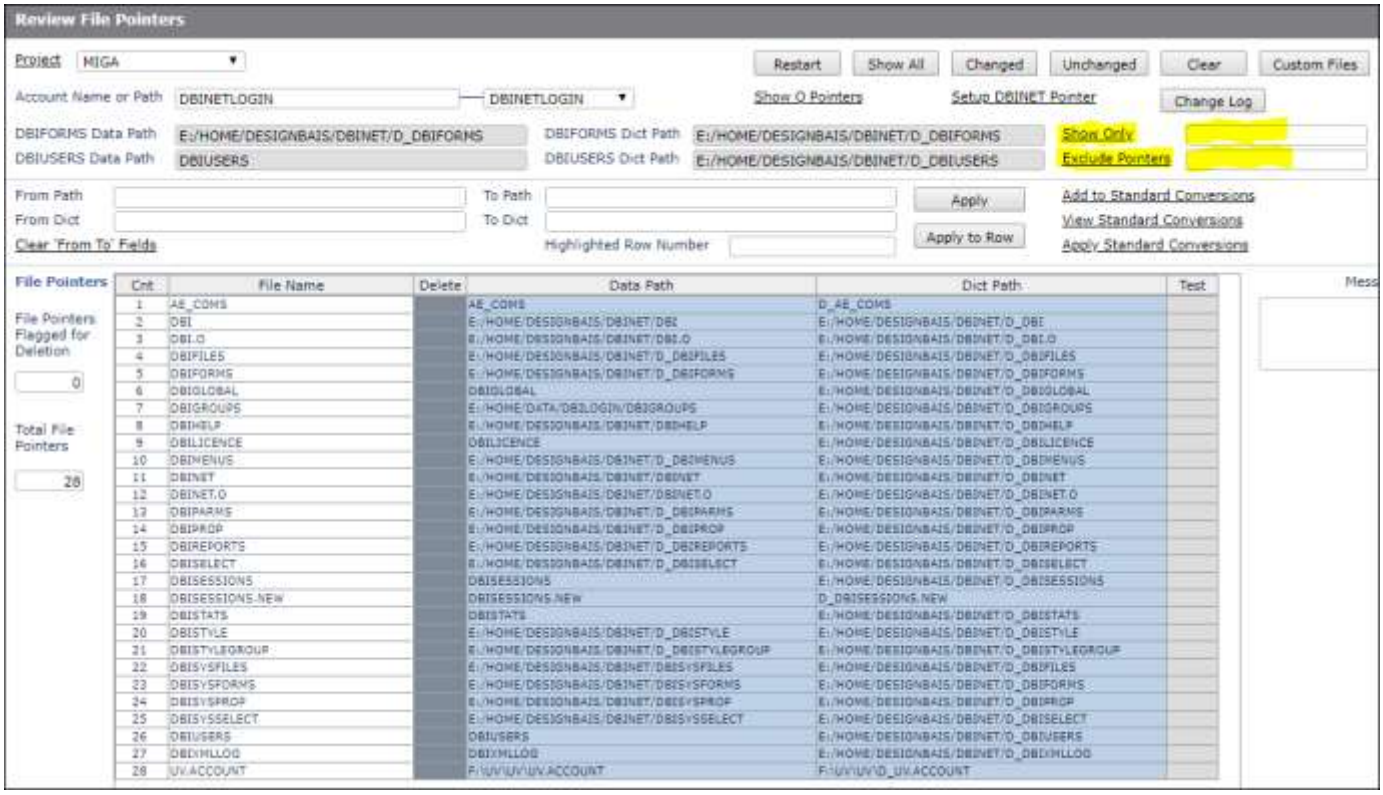
Review File Pointers

This function allows you to amend file and Q pointers in an account. This task is usually not required very often. More often you may want to display pointers just to check where your files are.

The function is accessed from the side menu.



You must create a *Project* code by clicking the Project button. This allows you to save a set of account names so that they are available to be selected from the *Account Name or Path* dropdown list.



Use the *Show Q Pointers* and *Show File Pointers* link to flip between the two types.

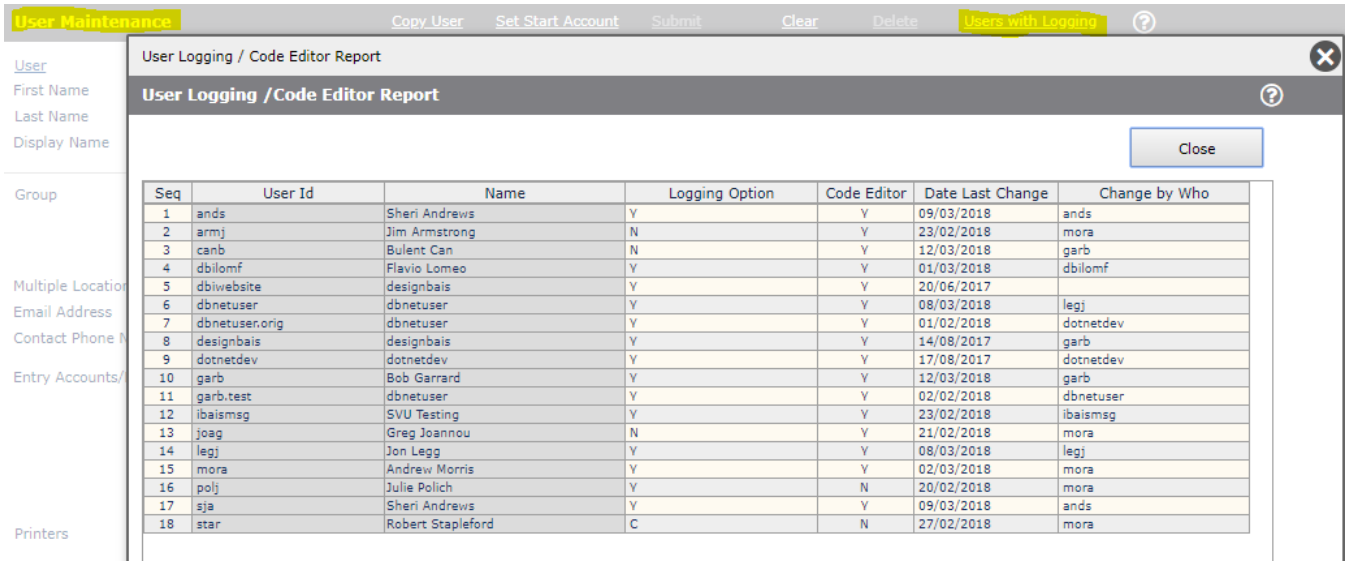
To limit the display to those pointers that are of interest use the *Show Only* text field to enter a string to limit the display to those pointers that contain the string.

Use the *Exclude Pointers* text field to exclude pointers containing the entered string.

You can flag pointers for deletion when *Submit* is clicked. The function identifies some pointers as temporary and automatically sets the delete flag. You can click these to retain the pointers if need be.

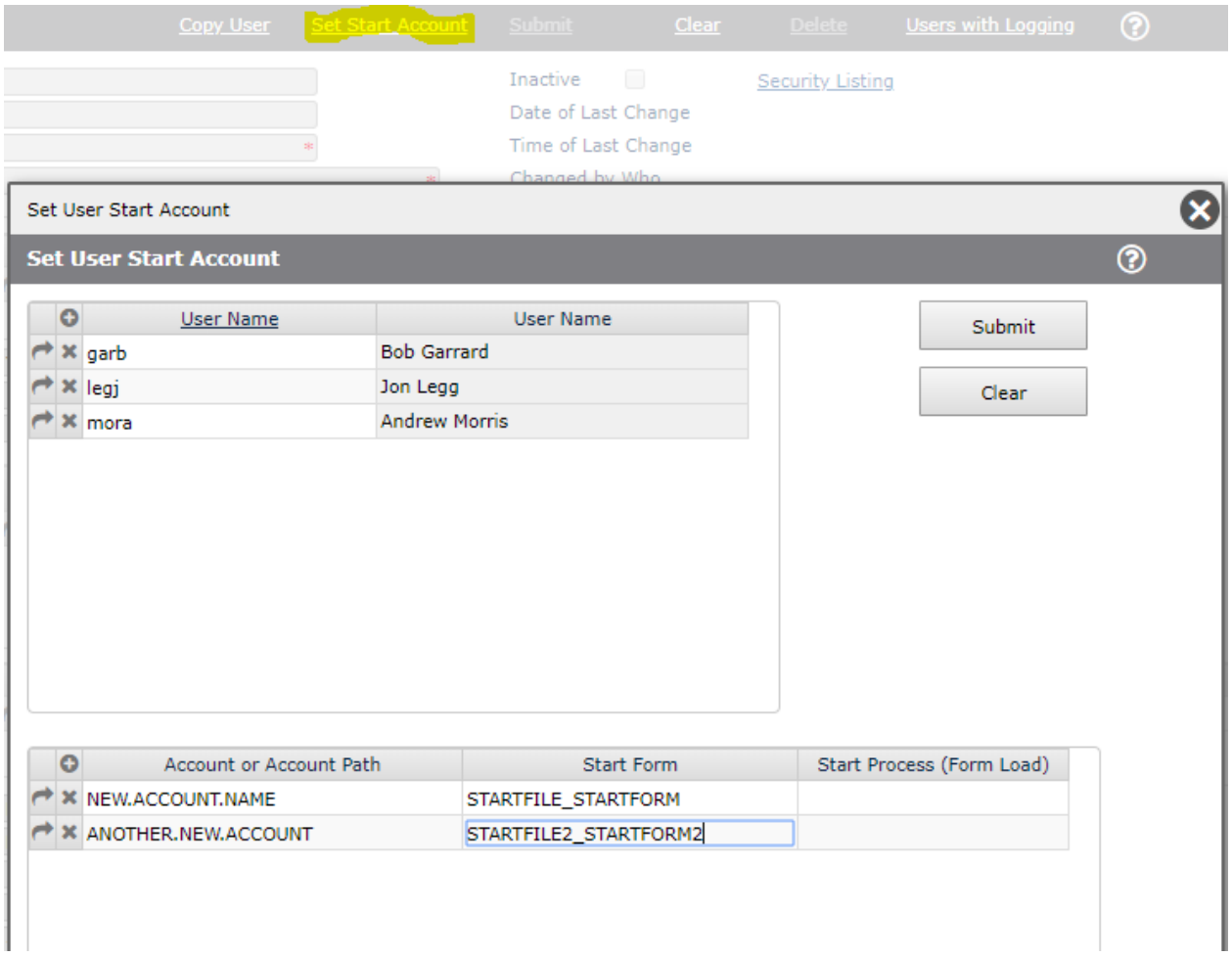
User Maintenance

New report of *Users with Logging* displays the subset of users that have logging turned on or have access to the Code Editor. Click option on User Id and Name takes you to the maintenance form.



Seq	User Id	Name	Logging Option	Code Editor	Date Last Change	Change by Who
1	ands	Sheri Andrews	Y	Y	09/03/2018	ands
2	armj	Jim Armstrong	N	Y	23/02/2018	mora
3	canb	Bulent Can	N	Y	12/03/2018	garb
4	dbilomf	Flavio Lomeo	Y	Y	01/03/2018	dbilomf
5	dbiwebsite	designbais	Y	Y	20/06/2017	
6	dbnetuser	dbnetuser	Y	Y	08/03/2018	legj
7	dbnetuser.orig	dbnetuser	Y	Y	01/02/2018	dotnetdev
8	designbais	designbais	Y	Y	14/08/2017	garb
9	dotnetdev	dotnetdev	Y	Y	17/08/2017	dotnetdev
10	garb	Bob Garrard	Y	Y	12/03/2018	garb
11	garb.test	dbnetuser	Y	Y	02/02/2018	dbnetuser
12	ibaismsg	SVU Testing	Y	Y	23/02/2018	ibaismsg
13	joag	Greg Joannou	N	Y	21/02/2018	mora
14	legj	Jon Legg	Y	Y	08/03/2018	legj
15	mora	Andrew Morris	Y	Y	02/03/2018	mora
16	polj	Julie Polich	Y	N	20/02/2018	mora
17	sja	Sheri Andrews	Y	Y	09/03/2018	ands
18	star	Robert Stapleford	C	N	27/02/2018	mora

New button *Set Start Account* allows you to set up multiple users with multiple accounts in one go. Only enabled if you are not maintaining a user record.



When you are maintaining a user then the *Display Start Account List* button allows you to see all the start accounts and start forms for a user. When there is a long list it is difficult to see if an account is already present in the grid. This report allows for sorting and duplicates can be deleted.

Seq	Start Account or Path	Delete	Start Form	Start Process - Form Load
3	BA.LOGINNET		DBIFORMS_DEVELOP	
4	BA.SOURCE.RELNET		DBIFORMS_DEVELOP	
16	BAIS.GL.LIVE		DBIFORMS_DEVELOP	
7	BAIS.GL.LIVENET		DBIFORMS_DEVELOP	
17	BAIS.LIVE		DBIFORMS_DEVELOP	
6	BAIS.LIVENET		DBIFORMS_DEVELOP	
2	BAIS.NET		SCOPTIONS_MAINMENU	
18	BAIS.PF.GL.LIVE		DBIFORMS_DEVELOP	
8	BAIS.PF.GL.LIVENET		DBIFORMS_DEVELOP	
19	BAIS.PF.LIVE		DBIFORMS_DEVELOP	
9	BAIS.PF.LIVENET		DBIFORMS_DEVELOP	
20	BAIS.SHR.LIVE		DBIFORMS_DEVELOP	
10	BAIS.SHR.LIVENET		DBIFORMS_DEVELOP	
31	DBINET.DEMO		DBCLIENT_DEMO	
1	DBINET.UPG		DBIFORMS_DEVELOP	
5	DBINETLOGIN		DBIFORMS_DEVELOP	
21	DESIGNBAIS.GL.LIVE		DBIFORMS_DEVELOP	
11	DESIGNBAIS.GL.LIVENET		DBIFORMS_DEVELOP	
22	DESIGNBAIS.LIVE		DBIFORMS_DEVELOP	
12	DESIGNBAIS.LIVENET		DBIFORMS_DEVELOP	
23	DESIGNBAIS.PF.GL.LIVE		DBIFORMS_DEVELOP	
13	DESIGNBAIS.PF.GL.LIVENET		DBIFORMS_DEVELOP	
24	DESIGNBAIS.PF.LIVE		DBIFORMS_DEVELOP	
14	DESIGNBAIS.PF.LIVENET		DBIFORMS_DEVELOP	
25	DESIGNBAIS.SHR.LIVE		DBIFORMS_DEVELOP	
15	DESIGNBAIS.SHR.LIVENET		DBIFORMS_DEVELOP	
27	PULSE.GL.RELNET		DBIFORMS_DEVELOP	
30	PULSE.PF.GL.RELNET		DBIFORMS_DEVELOP	
29	PULSE.PF.RELNET		DBIFORMS_DEVELOP	
32	PULSE.REL		DBIFORMS_DEVELOP	
26	PULSE.RELNET		DBIFORMS_DEVELOP	
28	PULSE.SHR.RELNET		DBIFORMS_DEVELOP	

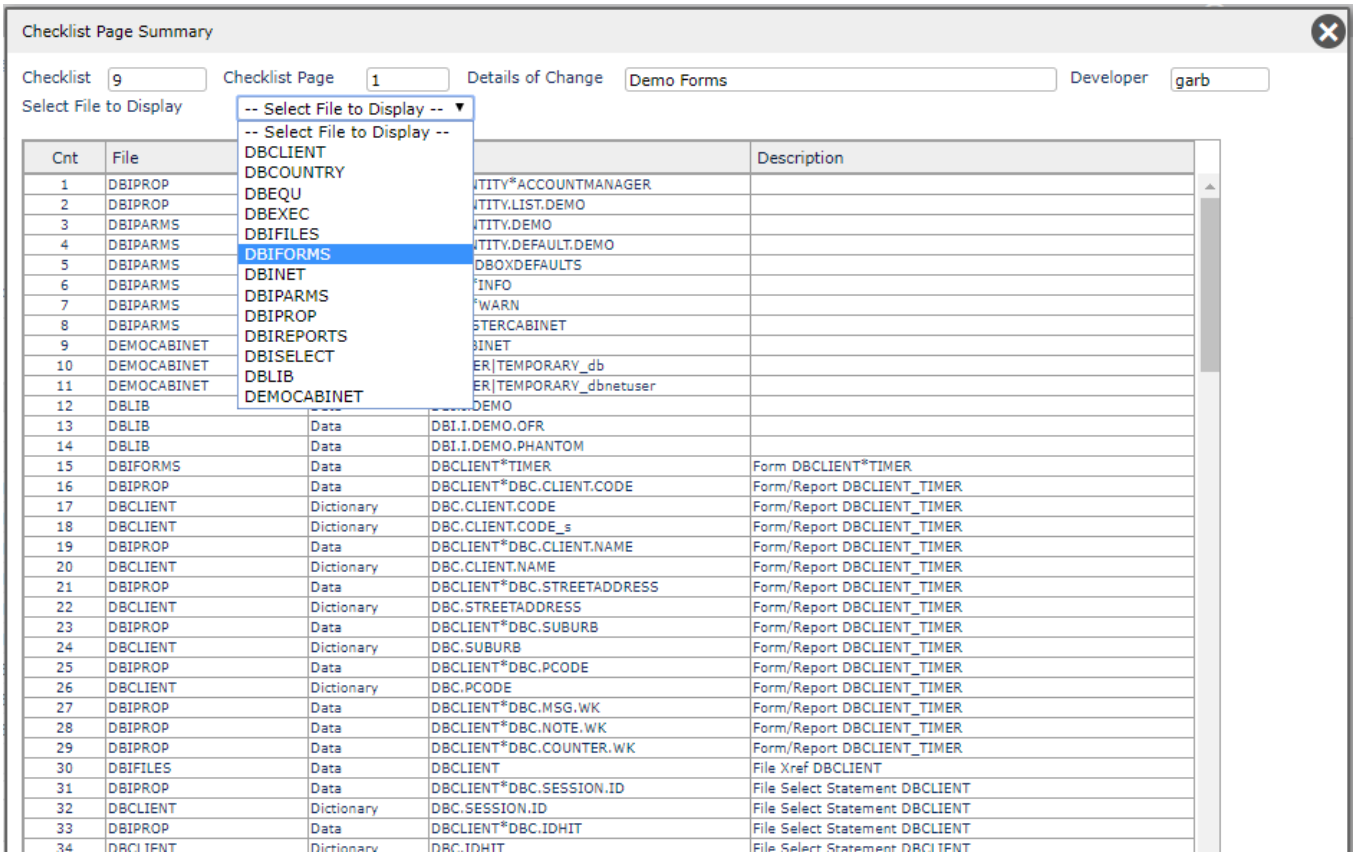
Note that there are now flags on the user file DBIUSERS for:

- Turn Logging On
- Timeout Message
- Code Editor Access
- Show Status Bar
- Log DBDS
- PIN Required
- Print Zoom
- Basic Library Files (for Checklist compile option)

Checklists

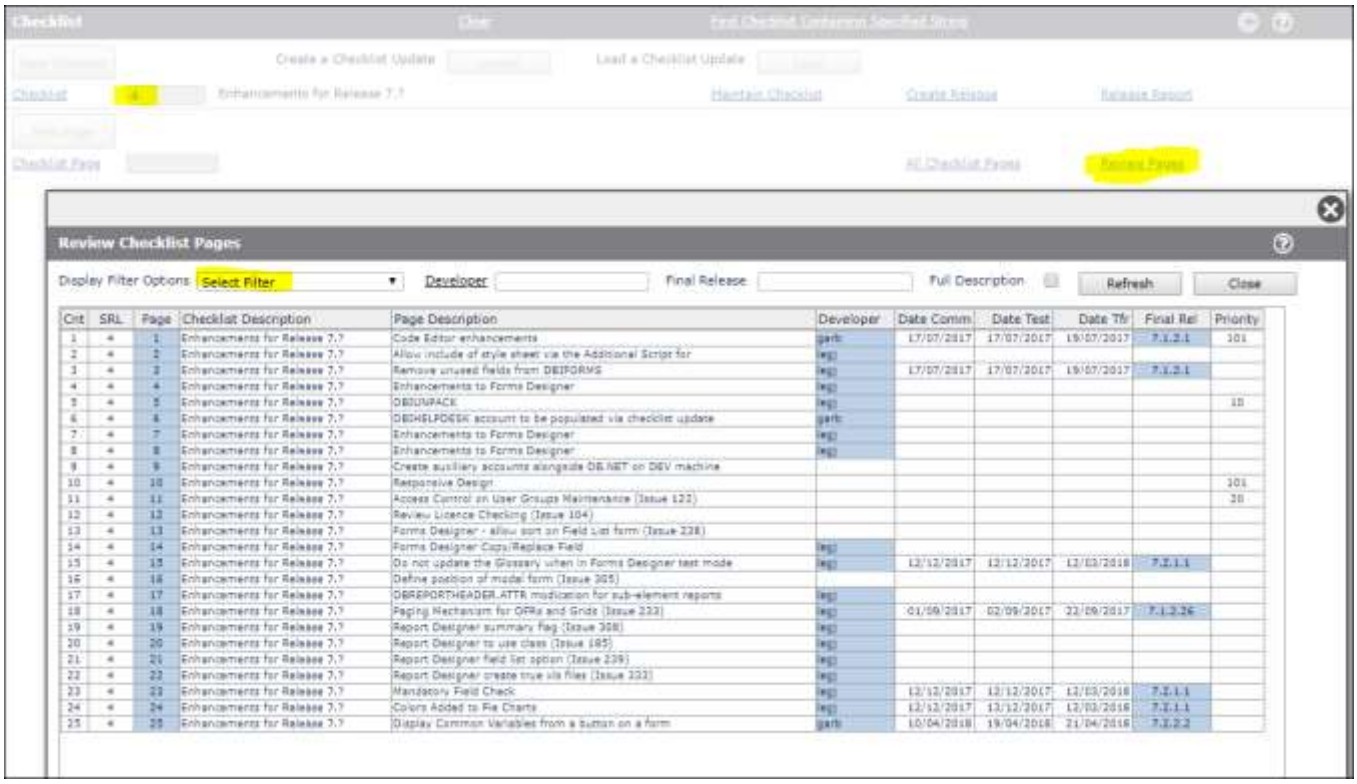
Checklists can now be refreshed. By this is meant that, on entry to a checklist page, any new elements, say new fields on a form that is already in the checklist, are automatically added to the checklist.

There is a *Checklist Page Summary* button to display the items that are on a checklist.

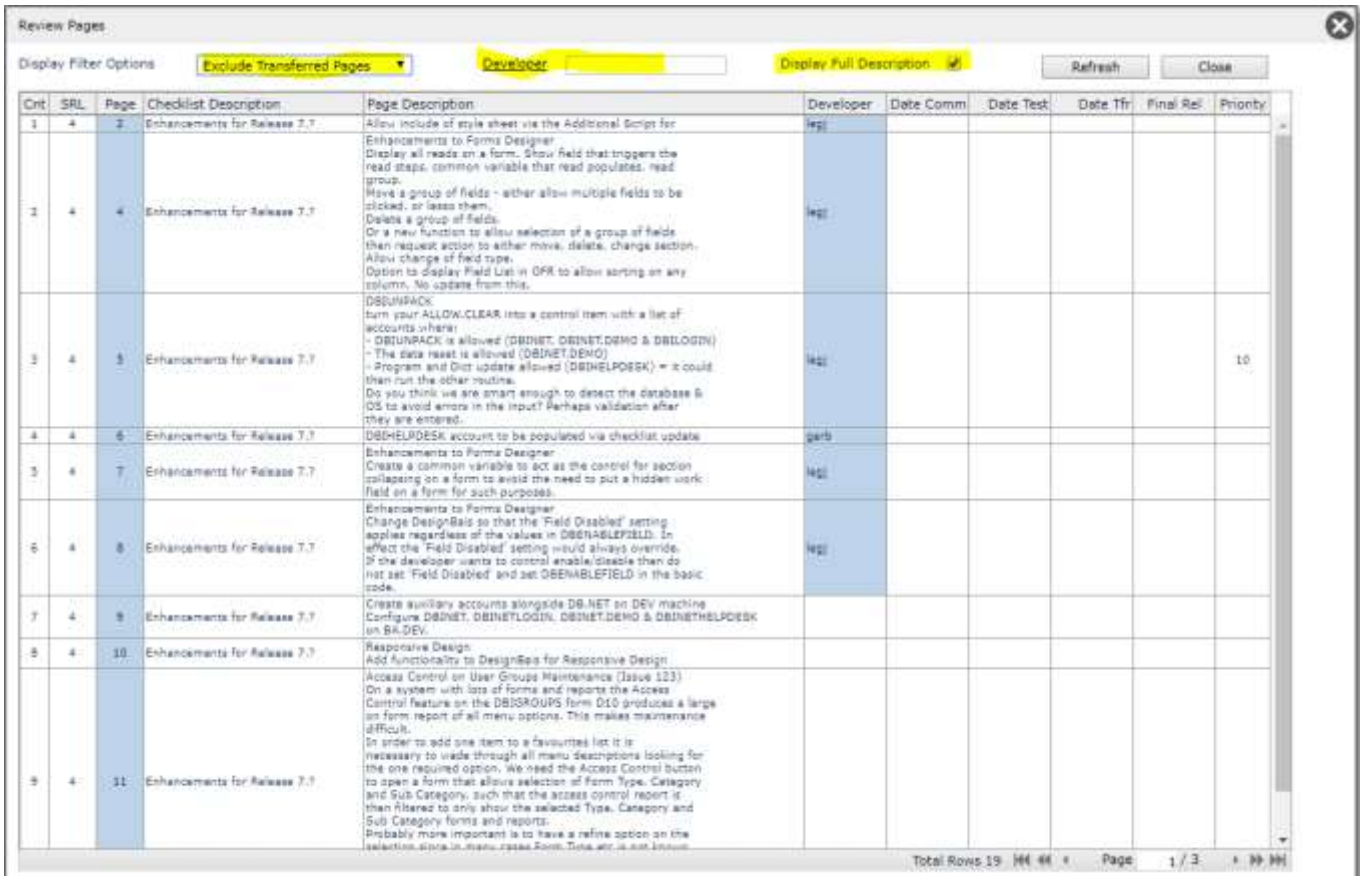


There is a *Create Release* option to allow a set of changes to be marked as belonging to a release, and updated with the release number, with a report.

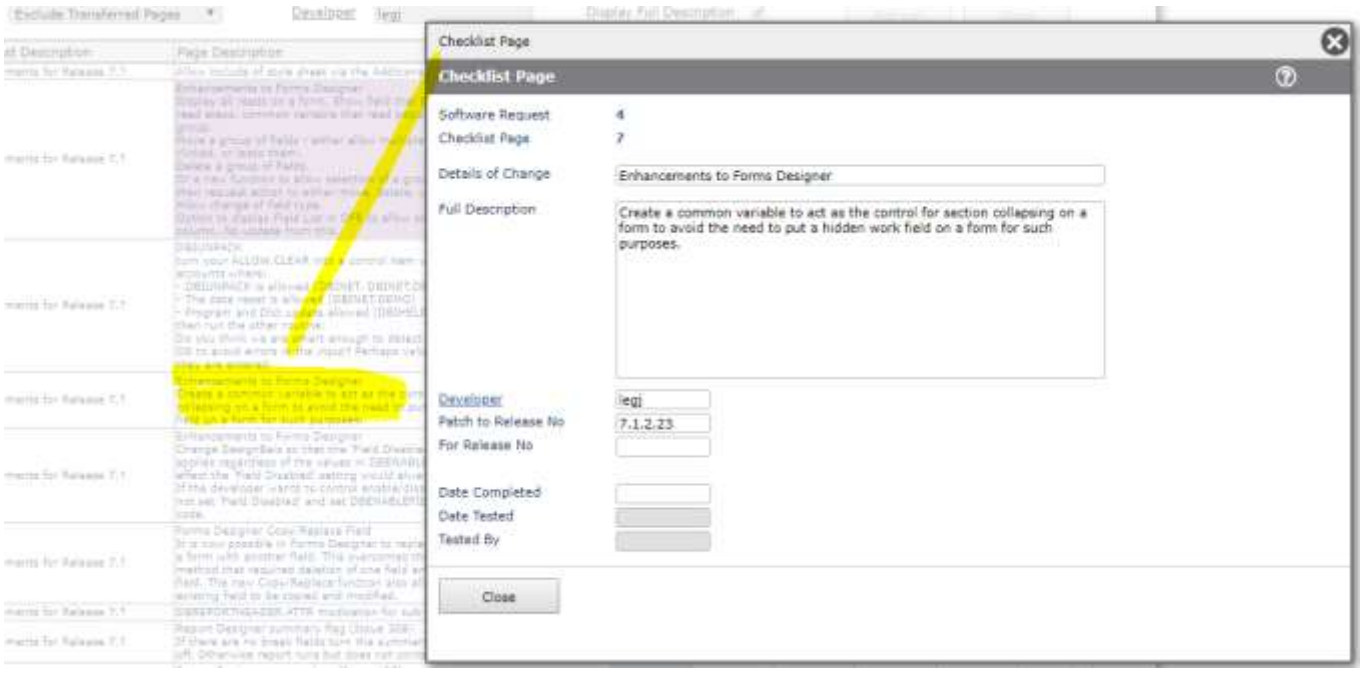
A *Review Pages* option allows you to view all pages within a checklist.



Use the *Select Filter* option to exclude completed tasks. You can also filter the display by *Developer* and you can optionally display the Full Description.

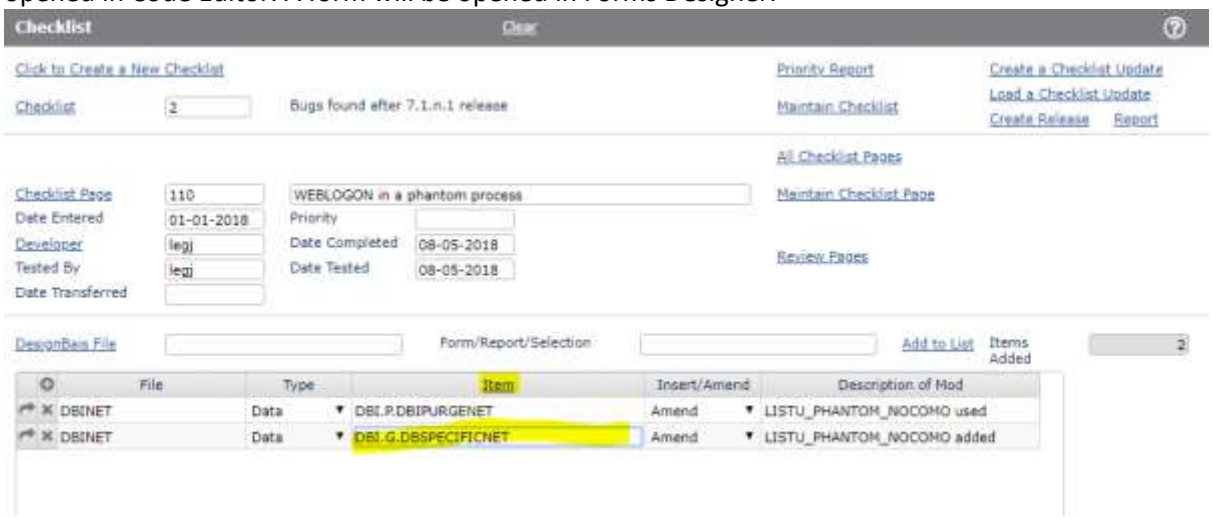


Click the Page or Page Description row to maintain that checklist page.



There is now direct access to the DesignBAIS Tools and Code Editor from *Checklist* form A10

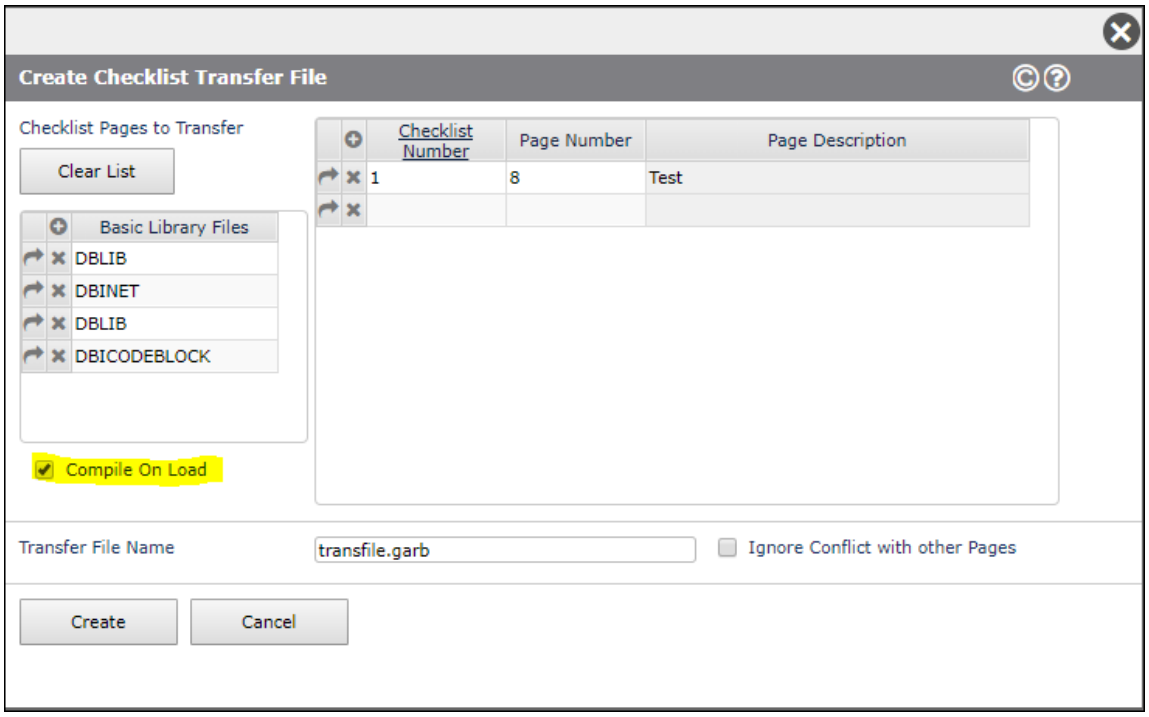
Click in the required row then click the column header process *Item*. A basic subroutine will be opened in Code Editor. A form will be opened in Forms Designer.



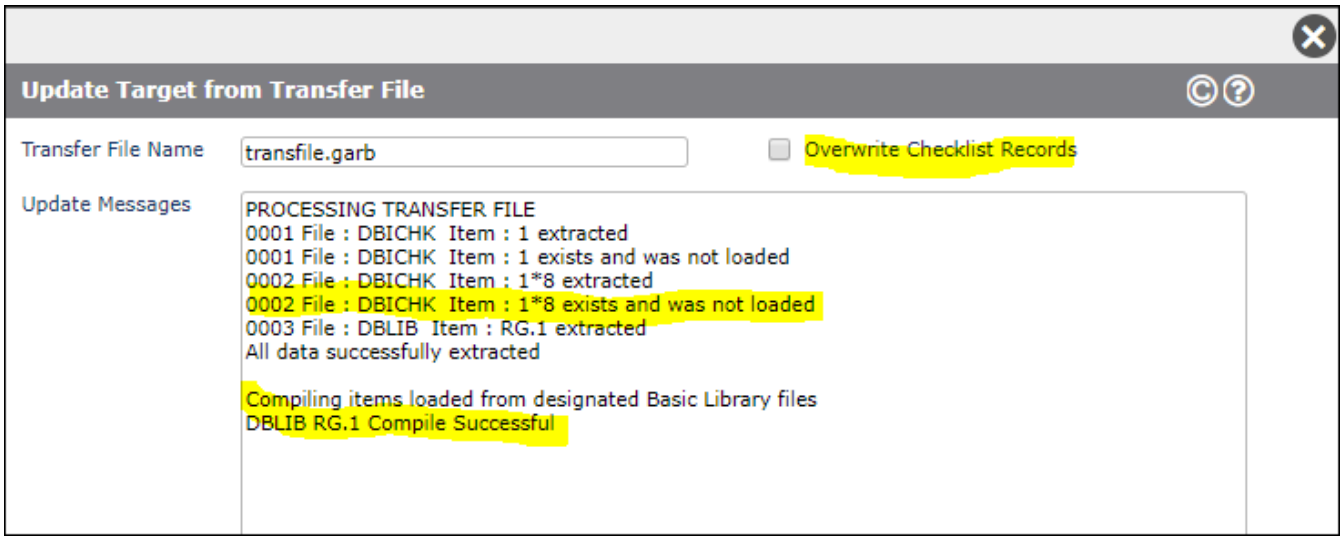
A Checklist update transfer file can now optionally compile basic routines on load.

Users with Code Editor Access can now enter a list of Basic Library Files on their user record.

The list of basic files will be displayed in the *Create Checklist Transfer File* form:

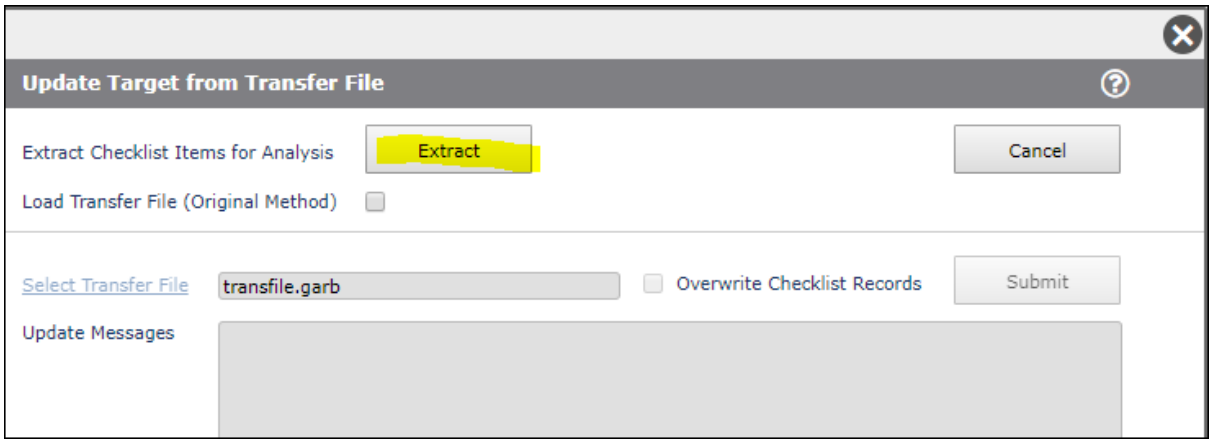


Click the *Compile On Load* checkbox to activate the compile on load function:

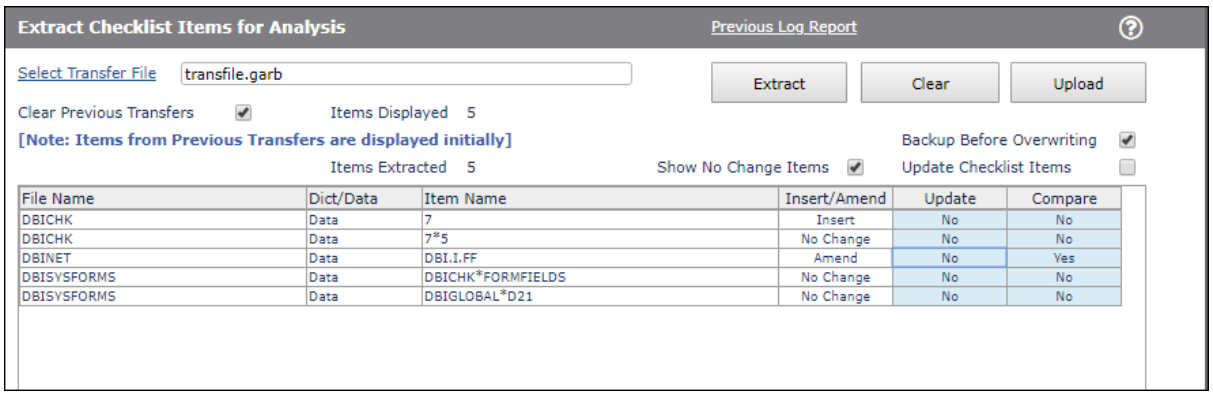


Note also that the Checklist record will only update the target DBICHK file if the *Overwrite Checklist Records* checkbox is clicked when loading the transfer file. In the example above the checklist record 1*8 is extracted but not loaded.

There is a new *Extract Checklist Items for Analysis* feature when loading a checklist transfer file. This feature allows you to review the contents of the transfer file before updating the target environment.



The following form will display:



After selecting the name of the Transfer file (there is now a lookup) click the *Extract* button. Note that prior to clicking *Extract* the contents of the previously loaded and extracted transfer file is displayed. This is to allow you to load multiple transfer files if necessary, prior to extracting the items for analysis. The *Clear Previous Transfers* checkbox is ticked by default and therefore previous transfers are removed when you click *Extract*.

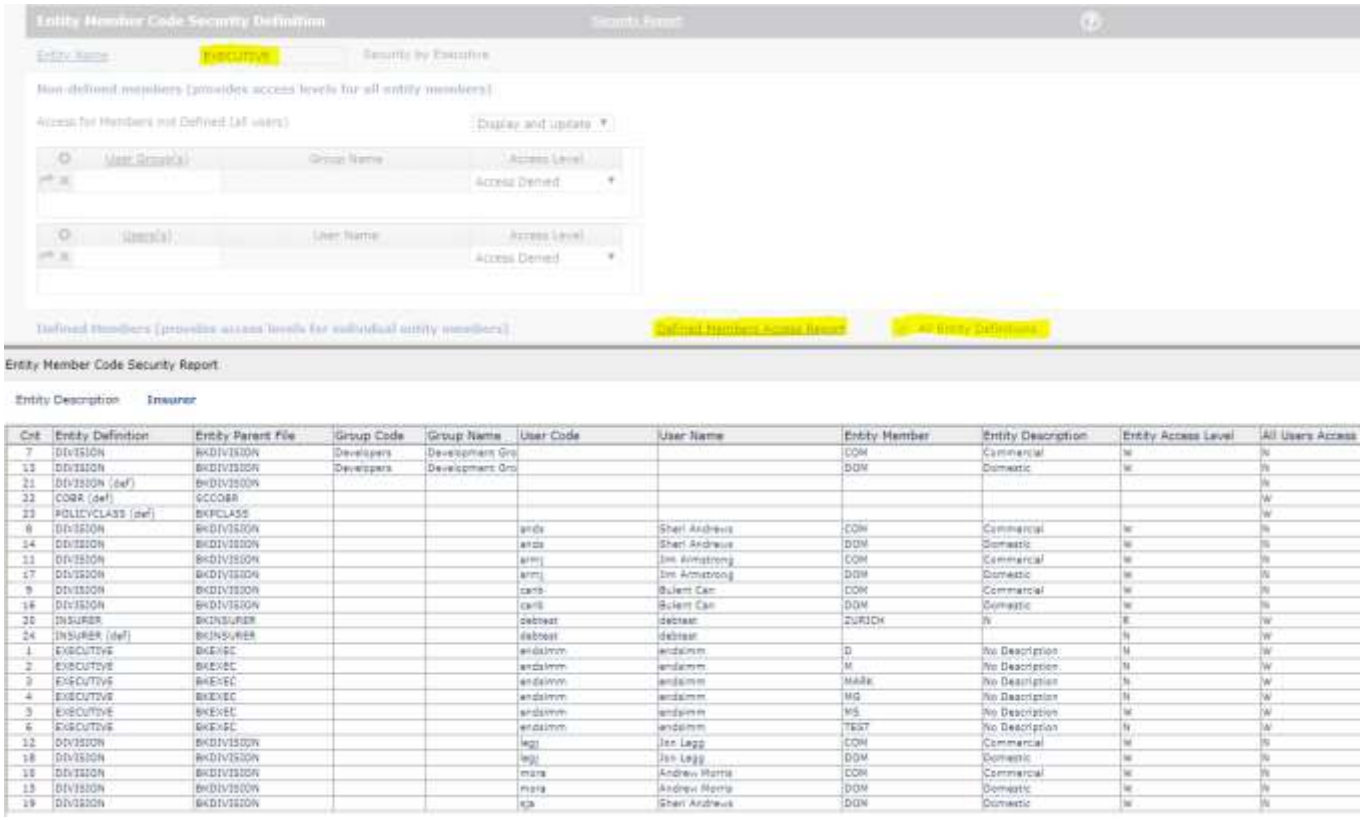
The *Show No Change Items* checkbox controls whether extracted items from the transfer file that match the same item in the target environment are displayed. Normally you do not need to see these since the *Install* procedure will only install items that are flagged with *Update* set to *Yes* and matching items will have *Update* set to *No* (it is not necessary to update an item with a matching item from the transfer file!).

You can compare an item before installing it if required. Click the required row in the *Compare* column.

The *Backup Before Overwriting* checkbox is on by default. When set the target items in the target environment are backed up to DBIBACKUP prior to being overwritten with the item from the transfer file.

Entity Security

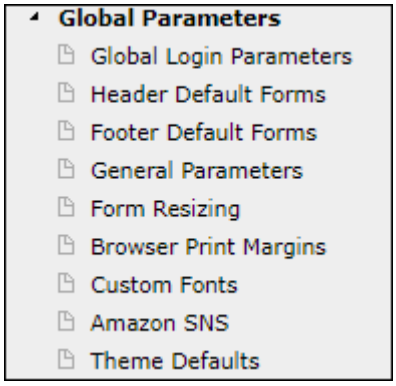
There is a new *Defined Members Access Report* which displays all entity security settings for one or all Entity Security definitions. The on-form report can be sorted on any column allowing you to review the access restrictions for any or all users in one go.



Entity Member Code Security Report

Entity Description: Insurer

Cnt	Entity Definition	Entity Parent File	Group Code	Group Name	User Code	User Name	Entity Member	Entity Description	Entity Access Level	All Users Access
7	DIVISION	BKDIVISION	Developers	Development Gro			COM	Commercial	W	N
13	DIVISION	BKDIVISION	Developers	Development Gro			DOM	Domestic	W	N
21	DIVISION (def)	BKDIVISION								N
22	CORR (def)	SCCOBR								W
23	POLICYCLASS (def)	BKPCLASS								W
8	DIVISION	BKDIVISION			andy	Sheri Andrews	COM	Commercial	W	N
14	DIVISION	BKDIVISION			andy	Sheri Andrews	DOM	Domestic	W	N
11	DIVISION	BKDIVISION			armj	Jim Armstrong	COM	Commercial	W	N
17	DIVISION	BKDIVISION			armj	Jim Armstrong	DOM	Domestic	W	N
9	DIVISION	BKDIVISION			cank	Bulent Can	COM	Commercial	W	N
16	DIVISION	BKDIVISION			cank	Bulent Can	DOM	Domestic	W	N
20	INSURER	BKINSURER			debsat	debsat	ZURICH	N	E	W
24	INSURER (def)	BKINSURER			debsat	debsat			N	W
1	EXECUTIVE	BKEXEC			andsimm	andsimm	D	No Description	N	W
2	EXECUTIVE	BKEXEC			andsimm	andsimm	M	No Description	N	W
3	EXECUTIVE	BKEXEC			andsimm	andsimm	MARK	No Description	N	W
4	EXECUTIVE	BKEXEC			andsimm	andsimm	MG	No Description	N	W
5	EXECUTIVE	BKEXEC			andsimm	andsimm	MS	No Description	W	W
6	EXECUTIVE	BKEXEC			andsimm	andsimm	TEST	No Description	N	W
12	DIVISION	BKDIVISION			legj	Jon Legg	COM	Commercial	W	N
18	DIVISION	BKDIVISION			legj	Jon Legg	DOM	Domestic	W	N
10	DIVISION	BKDIVISION			morra	Andrew Morris	COM	Commercial	W	N
15	DIVISION	BKDIVISION			morra	Andrew Morris	DOM	Domestic	W	N
19	DIVISION	BKDIVISION			sja	Sheri Andrews	DOM	Domestic	W	N

Global Parameters

All parameters are now maintained via forms accessed from the *Global Parameters* option on the Tools Menu.