
Forms Value-upgrade

— Modern and Intuitive —

Stroe Laura Ariadna

SoftBASE A/S

- Engineer
 - Graduated from Denmark Technical University in 2016, MSc in Efficient and Intelligent Software
 - Worked with Forms for 4 years
-

Project Description

The project consisted of cca 1400 Forms of different shape and complexity, using the SoftBASE [Value-upgrade](#).

The project in itself was voluminous and as expected, there were various challenges on the way but nonetheless all major milestones are met and important obstacles removed.

Among the challenges, notable are the lack of standardization across the application.

Project Description - Value-upgrade Process

As seen on the website, there are standard features which are added as part of the default value upgrade, but there are also optional features from which the client chooses.

For this project, the optional features were APEX linking and role based navigation.

The screenshot shows the 'Forms Value-upgrade' page on the SoftBASE website. The page is divided into several sections:

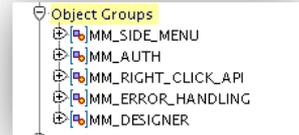
- Standard Features:**
 - About adding value:** All GUI, look & feel, components are built using Native Forms Components – no Pluggable Java Components. This secures that your legacy application stays compatible with future releases.
 - GUI – look & feel:** Look & feel derived from APEX, consistent with the general GUI's in a Web 2.0 environment and fit for large screens, High Definition – and zoom level set per user.
 - Integration components:** Five productivity components for: Host command, Web command, Gdrive link, APEX linking, Form to Form.
 - Workflow navigation:** Role based functionality for workflow, dynamic menus, dynamic layout and Context guided data entry.
 - Screen reporting:** Writing to comma separated files and HTML files.
- Optional Features:**
 - ... The process:**
 - Adjustment of Authentication / Authorization
 - Add Object group to every Form
 - Attach a Library to every Forms
 - Review regarding trigger conflicts
 - Automatic resizing of all Forms
 - ... The options:**
 - APEX linking
 - Role based navigation
 - On line Designer
 - Interface module
 - Language module

The page also includes a navigation menu at the top with links like 'SaaS Concept', 'Application Solutions', 'SaaS Terms', 'Oracle Technology', 'NEWS archive', and 'About us'. A central image shows a screenshot of the application interface with a table of data.

The Value-upgrade Features

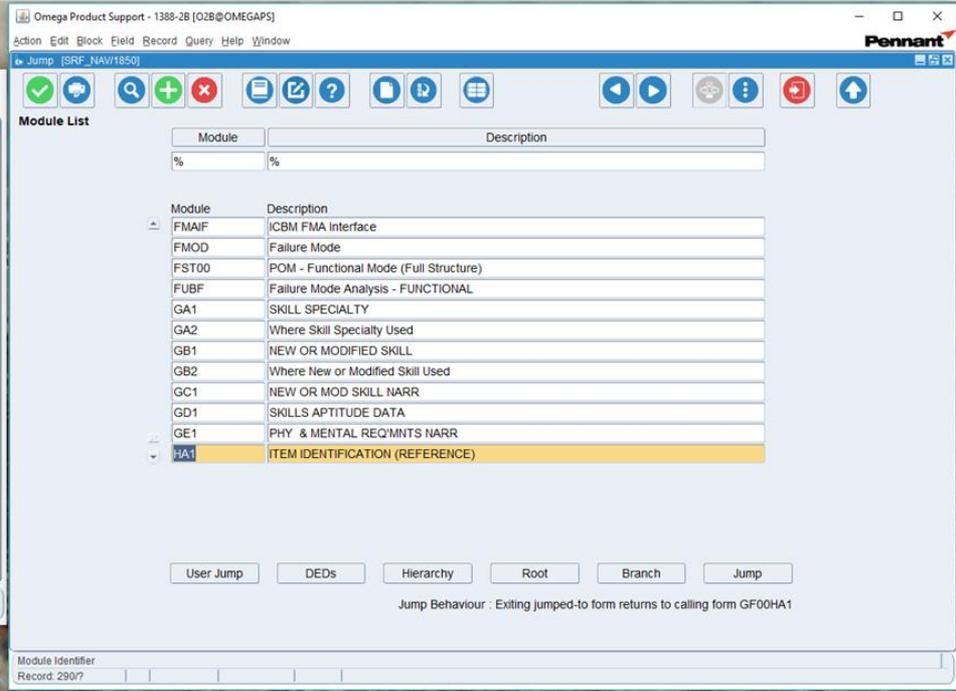
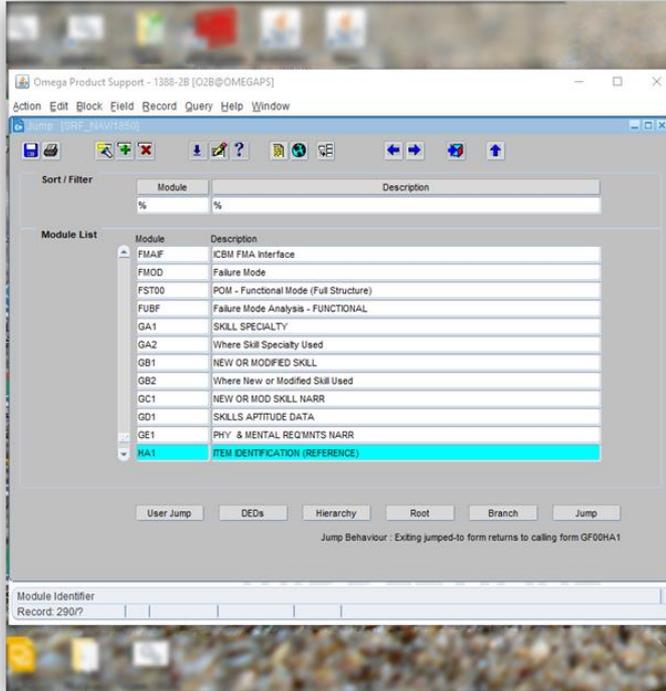
The features, both the standard ones and the optional ones are independent of each other therefore they can be added individually and they will work no matter the combination in which they are found.

The individualism is given by the fact that each feature has its objects in an object library dedicated to itself, and the code in libraries. This makes it easier to maintain, since there is only one source of truth.



All these features improve productivity by either making the system easier to use, more intuitive, faster or by offering a whole new way of doing something that otherwise required extra steps or even extra applications opened (ex. Google Integration or Apex Deep Linking offer pinpointed navigation to certain files/documents in applications outside of Forms, no manual opening of the application and searching for the file/document yourself).

Before and After



Value-upgrade - Look and Feel

- We automatically resize all Forms to a size suitable for HD screens, no matter the coordinate system in which the Forms are (inch, point, pixel).
- At this point, the sizes of items, scrollbars, buttons and toolbar, width/height or both become standardized across the application.
- The width of scrollbars becomes standard while the height is calculated according to the number of items and set.
- The toolbar buttons are resized to be able to fit 32x32 images.
- There is a balance between the icon colors and the Oracle Look and Feel color scheme.
- The round icons, in their simplicity, are easy on the eyes and quite intuitive.



Value-upgrade - Look and Feel

- The colors use the Oracle Look and Feel definitions of OLAF1, OLAF2, OLAF3, OLAFVeryLight , OLAFVeryDark and so on and are added to the Visual Attributes from where are used across the application, according to context.
- The Fonts used are the generic ones and are defined in the Enterprise Manager under “Fonts and Icon Mappings”.

Item Identification

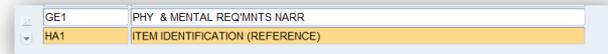
CAGE	Ref. No.	Name	Name Code
11599	109661	SHAFT,MAIN BEARING	
11599	109661X	SHAFT,MAIN DRIVE	
11599	109661Y	SHAFT,MAIN DRIVE	
11599	109796	COVER, SHAFT SEAL	

Item Identification

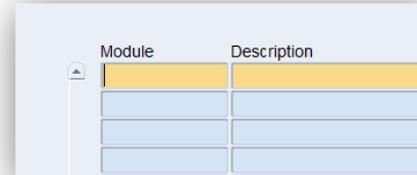
CAGE	Ref. No.	Name	Name Code
11599	109661	SHAFT,MAIN BEARING	
11599	109661X	SHAFT,MAIN DRIVE	
11599	109661Y	SHAFT,MAIN DRIVE	
11599	109796	COVER, SHAFT SEAL	

Value-upgrade - Look and Feel

- The purpose of the colors is not only visual aesthetics but also helps differentiating a type of field from another type of field.
 - Mandatory items from non mandatory items - helps avoiding error/notification messages which interrupt the fluid working process.
 - Description items and/or disabled items - give information regarding the data contained in the field from the start.
 - Current Record visual attribute - easy to follow.
 - Enter/Execute Query - makes it obvious that you are in a query mode and makes it possible to see which items are querriable.



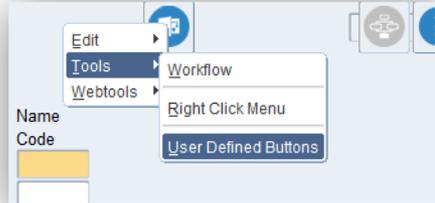
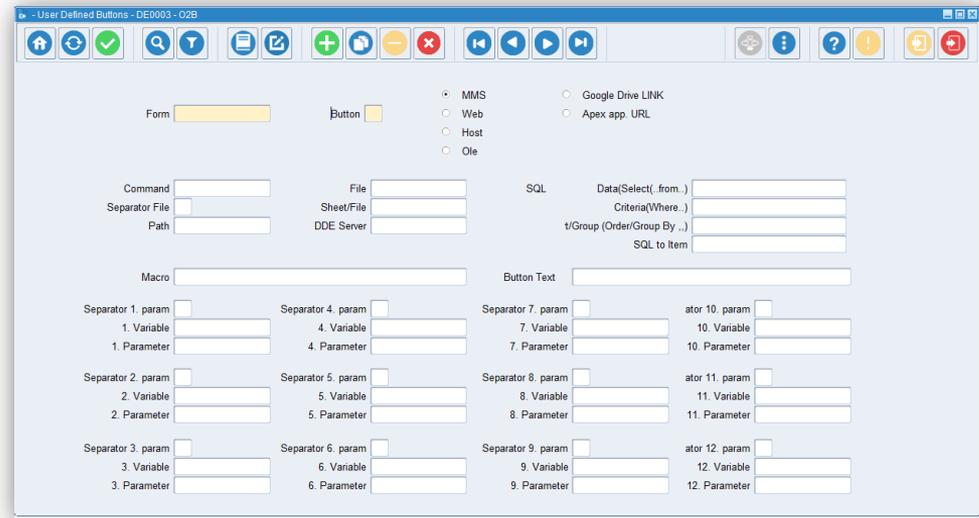
GE1	PHY & MENTAL REQ'MNTS NARR
HA1	ITEM IDENTIFICATION (REFERENCE)



Module	Description

VU - Standard Features - Right Click API

- The Right Click API is a right-click menu which is defined in a form and can be accessed by right clicking anywhere on the canvas .

Form Button

 MMS
 Web
 Host
 Ole

 Google Drive LINK
 Apex app URL

Command File SQL Data(Select(. from .))
 Separator File Sheet/File Criteria(Where..)
 Path DDE Server t/Group (Order/Group By ..)
 SQL to Item

Macro Button Text

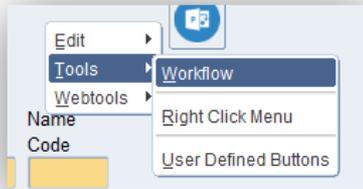
Separator 1. param <input type="text"/>	Separator 4. param <input type="text"/>	Separator 7. param <input type="text"/>	ator 10. param <input type="text"/>
1. Variable <input type="text"/>	4. Variable <input type="text"/>	7. Variable <input type="text"/>	10. Variable <input type="text"/>
1. Parameter <input type="text"/>	4. Parameter <input type="text"/>	7. Parameter <input type="text"/>	10. Parameter <input type="text"/>
Separator 2. param <input type="text"/>	Separator 5. param <input type="text"/>	Separator 8. param <input type="text"/>	ator 11. param <input type="text"/>
2. Variable <input type="text"/>	5. Variable <input type="text"/>	8. Variable <input type="text"/>	11. Variable <input type="text"/>
2. Parameter <input type="text"/>	5. Parameter <input type="text"/>	8. Parameter <input type="text"/>	11. Parameter <input type="text"/>
Separator 3. param <input type="text"/>	Separator 6. param <input type="text"/>	Separator 9. param <input type="text"/>	ator 12. param <input type="text"/>
3. Variable <input type="text"/>	6. Variable <input type="text"/>	9. Variable <input type="text"/>	12. Variable <input type="text"/>
3. Parameter <input type="text"/>	6. Parameter <input type="text"/>	9. Parameter <input type="text"/>	12. Parameter <input type="text"/>

VU - Standard Features - Right Click API

- Via the Right Click API you can navigate quickly between Forms but can also can define host commands and URLs with what parameters you need, and easily navigate to those.
- The Right Click API feature, as a whole, consists of a form, a few tables and procedures, and a set of popup menus which are put in an object library, and added to the toolbar and the canvas.
- All the objects needed for this functionality are native.

Value-upgrade - Standard Features - Workflow

- The workflow navigation feature, like the Right Click API, consists of a form in which you define where to navigate to from where. This form is accessed via the Right Click Menu on the Toolbar.



A screenshot of a software form titled 'PAGE_1'. The form contains several input fields and a list box. The fields are: 'Form' (GF00HA1), 'Block' (LDMHA), 'Field' (MM1), and 'Batch'. Below these is a 'Group' dropdown menu set to 'SEC.ADMIN'. Underneath the group menu is a list of fields: 'Dest. Program', 'Dest. Block', 'Dest. Field', 'Help/Hint Text', and 'Description'. The 'Help/Hint Text' field is highlighted in yellow. The form is accessed via the Right Click Menu on the Toolbar.

Value-upgrade - Standard Features - Workflow

- After the workflow is defined (per form per user group), the workflow button on the toolbar becomes active and pressing it will navigate the user to the form/block/item defined previously.
-  Becomes 
- The Workflow feature consists of a form which is given, a few tables, procedures which are added to the library and the button on the toolbar.
- All the Objects are native.

VU - Standard Features - Screen Reporting

The screenshot displays two windows from the VU software interface. The left window, titled 'Jump: [SRF_NAV1850]', shows a 'Module List' table with columns for 'Module' and 'Description'. The right window, titled 'SRF_NAV.csv - LibreOffice Calc', shows a spreadsheet with columns 'A' and 'B' containing the same data as the 'Module List' window.

Module List Window:

Module	Description
00CHANGE	Change Sign in
00FODA1	FORM SOURCE
00HASRNO	PART SERIAL NO
00IPDE	IPPF Load errors
00IPPC	IPPC Load errors
00IPPD	IPPD Load errors
00IPPI	IPPI Load errors
00IPPP	IPPP Load errors
00PL1	PHRASE LIBRARY
00WHERE	VIEW WHERE PART USED
12IPB	IPD Assembly
12LOG	Publisher Assembly Log Viewer

SRF_NAV.csv - LibreOffice Calc Window:

	A	B
1	Module	Description
2	00CHANGE	Change Sign in
3	00FODA1	FORM SOURCE
4	00HASRNO	PART SERIAL NO
5	00IPDE	IPPF Load errors
6	00IPPC	IPPC Load errors
7	00IPPD	IPPD Load errors
8	00IPPI	IPPI Load errors
9	00IPPP	IPPP Load errors
10	00PL1	PHRASE LIBRARY
11	00WHERE	VIEW WHERE PART USED
12	12IPB	IPD Assembly
13	12LOG	Publisher Assembly Log Viewer
14	21BF2	RCM Analysis (ADF2)
15	21BF2	RCM Task Requirement
16	21CMAC1	CM Audit Control
17	38CIL	Configuration Item Library
18	38CIM	Configuration Item Mapping
19	38WSR	LISS Work Sheet Report
20	42ILLUST1	Illustrations
21	42IPPD1	IPD-ICN Defaults
22	42VIPD1	Illustrated Parts Documents

VU - Standard Features - Screen Reporting

- There is a form in which you can define which block with which column you wish to be taken into consideration when writing to the file.
- To write to .csv or .html, or select the block and columns to write, right click on the toolbar and go to the desired entry in the webtools menu.
- The form and the code required is given as part of the installation.
- The screen reporting is possible on both single blocks and master detail blocks.
- This feature comes with 3 entries in the menu, and a few procedures in the library.

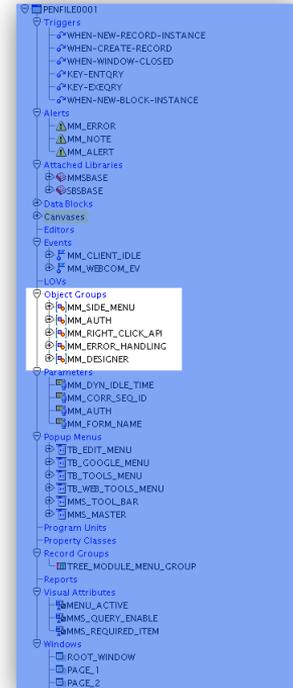


The Process - Continuous Access During Project

- Initially we offered a Linux server as a delivery/test server accessible by the client.
- The Client is Windows only therefore we prepared a Windows 12 server for them which they can access via VPN and at any time see and test the progress of the project.

The Process - Preparation - 2 .fmb files

- All the objects needed for the features were gathered in one .fmb file which will be given to the client. This step is simplified by the fact that each feature is already part of its own object library, thus this step only entails putting the relevant ones in the .fmb the client receives.
- Another .fmb file was made for the specific triggers or visual attributes the SoftBASE Forms need to run, if these can't be mixed with the others.



The Process - Preparation - 2 .pll files

- The procedures needed for the features to run smoothly are added to 2 libraries:
 - A library for the client's specific objects
 - A library for the Value-upgrade Forms (example the Value-upgrade Forms have their own toolbar, thus the functionality for that toolbar is found in this second library, because it is not related to the client's toolbar).

The Process - Preparation - All objects are created

- The tables which are needed for the SoftBASE Forms are created as views from the tables already existing on the database where possible.
- In the case of the user access tables, user groups and roles tables, for the Value upgrade features to work we just make views over existing similar tables tables, structuring and naming them however necessary.
- Other objects are freshly created (for example context menu table).

Process - Resizing - automated process

- When everything is ready to be added to the projects, the Forms have to be prepared.
- The resizing of the toolbars, canvases, items etc. is done automatically via several batches.
- The Forms are now ready to receive the features.

Process - Adding the Objects

- After the objects are gathered and adapted, there is only a matter of adding them where necessary. For example, the toolbar features will not be added to dialog pages where there is no toolbar.
- In the case of this project, there was a number of master canvases from which the Forms took the toolbars and visual attributes, therefore the objects are added to these toolbars, and they will be inherited by the Forms using these toolbars, they only need to have the object libraries added to them too.

Process - Refining

- The automated resizing process does its best to stay loyal to the original form arrangement of items and what not, but sometimes , after resizing, that's just not good enough anymore, because with a bigger screen, a bigger canvas, the items arrangement can be improved.
- This is the case for spread tables for example:

Results

- The project was split into 3 major steps:
 - Value-upgrade Preparation - gathering and adding all objects and Forms.
 - Upgrading Process - automated process done on Forms and canvases by the bulk.
 - Refining - per individual form.
- Following these 3 steps, in the end it amounted to 15-20 minutes per form.
- This was a big project, with a high complexity given by the lack of standardization and the high number of master canvas Forms, each bringing something new. The standardization was also done by SoftBASE as part of the Value-upgrade.

Results

- The requirements were clear, the code was not to be thrown out and replaced, no patching in code here and there just because.
- Everything was to be integrated in the product, resulting in the same familiar product but with added value.
- The product in its final form would work as before , have everything as before plus all the new features which must be in harmony with their product - obviously!

Results

- It is important that the old functionality does not change, that everything still works as it worked before (if not even better).
- The new functionality brings added value to the user experience which increases overall performance. With so many features at the press of a right click or a button, a user can quickly navigate from one place to another or print csv/html files making everyone's job easier.
- For a more detailed description of the features visit <https://www.softbase.dk/forms/>