

User Interface - Feature #3260

auto-conversion and enablement for the embedded web client

03/07/2017 12:55 PM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to User Interface - Feature #3259: add useful web components that can...		New	
Related to User Interface - Feature #3258: improve javascript upcalls from th...		New	
Related to User Interface - Feature #3218: enhance the embedded mode web GUI ...		Closed	
Related to User Interface - Feature #3261: enhanced browse that can optionall...		Closed	
Related to User Interface - Feature #3331: translate 4GL menu widget processi...		Closed	

History

#1 - 03/07/2017 01:08 PM - Greg Shah

Our current approach for embedded mode is that to leverage the converted ABL screens, we must write some driver ABL code to provide a callable interface for javascript upcalls.

In the hotel demo application (and others we have done), this has been manually written as edriver.p.

The question here is how much of this code could potentially be automatically generated across a large scale conversion project? Does the use of ADM/ADM2 provide a level of consistency that could be leveraged to create conversion rules to ease this effort, given some assumptions about how a set of screens are related to one another?

Constantin has provided these thoughts:

Actually, the initialization of the screens is somehow common :

use RUN init-object on one or more 4GL programs to build one or more frames:

```
RUN init-object IN THIS-PROCEDURE (
    INPUT 'adm/objects/p-navico.r':U ,
    INPUT FRAME F-Main:HANDLE ,
    INPUT 'Edge-Pixels = 2, SmartPanelType = NAV-ICON,
    Right-to-Left = First-On-Left':U ,
    OUTPUT h_p-navico ).
```

RUN set-position and RUN set-size to set its location size

```
RUN set-position IN h_p-navico ( 23.91 , 15.00 ) NO-ERROR.
RUN set-size IN h_p-navico ( 1.67 , 38.00 ) NO-ERROR.
```

RUN add-link to link ADM objects with each-other (this gets as parameter a handle obtained from RUN init-object).

```
RUN add-link IN adm-broker-hdl ( h_cust , 'Record':U , h_cust-2 ).
```

The second parameter I think maps to ADM methods (like send-records for record), which need to be collected from the window program, too.

All this code is in adm-create-objects, where there is a big CASE statement, with initialization code for each tab page.

A problem is we need to distinguish between frames built via RUN init-object which reside in the FRAME associated with the 'tab pages' and other options: this can be done if we limit the RUN init-object to calls which have a INPUT FRAME F-Main:HANDLE as second parameter (but there are exceptions to this case, there is a case where the frame, although shown in the tab, is parented by the window and not the frame as the other 'sibling screen frames').

after we determine which RUN init-object statements are in use, we can collect their output handles and gather the other code used with them we need a smart way to solve circular dependencies between pages (in our example with cust maintenance window, page 1 and 2 are inter-dependent and they both need to be initialized before executing RUN add-link).
coordinates passed to RUN set-position need to be translated to (1,1) (considering the left-most and top-most coordinates as 1 and adjusting all others based on these)

The above is somehow the easy part, which I think can be done via rules. The hard part is extracting enough ADM initialization code and other business-logic from the window program (in our case windows/<prog>.w) and other 'system' programs (like menuing) so that the code can work. But I think this can be done as a common framework for the embedded app, written by hand, and re-used by all these new generated 4GL code.

#2 - 03/07/2017 01:13 PM - Greg Shah

But I think this can be done as a common framework for the embedded app, written by hand, and re-used by all these new generated 4GL code.

One key question: how much of this can be written once and leveraged across multiple applications? We can modify the Possenet ADM/ADM2 code to standardize approaches and implement whatever hooks might be needed. This code could be converted and be part of the FWD distribution along with the helper code that matches up. Then the per-application effort might be making sure that the application screens are fit into the customized ADM/ADM2 framework.

#3 - 03/07/2017 01:18 PM - Greg Shah

Another idea we are considering is auto-converting standard ABL menus into a modern web client version. Eric is experimenting with some alternatives for the web component. The actual automation of the conversion is not hard. The trickiest part is any integration with automatically enabled legacy ADM/ADM2 screens in embedded mode. How much of the processing should map to the ABL menu widget processing and how much should be mapped to something else?

#4 - 03/07/2017 01:19 PM - Greg Shah

- Related to Feature #3259: add useful web components that can be easily integrated into the embedded web client added

#5 - 03/07/2017 01:19 PM - Greg Shah

- Related to Feature #3258: improve javascript upcalls from the embedded client added

#6 - 03/07/2017 01:19 PM - Greg Shah

- Related to Feature #3218: enhance the embedded mode web GUI to implement a good default approach for common cases added

#7 - 03/07/2017 01:25 PM - Greg Shah

- Related to Feature #3261: enhanced browse that can optionally selected as a replacement for the default ABL browse added

#8 - 07/18/2017 05:51 PM - Constantin Asofiei

A status for the automation of embedded windows.

The solution I found to automate the generation of embedded windows and their associated pages (tabs) is first to collect all the windows and their pages, and after this generate a Progress AST with associated code to switch between the pages - each ADM/ADM2 window will have a new Java program generated, in some package like com.foo.app.embedded. These ASTs will be generated between front and middle phases (using the front phase ASTs as source), and will be inserted into the conversion source list so that the conversion process will pick them up and convert them to Java programs. Beside these programs, it will generate a registry (XML or JSON) with all the windows and the pages, which will be imported by javascript and automate the UI.

Behind this, at runtime the embedded mode will use a 'driver' which will be composed of these parts:

1. a common part (via an include file I think and some 4GL programs) with code to activate a certain window/page
2. a hand-written/custom part, which will contain customer-specific business logic to initialize the app. I've tried to get this automated somehow, but is too much specific. In any case, while rewriting the embedded part for our WIP apps, if there are parts which look common I'll move them into common code. Also, I'll document what is needed for this process.
3. a part to 'decorate' a specific window with other custom pages/views (non-generated, like the room occupancy we have for Hotel GUI) - both at 4GL and JS level.

Currently I'm at the AST/code generation for the embedded windows/pages and also started the common driver part.

#9 - 07/18/2017 05:56 PM - Greg Shah

I like the approach.

#10 - 08/14/2017 01:09 PM - Constantin Asofiei

There are additional files (some dependent .p programs - fwd-embedded-driver.p and fwd-page-template.p and some include files, all in fwd/ folder) - where should I put these?

#11 - 08/14/2017 01:18 PM - Greg Shah

It depends on how they are used. Must they be converted at the same time as the application code? Or can they be converted in advance?

#12 - 08/14/2017 01:23 PM - Constantin Asofiei

Greg Shah wrote:

It depends on how they are used. Must they be converted at the same time as the application code? Or can they be converted in advance?

The fwd-page-template.p is required during conversion (I need the AST to be used as a template for each created embedded program) plus it

references an include file with custom code. Both programs have preprocessor code emitted depending on the ADM version in use. So yes, they are required at conversion.

Also, the embedded code is generated into fwd/embedded/ folder added as a prefix to the program's path - should I make this configurable?

#13 - 08/14/2017 02:30 PM - Greg Shah

they are required at conversion

I assume this code always needs to be carefully matched up with the specific embedded conversion rules in FWD. They are probably always needed to be present too, in any case where embedded mode is being enabled.

Thus it makes sense that these are checked in to the FWD bsr repo instead of to a given project's repo.

For now, place these files in src/4gl/embedded/. The propath may need some implicit pathing to find these.

Also, the embedded code is generated into fwd/embedded/ folder added as a prefix to the program's path - should I make this configurable?

There should be a flag (p2j.cfg.xml) to enable/disable automatic conversion of embedded mode.

I'd prefer the output in pkgroot + /embedded/.

#14 - 08/14/2017 02:40 PM - Constantin Asofiei

Greg Shah wrote:

they are required at conversion

I assume this code always needs to be carefully matched up with the specific embedded conversion rules in FWD. They are probably always needed to be present too, in any case where embedded mode is being enabled.

Exactly.

Thus it makes sense that these are checked in to the FWD bsr repo instead of to a given project's repo.

For now, place these files in src/4gl/embedded/. The propath may need some implicit pathing to find these.

OK. At runtime, we can leverage the propath for fwd-embedded-driver.p, but at conversion time I need the exact path for fwd-page-template.w. Plus, both files must be converted alongside the other legacy code and included in the final jar.

Also, the embedded code is generated into fwd/embedded/ folder added as a prefix to the program's path - should I make this configurable?

There should be a flag (p2j.cfg.xml) to enable/disable automatic conversion of embedded mode.

Currently I've added a new conversion mode (E0) which enables this - see ConversionDriver changes.

I'd prefer the output in pkgroot + /embedded/.

What I'm generating are ASTs which are placed in fwd/embedded/path/to/awindow.w.ast, where original path is path/to/awindow.w; the final Java source code will be using pkgroot automatically. I'll drop the fwd/ prefix.

#15 - 08/14/2017 02:42 PM - Constantin Asofiei

Code generation outstanding issues at this time:

1. windows with program parameters - I will add a warning and ignore these programs for now
2. programs which have adm-create-objects but the case has only 0 or 1 page number - these might be non-tabbed programs, need to confirm in the VM. Not sure how to include these.
3. programs which have adm-create-objects, the folder names have i.e. n entries, but the case is referring a page number greater than n - I'm ignoring this page for now...
4. programs which are in the windows/ folder, but there is no adm-create-objects - again, these might be non-tabbed programs.

Without these, the programs selected as embedded windows from the windows/ folder parse fully (for our older POC app).

Next items to finish:

1. changes in the POC's custom driver code for embedded
2. some changes/customization for the JS side

#16 - 08/16/2017 04:55 PM - Greg Shah

Code Review Task Branch 3260a Revision 11160

This is really impressive stuff. I do think that more documentation will be needed. In particular some example code that shows ADM/ADM2 screens and the embedded code that is generated.

I also have a sense that there may be a way to do more of this in the runtime and to eliminate the generated code, but for now I simply don't understand it thoroughly enough to know that it is or is not possible.

1. frame_generator.xml, AnnotatedAst, DumpTree, ConversionDriver, ECW, CommonAstSupport, PatternEngine, TemplateWorker, SymbolResolver, progress.g, CommonFrame, ControlEntity, p2j.embedded.js need history entries.

2. ConversionDriver.embedded(), ECW.evaluateCharacterExpression() (both of them), CommonAstSupport.splitToList(), CommonAstSupport.addNewAst(), PatternEngine.getNewAstFiles(), PatternEngine.addNewAst() need javadoc.

#17 - 08/16/2017 04:57 PM - Greg Shah

Code Review Task Branch 3260a Revision 11161

It looks good.

#18 - 08/21/2017 12:07 PM - Constantin Asofiei

Greg, with the embedded part I'm testing an approach which switches the approach from code generation: instead of generating new code, I'm adding only the required new procedures to the existing ABL programs (which are determined to be executed as embedded). The code in these procedures is the same as in the previous approach; this should be cleaner because I no longer have to 'pull' code from the legacy program to the new, generated, program - there are cases where the window programs have existing logic in the external program or other internal procedures which is not easy to pull, it complicates things a lot.

Also, what I'm trying to solve is this: there are cases where the default frame in the window (i.e. f-main) has explicit widgets added (like browse or others). In some cases, these look like legit widgets which need to be shown; in other cases, like Hotel GUI, the logout and exit buttons need to be ignored. I want to add some hints to determine which main frames have 'ignored' widgets (by their names) and ignore them (i.e. make them hidden from the app logic).

In other news, somehow I caught a deadlock in virtual session termination: the Queue instance was not being locked before starting to terminate the session; there is a fix for this in 3260a 11166.

#19 - 08/23/2017 04:59 PM - Constantin Asofiei

Greg, do you want me to keep the conversion changes related to add new, generated, ASTs to the source list? I don't think I need to generate new programs anymore (just code added to existing programs).

#20 - 08/23/2017 05:01 PM - Greg Shah

Are you referring to the helper methods in ConversionDriver? That would be OK.

I guess you don't need the new E0 mode anymore because the changes can be put into annotations?

#21 - 08/23/2017 05:09 PM - Constantin Asofiei

Greg Shah wrote:

Are you referring to the helper methods in ConversionDriver? That would be OK.

OK, I'll clean it up and rename some APIs to more generic names.

I guess you don't need the new E0 mode anymore because the changes can be put into annotations?

Hmm... I think this can be done as a distinct phase just at the beginning of ConversionDriver.back, conditionally via a add-embedded-windows flag at p2j.cfg.xml.

#22 - 08/23/2017 05:10 PM - Greg Shah

I think this can be done as a distinct phase just at the beginning of ConversionDriver.back, conditionally via a add-embedded-windows flag at p2j.cfg.xml.

OK, sounds good.

#23 - 08/25/2017 10:55 AM - Greg Shah

Code Review Task branch 3260a Revision 11172

I think the PatternEngine should probably have clearNewProgramFiles() to give the caller some control over the lifetime of the list. Otherwise I'm good with the changes.

#24 - 08/25/2017 12:07 PM - Constantin Asofiei

Greg, I've rebased 3260a with trunk 11161 and 3260a rev 11174 contains latest changes. The final approach is for the fwd-embedded-driver.p to rely on the ADM framework to manipulate the window, and just remove/hide any frames other than the one being reported as "main". In the program file for a window which is determined to be exposed as an embedded window, only these APIs are emitted:

1. getPageHandles - returns the handles associated with the programs for a certain page in this window

2. fwdGetMainFrame - returns the main frame handle
3. fwdGetWindow - returns the window handle

There are some issue, but it basically works with both the POC and Hotel GUI:

1. in ADM2.2 (Hotel GUI), the folder frame (the tabs) although is hidden by the FWD driver, is made visible again by ADM... need to find a way to resolve this.
2. in both cases: I rely on the WINDOW and FRAME handles to walk its children and hide as needed, but this isn't working properly in FWD. We need some fixes to make FIRST-CHILD and NEXT-SIBLING work properly in the UI, for WINDOW and FRAME.

#25 - 08/25/2017 12:17 PM - Greg Shah

Code Review Task Branch 3260a Revision 11174

I'm good with the changes. The result is quite good.

#26 - 08/25/2017 02:17 PM - Constantin Asofiei

Greg, I need to backout the virtual session deadlock fix - I think is causing problems when connecting. I'll save the logs and create a new task...

#27 - 08/25/2017 02:25 PM - Greg Shah

OK, then you can merge to trunk after that is backed out. It is the only thing that is "dangerous" in your update, and the rest passed regular ChUI runtime regression testing.

#28 - 08/25/2017 02:57 PM - Constantin Asofiei

Greg Shah wrote:

OK, then you can merge to trunk after that is backed out. It is the only thing that is "dangerous" in your update, and the rest passed regular ChUI runtime regression testing.

Backed out and merged to trunk rev 11162; 3260a was archived.

#29 - 09/01/2017 02:54 PM - Greg Shah

- Related to Feature #3331: translate 4GL menu widget processing into a standardized embedded web client menu system added

#30 - 09/13/2017 09:22 AM - Constantin Asofiei

Hynek, quick question: is it me or afterConfigUpdate is never called on server-side?

#31 - 09/13/2017 09:36 AM - Hynek Cihlar

Constantin Asofiei wrote:

Hynek, quick question: is it me or afterConfigUpdate is never called on server-side?

Any config change registered after ConfigSyncManager.markScopeStart() relies on AspectJ. Are your server classes processed with the AspectJ compiler?

#32 - 09/13/2017 09:46 AM - Constantin Asofiei

Hynek Cihlar wrote:

Constantin Asofiei wrote:

Hynek, quick question: is it me or afterConfigUpdate is never called on server-side?

Any config change registered after ConfigSyncManager.markScopeStart() relies on AspectJ. Are your server classes processed with the AspectJ compiler?

What I want to track is a change of frame visible state, if is changed by client-side. The server side sees the state change and does some work, I don't have any changes in the underlying config sync code, only this in FrameWidget:

```
public void afterConfigUpdate(FrameConfig beforeUpdate)
{
    super.afterConfigUpdate(beforeUpdate);

    if (beforeUpdate.visible && !config.visible)
    {
        // in this case, was made not-visible - move to last
        moveInChain(false);
    }
}
```

I was expecting that ConfigManager.applyConfigUpdates called on server-side by LogicalTerminal.applyChanges would in the end notify via afterConfigUpdate, but this is not happening - there is no config owner when ConfigManager.syncConfigChanges call at the end of applyConfigUpdates is executed.

#33 - 09/13/2017 10:14 AM - Hynek Cihlar

Constantin Asofiei wrote:

I was expecting that ConfigManager.applyConfigUpdates called on server-side by LogicalTerminal.applyChanges would in the end notify via afterConfigUpdate, but this is not happening - there is no config owner when ConfigManager.syncConfigChanges call at the end of applyConfigUpdates is executed.

Are the methods beforeConfigValueSet() and afterApplyConfig() in SyncConfigChangesAspect executed in your server process?

#34 - 09/13/2017 10:26 AM - Constantin Asofiei

Hynek Cihlar wrote:

Are the methods beforeConfigValueSet() and afterApplyConfig() in SyncConfigChangesAspect executed in your server process?

Yes, but only afterConfigSet sets a non-null owner.

At this time, my impression is that on server-side, currently all afterConfigUpdate code - and there is more than one widget which defines this API - is never executed.

#35 - 09/13/2017 10:33 AM - Hynek Cihlar

Constantin Asofiei wrote:

Hynek Cihlar wrote:

Are the methods beforeConfigValueSet() and afterApplyConfig() in SyncConfigChangesAspect executed in your server process?

Yes, but only afterConfigSet sets a non-null owner.

beforeConfigValueSet or afterApplyConfig should be called, too. Otherwise afterConfigUpdate won't get called.

#36 - 09/13/2017 10:53 AM - Constantin Asofiei

Hynek Cihlar wrote:

Constantin Asofiei wrote:

Hynek Cihlar wrote:

Are the methods `beforeConfigValueSet()` and `afterApplyConfig()` in `SyncConfigChangesAspect` executed in your server process?

Yes, but only `afterConfigSet` sets a non-null owner.

`beforeConfigValueSet` or `afterApplyConfig` should be called, too. Otherwise `afterConfigUpdate` won't get called.

Sorry, I meant `afterConfigSet` is not called at all, only `beforeConfigValueSet` and `afterApplyConfig` are called.

#37 - 09/13/2017 11:00 AM - Hynek Cihlar

Constantin Asofiei wrote:

`beforeConfigValueSet` or `afterApplyConfig` should be called, too. Otherwise `afterConfigUpdate` won't get called.

Sorry, I meant `afterConfigSet` is not called at all, only `beforeConfigValueSet` and `afterApplyConfig` are called.

Can you put a breakpoint to the call to `resolveConfigOwner()` in `markScopeEnd()` and check why the config owner is not resolved?

#38 - 09/18/2017 06:54 PM - Constantin Asofiei

Hynek Cihlar wrote:

Constantin Asofiei wrote:

beforeConfigValueSet or afterApplyConfig should be called, too. Otherwise afterConfigUpdate won't get called.

Sorry, I meant afterConfigSet is not called at all, only beforeConfigValueSet and afterApplyConfig are called.

Can you put a breakpoint to the call to resolveConfigOwner() in markScopeEnd() and check why the config owner is not resolved?

I'm postponing this, as is not affecting my use case.

#39 - 09/18/2017 06:56 PM - Constantin Asofiei

Greg, what I want to finish at this time is an improvement for the JS code, to allow unique JS files for each window (so each window can be decorated on its own). I think I will be able to finish it in the next couple of days.

#40 - 09/27/2017 07:31 PM - Constantin Asofiei

Greg, I have some Hotel GUI changes which are dependent on 3260b - where should I commit them for review?

#41 - 09/28/2017 09:03 AM - Greg Shah

I've created a temporary copy of Hotel GUI as ~/secure/code/p2j_repo/samples/hotel_gui_3260b Use that and when the changes are committed to hotel_gui, we will remove hotel_gui_3260b.

#42 - 09/28/2017 12:36 PM - Constantin Asofiei

3260b rev 11175 and hotel_gui_3260b rev 147 has latest compatible changes.

There is a bug, don't know why, but the page buttons disappear after you click and move mouse away from it... and re-appear if you hover the mouse over it.

#43 - 10/04/2017 08:14 AM - Constantin Asofiei

3260b rev 11177 passed runtime testing.

#44 - 10/04/2017 03:55 PM - Greg Shah

You can merge to 3260b trunk.

#45 - 10/04/2017 04:09 PM - Constantin Asofiei

3260b was merged to trunk rev 11173 and archived.

hotel_gui_3260b was promoted to hotel_gui (rev 147).

#46 - 10/20/2017 02:19 PM - Constantin Asofiei

There is at least one case where the window title for a ADM window is set as window:title = some-char-var.. In this simple example, the some-char-var is used ONLY here and the title is set in this var's INIT value. There might be other cases where the window title is a complex expression... how do you want to handle this? Is it OK to use hints and default to the program name, if hints not found?

#47 - 10/20/2017 02:50 PM - Greg Shah

Is there any way to allow this to be set at runtime and send the result down to the client?

#48 - 10/20/2017 02:54 PM - Constantin Asofiei

Greg Shah wrote:

Is there any way to allow this to be set at runtime and send the result down to the client?

The embedded app needs the ADM window title before executing that window, to be able to populate a drop-down (or any other widget) from where the window can be chosen (that's why I export the windows to a adm_windows.json file). So no, I can't compute this at runtime.

Although you did give me an idea, about the window's pages... I might be able to extract these from the 'afolder' ADM program handle, at runtime.

#49 - 10/23/2017 10:59 AM - Constantin Asofiei

Created 3260c which holds fixes/improvements related to ADM windows in embedded mode.

Revision 11182 contains:

1. ADM version is kept at the window, in adm_windows.json
2. fixed a problem with the window title, when it can't be evaluated to a constant string - the window title defaults to program name and can be overridden by the embedded app logic, when building the AMD windows container (i.e. combo-box).

#50 - 10/24/2017 10:11 AM - Greg Shah

Code Review Task Branch 3260c Revision 11183

I'm fine with the changes.

#51 - 10/24/2017 01:02 PM - Constantin Asofiei

Greg Shah wrote:

Code Review Task Branch 3260c Revision 11183

I'm fine with the changes.

Merged to trunk rev 11182 and archived.

#52 - 10/26/2017 11:25 AM - Constantin Asofiei

- % Done changed from 0 to 100

- Assignee set to Constantin Asofiei

Greg, this task can be closed.

#53 - 10/26/2017 12:06 PM - Greg Shah

- Status changed from New to Closed