



OpenROAD 6.2 – New Features in Detail, Part II

New in OpenROAD 6.2 – For OpenROAD developers.
Fourth of four presentations. Assumes attendance at the first.

Sean Thrower

June, 2015

Confidential © 2014 Action Corporation

Disclaimer

This document is for informational purposes only and is subject to change at any time without notice. The information in this document is proprietary to Actian and no part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of Actian.

This document is not intended to be binding upon Actian to any particular course of business, pricing, product strategy, and/or development. Actian assumes no responsibility for errors or omissions in this document. Actian shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. Actian does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

Contents

- OpenROAD 6.2 – New generic features in more detail, part II
 - The fourth of four presentations covering OpenROAD 6.2
 - This presentation is the second of two reviewing in detail the new features in this release
 - Features illustrated in the presentation require the first OpenROAD 6.2 patch
 - p14746 or later



This presentation assumes that you have seen the Overview presentation.

If you have not, we recommend that you first view the recording of the Overview presentation, available at

<https://actian.webex.com/actian/j.php?MTID=m013e2f1d62551cb8f21d7f92dd145f6f>

p14746 OR 6.2.0 (int.w32/00)

OpenROAD 6.2 Objectives

- OpenROAD 6.2 is the outcome of a set of objectives pursued systematically over the last few years:
 - I - Provide specific support for certain in-demand business requirements
 - II - Reduce the cost of OpenROAD development
 - III - Improve the deployment of OpenROAD
 - IV - Provide generic enabling facilities to underpin the new features and also future ones
 - V - Implement all these changes in a way that logically extends the OpenROAD metamodel and fills important gaps in it
- The fourth objective is addressed in this presentation, completing the coverage begun in the third presentation
 - The fifth objective is also covered briefly, along with more on some of the many client-driven enhancements also in OpenROAD 6.2.

In the Overview Presentation (reminder):

→ Meeting Business Requirements

- Restyling to up-to-date look-and-feel(s)
- Same-code (unchanged code) transformations
- Generated userclasses and displays
- Active-map and Booking/Allocation capabilities
- Richer out-of-the-box capabilities
- Easier deployment

→ Improving OpenROAD ROI

5

Confidential © 2014 Actian Corporation



Meeting OpenROAD 6.2 Objectives ...

- IV - Providing generic enabling facilities to underpin the new client-oriented features, and future ones



6

Confidential © 2014 Actian Corporation

Actian

IV - Providing generic enabling facilities to underpin the business-need features, both current and future

1. Bitmapped backgrounds with built-in bordering and double-buffering
2. Compound bitmaps, sprites and animations
3. Tagged Values/Items
4. Storable defined-behaviours
5. Helper classes
6. Many enabling property and method changes to field and data classes
 - TreeViews, TableFields, TabFolders; String & BitmapObjects; and much more
7. Enhanced PropertyChanger facilities
8. Database and display heuristics
9. Downloadable IngresNet
10. LoadnRun deployment

Items 1 to 5 were all covered in the third presentation

Items 9 and 10 were covered in the second presentation

□ System Class and 4GL enabling-changes ⑥

→ Display class changes

- Bitmapped background changes for all fields (already covered)
- Specific changes for TreeViewField, TableField, TabFolder, EntryField, SubForm, ControlButton, and others

→ Other class changes

- BitmapObject, FieldObject, LongByteObject, Array

→ New classes

- New system classes TaggedValue, HashTableEntry, XMLParserCallbacks (SAX)
- Helper userclasses (already covered)

→ New 4GL events and variables

- (Child)MouseDown and MouseUp, (Child)MouseEnterNode and MouseExitNode
- Curexec (alternative to curframe, curprocedure, curmethod)

TreeViewField changes

→ 56 new or changed attributes and methods

→ Highlights:

- Rightclick node selection (optional)
- Branch move and copy, drag and drop
- Branch hide
- Node checkboxes
- Font and background control at all levels
- Managed unique-key access to nodes
- Simple to code

Action

Demo:

```
w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -ctreeviewfieldchanges
```

Click the top button to populate the tree with nodes representing folders

Note the checkboxes, FullRowSelect highlight, customized nodeheight and inset

Holding down the Shift button, Drag the "Development" node onto the "ST" node

Note that the node is inserted at that location

Drag the "Development" node onto the "Tools" node

Note that the node is attached to that node

Click the second button to populate the tree with nodes representing files

Note the individual-node font control, background control

RightClick an unselected node

Note the focus switches to the node while the dropdown is visible

Click whitespace to release the dropdown

Note the focus switches back to the previously-selected node node

Changed Properties: 56 new or changed attributes and methods, 9 new reasoncodes:

In TreeViewField:

- Rightclick node selection (which you can switch off)
- **New attributes:** DragCopyCursor, DragMoveCursor, DropCursor, ExactNodeHeight, FullRowSelect, HasCheckboxes, HasHorizontalScrollBar, Indent, IsBold, IsItalic, IsPlain, IsUnderlined, NodeHeight, RightClickBehavior, SelBgColor, SelFgColor, SelIsBold, SelNoFocusBgColor, SelNoFocusFgColor, ShowSelection, Style, TopNode, TreeHeight, TreeWidth, TypeFace, TypeFaceName, TypeSize attributes
- **New method:** ExpandAll

In Tree:

- **New attribute:** CheckedNodes
- **New methods:** CopyNode, MoveNode, SetBranchChecked, Sort

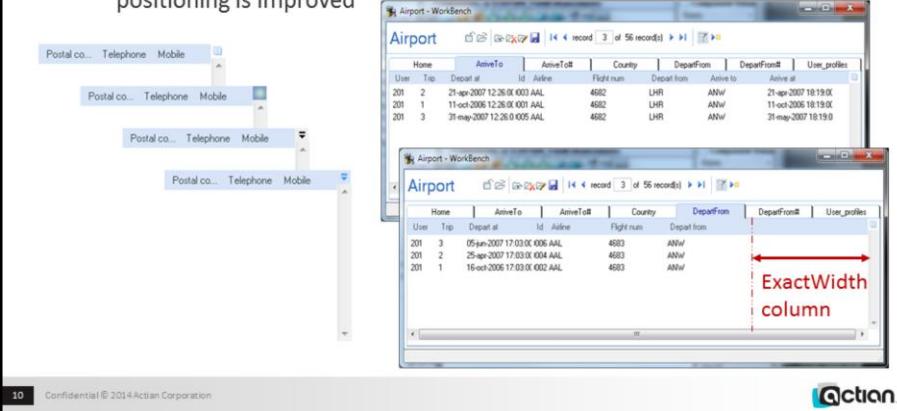
- Enhanced method: AddNode

In TreeNode:

- New attributes: BgColor, DragCopyCursor, DragMoveCursor, DropCursor, ExpandedBitmapLabelIndex, FgColor, Height, IsBold, IsChecked, IsDragNode, IsDropNode, IsItalic, IsPlain, IsUnderlined, KeyLabel, Level, SelBgColor, SelNoFocusBgColor, SelNoFocusFgColor, TypeFace, TypeFaceName, TypeSize, Width, Xleft, Ytop

TableField: Exact Width and Height, ControlButton appearance

- TableFields can now be aligned with other fields
- Extended exact-width uses unnamed column, matching header and background of previous column
- ControlButton bitmap can be replaced, other attributes can be changed, positioning is improved



Demo:

w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -ctablefieldexactwidth

Click second toolbar icon ("Open dataset")

Type 27 in the "record" field and hit return

- JFK airport details will display

Select each tab in turn

- 4 of these are tablefields, each of different default width, but all showing the same displayed width, providing a clean display

Demo:

w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -ccontrolbuttons

- The four tablefields each display a different controlbutton

Click each controlbutton in turn to show the menu (click any option to close the menu)

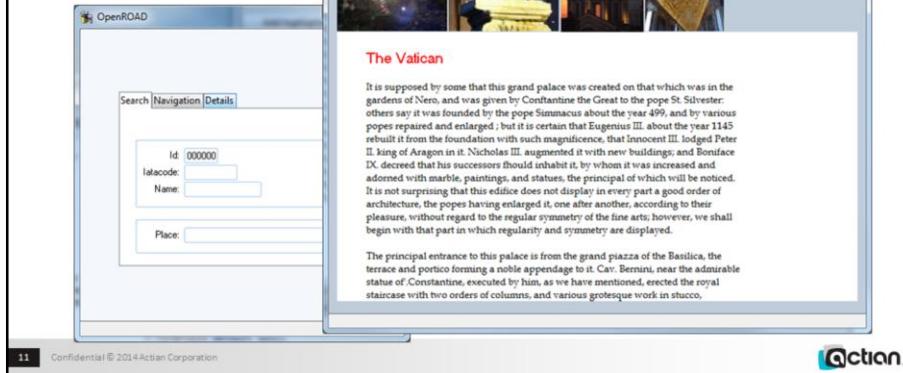
Note that the first controlbutton is the original image, honouring the FgColor and BgColor

Note that the third and fourth controlbuttons are identical images, honouring the FgColor and BgColor

Tabfolder: Bitmapped and Highlighted Tabs

- Bitmapped tabs can be any size or appearance, and support transparency and mouseover highlighting

- COR_BITMAPPED
- POS_CENTER



Demo:

w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -ctabfolderbitmappeds

- The tabfolder displays rectangular bitmapped tabs

Click each tab in turn

Note the tab highlighted text and the page display

Note that the tabbar can now (6.2) be coloured or clear

Demo:

w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -cbdptabhighlighting

- The tabfolder displays gradianted bitmapped tabs

Click the "Add tab bitmaps" button

- The unselected tabs display a bordered double-gradient bitmapped-background

Select different tabs to confirm this

Click the "Add highlighting" button

Mouse across the unselected tabs

- The moused tab highlights during mouseover

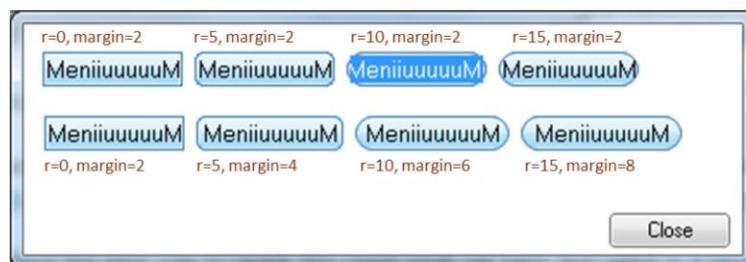
Mouse across the selected tab

- The moused tab does not highlight

Note that this is the characteristic Windows7 behaviour for tabfolders

EntryField: Independent Margins

- Each margin can be any pixel size (including negative)
 - LeftInnerMargin, TopInnerMargin, RightInnerMargin, BottomInnerMargin
 - Default is 2px – previously this could not be reset
 - Need to reset margins for rounded-border fields, otherwise the text overlaps



SubForm: SizeToFit

→ Fields can now be automatically right-aligned to the frame

- By right-aligning the field to a SubForm with `SizeToFit = STF_FRAMEHORIZONTAL`
- This works on the TopForm and in MainBar toolbars
- SubForm automatically adjusts to field or frame resize
- Settings are
 - `STF_SELF` (no alignment)
 - `STF_PARENT`, `STF_PARENTHORIZONTAL`, `STF_PARENTVERTICAL`
 - `STF_FRAME`, `STF_FRAMEHORIZONTAL`, `STF_FRAMEVERTICAL`



Demo:

w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -csubformsizetofit

Note the top toolbar help button, already right-aligned within a subform that has `SizeToFit = STF_FRAMEHORIZONTAL`

Click the ApplySizeToFit button to make the pink subform `STF_PARENT`

Note the pink subform fills its parent to the right and bottom

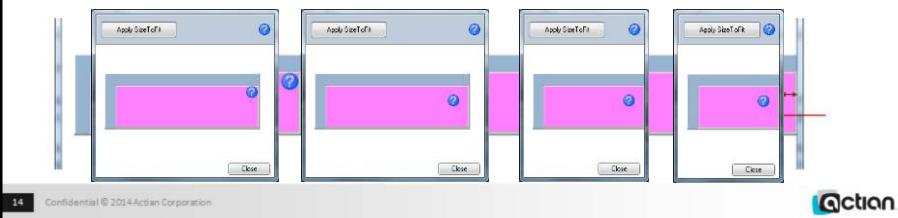
Resize the frame from the right to truncate the pink subform and make the help button disappear

Click the ApplySizeToFit button again to make the pink subform `STF_FRAME`

Note that the pink subform has resized to align with the frame (the help button has reappeared)

SubForm: ChildMargins

- Like StackFields, SubForms can now specify margins for each edge
 - ChildLeftMargin
 - ChildTopMargin
 - ChildRightMargin
 - ChildBottomMargin
- This enables:
 - Better positioning of SubForm childfields for more exact visual effects
 - SubForms with bitmapped backgrounds that include borders can use the margin setting to prevent gravity-locked childfields from overlaying the border



Demo:

w4gldev rundbapp remotehost::or62demos d201504_fieldenhancements -csubformchildmargins

Click the help button on the pink subform to apply a top and right margin to the subform

Note that the help button is inset 3 pixels

Click the ApplySizeToFit button twice to make the pink subform STF_FRAME

Note the help button is inset by the same amount in the resized pink subform

Resize the frame from the right to resize the pink subform

Note the help button is still inset by the same amount in the resized pink subform

Click the help button repeatedly on the pink subform to increase the margins by 3 pixels each time

Resize the frame from the right to resize the pink subform, repeatedly

Note the help button inset is always honoured

SubForm: OS_GROUPBOX OutlineStyle

- Groupboxes in Windows 7 and other styles have rounded corners and non-black borders, and may have no text label
- Setting the SubForm OutlineStyle to OS_GROUPBOX:
 - Rounds the corners
 - Allows border colour to be set using OutlineColor
- Setting the GroupBoxLabel value to " " (a single space):
 - Rounds the corners without adding any textlabel



GroupBoxLabel:	'Label'	'Label'	" "
OutlineColor:	CC_GREEN	CC_PALE_BLUE	CC_PALE_RED
OutlineStyle:	OS_SOLID	OS_GROUPBOX	OS_GROUPBOX

CompositeField: FieldsByProperty

→ Enables location of any field(s) by any flag, tag or property

- Avoids the pre-6.2 need to name or reference fields purely in order to re-find them later on
- Can search within the searchset
- Efficient and accurate, enables much more flexible generic coding when manipulating fields

→ Examples

- Return every field on the form, however deeply nested

```
fields = curframe.TopForm.FieldsByProperty();
```

- Return every field tagged as part of an address

```
fields = curframe.TopForm.FieldsByProperty(tagname='address');
```

- Return every tabpage on the form

```
fields = curframe.TopForm.FieldsByProperty(attributename='classname',  
searchstring='tabpage', casesensitive=False);
```

FormField: BgDisplayPolicy, BgPattern, UpdBackground

→ BgDisplayPolicy attribute

- Now determines whether a FormField's background is image- or property-based
 - BDP_DEFAULT, BDP_FIXED, BDP_RELATIVE: property-based
 - BDP_CORNERED, BDP_BORDEREDSCALED, BDP_BORDEREDTILED: image-based
- Image-based backgrounds (bitmapped backgrounds) are new in 6.2 and were covered in the first of these two "new features" presentations

→ BgPattern attribute

- New FP_BITMAPCLEAR setting enables field backgrounds to be transparent

→ UpdBackground method

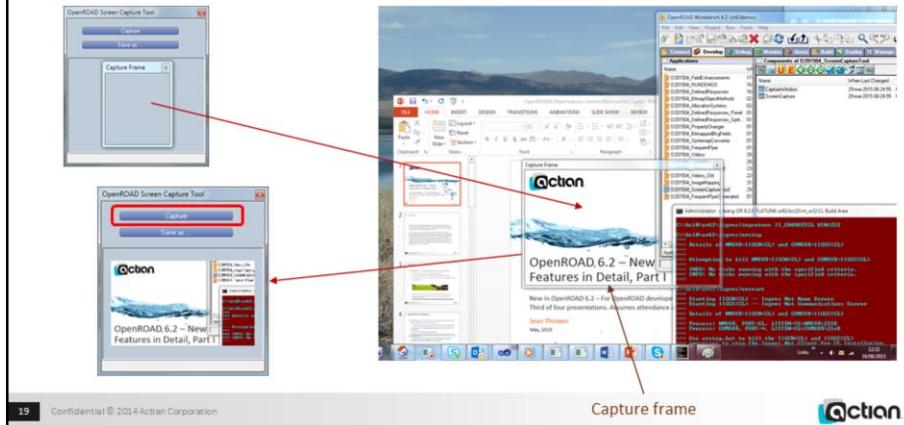
- New parameters to adjust image-based backgrounds at runtime
 - Imageindex – which of the BgBitmap's sub-images to display
 - Cornersize – the size of the corner-area of the field's background
 - Opacity – the opacity of this field's background

BitmapObject: Marker attribute

- Read/write attribute, varchar(1024)
- Can serve a variety of purposes:
 - Enables bitmaps to be uniquely identified ...
 - Using Marker as a tag or name
 - Or to be annotated ...
 - Using Marker to contain descriptive information
 - Or to be prepared for later instantiation if needed
 - Using Marker to hold the DbHandle or FileHandle value
- Using Marker, you can now efficiently retrieve large datasets from the database, even when they include bitmap handles
 - Previously for example, 1 select returning 1000 such Employee records would unavoidably trigger another 1000 selects, one for each bitmap ...
 - Even though the business requirement only needed the bitmap for individual employees picked out by the enduser from the OpenROAD frame using other criteria.

BitmapObject Methods: CaptureBitmap

- Captures the specified area of the screen or window as a bitmap
 - And replaces it with another bitmap if desired – very useful
- Easy to implement Screen Capture tool (using FP_CLEAR TopForm)



Demo:

w4gldev rundbapp remotehost::or62demos d201504_screencapturetool -cscreencapture

Drag the capture window to the location to be captured

Resize (and reposition) the capture window until it displays the exact area to be captured

Click the Capture button

The captured area appears as a thumbnail in the frame

Click the Save as ... button

The File Save dialog appears

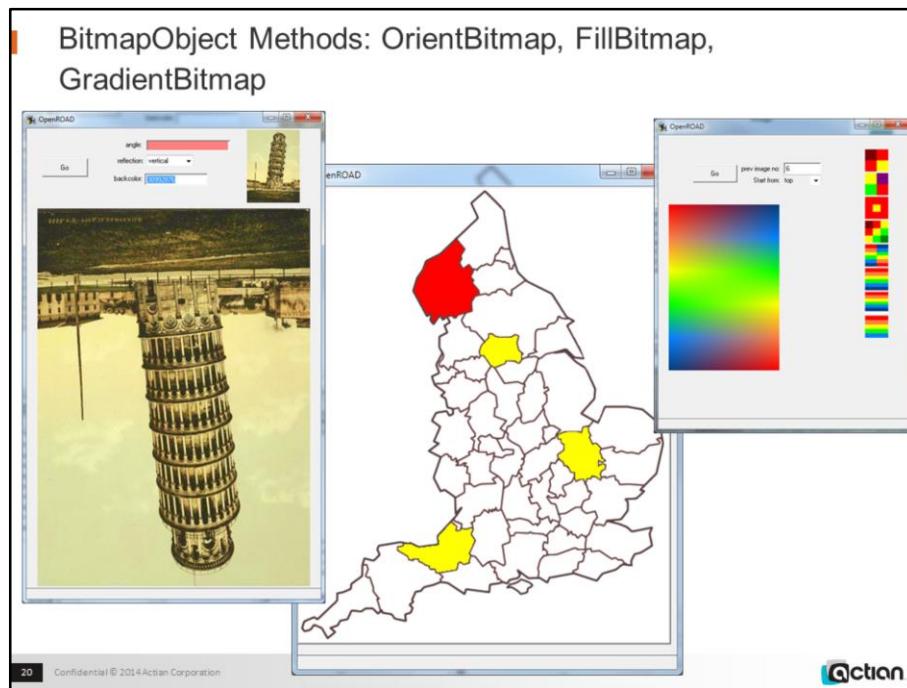
Choose or enter a folder and a "bmp" frame name to write the captured image out as, and click Save

The image is written to file, and the File Save dialog disappears

RightClick the Save as ... button

The image from the file is displayed in the centre of the screen

Note that the file image is identical to the captured area (except that that may be greyed, depending on the utility displaying the file image)



Demo (OrientBitmap):

w4gldev rundbapp remotehost::or62demos d201504_bitmapobjectmethods -corientbitmap

Change the reflection choice to “vertical” and click the Go button

The image is displayed first normally, then inverted

Change the reflection choice back to “none” and set the angle to around 45 and click the Go button

The image is displayed first normally, then at 45 degrees to normal

Change the backcolor to 255 (red) and click the Go button

The image is displayed within a red rectangle

Demo (GradientBitmap):

w4gldev rundbapp remotehost::or62demos d201504_bitmapobjectmethods -cgradientbitmap

Click the Go button

Note that the seed image with the black outline has been used to create the gradient

Note that the actual seed image is tiny

RightClick the Go button to show a gradient arising from a text specification

Note that GradientBitmap will accept text specifications based on the CSS standard

Note that this does not (yet) include fixed-color areas or radial gradients

Demo (FillBitmap):

w4gldev rundbapp remotehost::or62demos d201504_bitmapobjectmethods -cfillbitmap

Click within a county region

The region is filled with red

Note that the boundary is captured (as contiguous pixel coordinates)

Note that the fill can be background or boundary or wholeimage-based, exact-colour or colour-

range

Click another county region

The new region is filled with red, the previous region with yellow

Other BitmapObject Methods

- All these have rich and useful sets of capabilities

- **BlankBitmap**

- Resets bitmap to any size and background colour

- **FormatBitmap**

- Converts other formats to 24bit bitmaps

- **ComposeBitmap, Elements, GetDisplayedElement**

- Compose or decompose bitmaps for compound-bitmap processing

- **ResizeBitmap, CornerBitmap, LocateBitmap**

- Used to paint field backgrounds

- Resize supports Halftone and Bilinear; LocateBitmap can match transparent bitmaps

- **ExtractBitmap, ReplaceBitmap, GetPixelColor, SetPixelColor**

- Used to manipulate bitmap contents

StringObject Methods

- Already in 5.1 release, but they come into their own in 6.2.
- Split
 - Splits a StringObject into an array of StringObject
 - Powerful method with parsing capabilities
 - Can apply multiple delimiters
 - 400 times faster than equivalent 4GL processing
- Join
 - Concatenates an array of StringObject into a single StringObject; used with Split
- ToString procedure in Core
 - Returns a StringObject containing the passed text, usually for use with Split
- Example: To get the filename from a full pathname:

```
fname = ToString(text=fullpathname).Split(delimiter=“\”, backwards=TRUE)[1].Value;
```

New HashTableEntry Class

- Provides HashTable contents as an array of key-object items
 - So the contents of a hashtable can be accessed sequentially as well as by key lookup
- Accessed via HashTable Entries attribute and GetEntries, SetEntries methods
- HashTableEntry contents are saved when the HashTable is saved
 - Essential so that resources stored in the TaggedValueKeyedItems attribute (a HashTable) will be preserved when the host component is saved, imaged or exported.

New MainBar Class

→ MainBars have been in OpenROAD for several releases

- Mentioned in the Workbench User Guide (as Toolbars – see Creating Toolbars)

→ A MainBar is a StackField that can be attached to the side (top, bottom, left, or right) of a frame,

- To provide locations for toolbars and ribbons and menus
- Separate from TopForm

→ FrameExec and FrameSource have four Array of MainBar attributes:

- MainBarLeft, MainBarTop, MainBarRight, MainBarBottom
- Enabling each frame edge to have multiple MainBars
- Each MainBar can contain multiple fields of any type (except FrameForm and MainBar)

→ Use the (newly enhanced) ToolBar Editor in Workbench to create MainBar toolbars

24 Confidential © 2014 Actran Corporation

New MouseData Class and Mouse Events

- New (Child)MouseDown, (Child)Mouseup events
 - And a new MouseData class to hold data about these events
- These events are not queued
 - They fire before any of the queued events (as do KeyDown and SliderMove)
- If you cancel a MouseDown or MouseUp event, you cancel the associated queued event (Click for example)
 - Just set the MouseData CancelEvent attribute to TRUE
- The MouseData object also provides:
 - The timestamp (integer8, millisecond resolution)
 - The mouse exact location
 - What keys were being pressed at the time
 - And more

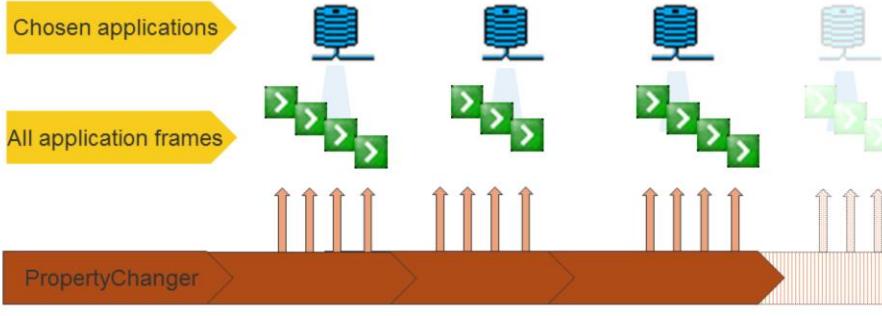
New XMLParserCallbacks Class

- ➔ Used to defeat out-of-memory situations
 - While parsing very large XML files
- ➔ User-defined 4GL procedures
 - Processes each parsed element
 - So the method doing the parsing can safely remove the XMLElement from its parent
 - Allowing for efficient memory use while processing the XML document
- ➔ Uses SAX (under the covers)

❑ Enhanced PropertyChanger facilities ⑦

→ PropertyChanger accesses every frame in every selected application and applies changes to the fields, frame and code

- Changes may be built-in or client-defined
 - Preferably as calls to methods in a dedicated userclass



... Enhanced PropertyChanger facilities ...

→ The two built-in PropertyChanger upgrades are:

- Upgrade from OpenROAD 4.1
- Apply Windows-7-like styling to existing applications

■ To upgrade from OpenROAD 4.1, start with the PropertyChanger frame

- Any client customization of 4.1 upgrade should call methods or procedures from this starting component

■ To apply Windows-7-like styling, start with the W7Styler frame

→ You can create and apply other custom conversions, too

- Other styles
- Company rebrands

■ To do this, use different starting components, based on the original PropertyChanger

- First, examine the simple conversion coding invoked by PropertyChanger,
- and the sophisticated conversion coding invoked by W7Styler

PropertyChanger and Windows7-styling

- ➔ The W7Styler frame differs little from the PropertyChanger frame
 - The real differences are in the methods it calls
- ➔ The processing called by W7styler applies style settings from the w7style frametemplate's stylesheet
 - Much of this is near-identical to Workbench field restyling
 - See the style_attributes, tablefield_style_attributes, and tabfolder_style_attributes userclasses
 - However some field restyling needs custom processing, and all the Windows7 animation effects (pulsing, mouseover and focus highlighting) have to be applied as taggedvalue-stored "inputhead-behaviors"
 - All this is handled by methods in the Style and Windows7Style userclasses
- ➔ The w7style frametemplate was customized for Windows7-styling using a setup frame (SetW7Stylesheet)
 - SetW7Stylesheet is included in PropertyChanger, for you to adapt if needed

Using PropertyChanger

- Import the PropertyChanger application into your OpenROAD repository from II_SYSTEM\ingres\propertychanger
 - It does not need to include, or be included by, any of your applications
- Run the PropertyChanger application, specifying the starting component appropriate to the customization you require
 - from Workbench, using Debug | Go
 - From the commandline, using w4gldev rundbapp <database> propertychanger
 - From the commandline (after imaging) using w4gldev runimage propertychanger.img
- Follow the instructions in the Wizard. Other than selecting the application(s), using the default settings is usually best
 - Once converted, applications are ready to run
 - No need for PropertyChanger in the runtime application

Windows7-restyling and the new OpenROAD features

→ The new OpenROAD 6.2 ability to W7-restyle existing applications without changing the original code, required every one of these new OpenROAD 6.2 generic features:

- Bitmapped backgrounds
- Compound bitmaps
- TaggedValues
- Stored Defined Behaviors
- Helper Classes
- Setup Frames
- Extended PropertyChanger

Database and Display Heuristics ⑧

The userclass and display generators

... apply heuristics to the database data and metadata,

... to derive a layer of relationship, category, subtype and other metadata,

... with which to define business userclasses and frame layouts

The screenshot shows the OpenROAD application interface. On the left, there's a tree view of database objects like 'user_profile', 'User', 'Email', etc. In the center, there's a 'Customization' dialog with tabs for 'Description', 'Class / Attribute', and 'Database properties'. The 'Class / Attribute' tab is active, showing settings for 'Category: contact', 'Subtype: email', and 'Rw/Rv'. Below it, a 'Display properties' section includes 'Sequence: 4', 'Display_order: 12', and checkboxes for 'Main' and 'Sub'. At the bottom, there are 'Save as named display' and 'Set as default' buttons. To the right, a Notepad window titled 'demodb.refs - Notepad' displays a large block of SQL-like metadata related to foreign keys and references between tables like 'user_profile', 'flight', 'route', 'airline', and 'country'. Arrows point from the text 'Attribute metadata' to the customization dialog and from the text 'Relationship metadata, derived from constraints, storage structures, indexes, etc' to the Notepad window.

Attribute metadata

Relationship metadata, derived from constraints, storage structures, indexes, etc

... Database and Display Heuristics ...

Heuristics are applied to the database to mine its data and metadata for relevant business information, shaping the Business Userclass being created.

One output is the set of referential joins that interrelate all tables to which the seed table connects, directly or indirectly.

These can be exported, amended by the developer, and used instead of the mining output, especially for non-Ingres databases.

The heuristics themselves can be customized by the developer.

33 Confidential © 2014 Actian Corporation 

→ The heuristics use for example:

- Constraints, storage structures, indexes, primary, lookup, and indexed foreign keys
- Column names and data cardinality
- Collision resolution rules

Demo:

```
w4gldev runimage workbnch.img -Tall -/appflags profile=or62demos application=d201504_frequentflyer
```

Create and edit a userclass called User_profile

Select Attributes | New | From Database

Choose the user_profile table

Tick Trim Prefix, set the prefix to u_p, tick Capitalize and Cascade

Click Advanced options, tick Write referential ...

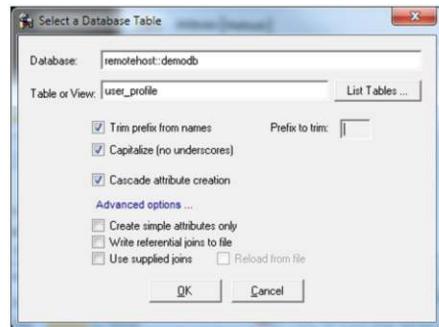
Click OK

Edit the OS file "c:\temp\demodb.refs"

Note that this file contains the relationships that will define which userclasses are created, and which userclass-datatype attributes are created for those userclasses: in effect, this is defining how the application's business classes overlap and interact.

Note that this file can be customized to correct any misinterpretations resulting from the heuristic processing, or to supply relationship definitions for non-Ingres databases whose system table interface may not match that provided by Ingres.

... Database and Display Heuristics – overrides



determine_key_relationships

OpenROAD will find any procedure of this name and use it to determine referential keys, primary and lookup keys, column prefix, instead of the default procedure.

db_refjoins_addlist

db_refjoins_stoplist

OpenROAD will find any include script of this name and use it add or remove referential joins from the list being built for the table being analysed.

determine_attribute_category

determine_attribute_subtype

OpenROAD will find any procedures of these names and use them to determine the category or subtype of a particular attribute instead of using the default procedure.

override_datatype

OpenROAD will find any procedure of this name and use it to determine attribute datatypes instead of the default procedure.



... Database and Display Heuristics – overrides

→ override_datatype

- Needs to be overridden if for example the development language is not English:

```
case Lowercase(datatype) of
  'longbyteobject', 'varchar(256)':
  {
    if  (attname like '%pic%' or
        attname like '%image%' or
        attname like '%icon%' or
        attname like '%map%' or
        attname like '%graph%' or
        attname like '%photo%') then
      datatype = 'bitmapobject';
    endif;
  }
```

... Database and Display Heuristics – overrides

- ➔ `determine_attribute_category, determine_attribute_subtype`
 - Need to be overridden if for example the development language is not English:

```
if attribute.Datatype = 'bitmapobject' then
    case typecode of      // "typecode" is a DataTypeCode (DTC_) setting
        20,21,26,27: typecode = 25;
        default:      ;
        endcase;
    endif;

    if (typecode = 30) and
        (attribute_name like '%id' or attribute_name like '%no' or
         attribute_name like '%num') then
        attribute.SetTaggedValue(tag='category', textvalue='identifier');
    elseif (typecode = 20 or typecode = 21 or typecode = 26 or typecode = 27) and
        ...

```

... Database and Display Heuristics – overrides

The screenshot shows two windows from the OpenROAD application. On the left is the 'Select a User Class' dialog, which lists applications and user classes. An arrow points from the 'Customize' button in this dialog to the 'Customization' tab of the 'OpenROAD' window on the right. The right window displays the customization settings for the 'User_profile' class, specifically for the 'Email' attribute. It shows various properties like Category (contact), Subtype (email), and values (R/r). The 'Display properties' section includes a sequence (4) and a display profile (1234). There are checkboxes for 'Main', 'Sub', 'Th', 'List', 'Title', and 'Print'.

Demo:

w4gldev runimage workbnch.img -Tall -/appflags profile=or62demos

application=d201504_frequentflyergenerated

Create and edit User_profileDetails_1 frame with active_display template

Choose Insert | Display from User Class

Click "Customize" to display the relevant dialog

Select the "Email" node in the left hand tree

The email attribute's extended properties are displayed

Note that this "Customize" dialog provides a way to override the generated userclass settings either temporarily or permanently. Typical changes are to mark some attributes as never displayed, or to correct the allocated category or subtype of an attribute.

Note that the temporary overrides can be saved and subsequently reused as named customizations

Select other nodes in the left hand tree

The corresponding extended properties are displayed

Note that these differ, depending on whether the attribute is of object type

- ✓ Providing generic enabling facilities to underpin the business-need features, both current and future

1. Bitmapped backgrounds with built-in bordering and double-buffering
2. Compound bitmaps, sprites and animations
3. Tagged Values/Items
4. Storable defined-behaviours
5. Helper classes and Setup frames
6. Many enabling property and method changes to field and data classes
 - TreeViews, TableFields, TabFolders; String & BitmapObjects; and much more
7. Enhanced PropertyChanger facilities
8. Database and display heuristics
9. Downloadable IngresNet
10. LoadnRun deployment

Items 1 to 5 were all covered in the third presentation

Items 9 and 10 were covered in the second presentation



Meeting OpenROAD 6.2 Objectives ...

- V - Implement changes in a way that logically extends the OpenROAD metamodel and fills important gaps in it



39

Confidential © 2014 Actian Corporation



V - Extending the OpenROAD Metamodel

1. Field background modelled as a single dynamic image
 - Style/Response complexities encapsulated via method interface (like DBSessionObject)
2. Property-and tag-based field identification
 - Extending OpenROAD's reuse capability
3. Field, attribute and component tags (tagged values), providing
 - Extra properties (state values, custom properties), permanent or runtime-only
 - Stored resources (named-items and named-collections of any complexity)
4. Stored behaviors that execute independently of 4GL events
 - Design- or runtime-built, customizable, intrinsic and/or 4gl-procedure-based
5. SetupFrame mechanism to apply setup processing at design time
 - Instead of contaminating runtime code

The conceptual aspect is important: a great strength of OpenROAD is the consistency and careful definition of its object-oriented metamodel.

TaggedValues are a well established and very useful feature of UML object orientation models, missing from OpenROAD until now. As well as holding ancillary information for which there is no current provision (which database column contains the data for this attribute? what are its allowed values? what was the value of this field before the use changed it?) they enable resources (data and behaviors of any complexity) to be stored, local to their use.

A different concept underpins the bitmap-related changes: the simplification of field appearance into "background" rather than the previous constellation of arbitrarily distinguished properties (still supported), and the provision instead of support for multiple backgrounds (images) and mouse-sensitive mobile subimages (sprites). The validation of this approach is the ability to simulate the highly complex Windows 7 interface using only the new generic capabilities.

Some other OpenROAD 6.2 changes ...

- For others, see the OpenROAD Release Summary



41

Confidential © 2014 Actran Corporation

 **Actran**

Debugging Changes

→ Hitting the SK_BREAK (Pause/Break) key

- In Workbench: stops execution, opens the debugger at the executing line
 - Ideal for breaking into loops to investigate
- In runtime image: writes the 4GL stack information (exec name, line number, etc) to the trace window and log

FrameExec/ProcExec Changes

→ CurExec system variable

- CurFrame, CurProcedure and CurMethod all reference the currently executing ProcExec-based function
 - CurFrame for ghostexecs or frameexecs, CurProcedure for local or global procedures, CurMethod for methodexecs
- Many of the things they are used for are the same, but you can't use the same code each time:
 - curprocedure.Trace() fails if you are in a method's 4GL, for example
- CurExec solves that problem, as it works everywhere:
 - In a procedure, curexec = curprocedure; in a method, curexec = curmethod, etc

→ InInitialize attribute

- Indicates whether the current frame is still within its initialize code
 - CurExec.InInitialize will be TRUE or FALSE
 - You can be in a procedure or method when you check this
 - If it is FALSE, the frame is not yet "realized", which affects some display properties

PNG and 32-Bit Icon Support

- OpenROAD supports image manipulation up to 24 bit
 - Not 32 bit
 - All the new BitmapObject methods use 24 bit images
- However, this does not prevent us reading and displaying 32 bit images
 - So we have added support for PNG and 32 bit ICO images
 - **PNG support includes:**
 - Read, Write, Display (including alpha-channel)
 - Minor restrictions on some sub-formats
 - **32 bit ICO support includes:**
 - Read, display transparently as window icon (including alpha), display on form (ignoring alpha)
 - These images can also be manipulated within OpenROAD, but are converted to 24 bit for this

Datatype support

- Full support for integer
 - Now your OpenROAD timestamps need not expire in this millennium ... ☺
- Controlled support for nchar and nvarchar
 - UTF8 support means these are not needed, and are now disabled by default
 - Set II_W4GL_ALLOW_NVARCHAR to TRUE to re-enable their use

New Environment Variables

- **II_W4GL_EXPORT_WITH_DEFAULTS**
 - Includes default attribute values in XML exports
- **II_TIMER_FOR_CURSOR**
 - Reduces cursor flashing when userevents are used for timer purposes
 - Suggested setting 4000 (milliseconds to wait before displaying busy cursor)
- **II_W4GL_TRACE_PAUSE_ON_ERROR**
 - Triggers an Error Popup Box for unhandled runtime exceptions, so developers and QA testers can catch such errors
- **II_W4GL_ALLOW_NVARCHAR**
 - Re-enables the use of nchar and nvarchar data types
- **II_W4GL_PASS_NVARCHAR_AS_WCHAR**
 - Enables passing nvarchar as wide chars instead of chars to 3GL procedures

New W4gldev Commandline Utilities and Flags

→ New Utilities

- **CreateApp**
 - Creates an empty application
- **RenameApp**
 - Renames an application

→ New Flags

- **-nowindows flag (lets the application run without requiring the window manager)**
 - Previously available for: runimage and rundbapp
 - Now also available for: backupapp, compileapp, createapp, destroyapp, documentapp, makeimage, purgeapp, queryimage, renameapp, versionapp

Case-sensitive Array Sort

- New ArrayObject attribute `SortCasesensitive`
 - When set, the array will always be sorted case-sensitive

What was covered

- ➔ OpenROAD 6.2 – New generic features in more detail, Part II
 - The fourth of four presentations covering OpenROAD 6.2
 - ➔ This presentation was the second of two reviewing in detail the new features in this release
 - Features illustrated in the presentation require the first OpenROAD 6.2 patch
 - p14746 or later



Thank you

- Sean Thrower, OpenROAD Engineering
sean.thrower@actian.com