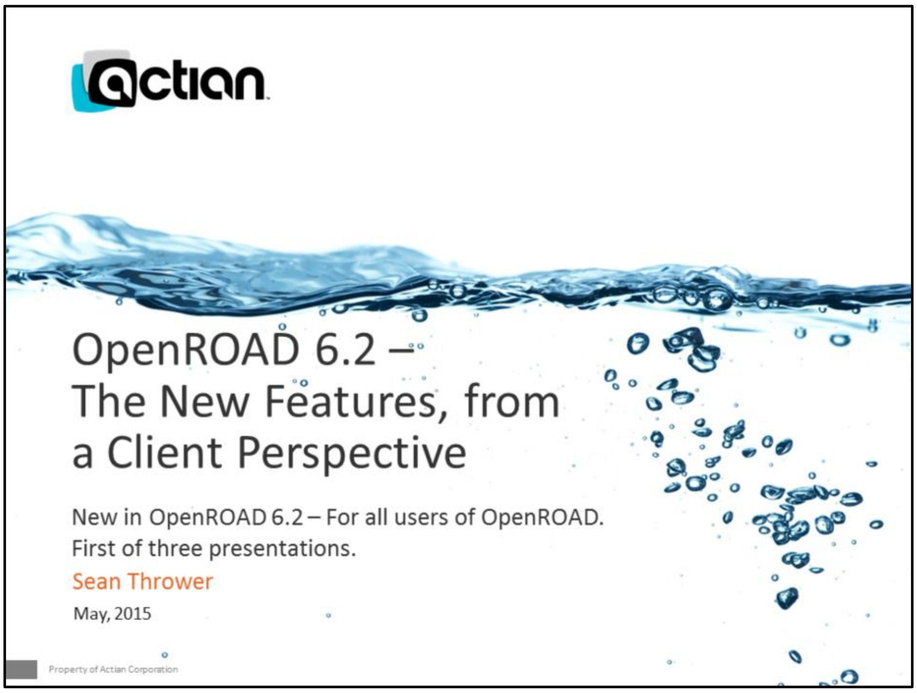
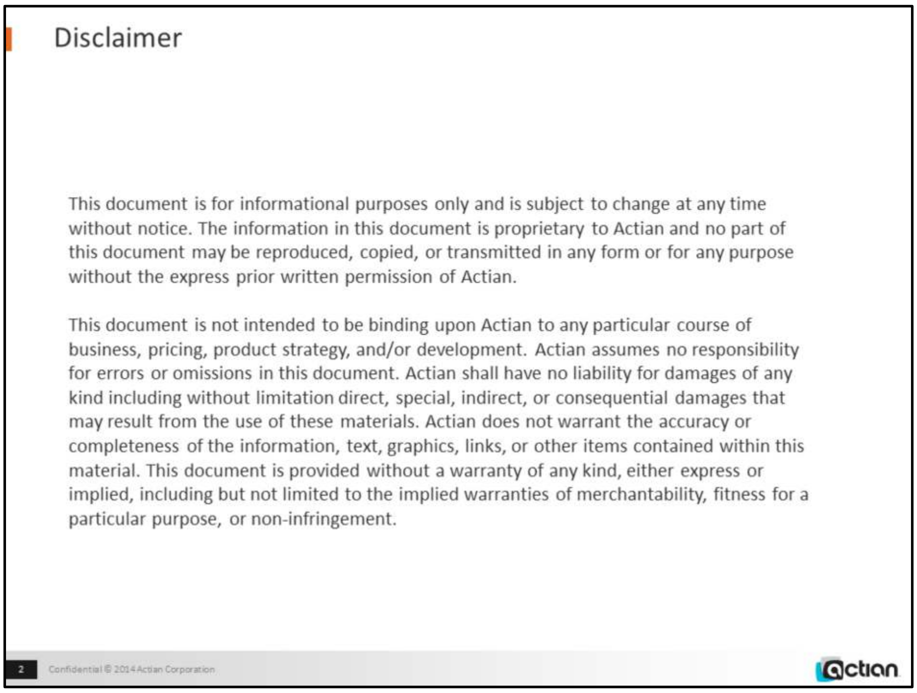


OpenROAD 6.2 – The New Features, from a Client Perspective

New in OpenROAD 6.2 – For all users of OpenROAD.
First of three presentations.

Sean Thrower

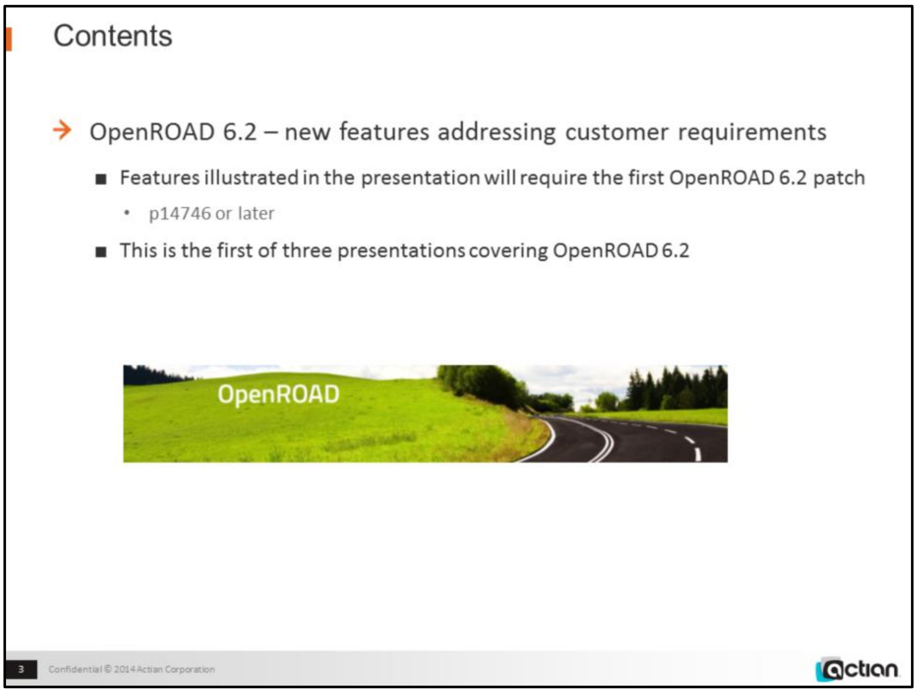
May, 2015



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.

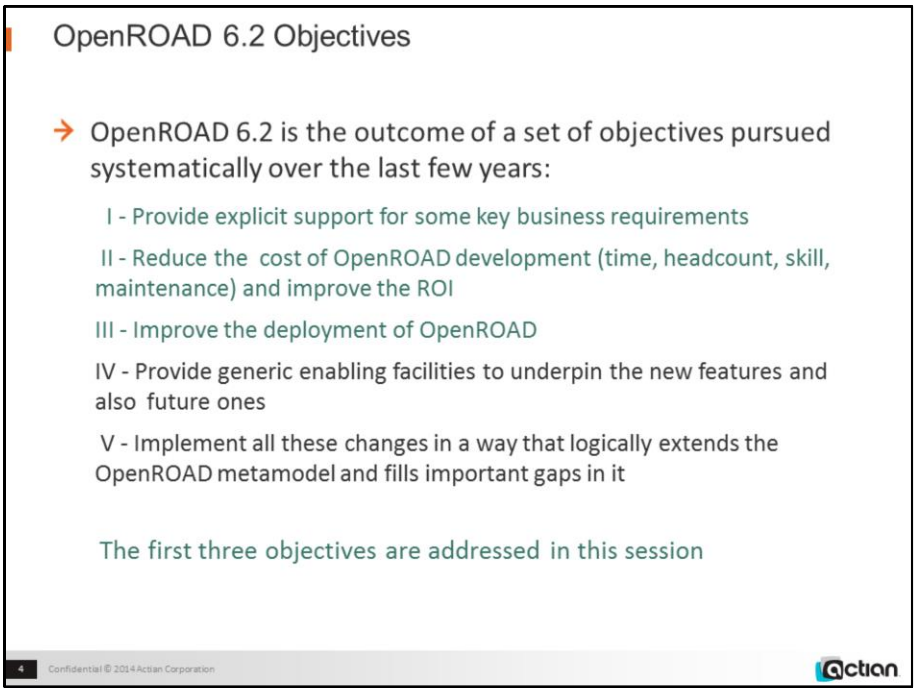
This document is not intended to be binding upon Actian to any particular course of business, pricing, product strategy, and/or development. Actian assumes no responsibility for errors or omissions in this document. Actian shall have no liability for damages of any kind including without limitation direct, 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 features addressing customer requirements
  - Features illustrated in the presentation will require the first OpenROAD 6.2 patch
    - p14746 or later
  - This is the first of three presentations covering OpenROAD 6.2

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 explicit support for some key business requirements

  II - Reduce the cost of OpenROAD development (time, headcount, skill, maintenance) and improve the ROI

  III - Improve the deployment of OpenROAD

  IV - Provide generic enabling facilities to underpin the new features and also future ones

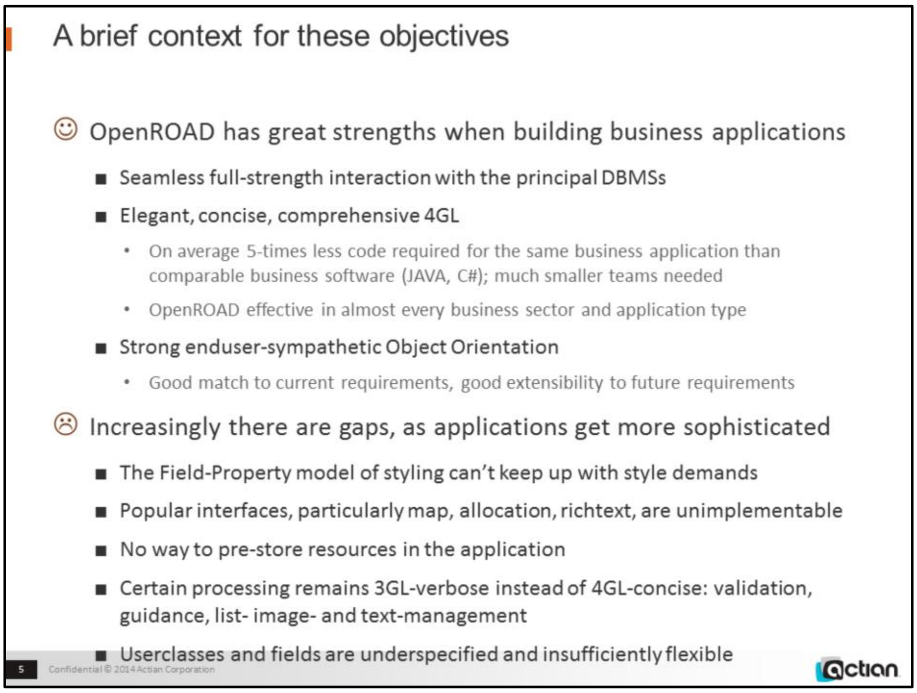
  V - Implement all these changes in a way that logically extends the OpenROAD metamodel and fills important gaps in it

  The first three objectives are addressed in this session

4 Confidential © 2014 Actian Corporation

Actian

In addition there is the usual long list of client-driven enhancements.


A brief context for these objectives

☺ OpenROAD has great strengths when building business applications

- Seamless full-strength interaction with the principal DBMSs
- Elegant, concise, comprehensive 4GL
  - On average 5-times less code required for the same business application than comparable business software (JAVA, C#); much smaller teams needed
  - OpenROAD effective in almost every business sector and application type
- Strong enduser-sympathetic Object Orientation
  - Good match to current requirements, good extensibility to future requirements

☹ Increasingly there are gaps, as applications get more sophisticated

- The Field-Property model of styling can't keep up with style demands
- Popular interfaces, particularly map, allocation, richtext, are unimplementable
- No way to pre-store resources in the application
- Certain processing remains 3GL-verbose instead of 4GL-concise: validation, guidance, list- image- and text-management
- Userclasses and fields are underspecified and insufficiently flexible

5 Confidential © 2014 Actian Corporation

DBMS interaction
- Inline SQL, QueryObjects
- Powerful model of data-mapping, separating data from field

4GL
- Event model considerably superior to rivals

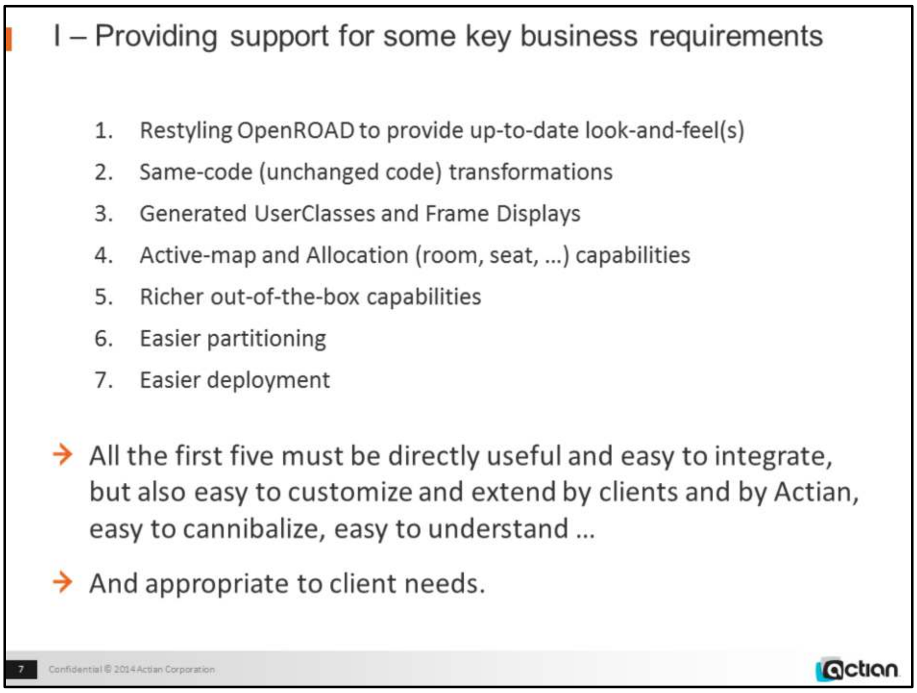
OO
- Endusers' OO "work-model" will map 1:1 with OR's implementation: single inheritance, single interface, 2-level overlapping classes. Limited datatypes though

Field-Property model
- No provision for rounding, asymmetry, sectors and sprites, intrinsic animations

Resources
- Binary files (docs, software, data, applications)
- Pre-defined scope-escaped functions, dynamic code, lists and images
- Comments
- Ancillary properties, including state and "informal" data

Meeting the OpenROAD 6.2 Objectives …

- I – Providing explicit support for some key business requirements



I – Providing support for some key business requirements

1. Restyling OpenROAD to provide up-to-date look-and-feel(s)
2. Same-code (unchanged code) transformations
3. Generated UserClasses and Frame Displays
4. Active-map and Allocation (room, seat, …) capabilities
5. Richer out-of-the-box capabilities
6. Easier partitioning
7. Easier deployment

→ All the first five must be directly useful and easy to integrate, but also easy to customize and extend by clients and by Actian, easy to cannibalize, easy to understand …

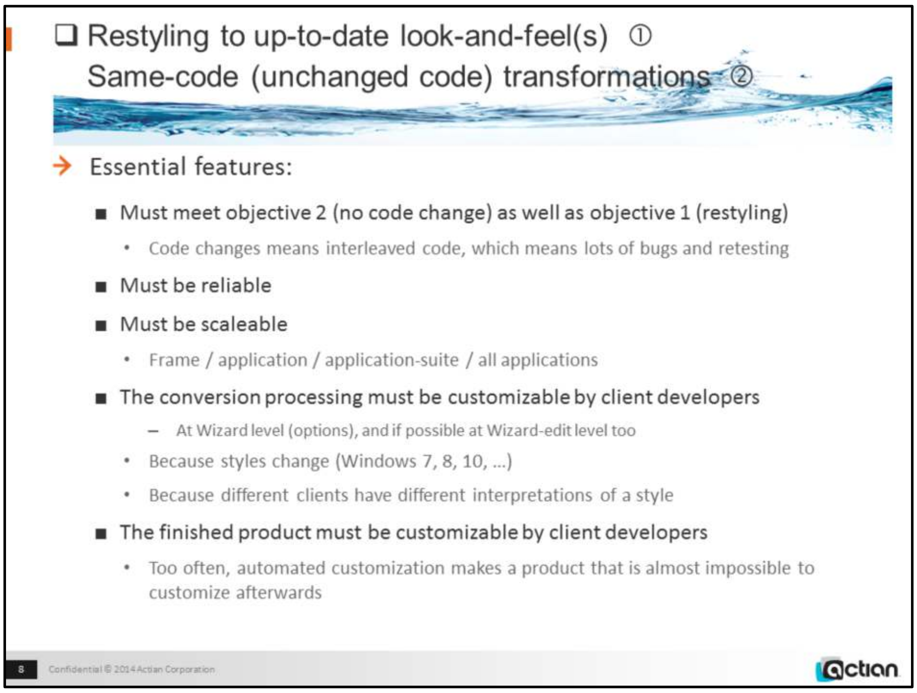
→ And appropriate to client needs.

7 Confidential © 2014 Actian Corporation

Under the first category comes the detailed simulation of Windows 7 styling, not only for new frames, but even existing applications; the W7Styler will convert all the frames in an application without needing to change a single line of code.
Other first-category changes are direct support for visual allocation systems (airline or hall seating, hotel rooms, etc); for "active" maps and diagrams, including irregular regions; and for the automatic generation of rich business classes and functional frame displays that can be used as-is or readily developed further to supply or replace the many bread-and-butter enduser frames that production applications include.

The first five of these features in particular must be, and are designed to be:
- directly useful
- easy to integrate
- easy to customize and extend by clients and by Actian
- easy to cannibalize
- easy to understand

… as well as actually being needed by clients!



❑ Restyling to up-to-date look-and-feel(s) ① Same-code (unchanged code) transformations ②

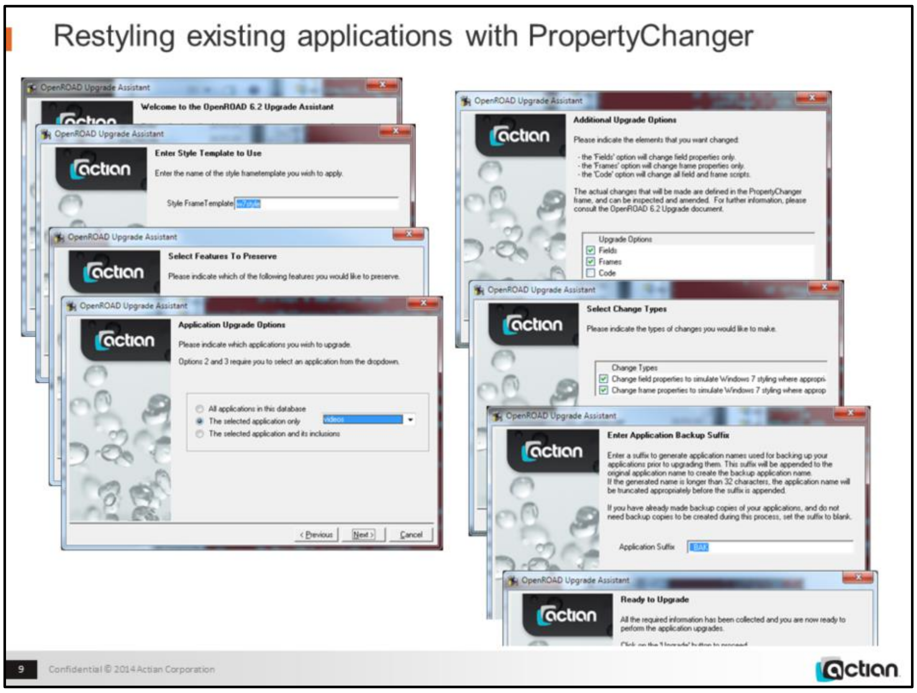
→ Essential features:

- Must meet objective 2 (no code change) as well as objective 1 (restyling)
  - Code changes means interleaved code, which means lots of bugs and retesting
- Must be reliable
- Must be scaleable
  - Frame / application / application-suite / all applications
- The conversion processing must be customizable by client developers
  - At Wizard level (options), and if possible at Wizard-edit level too
  - Because styles change (Windows 7, 8, 10, ...)
  - Because different clients have different interpretations of a style
- The finished product must be customizable by client developers
  - Too often, automated customization makes a product that is almost impossible to customize afterwards

8 Confidential © 2014 Actian Corporation

Numbers in circles in the title refer to the numbering of the sub-objectives on the previous page

Demo manager:
(Running this before the presentation displays a “topmost” toggle-menu for all of the demos listed in this presentation’s Notes sections)
w4gldev rundbapp remotehost::or62demos d201504_rundemos



Restyling existing applications with PropertyChanger

OpenROAD Upgrade Assistant

Welcome to the OpenROAD 6.2 Upgrade Assistant

OpenROAD Upgrade Assistant

Enter Style Template to Use

Enter the name of the style frametemplate you wish to apply.

Style FrameTemplate: w7style

OpenROAD Upgrade Assistant

Select Features To Preserve

Please indicate which of the following features you would like to preserve.

OpenROAD Upgrade Assistant

Application Upgrade Options

Please indicate which applications you wish to upgrade.

Options 2 and 3 require you to select an application from the dropdown.

- ○ All applications in this database
- ● The selected application only — videos
- ○ The selected application and its inclusions

< Previous | Next > | Cancel

OpenROAD Upgrade Assistant

Additional Upgrade Options

Please indicate the elements that you want changed:

- the 'Fields' option will change field properties only.
- the 'Frames' option will change frame properties only.
- the 'Code' option will change all field and frame scripts.

The actual changes that will be made are defined in the PropertyChanger frame, and can be inspected and amended. For further information, please consult the OpenROAD 6.2 Upgrade document.

Upgrade Options
- ✓ Fields
- ✓ Frames
- ☐ Code

OpenROAD Upgrade Assistant

Select Change Types

Please indicate the types of changes you would like to make.

Change Types
- ✓ Change field properties to simulate Windows 7 styling where appropri
- ✓ Change frame properties to simulate Windows 7 styling where approp

OpenROAD Upgrade Assistant

Enter Application Backup Suffix

Enter a suffix to generate application names used for backing up your applications prior to upgrading them. This suffix will be appended to the original application name to create the backup application name.
If the generated name is longer than 32 characters, the application name will be truncated appropriately before the suffix is appended.

If you have already made backup copies of your applications, and do not need backup copies to be created during this process, set the suffix to blank.

Application Suffix: _BAK

OpenROAD Upgrade Assistant

Ready to Upgrade

All the required information has been collected and you are now ready to perform the application upgrades.

9 Confidential © 2014 Actian Corporation

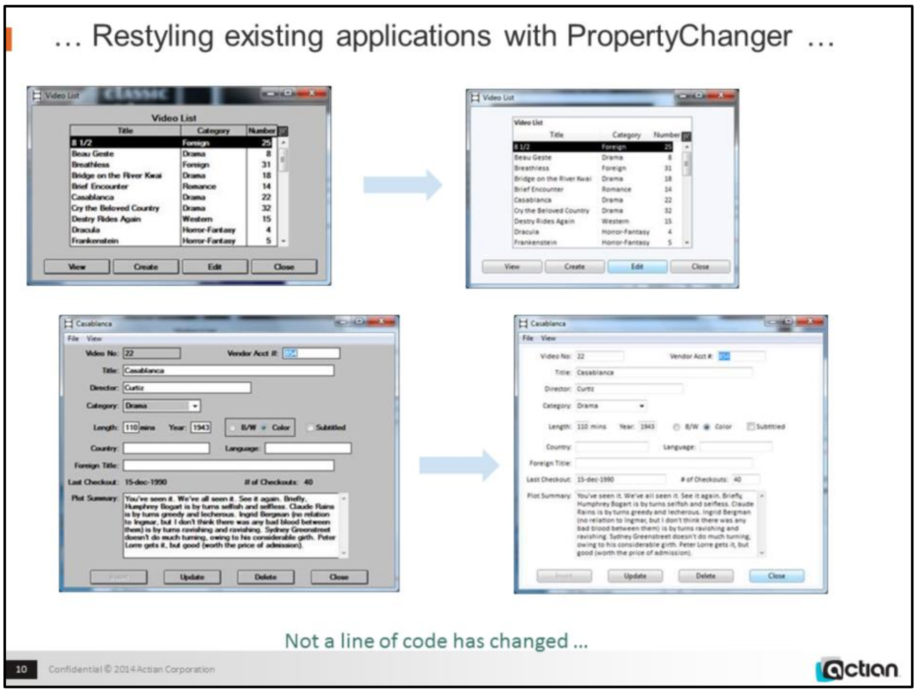
Demo:

w4gldev rundbapp remotehost::or62demos propertychanger –cw7styler
Default all options except application (choose D201504_VIDEOS)

- The conversion takes around 2 minutes

Then compare the restyled application with the old (_BAK) version:
w4gldev rundbapp remotehost::or62demos D201504_VIDEOS_BAK
w4gldev rundbapp remotehost::or62demos D201504_VIDEOS

Use Ctrl-Tab to move the inputfocus onto the buttonfields to show pulsing.
Mouse over the buttonfields and tablefield headers to show highlighting.



… Restyling existing applications with PropertyChanger …

Video List

Title	Category	Number
8 1/2	Foreign	25
Beau Geste	Drama	8
Breathless	Foreign	31
Bridge on the River Kwai	Drama	18
Brief Encounter	Romance	14
Casablanca	Drama	22
Cry the Beloved Country	Drama	32
Destry Rides Again	Western	15
Dracula	Horror-Fantasy	4
Frankenstein	Horror-Fantasy	5

Video No: 22
Vendor Acct #:
Title: Casablanca
Director: Curtiz
Category: Drama
Length: 110 mins
Year: 1943
B/W [✓ Color]
Subtitled
Country:
Language:
Foreign Title:
Last Checkout: 15-dec-1990
# of Checkouts: 40
Plot Summary: You've seen it. We've all seen it. See it again. Briefly, Humphrey Bogart is by turns selfish and selfless. Claude Rains is by turns greedy and lecherous. Ingrid Bergman (no relation to Ingmar, but I don't think there was any bad blood between them) is by turns ravishing and ravishing. Sydney Greenstreet doesn't do much turning, owing to his considerable girth. Peter Lorre gets it, but good (worth the price of admission).

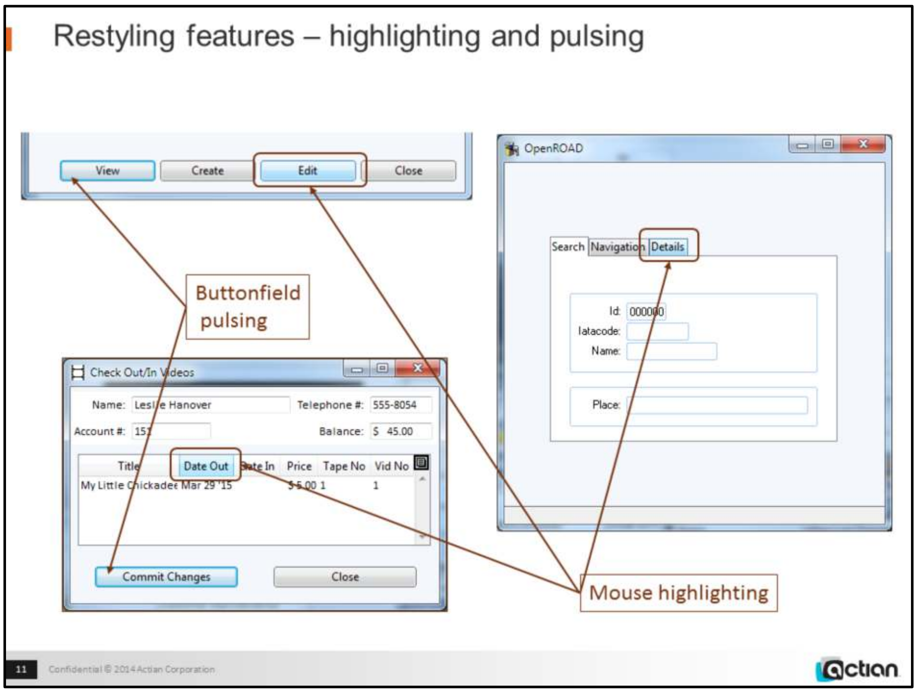
Not a line of code has changed …

10 Confidential © 2014 Actian Corporation

Restyling the old Videos application:

- no code change – not a single line.

Same result when tested on various Workbench applications, and client samples.



Restyling features – highlighting and pulsing

Id: 000000
Iatacode:
Name:
Place:

Name: Leslie Hanover   Telephone #: 555-8054
Account #: 151   Balance: $ 45.00

Title	Date Out	Date In	Price	Tape No	Vid No
My Little Chickadee	Mar 29 '15		$ 5.00	1	1

11 Confidential © 2014 Actian Corporation

Demo:
D201504_VideosConverted
Run application
Choose Check Out option
Enter 151 as customer account
Ctrl-Shift-Tab to move focus to “Commit Changes”
Hover mouse over Date Out column header

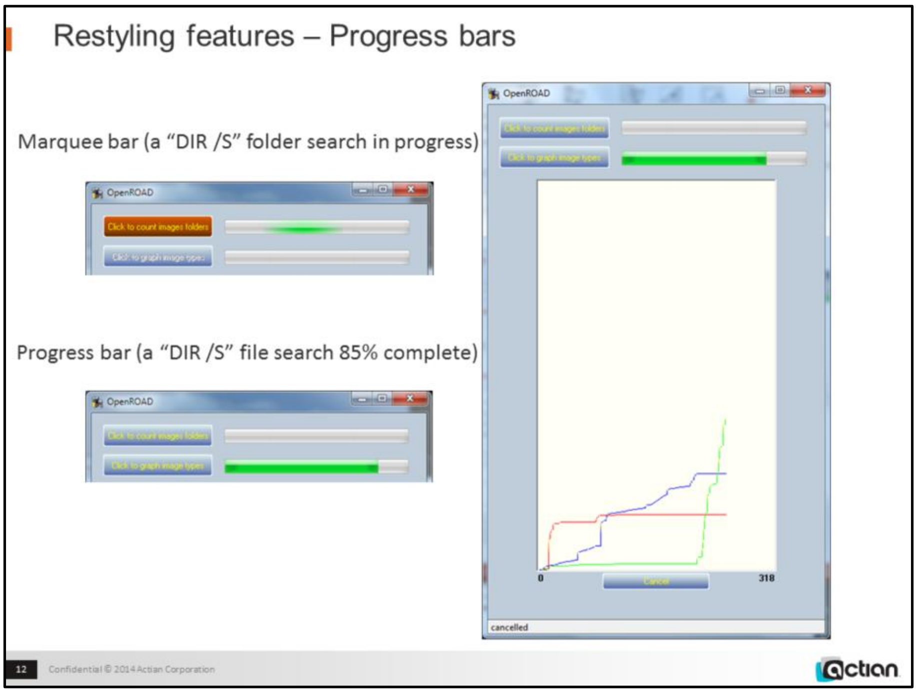
Demo:
D201504_BitmappedTabfolderTabs -cBDPTabHighlighing

See also Videos conversion.

Restyling applied to ButtonFields, EntryFields, TableField headers, TabFolder tabs, SubForms, other compositefields, FreeTrims, Mainbars, RectangleShapes, ControlButtons.

Most other fieldtypes are already Windows7-like style, since we used native widgets for them.

Font changed to Segoe UI 9


Restyling features – Progress bars

Marquee bar (a "DIR /S" folder search in progress)

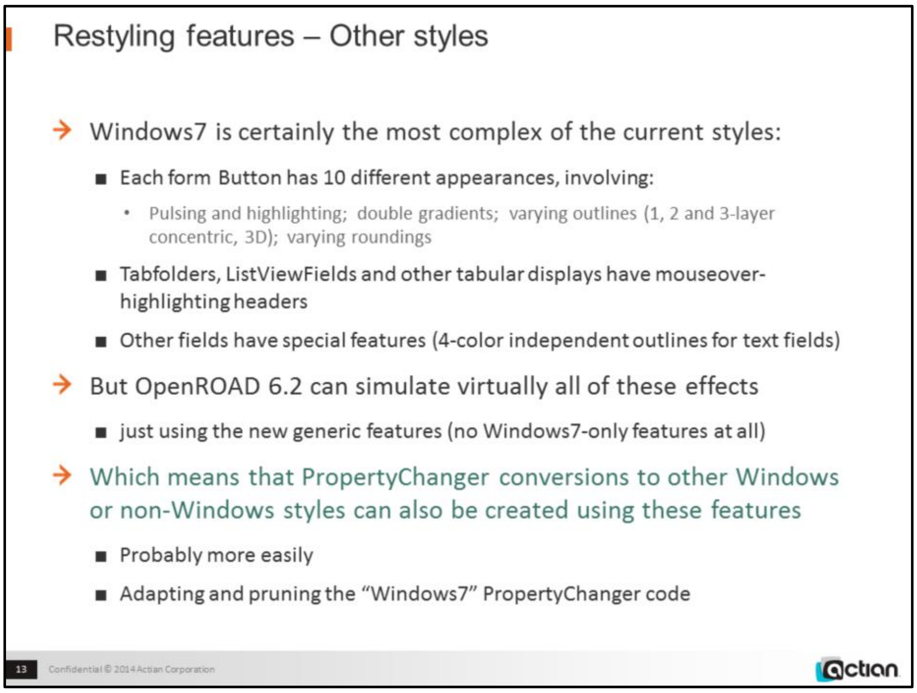
Progress bar (a "DIR /S" file search 85% complete)

12 Confidential © 2014 Actian Corporation

Demo:
D201504_DefinedResponses_Sprites –cprogressbars_coded

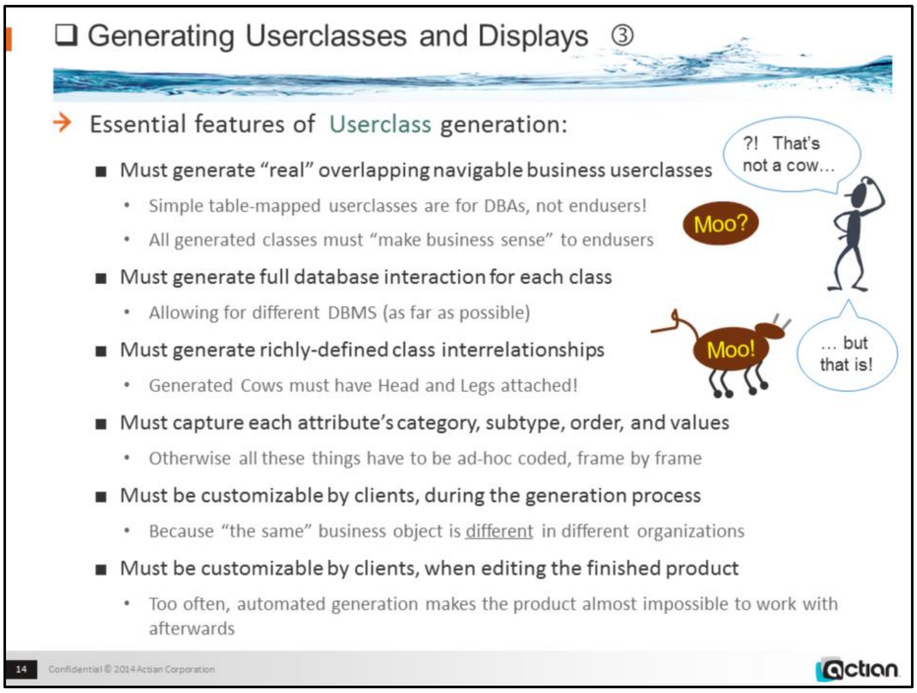
Marquee bars can be either "chase" or "zigzag" in appearance

Other points (covered in second presentation):
- The difference between chase and zigzag bars is just a handful of lines of 4GL code in the setup (no code is required at runtime).
- Progress bars need the 4GL to tell them what value to display, and when; no other runtime code is required.
- The graph itself is pure OpenROAD, no code other than the x,y calculations and a couple of system method calls



Restyling features – Other styles

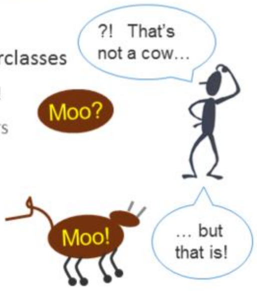
- Windows7 is certainly the most complex of the current styles:
  - Each form Button has 10 different appearances, involving:
    - Pulsing and highlighting; double gradients; varying outlines (1, 2 and 3-layer concentric, 3D); varying roundings
  - Tabfolders, ListViewFields and other tabular displays have mouseover-highlighting headers
  - Other fields have special features (4-color independent outlines for text fields)
- But OpenROAD 6.2 can simulate virtually all of these effects
  - just using the new generic features (no Windows7-only features at all)
- Which means that PropertyChanger conversions to other Windows or non-Windows styles can also be created using these features
  - Probably more easily
  - Adapting and pruning the "Windows7" PropertyChanger code



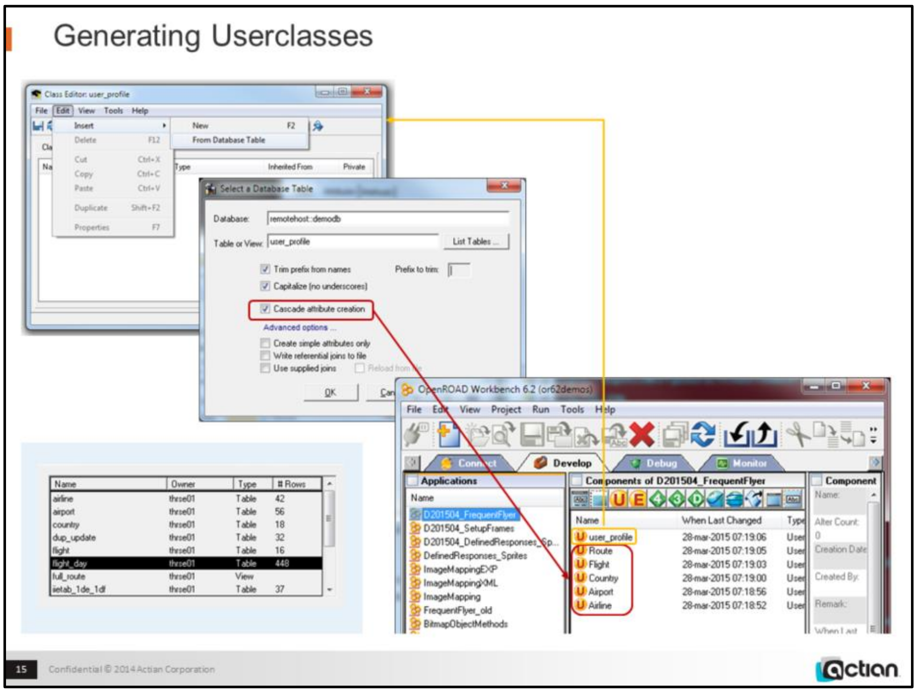
❑ Generating Userclasses and Displays ③

→ Essential features of Userclass generation:

- Must generate "real" overlapping navigable business userclasses
  - Simple table-mapped userclasses are for DBAs, not endusers!
  - All generated classes must "make business sense" to endusers
- Must generate full database interaction for each class
  - Allowing for different DBMS (as far as possible)
- Must generate richly-defined class interrelationships
  - Generated Cows must have Head and Legs attached!
- Must capture each attribute's category, subtype, order, and values
  - Otherwise all these things have to be ad-hoc coded, frame by frame
- Must be customizable by clients, during the generation process
  - Because "the same" business object is <u>different</u> in different organizations
- Must be customizable by clients, when editing the finished product
  - Too often, automated generation makes the product almost impossible to work with afterwards



actian



Generating Userclasses

Select a Database Table

Database: remotehost::demodb

Table or View: user_profile

- [✓] Trim prefix from names — Prefix to trim:
- [✓] Capitalize (no underscores)
- [✓] Cascade attribute creation
- [ ] Create simple attributes only
- [ ] Write referential joins to file
- [ ] Use supplied joins

Name	Owner	Type	# Rows
airline	thrse01	Table	42
airport	thrse01	Table	56
country	thrse01	Table	18
dup_update	thrse01	Table	32
flight	thrse01	Table	16
flight_day	thrse01	Table	448
full_route	thrse01	View	
iietab_1de_1df	thrse01	Table	37

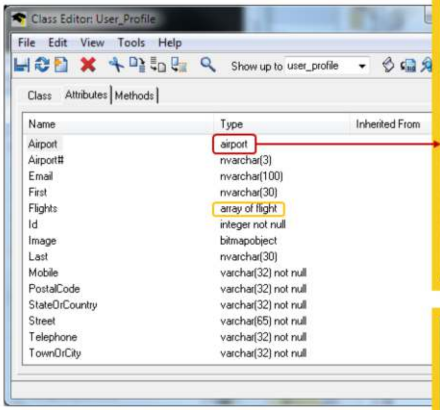
Components of D201504_FrequentFlyer

Name	When Last Changed	Type
user_profile	28-mar-2015 07:19:06	User
Route	28-mar-2015 07:19:05	User
Flight	28-mar-2015 07:19:03	User
Country	28-mar-2015 07:19:00	User
Airport	28-mar-2015 07:18:56	User
Airline	28-mar-2015 07:18:52	User

15 Confidential © 2014 Actian Corporation

Demo:
w4gldev runimage workbench.img –Tall -/appflags profile=or62demos application=d201504_frequentflyer
Create and edit User_profile userclass
Choose Attributes | Insert | From Database Table
Continue as shown

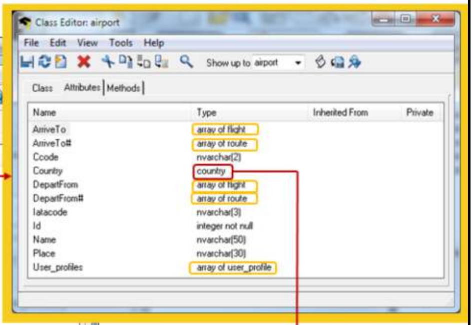
… Generating Userclasses …



Class Editor: User_Profile

Show up to user_profile

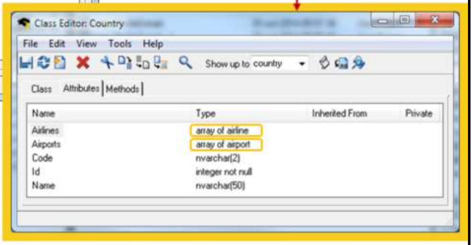
Name	Type	Inherited From
Airport	airport	
Airport#	nvarchar(3)	
Email	nvarchar(100)	
First	nvarchar(30)	
Flights	array of flight	
Id	integer not null	
Image	bitmapobject	
Last	nvarchar(30)	
Mobile	varchar(32) not null	
PostalCode	varchar(32) not null	
StateOrCountry	varchar(32) not null	
Street	varchar(65) not null	
Telephone	varchar(32) not null	
TownOrCity	varchar(32) not null	



Class Editor: airport

Show up to airport

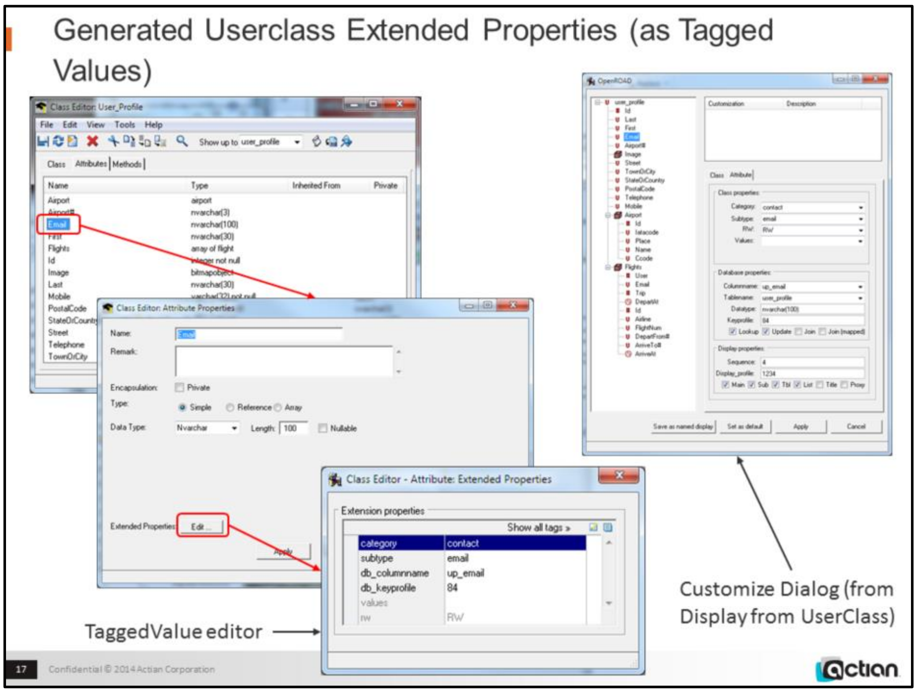
Name	Type	Inherited From	Private
ArriveTo	array of flight		
ArriveTo#	array of route		
Ccode	nvarchar(2)		
Country	country		
DepartFrom	array of flight		
DepartFrom#	array of route		
Iatacode	nvarchar(3)		
Id	integer not null		
Name	nvarchar(50)		
Place	nvarchar(30)		
User_profiles	array of user_profile		



Class Editor: Country

Show up to country

Name	Type	Inherited From	Private
Airlines	array of airline		
Airports	array of airport		
Code	nvarchar(2)		
Id	integer not null		
Name	nvarchar(50)		



Generated Userclass Extended Properties (as Tagged Values)

Class Editor: User_Profile

Show up to: user_profile

Name	Type	Inherited From	Private
Airport	airport		
Airport#	nvarchar(3)		
Email	nvarchar(100)		
First	nvarchar(30)		
Flights	array of flight		
Id	integer not null		
Image	bitmapobject		
Last	nvarchar(30)		
Mobile	varchar(32) not null		
PostalCode			
StateOrCountry			
Street			
Telephone			
TownOrCity			

Class Editor: Attribute Properties

Name: Email
Remark:
Encapsulation: ☐ Private
Type: ● Simple ○ Reference ○ Array
Data Type: Nvarchar Length: 100 ☐ Nullable
Extended Properties: Edit ...

Class Editor - Attribute: Extended Properties

Extension properties

category	contact
subtype	email
db_columnname	up_email
db_keyprofile	84
values	
rw	RW

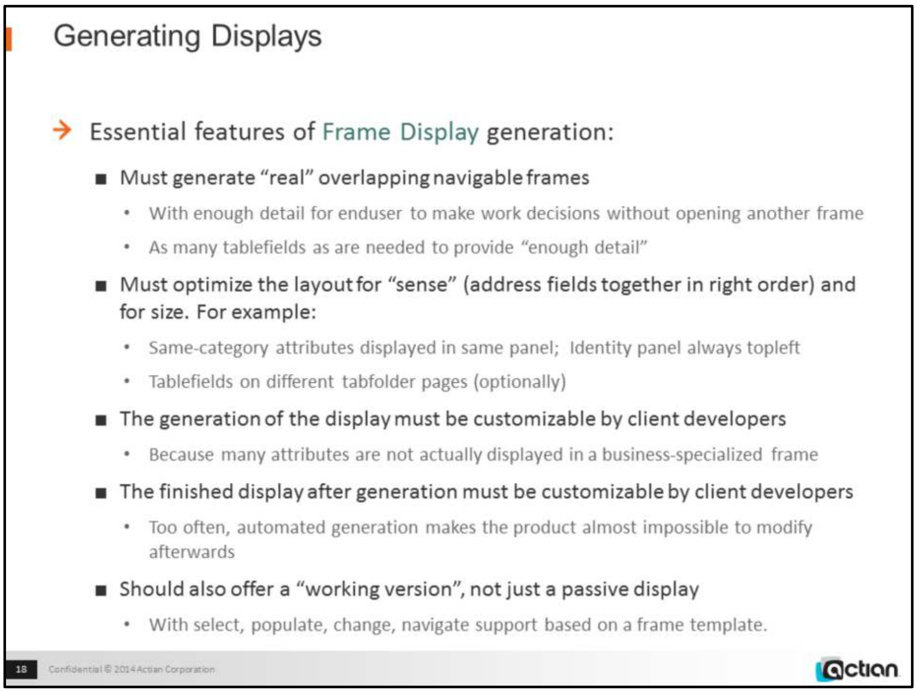
TaggedValue editor

OpenROAD

Category: contact
Subtype: email
RW: RW
Values:
Columnname: up_email
Tablename: user_profile
Datatype: nvarchar(100)
Keyprofile: 84
☑ Lookup ☑ Update ☐ Join ☐ Join (mapped)
Sequence: 4
Display_profile: 1234
☑ Main ☑ Sub ☑ Tbl ☑ List ☐ Title ☐ Proxy

Customize Dialog (from Display from UserClass)

The TaggedValues arrays for the userclass and for each property contain much useful ancillary information,
And the userclass's Queries array contains ready-to-run QueryObjects

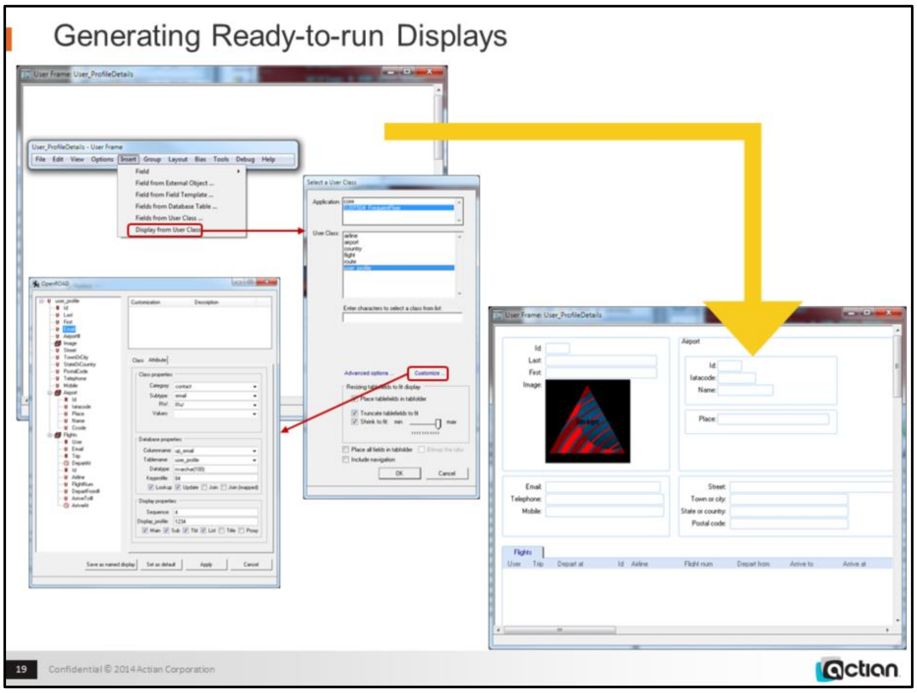


Generating Displays

→ Essential features of Frame Display generation:

- Must generate "real" overlapping navigable frames
  - With enough detail for enduser to make work decisions without opening another frame
  - As many tablefields as are needed to provide "enough detail"
- Must optimize the layout for "sense" (address fields together in right order) and for size. For example:
  - Same-category attributes displayed in same panel; Identity panel always topleft
  - Tablefields on different tabfolder pages (optionally)
- The generation of the display must be customizable by client developers
  - Because many attributes are not actually displayed in a business-specialized frame
- The finished display after generation must be customizable by client developers
  - Too often, automated generation makes the product almost impossible to modify afterwards
- Should also offer a "working version", not just a passive display
  - With select, populate, change, navigate support based on a frame template.

18 Confidential © 2014 Actian Corporation



Generating Ready-to-run Displays

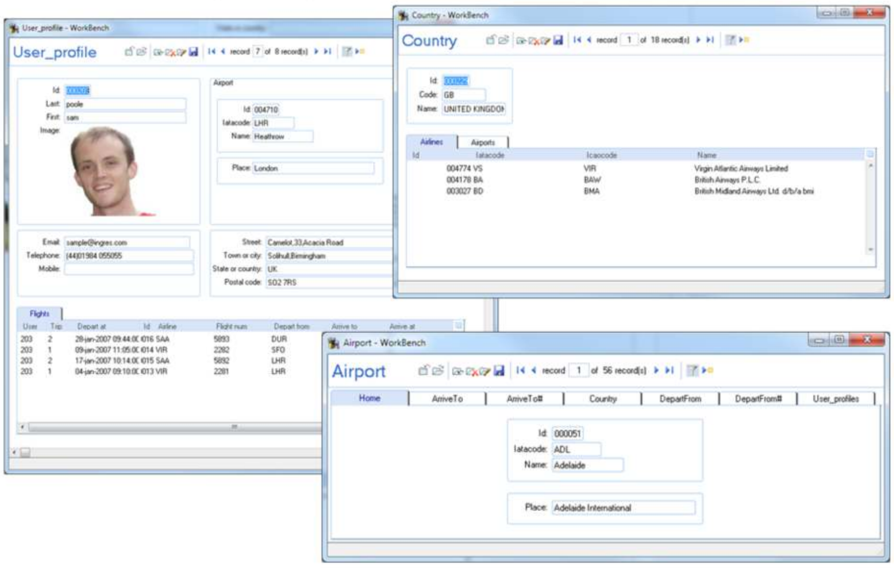
Demo:

w4gldev runimage workbench.img –Tall -/appflags profile=or62demos application=d201504_frequentflyer

Create and edit User_profileDetails frame from the active_display frametemplate
Choose Insert | Display from User Class
    The Select a User Class dialog appears
Select d201504_frequentflyer from the Application list
Select user_profile from the User Class list
    Note that there are rich Advanced and Customize options to tune the choice of fields to display, their properties, and the overall layout, and which capabilities to include (interframe navigation, for example)
Click OK and wait for the display to be generated
    The generated frame shows the userclass attributes as fields, grouped into panels by their category (identity fields, location fields, contact fields, etc), with object panels on the right hand side (Airport).
    Tablefields are at the bottom, as pages of a tabfolder (most business objects in the real world have several array attributes, not just one or none).
    Field order is preserved, so that name fields and address fields are in the right order.
    The size of the frame is as close as possible to whatever size you gave the Frame Editor.

<continued on next slide>

… Generating Ready-to-run Displays …



User_profile - WorkBench

User_profile — record 7 of 8 record(s)

Id:	000203
Last:	poole
First:	sam
Image:	

Airport

Id:	004710
Iatacode:	LHR
Name:	Heathrow
Place:	London

Email:	sample@ingres.com
Telephone:	(44)01984 055055
Mobile:	

Street:	Camelot,33,Acacia Road
Town or city:	Solihull,Birmingham
State or county:	UK
Postal code:	SO2 7RS

Flights

User	Trip	Depart at	Id	Airline	Flight num	Depart from	Arrive to	Arrive at
203	2	28-jan-2007 09:44:0C	016	SAA	5893	DUR		
203	1	09-jan-2007 11:05:0C	014	VIR	2282	SFO		
203	2	17-jan-2007 10:14:0C	015	SAA	5892	LHR		
203	1	04-jan-2007 09:10:0C	013	VIR	2281	LHR		

Country - WorkBench

Country — record 1 of 18 record(s)

Id:	000225
Code:	GB
Name:	UNITED KINGDOM

Airlines | Airports

Id	Iatacode	Icaocode	Name
004774	VS	VIR	Virgin Atlantic Airways Limited
004178	BA	BAW	British Airways P.L.C.
003027	BD	BMA	British Midland Airways Ltd. d/b/a bmi

Airport - WorkBench

Airport — record 1 of 56 record(s)

Home | ArriveTo | ArriveTo# | Country | DepartFrom | DepartFrom# | User_profiles

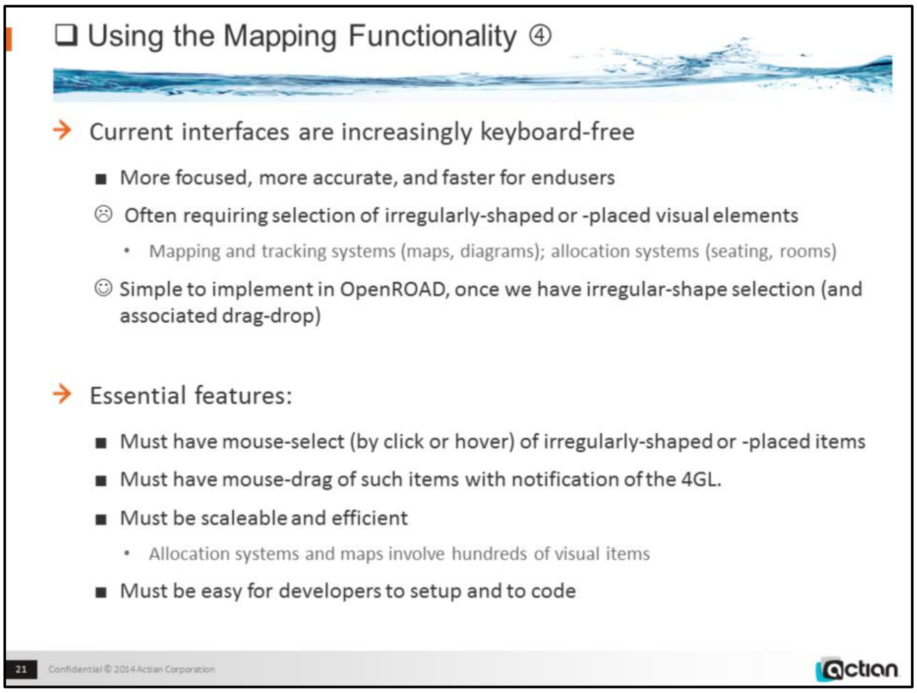
Id:	000051
Iatacode:	ADL
Name:	Adelaide
Place:	Adelaide International

20 Confidential © 2014 Actian Corporation

Demo (continued):

Hit F5 to run the frame
Rightclick the Open (second) toolbar icon and choose the Selected Item option
Doubleclick the sample@ingres.com entry
    The corresponding user is displayed
    Note that the frame can be edited and the changes saved
Leftclick the Open (second) toolbar icon
    All the users are selected, and the first one displayed
Click the Next (ninth, right-arrow) toolbar icon repeatedly
    Each user in the set is displayed in turn
Click the Menu icon in the topright of the Airport panel and choose the Open option
    The generated Airport frame is displayed, populated with the airport details, including the lists of arrivals and departures, both the scheduled ones and the actual flights this user has taken
    Note that the generation process is completely generic - you would need to change the Airport frame tabs to Flight Arrivals, Flight Departures Scheduled Arrivals, Scheduled Departures.

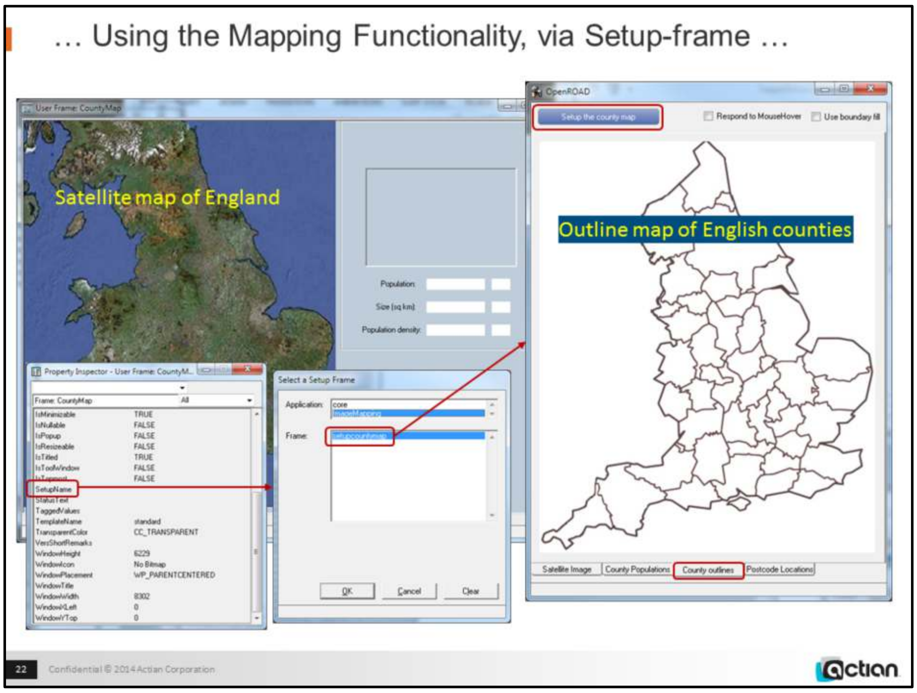
Each generated frame contains about 60 lines of code – the RequestManager helper class in Core does all the real 4GL work, including database interaction, navigation, item selection and traverse. Each of these functions can be overridden in the frame by a local procedure.
RequestManager is a UserClass, so it can be inherited and customized.



❑ Using the Mapping Functionality ④

- Current interfaces are increasingly keyboard-free
  - More focused, more accurate, and faster for endusers
  - ☹ Often requiring selection of irregularly-shaped or -placed visual elements
    - Mapping and tracking systems (maps, diagrams); allocation systems (seating, rooms)
  - ☺ Simple to implement in OpenROAD, once we have irregular-shape selection (and associated drag-drop)

- Essential features:
  - Must have mouse-select (by click or hover) of irregularly-shaped or -placed items
  - Must have mouse-drag of such items with notification of the 4GL.
  - Must be scaleable and efficient
    - Allocation systems and maps involve hundreds of visual items
  - Must be easy for developers to setup and to code



… Using the Mapping Functionality, via Setup-frame …

Frame: CountyMap — All

IsMinimizable	TRUE
IsNullable	FALSE
IsPopup	FALSE
IsResizeable	FALSE
IsTitled	TRUE
IsToolWindow	FALSE
IsTopmost	FALSE
SetupName	
StatusText	
TaggedValues	
TemplateName	standard
TransparentColor	CC_TRANSPARENT
VersShortRemarks	
WindowHeight	6229
WindowIcon	No Bitmap
WindowPlacement	WP_PARENTCENTERED
WindowTitle	
WindowWidth	8302
WindowXLeft	0
WindowYTop	0

Application: core, imageMapping

Frame: setupcountymap

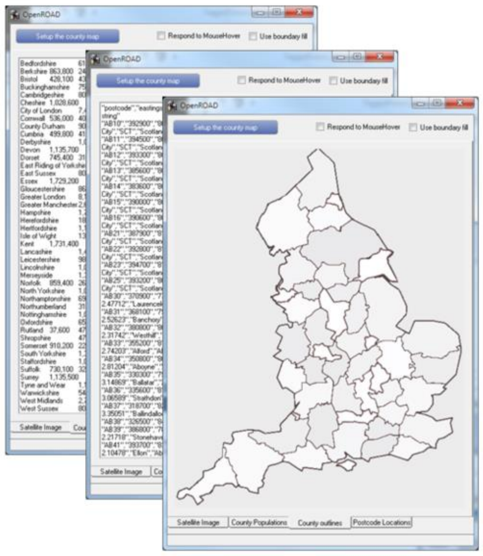
22 Confidential © 2014 Actian Corporation

Demo:
This demo uses the new setup-frame capability, to minimize runtime code

w4gldev runimage workbench.img –Tall -/appflags profile=or62demos application=d201504_imagemapping component=countymap, command=open

Select the SetupName entry in the Property Inspector
    The Setup Frame dialog will appear
Select the "D201504_ImageMapping" application and the "setupcountymap" frame, and click OK
    The SetupCountyMap setup frame will run
Click the "County Outlines" tab
    An outline map of English counties will appear
Click the "Setup the county map" button
    After a few seconds each county will be coloured a different shade of grey
Close the setup frame
    Note that the setupframe has analysed the data it has and has setup the CountyMap frame to run whenever the enduser needs it.
<instructions continue 2 slides on>

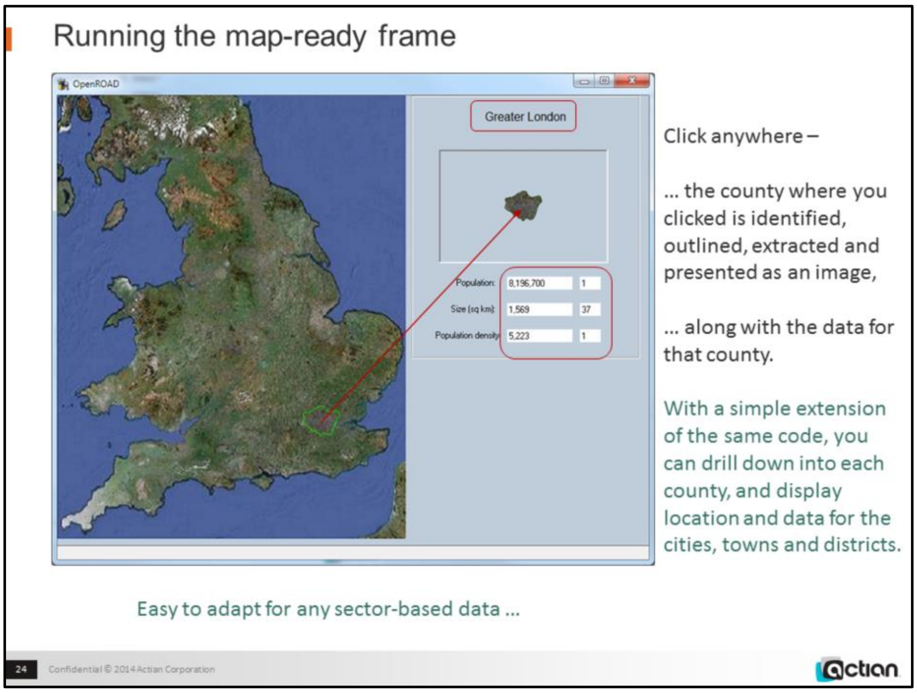
Running the Mapping setup-frame



```
Bedfordshire      61
Berkshire 863,800 24
Bristol   428,100 43
Buckinghamshire   75
Cambridgeshire    80
Cheshire  1,028,600
City of London    7,4
Cornwall  536,000 40
County Durham     90
Cumbria   499,800 41
Derbyshire        1,0
Devon     1,135,700
Dorset    745,400 31
East Riding of Yorkshi
East Sussex       80
Essex     1,729,200
Gloucestershire   86
Greater London    8,1
Greater Manchester 2,6
Hampshire         1,7
Herefordshire     18
Hertfordshire     1,1
Isle of Wight     13
Kent      1,731,400
Lancashire        1,4
Leicestershire    98
Lincolnshire      1,0
Merseyside        1,3
Norfolk   859,400 26
North Yorkshire   1,0
Northamptonshire  69
Northumberland    31
Nottinghamshire   1,0
Oxfordshire       65
Rutland   37,600  47
Shropshire        47
Somerset 910,200  22
South Yorkshire   1,3
Staffordshire     1,0
Suffolk   730,100 32
Surrey    1,135,500
Tyne and Wear     1,1
Warwickshire      54
West Midlands     2,7
West Sussex       80
```

```
"postcode","eastings
string"
"AB10","392900","8
City","SCT","Scotlan
"AB11","394500","8
City","SCT","Scotlan
"AB12","393300","8
City","SCT","Scotlan
"AB13","385600","8
City","SCT","Scotlan
"AB14","383600","8
City","SCT","Scotlan
"AB15","390000","8
City","SCT","Scotlan
"AB16","390600","8
City","SCT","Scotlan
"AB21","387900","8
City","SCT","Scotlan
"AB22","392800","8
City","SCT","Scotlan
"AB23","394700","8
City","SCT","Scotlan
"AB25","393200","8
City","SCT","Scotlan
"AB30","370900","7
2.47712","Laurencek
"AB31","368100","7
2.52623","Banchory"
"AB32","380800","8
2.31742","Westhill",
"AB33","355200","8
2.74203","Alford","A
"AB34","350800","8
2.81204","Aboyne",
"AB35","330300","7
3.14869","Ballatar",
"AB36","335600","8
3.06589","Strathdon"
"AB37","318700","8
3.35051","Ballindallo
"AB38","329500","8
"AB39","386800","7
2.21718","Stonehav
"AB41","393700","8
2.10478","Ellon","Ab
```

- Prepares the County Demographics frame
  - The setup frame is use-and-lose
- All done using tagged values and new bitmap and stringobject methods
  - Code is entirely generic 4GL
  - Data is simple and freely available online



Running the map-ready frame

Population:	8,196,700	1
Size (sq km):	1,569	37
Population density:	5,223	1

Click anywhere –

... the county where you clicked is identified, outlined, extracted and presented as an image,

... along with the data for that county.

With a simple extension of the same code, you can drill down into each county, and display location and data for the cities, towns and districts.

Easy to adapt for any sector-based data ...

Demo (continued):

Run the CountyMap frame

Click any point in SouthEast England on the satellite image

The county under the mouse is outlined in green, the name and demographic data for that county appear on the right, and a satellite image of that county appears above the data

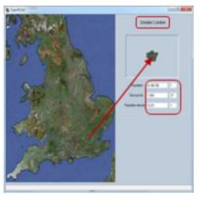
Click outher points in SouthEast England to show different counties and their data

Note that this is a genuinely useful facility, presenting genuinely useful data, using a simple generic mechanism adaptable to a wide range of business cases.

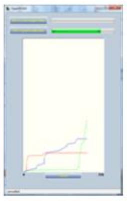
Select CountyMap, then choose File | Revert to last saved

❑ Dynamic Tracking Systems ④

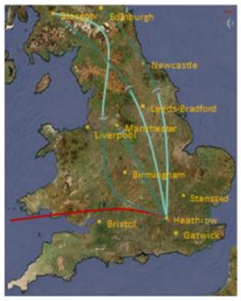
- If we add OpenROAD 6.2 Sprites to the Mapping capability …
- We can implement dynamic tracking systems
  - Flight tracking
  - Transport/delivery tracking
- With all the same advantages of simplicity, scaling, and ease of use



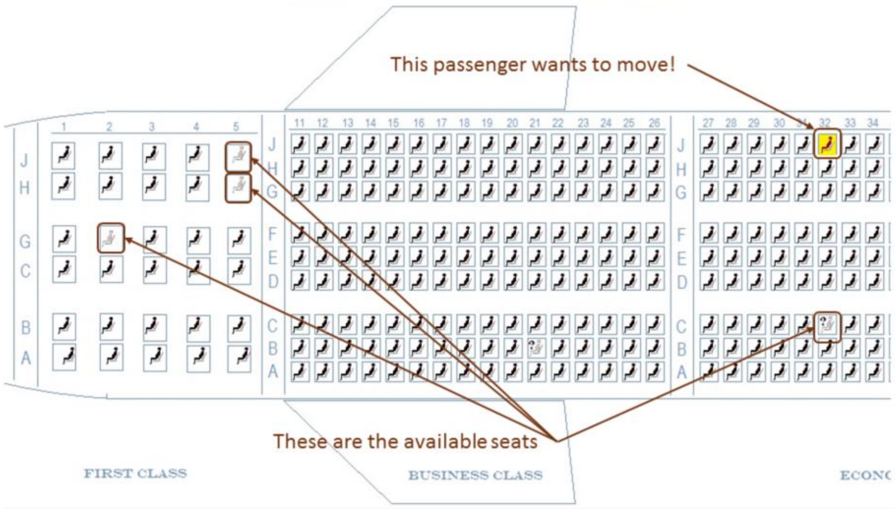
+



=



❑ Seat/Room Allocation Systems ④

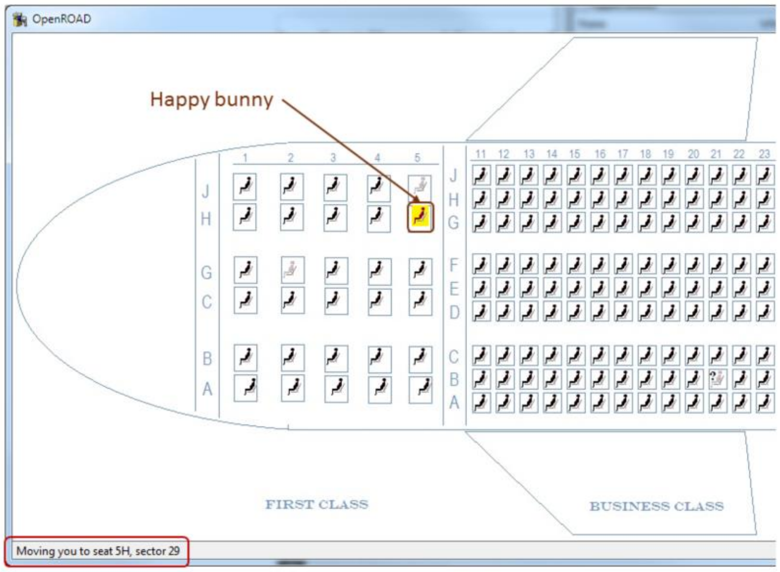


Demo:
D201504_AllocationSystems -cAirlineSeating
Drag the “yellow” passenger to one of the available seats, and drop the passenger there
See next slide for result:

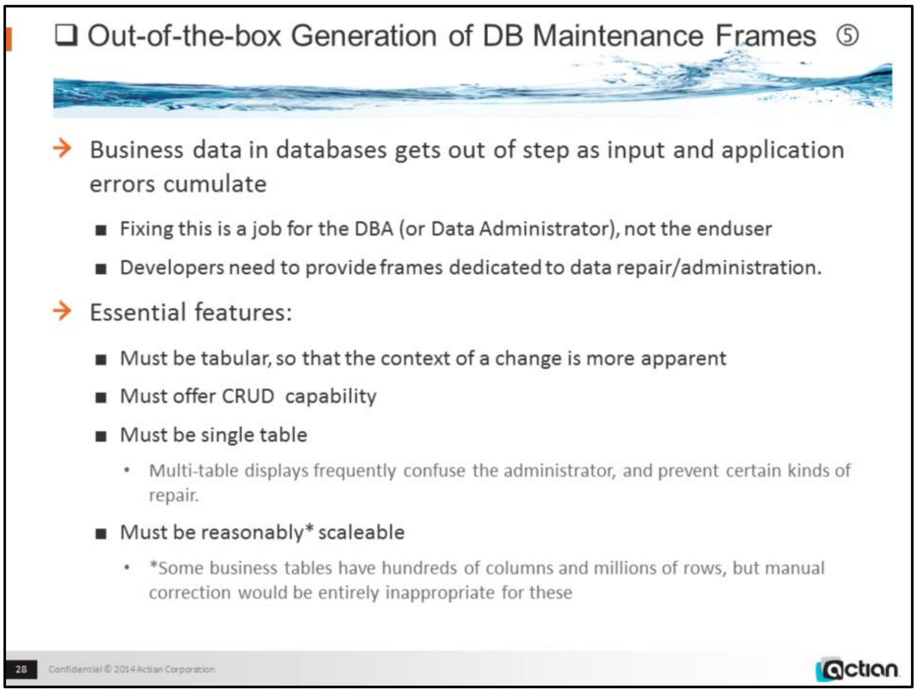
- The passenger now occupies that seat, the previous seat is now available, a status bar message identifies the new seat.

One field, 1 bitmap, 4 sprite images. There are no “passengers” on the initial bitmap.

… Seat/Room Allocation Systems …



Moving you to seat 5H, sector 29

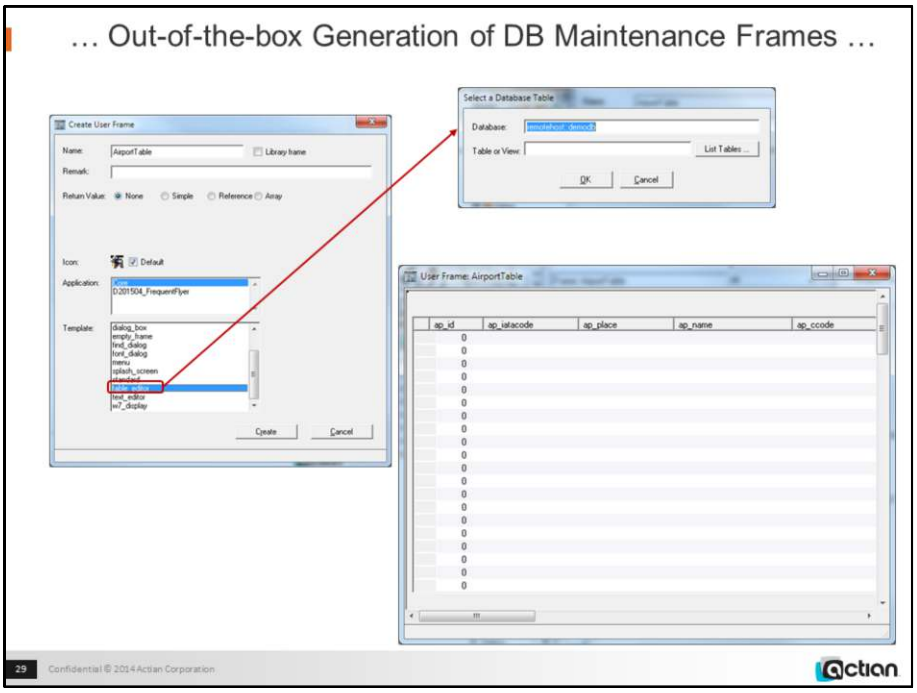

❑ Out-of-the-box Generation of DB Maintenance Frames ⑤

→ Business data in databases gets out of step as input and application errors cumulate

- Fixing this is a job for the DBA (or Data Administrator), not the enduser
- Developers need to provide frames dedicated to data repair/administration.

→ Essential features:

- Must be tabular, so that the context of a change is more apparent
- Must offer CRUD capability
- Must be single table
  - Multi-table displays frequently confuse the administrator, and prevent certain kinds of repair.
- Must be reasonably* scaleable
  - *Some business tables have hundreds of columns and millions of rows, but manual correction would be entirely inappropriate for these



… Out-of-the-box Generation of DB Maintenance Frames …

Create User Frame

Name: AirportTable
Library frame
Remark:
Return Value: [✓ None] Simple Reference Array
Icon: [✓ Default]
Application: D201504_FrequentFlyer
Template: dialog_box, empty_frame, find_dialog, font_dialog, menu, splash_screen, standard, [✓ table_editor], text_editor, w7_display

Select a Database Table

Database: remotehost::demodb
Table or View:

User Frame: AirportTable

	ap_id	ap_iatacode	ap_place	ap_name	ap_ccode
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				
	0				

Demo:

```
w4gldev runimage workbench.img –Tall -/appflags profile=or62demos application=d201504_outofbox
```

Create frame AirportMaintenance using table_editor template

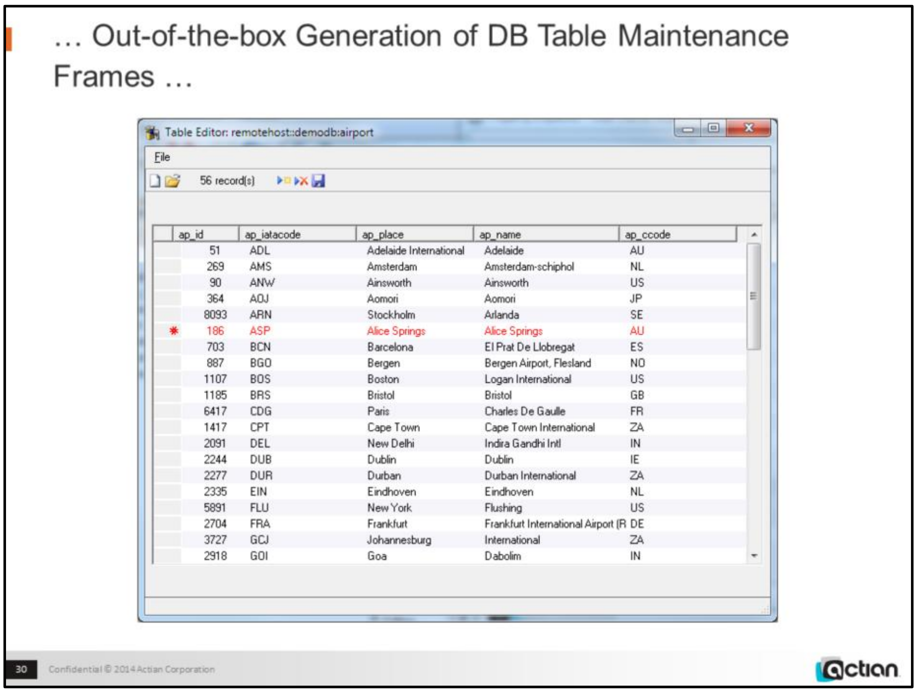
 A table-selection dialog appears

Specify "airport" table from the table list and click OK

 A tablefield-based display is created, whose columns correspond to the airport table's columns

Save the frame

<instructions continue on next slide>



… Out-of-the-box Generation of DB Table Maintenance Frames …

Table Editor: remotehost::demodb:airport

File

56 record(s)

	ap_id	ap_iatacode	ap_place	ap_name	ap_ccode
	51	ADL	Adelaide International	Adelaide	AU
	269	AMS	Amsterdam	Amsterdam-schiphol	NL
	90	ANW	Ainsworth	Ainsworth	US
	364	AOJ	Aomori	Aomori	JP
	8093	ARN	Stockholm	Arlanda	SE
*	186	ASP	Alice Springs	Alice Springs	AU
	703	BCN	Barcelona	El Prat De Llobregat	ES
	887	BGO	Bergen	Bergen Airport, Flesland	NO
	1107	BOS	Boston	Logan International	US
	1185	BRS	Bristol	Bristol	GB
	6417	CDG	Paris	Charles De Gaulle	FR
	1417	CPT	Cape Town	Cape Town International	ZA
	2091	DEL	New Delhi	Indira Gandhi Intl	IN
	2244	DUB	Dublin	Dublin	IE
	2277	DUR	Durban	Durban International	ZA
	2335	EIN	Eindhoven	Eindhoven	NL
	5891	FLU	New York	Flushing	US
	2704	FRA	Frankfurt	Frankfurt International Airport (R	DE
	3727	GCJ	Johannesburg	International	ZA
	2918	GOI	Goa	Dabolim	IN

Demo (continued):

Choose File | Open

The table populates with all the data from the database airport table

Select one or many records

Selected records have red text, and the current row ismarked with a red asterisk

Scroll the data

Insert a row (using the File menu or the toolbar icons)

Type new data into the added row

Save the new record

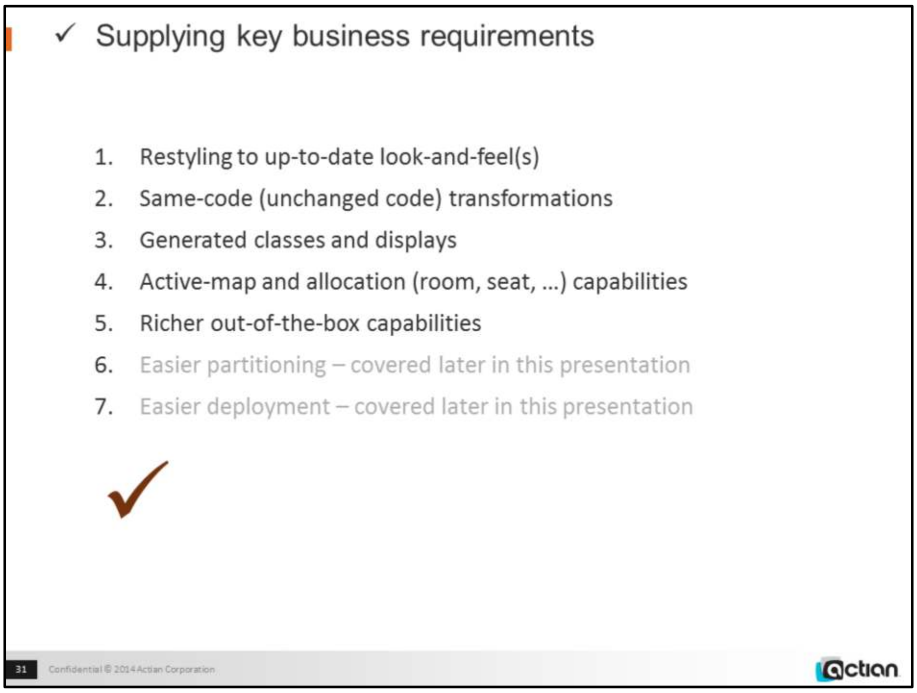
Close and reopen the frame and choose File | Open and scroll if necessary to confirm that the record you created is in the database

The added record is there

Delete the record and save

Close and reopen the frame and choose File | Open and scroll if necessary to confirm that the record you deleted is no longer in the database

The added record is gone



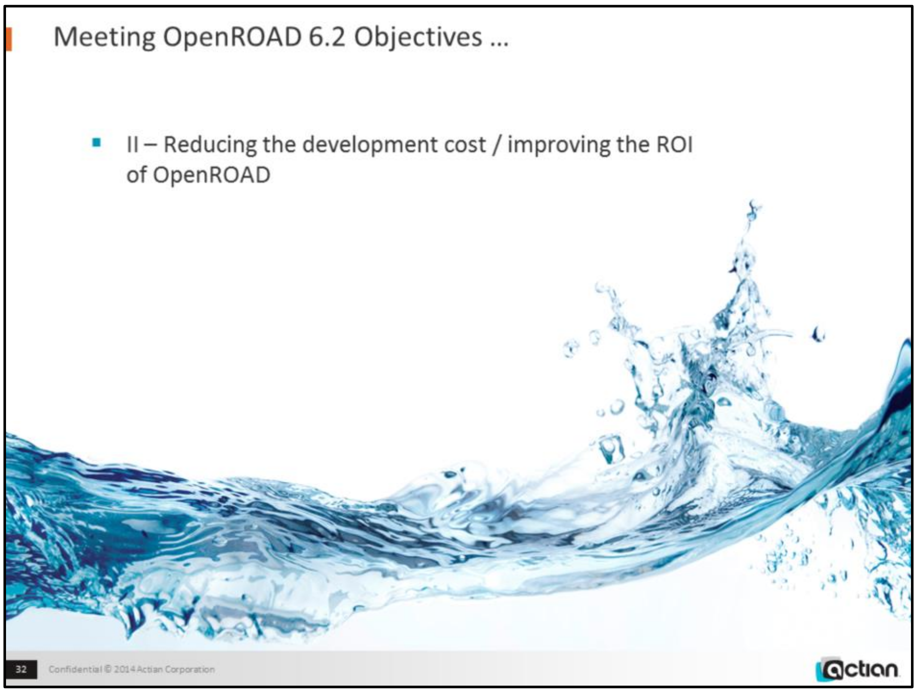
✓ Supplying key business requirements

1. Restyling to up-to-date look-and-feel(s)
2. Same-code (unchanged code) transformations
3. Generated classes and displays
4. Active-map and allocation (room, seat, …) capabilities
5. Richer out-of-the-box capabilities
6. Easier partitioning – covered later in this presentation
7. Easier deployment – covered later in this presentation

✓

31

actian



Meeting OpenROAD 6.2 Objectives …

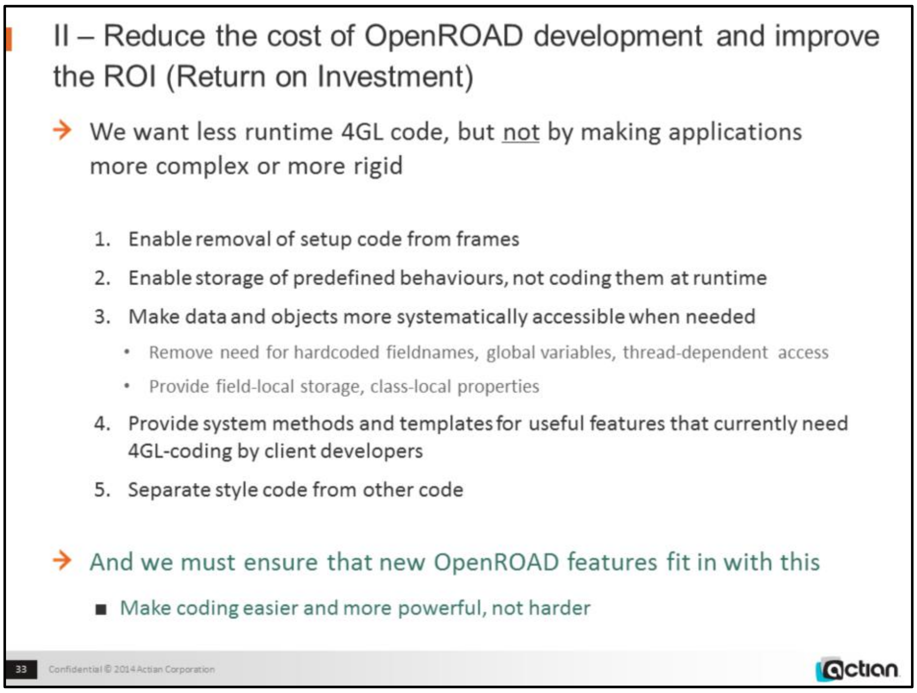
- II – Reducing the development cost / improving the ROI of OpenROAD

32 Confidential © 2014 Actian Corporation

OpenROAD 6.2 introduces many generic enabling-changes, including:

New bitmap, string, field, helper-class methods
New intrinsic behaviors
More system classes saved with components
Prestored behaviors + setupframe
Custom restyling support
Useful mapping support

These changes directly address the cost-reduction requirement. In particular the ability to prestore TaggedValues and Defined Behaviors, and the frame-handling features that RequestManager provides, enable dramatic simplification and reduction in the 4GL code required to manage frame displays, and its destabilizing effect on other frame code.
It is now possible and straightforward to provide rich custom restyling of existing applications without changing the existing code: Windows 7–like restying is provided as a click-driven conversion out of the box; other restylings are equally possible to provide. The mapping support enables straightforward implementation of seat and room-allocation systems, and useful implementations of simple geographic-location systems.



II – Reduce the cost of OpenROAD development and improve the ROI (Return on Investment)

→ We want less runtime 4GL code, but not by making applications more complex or more rigid

1. Enable removal of setup code from frames
2. Enable storage of predefined behaviours, not coding them at runtime
3. Make data and objects more systematically accessible when needed
   - Remove need for hardcoded fieldnames, global variables, thread-dependent access
   - Provide field-local storage, class-local properties
4. Provide system methods and templates for useful features that currently need 4GL-coding by client developers
5. Separate style code from other code

→ And we must ensure that new OpenROAD features fit in with this

- Make coding easier and more powerful, not harder

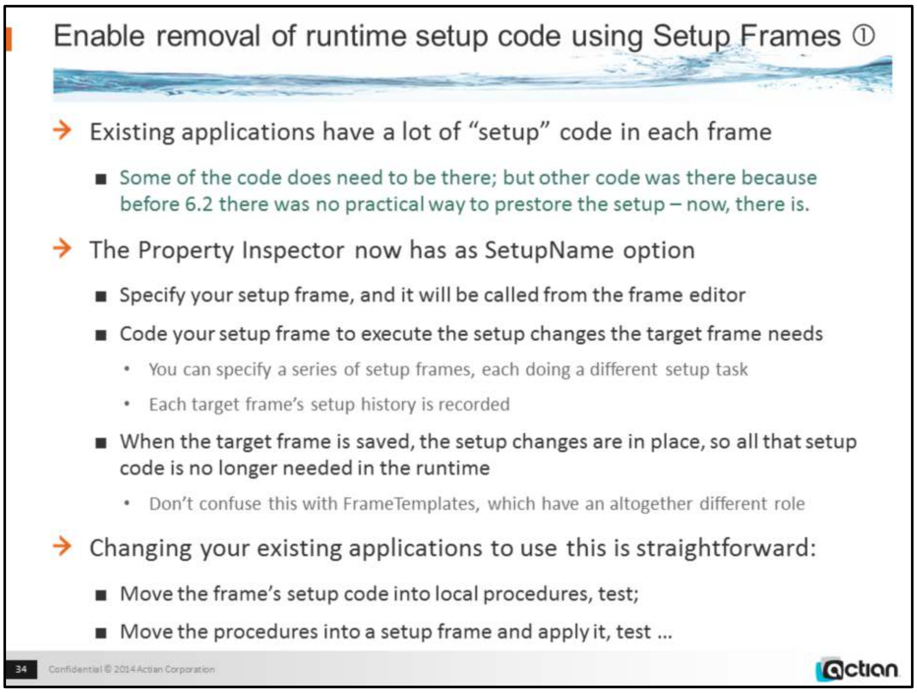
33 Confidential © 2014 Actian Corporation

The result to be
- More exactly correct
- More lowcost-extensible
- More lowcost-maintainable
- More saleable to management (internal and external)
- More timely

The development to be
- Less risky
- Easier to implement
- Less skill required
- Less rollback cost
- Smaller
- Less copy-paste-amend
- Less compromise required
- Less “provideds” (provided all frames look like X, provided architecture is master-detail, etc etc)
- More testable
- Less bug-prone

Must be directly useful, easy to integrate, easy to understand.
Must make a difference to the cost not only of new development and use of the new features, but also ongoing maintenance of existing features

Should reduce the number of coded events needed to achieve a single effect



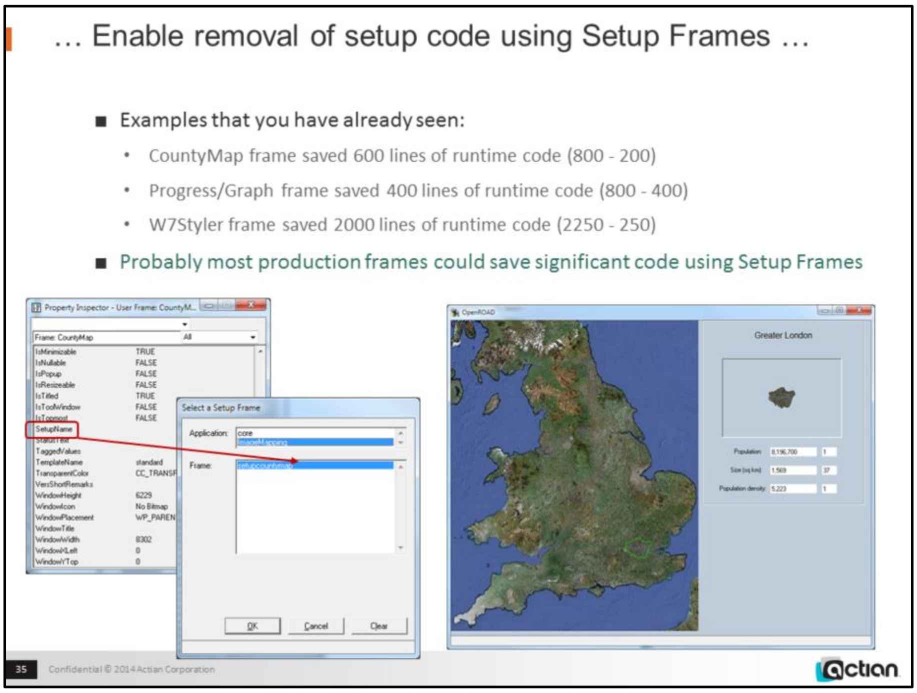
Enable removal of runtime setup code using Setup Frames ①

- Existing applications have a lot of “setup” code in each frame
  - Some of the code does need to be there; but other code was there because before 6.2 there was no practical way to prestore the setup – now, there is.
- The Property Inspector now has as SetupName option
  - Specify your setup frame, and it will be called from the frame editor
  - Code your setup frame to execute the setup changes the target frame needs
    - You can specify a series of setup frames, each doing a different setup task
    - Each target frame’s setup history is recorded
  - When the target frame is saved, the setup changes are in place, so all that setup code is no longer needed in the runtime
    - Don’t confuse this with FrameTemplates, which have an altogether different role
- Changing your existing applications to use this is straightforward:
  - Move the frame’s setup code into local procedures, test;
  - Move the procedures into a setup frame and apply it, test …

34 Confidential © 2014 Actian Corporation

Numbers in circles in the title refer to the numbering of the sub-objectives on the previous page.

The new 6.2 capabilities will increase the setup-code requirements …



… Enable removal of setup code using Setup Frames …

- Examples that you have already seen:
  - CountyMap frame saved 600 lines of runtime code (800 - 200)
  - Progress/Graph frame saved 400 lines of runtime code (800 - 400)
  - W7Styler frame saved 2000 lines of runtime code (2250 - 250)
- Probably most production frames could save significant code using Setup Frames

Property Inspector - User Frame: CountyM...

Frame: CountyMap | All

IsMinimizable	TRUE
IsNullable	FALSE
IsPopup	FALSE
IsResizeable	FALSE
IsTitled	TRUE
IsToolWindow	FALSE
IsTopmost	FALSE
SetupName	
StatusText	
TaggedValues	
TemplateName	standard
TransparentColor	CC_TRANSP
VersShortRemarks	
WindowHeight	6229
WindowIcon	No Bitmap
WindowPlacement	WP_PAREN
WindowTitle	
WindowWidth	8302
WindowXLeft	0
WindowYTop	0

Select a Setup Frame

Application: core, imageMapping

Frame: setupcountymap

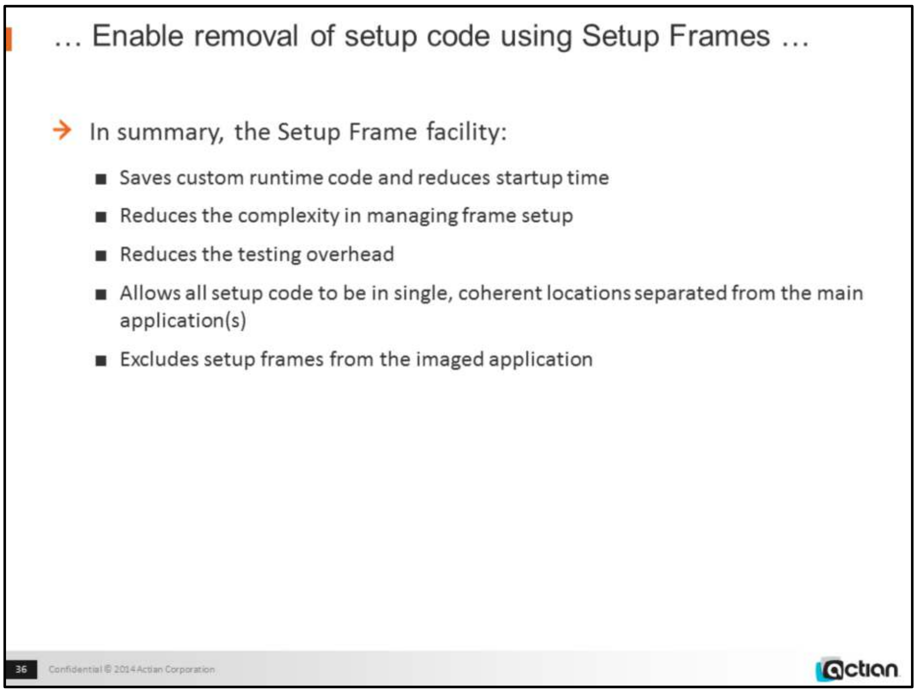
Greater London

Population:	8,196,700	1
Size (sq km):	1,569	37
Population density:	5,223	1

Demo:

w4gldev runimage workbench.img –Tall -/appflags profile=or62demos application=d201504_imagemapping component=countymap, command=open

Continue as shown



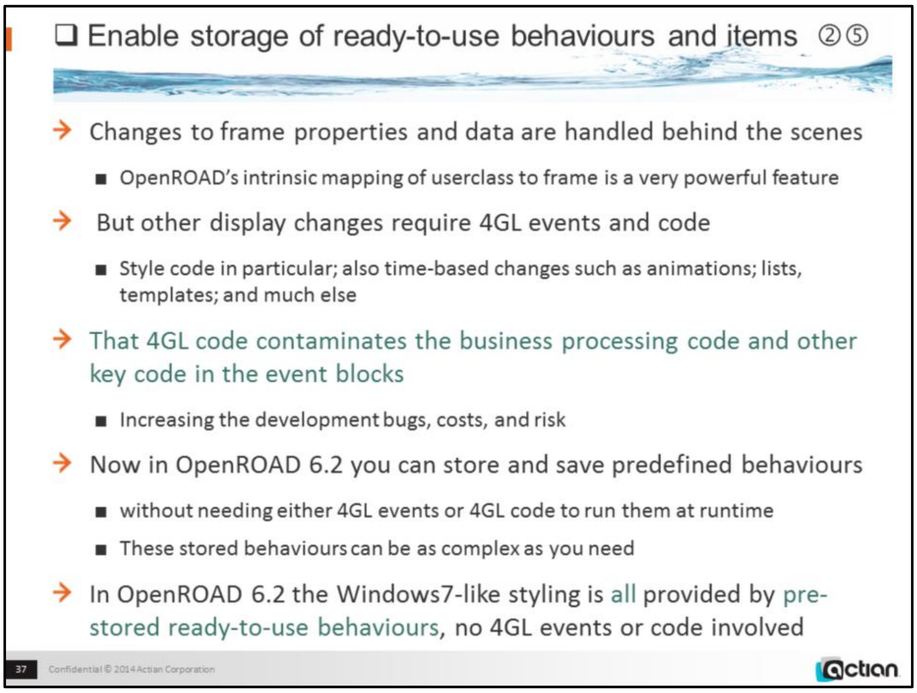
… Enable removal of setup code using Setup Frames …

- In summary, the Setup Frame facility:
  - Saves custom runtime code and reduces startup time
  - Reduces the complexity in managing frame setup
  - Reduces the testing overhead
  - Allows all setup code to be in single, coherent locations separated from the main application(s)
  - Excludes setup frames from the imaged application

36 Confidential © 2014 Actian Corporation

The new Setup Frame facility saves custom runtime code and reduces startup time. Setup Frame lets developers predefine frame setup code and put all setup code in single, coherent locations separate from the runtime application. All the changes to the frame are applied in advance so that when a frame starts up at runtime, everything is ready to go – significantly reducing startup time and reducing the complexity in managing frame setup.

Not only is the setup code no longer contaminating the frame at runtime, but it doesn't need to be tested any more as part of bug-fixing, upgrades, etc.
Even the initial testing is simpler, since it doesn't have to be "foolproof", it just has to work correctly when used correctly.


❑ Enable storage of ready-to-use behaviours and items ②⑤

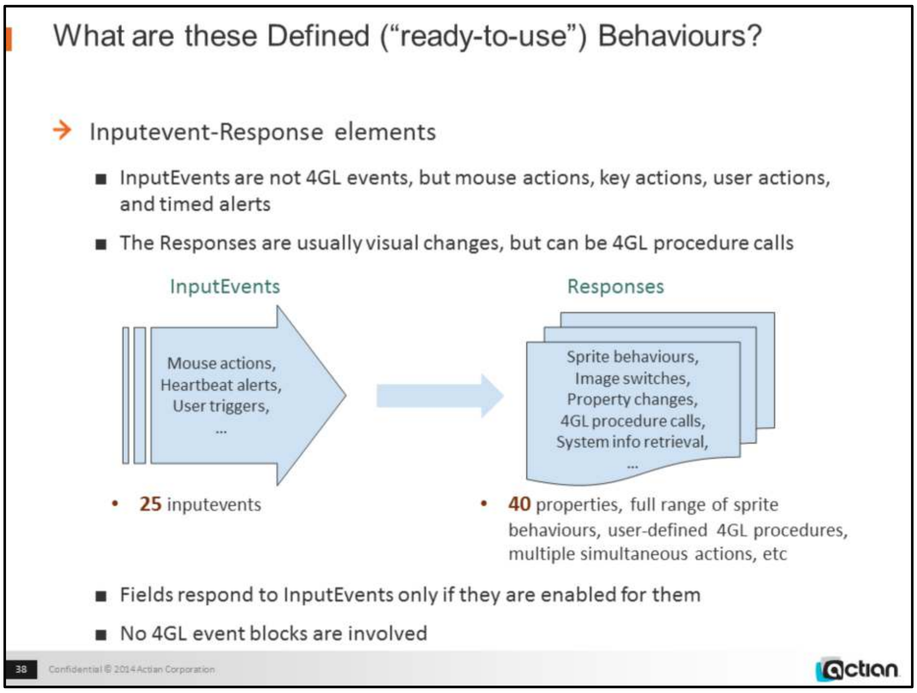
- Changes to frame properties and data are handled behind the scenes
  - OpenROAD's intrinsic mapping of userclass to frame is a very powerful feature
- But other display changes require 4GL events and code
  - Style code in particular; also time-based changes such as animations; lists, templates; and much else
- That 4GL code contaminates the business processing code and other key code in the event blocks
  - Increasing the development bugs, costs, and risk
- Now in OpenROAD 6.2 you can store and save predefined behaviours
  - without needing either 4GL events or 4GL code to run them at runtime
  - These stored behaviours can be as complex as you need
- In OpenROAD 6.2 the Windows7-like styling is all provided by pre-stored ready-to-use behaviours, no 4GL events or code involved

37 Confidential © 2014 Actian Corporation

What data and objects can be stored in OpenROAD?
Prior to 6.2, all saved, imaged or exported applications could contain:
- fields, lists, trees, strings, components

The applications can now also contain:
- taggedvalues, hashtables, longbyte objects, prochandles, arrays

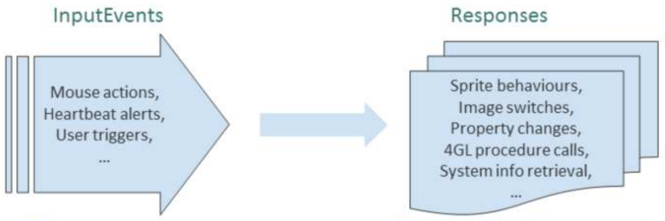
… which means that imaged, saved, and exported applications can now contain data of any size, kind or complexity



What are these Defined ("ready-to-use") Behaviours?

→ Inputevent-Response elements

- InputEvents are not 4GL events, but mouse actions, key actions, user actions, and timed alerts
- The Responses are usually visual changes, but can be 4GL procedure calls



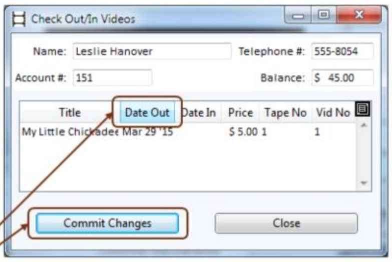
- **25** inputevents
- **40** properties, full range of sprite behaviours, user-defined 4GL procedures, multiple simultaneous actions, etc

- Fields respond to InputEvents only if they are enabled for them
- No 4GL event blocks are involved

… Enable storage of ready-to-use behaviours and items …

→ Pre-storing the Windows-7-like styling in OpenROAD 6.2 saved another 500 lines of runtime code (in addition to the Setup savings)

- and made the processing simpler and more reliable



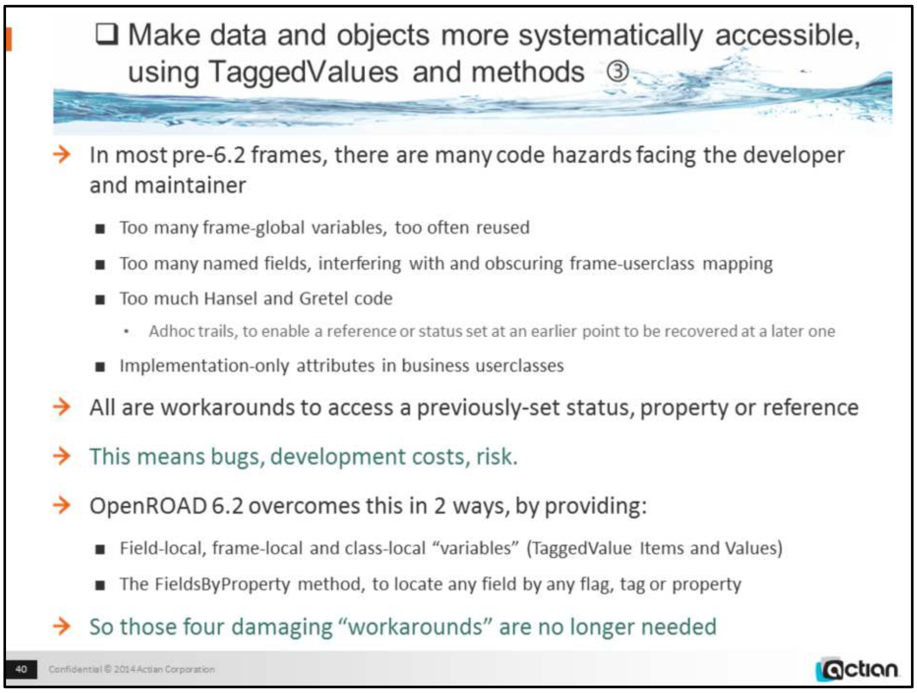
Name: Leslie Hanover Telephone #: 555-8054

Account #: 151 Balance: $ 45.00

Title	Date Out	Date In	Price	Tape No	Vid No
My Little Chickadee	Mar 29 '15		$ 5.00	1	1

No 4GL code or events required

actian



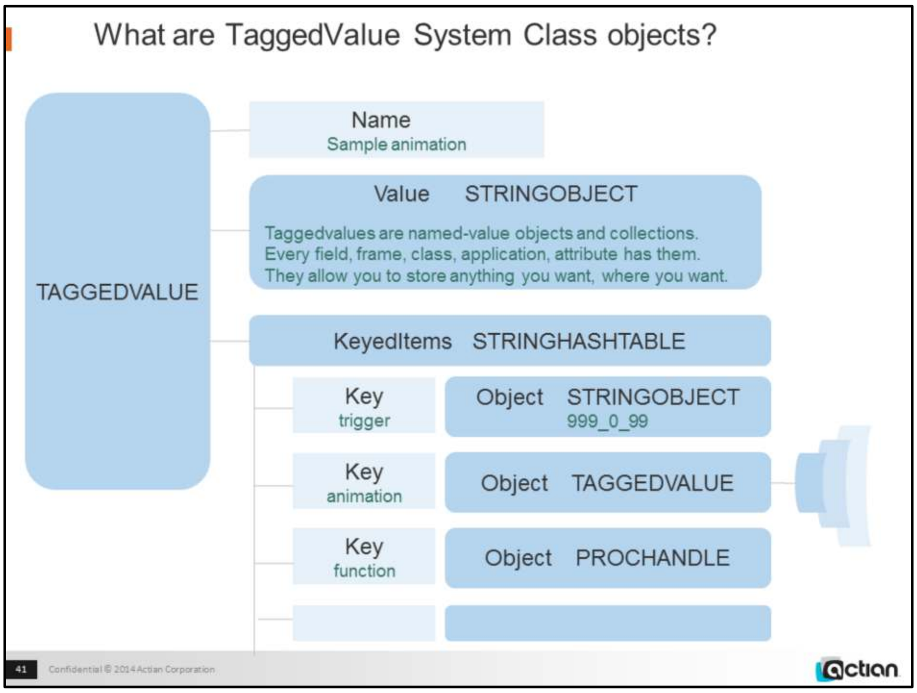
❑ Make data and objects more systematically accessible, using TaggedValues and methods ③

- In most pre-6.2 frames, there are many code hazards facing the developer and maintainer
  - Too many frame-global variables, too often reused
  - Too many named fields, interfering with and obscuring frame-userclass mapping
  - Too much Hansel and Gretel code
    - Adhoc trails, to enable a reference or status set at an earlier point to be recovered at a later one
  - Implementation-only attributes in business userclasses
- All are workarounds to access a previously-set status, property or reference
- This means bugs, development costs, risk.
- OpenROAD 6.2 overcomes this in 2 ways, by providing:
  - Field-local, frame-local and class-local “variables” (TaggedValue Items and Values)
  - The FieldsByProperty method, to locate any field by any flag, tag or property
- So those four damaging “workarounds” are no longer needed

40 Confidential © 2014 Actian Corporation

Examples of Hansel and Gretel code / Adhoc trails:

- passing an incrementing counter to a userevent as the MessageInteger
- including the datatype of a variable in the variable’s name



What are TaggedValue System Class objects?

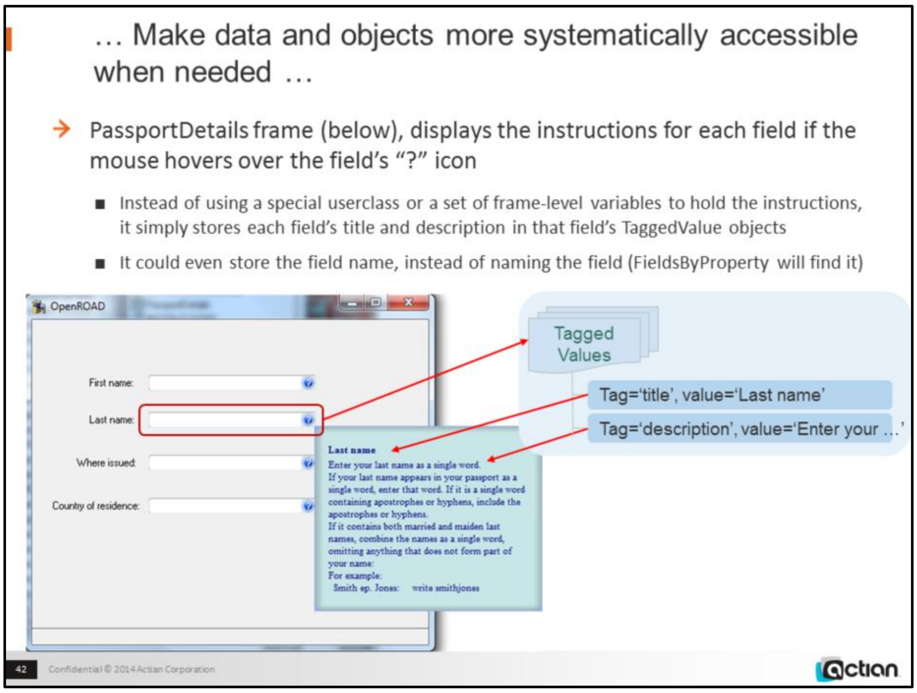
TAGGEDVALUE
Name
Sample animation
Value STRINGOBJECT
Taggedvalues are named-value objects and collections.
Every field, frame, class, application, attribute has them.
They allow you to store anything you want, where you want.
KeyedItems STRINGHASHTABLE
Key
trigger
Object STRINGOBJECT
999_0_99
Key
animation
Object TAGGEDVALUE
Key
function
Object PROCHANDLE

TaggedValue arrays are attached to, and saved with, all fields, attributes, components, applications.

There are TaggedValue-manipulation methods in each of these classes.

Each TaggedValue can hold (and save) keyed data and objects, in any combination or amount.

Almost all useful System Class objects can be saved, but not User Class objects.



… Make data and objects more systematically accessible when needed …

- PassportDetails frame (below), displays the instructions for each field if the mouse hovers over the field's "?" icon
  - Instead of using a special userclass or a set of frame-level variables to hold the instructions, it simply stores each field's title and description in that field's TaggedValue objects
  - It could even store the field name, instead of naming the field (FieldsByProperty will find it)

OpenROAD

First name:

Last name:

Where issued:

Country of residence:

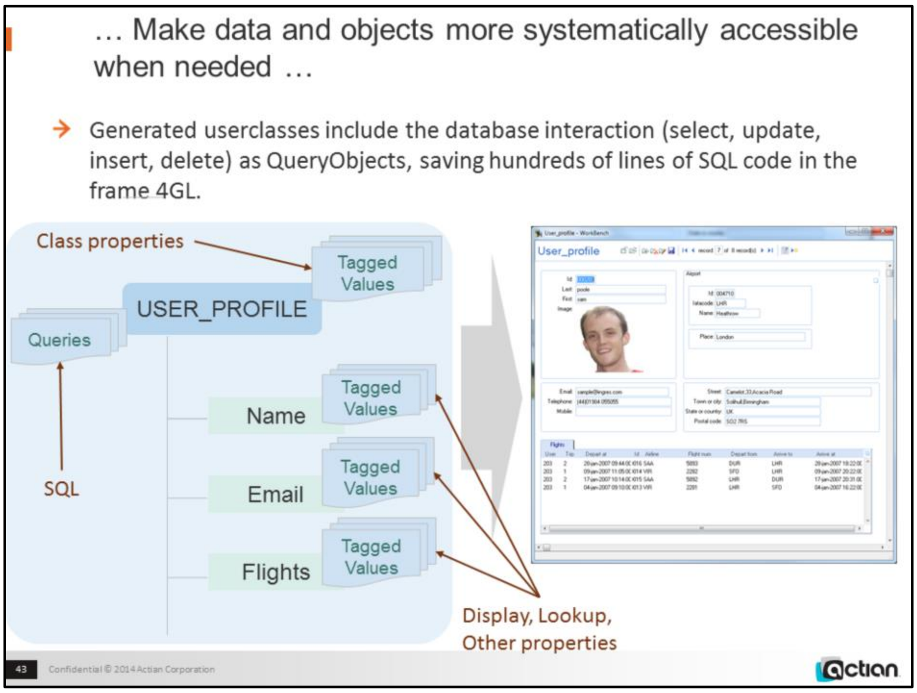
Last name
Enter your last name as a single word.
If your last name appears in your passport as a single word, enter that word. If it is a single word containing apostrophes or hyphens, include the apostrophes or hyphens.
If it contains both married and maiden last names, combine the names as a single word, omitting anything that does not form part of your name:
For example:
 Smith ep. Jones: write smithjones

Tagged Values
Tag='title', value='Last name'
Tag='description', value='Enter your …'

Demo:
D201504_DefinedResponses_Panel -cPassportDetails
Hover mouse over any "?" sprite
  the infopanel frame appears alongside the sprite, displaying that input field's description.
Move the mouse off the sprite
  the infopanel frame disappears.

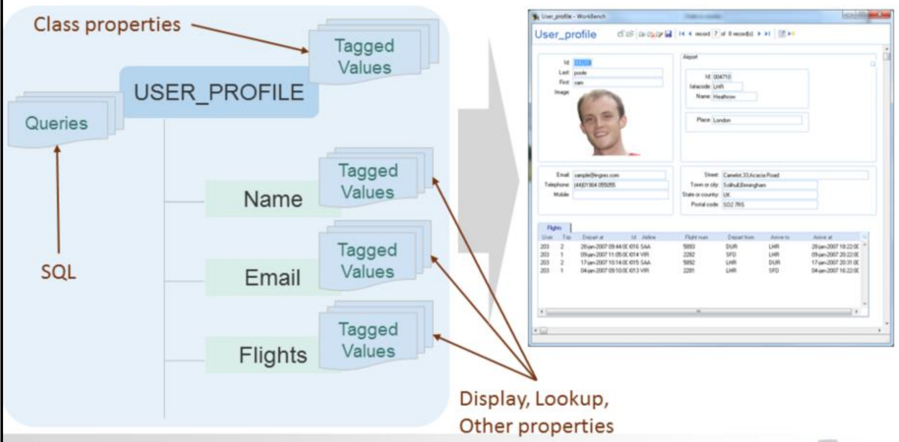
Demo:
w4gldev runimage workbench.img –Tall -/appflags profile=or62demos application=d201504_definedresponses_panel component=passportdetails, command=openscript
  Note no code

Other simple examples of taggedvalue uses:
- a field's previous value, held in one of that field's taggedvalues
- A field's public name, to use in messages.



… Make data and objects more systematically accessible when needed …

- Generated userclasses include the database interaction (select, update, insert, delete) as QueryObjects, saving hundreds of lines of SQL code in the frame 4GL.



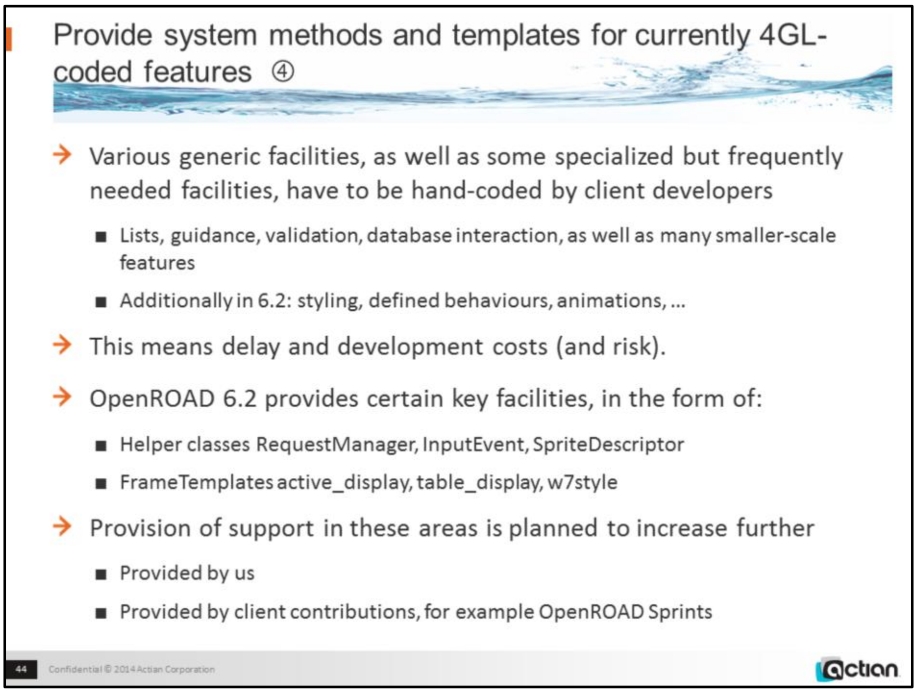
Id: 203
Last: poole
First: sam
Image:

Airport
Id: 004710
Iatacode: LHR
Name: Heathrow
Place: London

Email: sample@ingres.com
Telephone: (44)01904 055055
Mobile:

Street: Camelot,33,Acacia Road
Town or city: Solihull,Birmingham
State or country: UK
Postal code: SO2 7RS

User	Trip	Depart at	Id	Airline	Flight num	Depart from	Arrive to	Arrive at
203	2	20-jan-2007 09:44:0(	016	SAA	5893	DUR	LHR	20-jan-2007 18:22:0(
203	1	09-jan-2007 11:05:0(	014	VIR	2282	SFO	LHR	09-jan-2007 20:22:0(
203	2	17-jan-2007 10:14:0(	015	SAA	5892	LHR	DUR	17-jan-2007 20:31:0(
203	1	04-jan-2007 09:10:0(	013	VIR	2281	LHR	SFO	04-jan-2007 16:22:0(



Provide system methods and templates for currently 4GL-coded features ④

- Various generic facilities, as well as some specialized but frequently needed facilities, have to be hand-coded by client developers
  - Lists, guidance, validation, database interaction, as well as many smaller-scale features
  - Additionally in 6.2: styling, defined behaviours, animations, ...
- This means delay and development costs (and risk).
- OpenROAD 6.2 provides certain key facilities, in the form of:
  - Helper classes RequestManager, InputEvent, SpriteDescriptor
  - FrameTemplates active_display, table_display, w7style
- Provision of support in these areas is planned to increase further
  - Provided by us
  - Provided by client contributions, for example OpenROAD Sprints

… Provide system templates for currently 4GL-coded features…

→ The table_display frame template creates frames

- that support simple Data Administration for individual tables,
- without requiring any coding from the developer,
- and are completely standalone
  - no need to create other frames, userclasses or applications

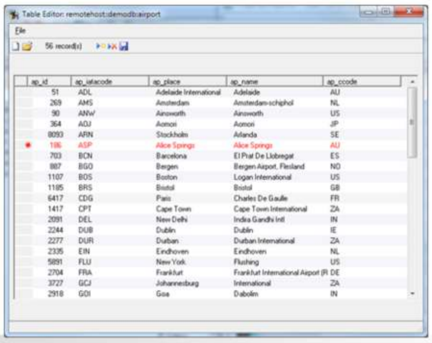


Table Editor: remotehost::demodb::airport

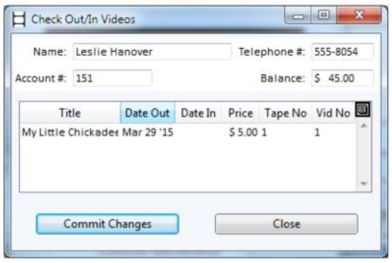
File

56 record(s)

	ap_id	ap_iatacode	ap_place	ap_name	ap_ccode
	51	ADL	Adelaide International	Adelaide	AU
	269	AMS	Amsterdam	Amsterdam-schiphol	NL
	90	ANW	Ainsworth	Ainsworth	US
	364	AOJ	Aomori	Aomori	JP
	8093	ARN	Stockholm	Arlanda	SE
●	186	ASP	Alice Springs	Alice Springs	AU
	703	BCN	Barcelona	El Prat De Llobregat	ES
	887	BGO	Bergen	Bergen Airport, Flesland	NO
	1107	BOS	Boston	Logan International	US
	1185	BRS	Bristol	Bristol	GB
	6417	CDG	Paris	Charles De Gaulle	FR
	1417	CPT	Cape Town	Cape Town International	ZA
	2091	DEL	New Delhi	Indira Gandhi Intl	IN
	2244	DUB	Dublin	Dublin	IE
	2277	DUR	Durban	Durban International	ZA
	2335	EIN	Eindhoven	Eindhoven	NL
	5891	FLU	New York	Flushing	US
	2704	FRA	Frankfurt	Frankfurt International Airport (R	DE
	3727	GCJ	Johannesburg	International	ZA
	2918	GOI	Goa	Dabolim	IN

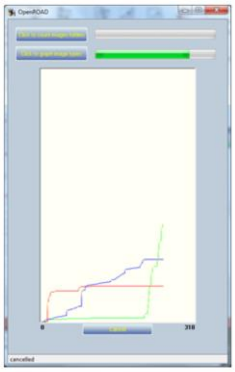
… Provide system methods for 4GL-coded features …

- The InputEvent and SpriteDescriptor "helper" userclasses contain wrapper methods that make Defined Behaviours, animations and styling easy to implement and maintain
  - Avoiding complex custom 4GL coding

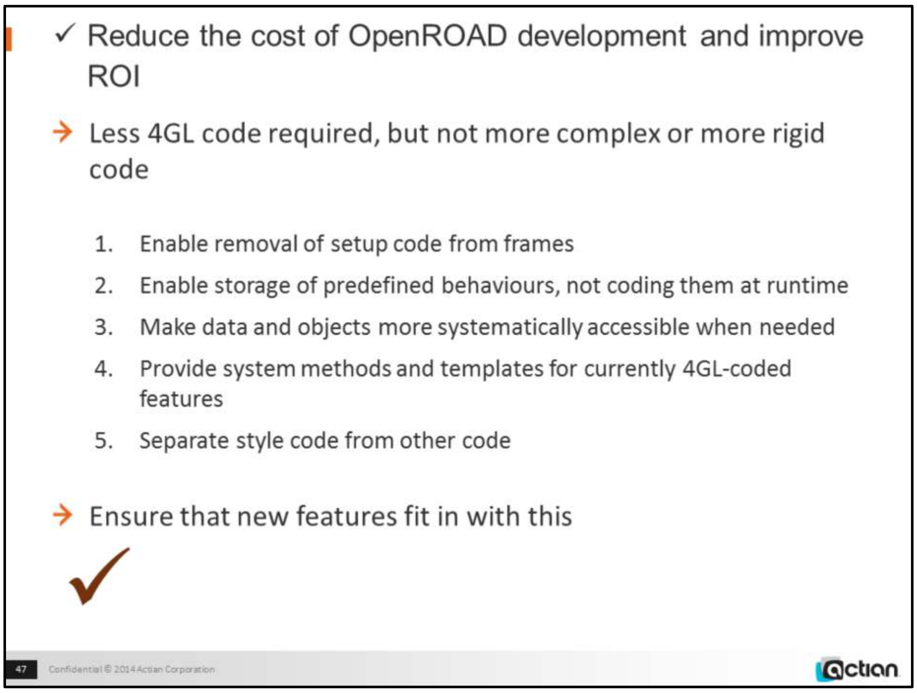


Name: Leslie Hanover Telephone #: 555-8054
Account #: 151 Balance: $ 45.00

Title	Date Out	Date In	Price	Tape No	Vid No
My Little Chickadee	Mar 29 '15		$ 5.00	1	1



cancelled



✓ Reduce the cost of OpenROAD development and improve ROI

→ Less 4GL code required, but not more complex or more rigid code

1. Enable removal of setup code from frames
2. Enable storage of predefined behaviours, not coding them at runtime
3. Make data and objects more systematically accessible when needed
4. Provide system methods and templates for currently 4GL-coded features
5. Separate style code from other code

→ Ensure that new features fit in with this

✓

actian

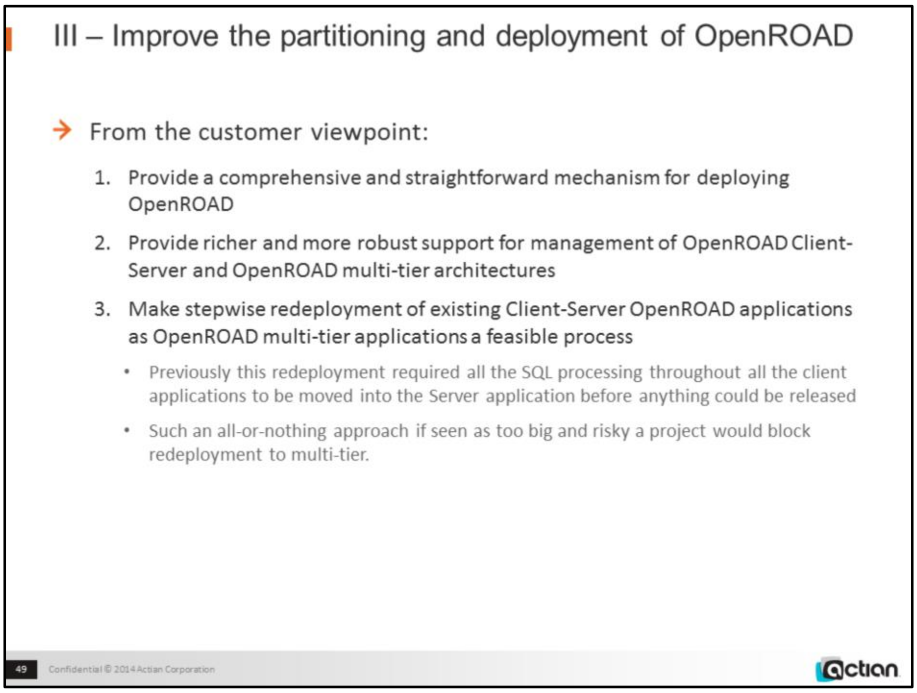
Meeting OpenROAD 6.2 Objectives …

- III – Improving OpenROAD partitioning and deployment
  - Covered in detail in the second of these presentations.
  - Brief summary here

48 Confidential © 2014 Actian Corporation

LoadnRun provides a comprehensive and straightforward mechanism for deploying OpenROAD.

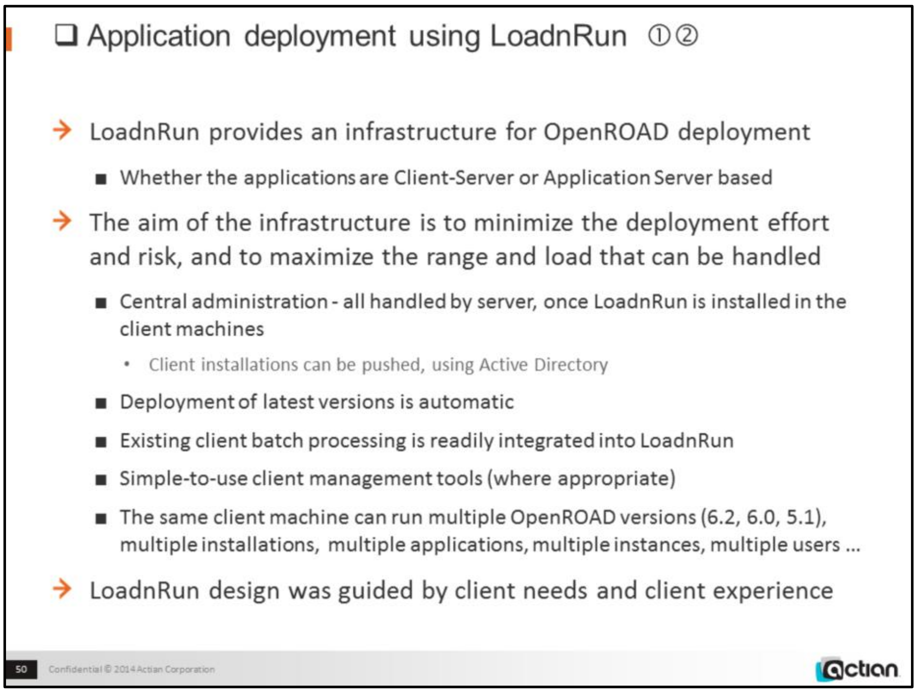
Dowloadable IngresNet provides a very useful step in redeploying existing Client-Server OpenROAD applications as OpenROAD client-OpenROAD server applications



III – Improve the partitioning and deployment of OpenROAD

→ From the customer viewpoint:

1. Provide a comprehensive and straightforward mechanism for deploying OpenROAD
2. Provide richer and more robust support for management of OpenROAD Client-Server and OpenROAD multi-tier architectures
3. Make stepwise redeployment of existing Client-Server OpenROAD applications as OpenROAD multi-tier applications a feasible process
   - Previously this redeployment required all the SQL processing throughout all the client applications to be moved into the Server application before anything could be released
   - Such an all-or-nothing approach if seen as too big and risky a project would block redeployment to multi-tier.

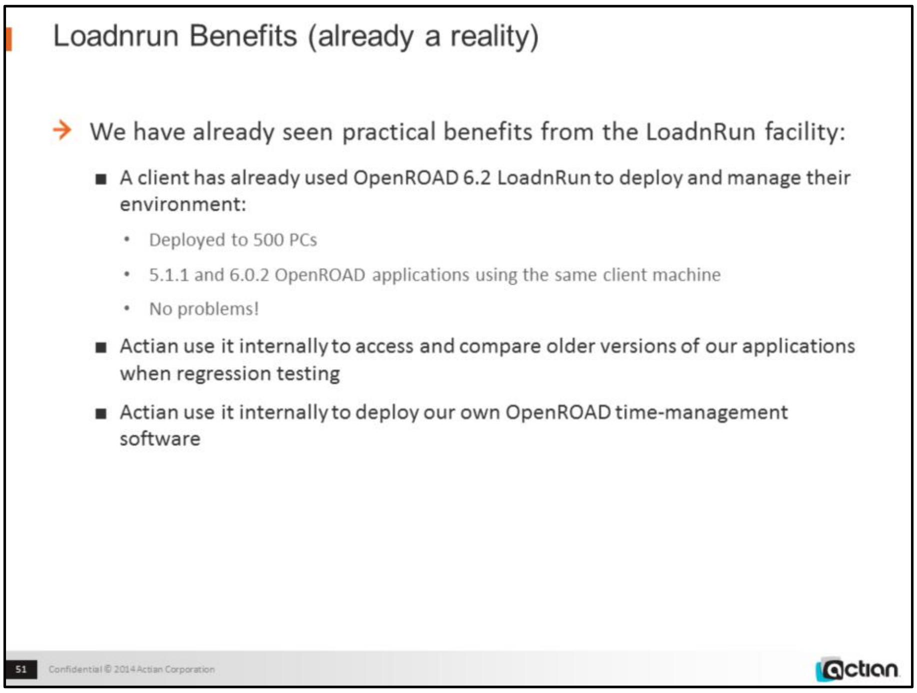


❑ Application deployment using LoadnRun ①②

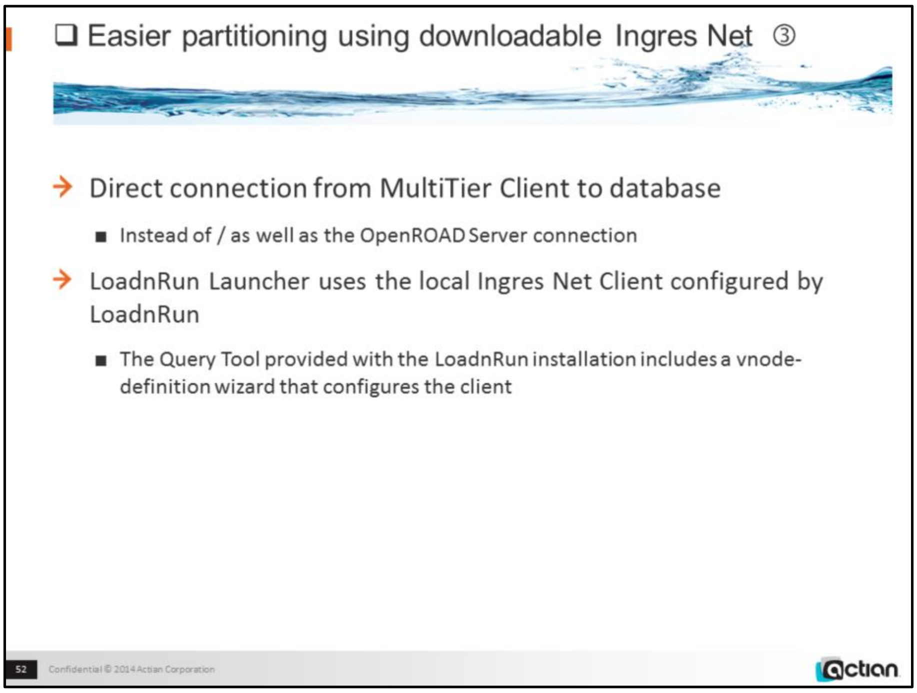
- LoadnRun provides an infrastructure for OpenROAD deployment
  - Whether the applications are Client-Server or Application Server based
- The aim of the infrastructure is to minimize the deployment effort and risk, and to maximize the range and load that can be handled
  - Central administration - all handled by server, once LoadnRun is installed in the client machines
    - Client installations can be pushed, using Active Directory
  - Deployment of latest versions is automatic
  - Existing client batch processing is readily integrated into LoadnRun
  - Simple-to-use client management tools (where appropriate)
  - The same client machine can run multiple OpenROAD versions (6.2, 6.0, 5.1), multiple installations, multiple applications, multiple instances, multiple users ...
- LoadnRun design was guided by client needs and client experience

50 Confidential © 2014 Actian Corporation

Numbers in circles in the title refer to the numbering of the sub-objectives on the previous page

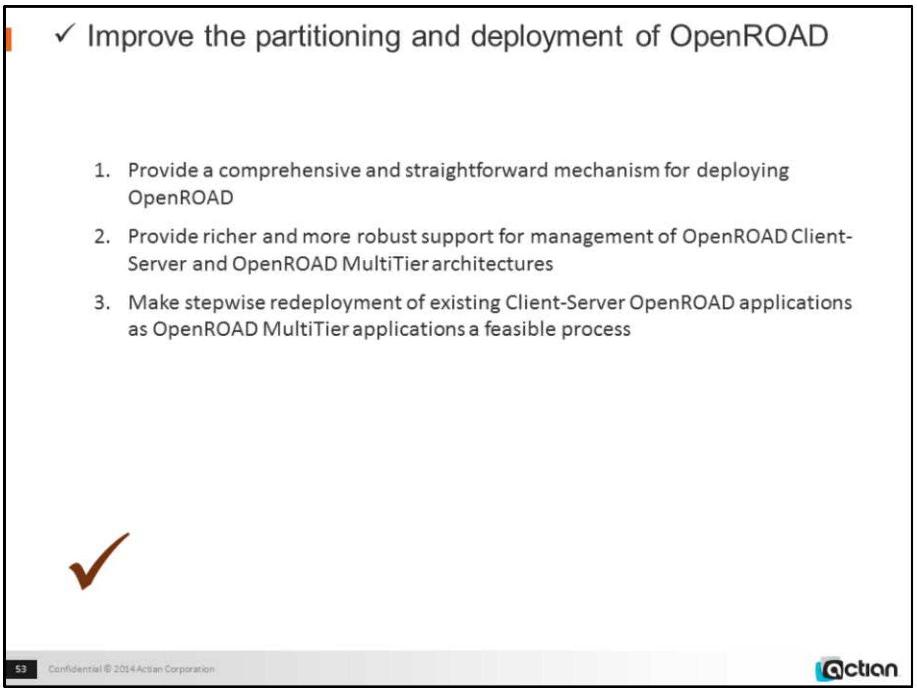

Loadnrun Benefits (already a reality)

- We have already seen practical benefits from the LoadnRun facility:
  - A client has already used OpenROAD 6.2 LoadnRun to deploy and manage their environment:
    - Deployed to 500 PCs
    - 5.1.1 and 6.0.2 OpenROAD applications using the same client machine
    - No problems!
  - Actian use it internally to access and compare older versions of our applications when regression testing
  - Actian use it internally to deploy our own OpenROAD time-management software



❑ Easier partitioning using downloadable Ingres Net ③

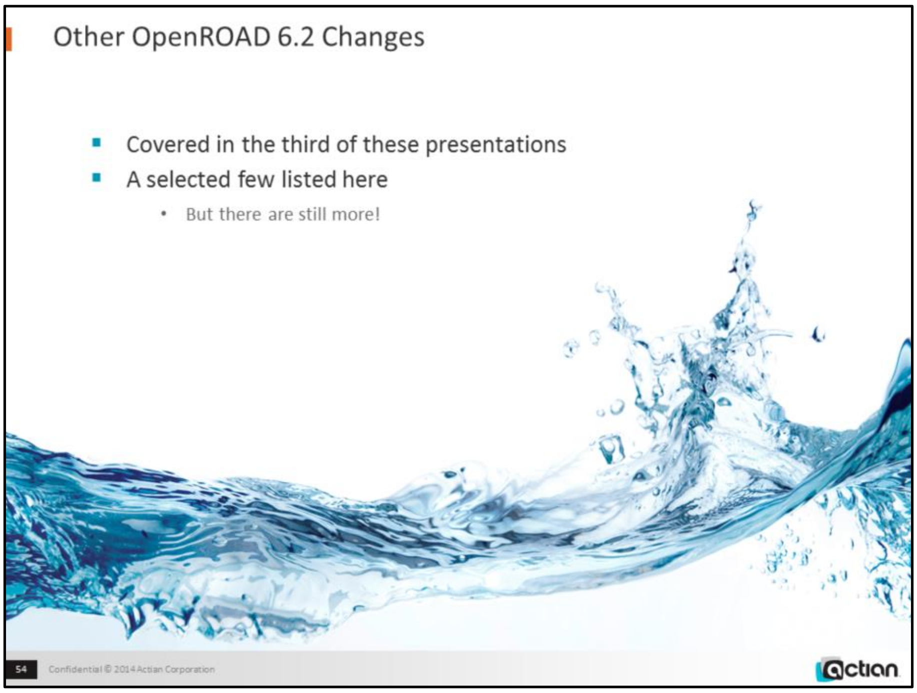
- Direct connection from MultiTier Client to database
  - Instead of / as well as the OpenROAD Server connection
- LoadnRun Launcher uses the local Ingres Net Client configured by LoadnRun
  - The Query Tool provided with the LoadnRun installation includes a vnode-definition wizard that configures the client



✓ Improve the partitioning and deployment of OpenROAD

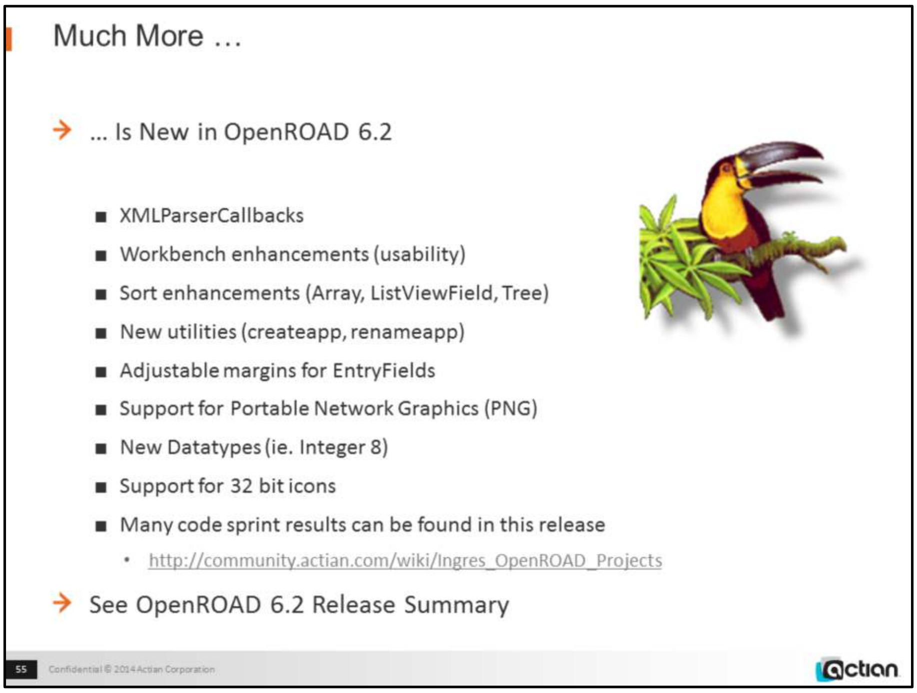
1. Provide a comprehensive and straightforward mechanism for deploying OpenROAD
2. Provide richer and more robust support for management of OpenROAD Client-Server and OpenROAD MultiTier architectures
3. Make stepwise redeployment of existing Client-Server OpenROAD applications as OpenROAD MultiTier applications a feasible process

✓



Other OpenROAD 6.2 Changes

- Covered in the third of these presentations
- A selected few listed here
  - But there are still more!



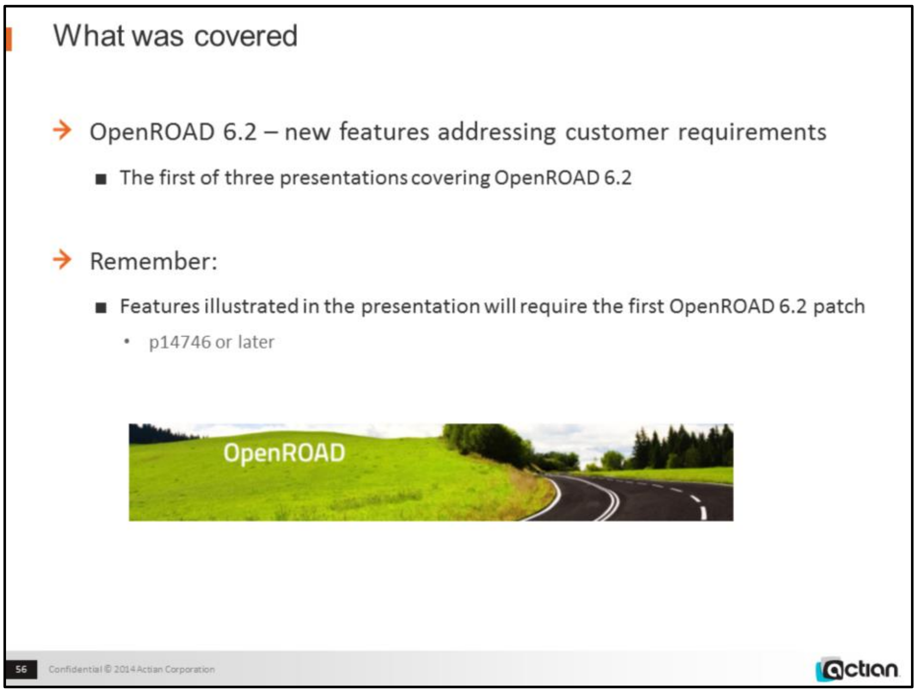
Much More …

- … Is New in OpenROAD 6.2
  - XMLParserCallbacks
  - Workbench enhancements (usability)
  - Sort enhancements (Array, ListViewField, Tree)
  - New utilities (createapp, renameapp)
  - Adjustable margins for EntryFields
  - Support for Portable Network Graphics (PNG)
  - New Datatypes (ie. Integer 8)
  - Support for 32 bit icons
  - Many code sprint results can be found in this release
    - http://community.actian.com/wiki/Ingres_OpenROAD_Projects
- See OpenROAD 6.2 Release Summary

55 Confidential © 2014 Actian Corporation

And more still …

- TreeViewField, ControlButton fields
- ToolTipText size
- Curexec system variable



What was covered

- OpenROAD 6.2 – new features addressing customer requirements
  - The first of three presentations covering OpenROAD 6.2
- Remember:
  - Features illustrated in the presentation will require the first OpenROAD 6.2 patch
    - p14746 or later

Thank you

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

57 Confidential © 2014 Actian Corporation