



## Italian User Group 2010

### RIA Web Development in Uniface Workshop

Mike Taylor

Uniface Solution Specialist

# Uniface 9.4.01

## Uniface and Rich Internet Applications

*“Rich Internet Applications seek to deliver the same features and functions normally associated with the desktop without the need to download and install software at the client.”*

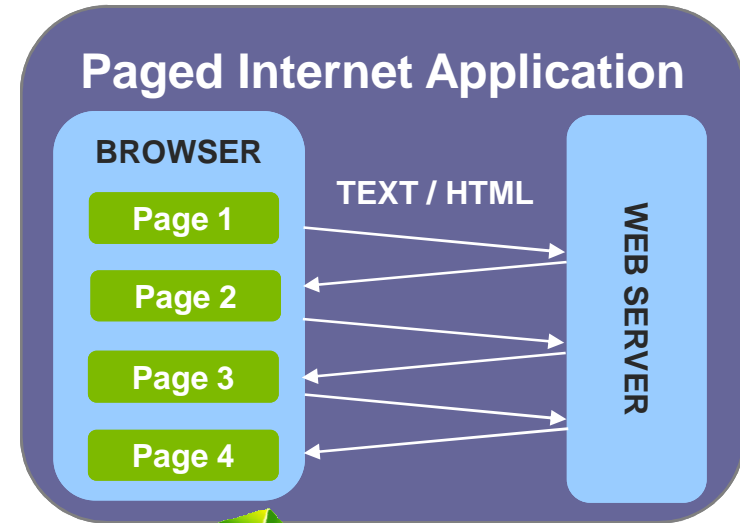
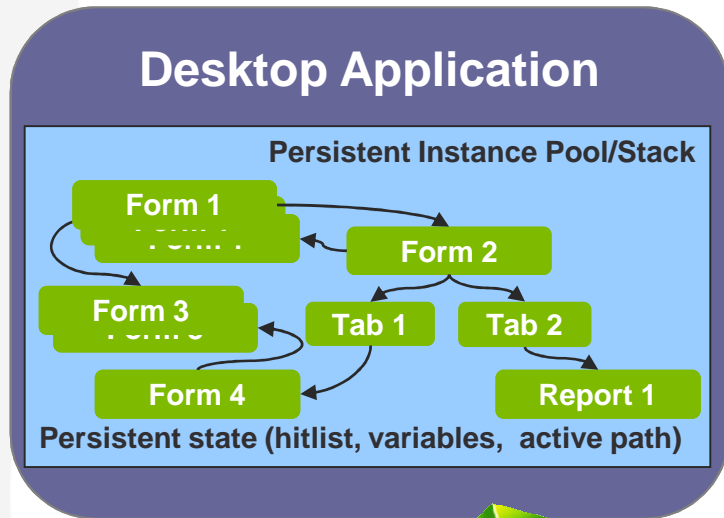
# *Objectives*

The Uniface Component Types

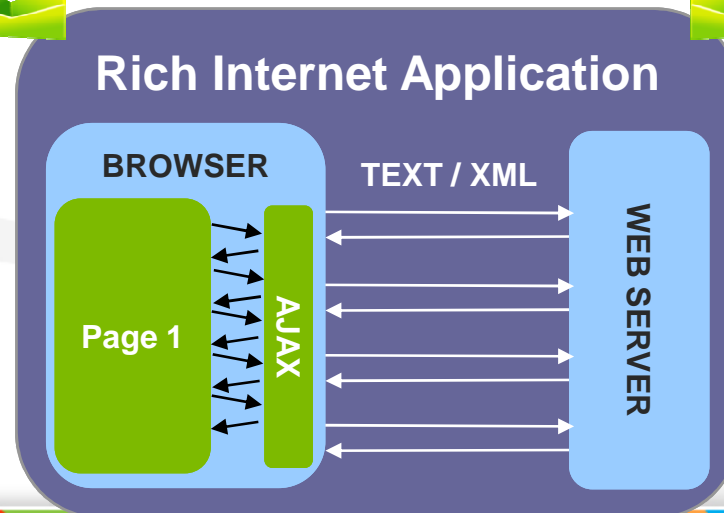


Uniface 9.4 and Rich Internet Applications

# Desktop → PIA → RIA



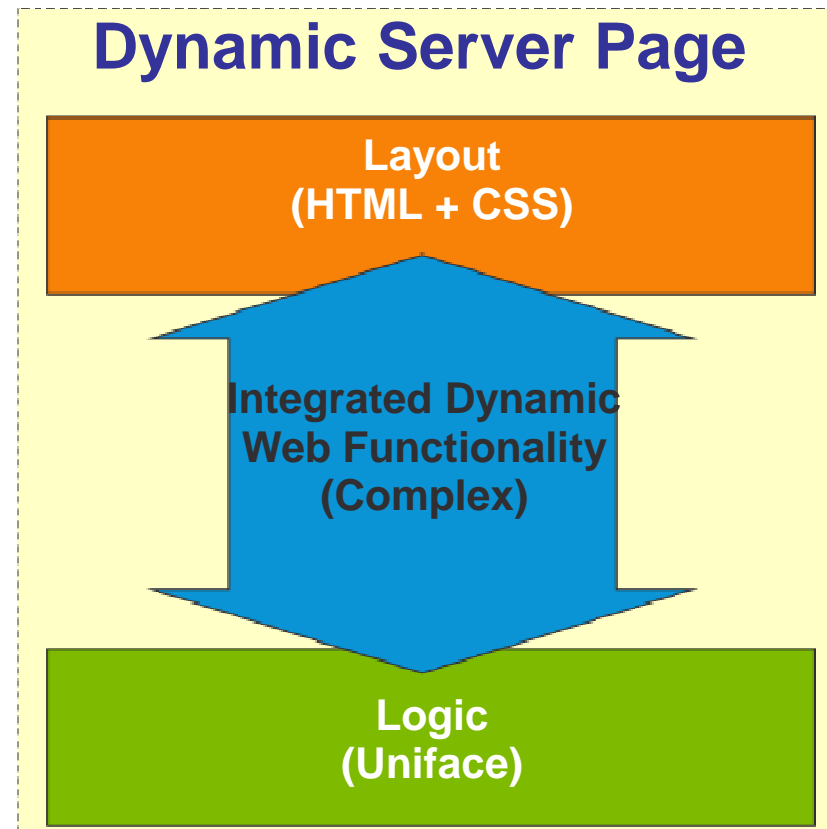
Rich interface  
High usability  
Reactive screens  
State persistence



Easier deployment  
No client footprint  
Multi channel  
Dynamic partial updates

# DSP Overview

- Model-driven development
- Proc is primary language
- Layout-logic split
- Dynamic page updates (AJAX)
  - Synchronous & Asynchronous
- Binding/HTML generator wizard
- Layout mash-up enablement
- Client-side declarative validation
- Procedural error handling
- Security features

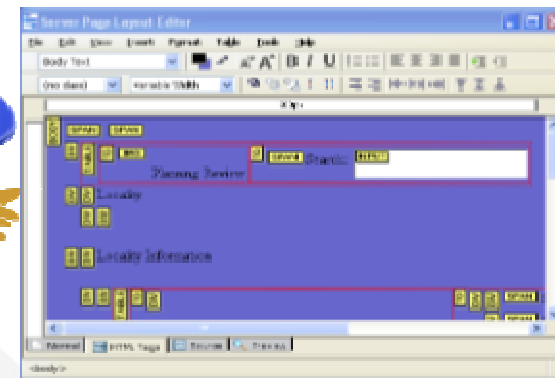
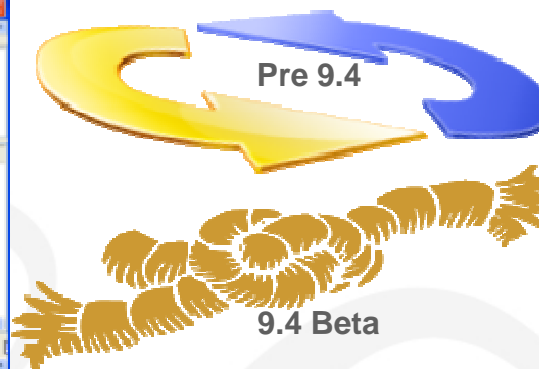
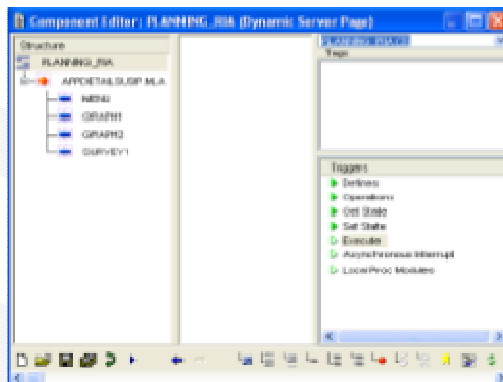


# Server Pages

The image displays a screenshot of the Uniface software interface. On the left, a navigation pane is divided into four sections: **PRESENTATION**, **BUSINESS LOGIC**, and **DATA ACCESS**. The **PRESENTATION** section contains icons for Report, Form, Dynamic Server Page, and Static Server Page. The **BUSINESS LOGIC** section contains icons for Entity Service, Session Service, and Service. The **DATA ACCESS** section contains icons for Model, Load Definitions, and Entity. A red circle highlights the **Static Server Page** icon in the **PRESENTATION** section. On the right, a **UNIFACE SHORTCUTS** panel is visible. Below it, a dropdown menu titled **Uniface Nine** is open, showing a list of options: Model, Start-up Shell, Form, Dynamic Server Page, Static Server Page, Report, XML Client, Service, Session Service, Entity Service, Process Studio, Component Integration, Signature, Library, and Templates. A red circle highlights the **Dynamic Server Page** and **Static Server Page** options in the dropdown menu.

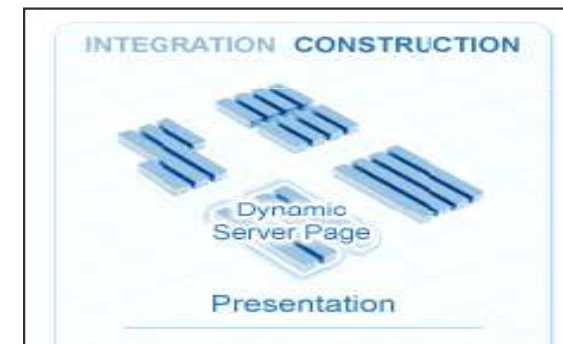
# Component Editor

- Logic Structure & Layout Separate
  - No more synchronisation
- Simple Binding
  - As easy as “copy and paste”



# DSP – Dynamic Server Page

- Incremental/Full page updates
- Rich widgets with client-side inline validation
- ID-based binding between data and layout
- Well-formed XHTML
- Extended Triggers:
  - OnChange – majority widgets
  - OnBlur, OnClick, OnDbClick, OnFocus, OnKeyPress – AttributesOnly widget
- Widget property manipulation via proc



# DSP Presentation

- Web layout independent of DSP structure
- Layout Editor x Component Editor
- “Copy as HTML” functionality
- Web layout stored in repository (or as .hts)
- CSS and HTML assist in designing modern web applications

# DSP Widgets

- Two groups: **Uniface** and **DOJO** (rich controls)
- DOJO – open source library of Javascript, CSS and AJAX-enabled objects
- AJAX requests provide incremental DSP updates
- Both DOJO and Uniface widgets have +50 properties with their HTML or CSS equivalent
- Dynamically change properties via Proc

# New Proc (1)

- web

Determines if operation/trigger can be called from a web browser

Public/Partner web

- callfieldtrigger

Explicitly calls a trigger on a field.

callfieldtrigger "DETAIL", "BTN\_STORE.BUTTONS.PROJECTS"

- webmessage

Displays a message in a dialog box in the browser.

Webmessage/info "Store Successful"

# Demo

## “Creating a Simple DSP”

Celebrating  
25 Years

uniface

# Request-Response Exchange

**User clicks button → browser sends request to server → server processes request and generates response → response sent to browser → browser processes response ...**

## Dynamic

- Response updates within a loaded web page
- Browser needs to know about the input data elements for request and target data elements for the response

## Static

- Loads a web page (same or another)

# USP x DSP - Differences

## USP

- data embedded in (X)HTML layout (webgen)
- when page is submitted, data is extracted (webget)
- data cannot be returned separately from page layout

## DSP

- data separate from layout
- data can be submitted between browser and server as JSON
- DSP can contain several DSP's via DSPContainer widget
- Partial asynchronous page updates
- websave, webload, weblayout, webdefinitions, websetocc ...

# New Proc (2)

- Manipulate CSS and HTML property names used for layout properties

```
$fieldproperties (BTN_RETRIEVE.BUTTONS) =  
  "style:display=none"
```

- Extended Triggers

```
trigger OnChange
```

```
public web
```

```
  If .....
```

# Demo

Celebrating  
25 Years

uniface

# DSPContainer Widget

- Enable child DSPs to be displayed within single parent DSP
- Isolate and limit the data that needs to be updated in a web page. For example, while one DSP is being updated, the rest of the page remains the same and the user can continue with other activities.

# Demo

Celebrating  
25 Years

uniface

# Scoping

- declares data that is included in the request-response exchange
- all declared data is blocked for the duration of the exchange
- avoids “race conditions”

*A **race condition** occurs when multiple processes access and manipulate the same data concurrently, and the outcome of the execution depends on the particular order in which the access takes place*

# New Proc (3)

- **Scope**

Declares the beginning of a scope declaration which specifies the data that is to be included in a DSP request-response exchange.

Scope

input

output

operation

endscope

# Demo

Celebrating  
25 Years

uniface

# AttributesOnly Widget

- An abstract field to which any HTML element, such as a <div> or a <table> can be bound
- Makes it possible to manipulate the properties of that element in Proc code
- By changing the \$fieldproperties of the AttributesOnly widget, the bound object will be affected
- Simple to use, very powerful !

# Demo

Celebrating  
25 Years

uniface

# Benefits of Uniface RIA

- Traditional Uniface productivity comes to RIA development
- Cooperative development with existing web design and development teams
- Flexible architecture encourages creativity
- Existing Uniface skills can be quickly extended to RIA development

# Thank You & Questions

