



CA Plex .NET Client Tips and Techniques

KT Technologies

Misc Track

Kiyoshi Terasawa



FOR INFORMATION PURPOSES ONLY

Terms of this presentation

This presentation was based on current information and resource allocations as of April 2013 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Certain information in this presentation may outline CA's general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this presentation "as is" without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages. CA confidential and proprietary. No unauthorized copying or distribution permitted.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. CA confidential and proprietary. No unauthorized copying or distribution permitted.



Kiyoshi Terasawa

CA KT Technologies

terasawakiyoshi@gmail.com

www.linkedin.com/in/kiyoshiterasawa

About this Session

Make a clear line what CA Plex can do and what cannot

Don't wait CA forever until they feed you, just make an action

You can get both benefit from CA Plex and native world

Install Visual Studio Community

Well, it is free! (If you are qualified...)



Get and Install

- This is your homework
 - Create Microsoft account if you don't have one
 - Download VS2013 Community Edition
 - Install it

Set Up Debugging Environment

Learn and use it! Forget AD Debugger from CA Plex

Debugging your Plex .NET function

- No AD Debugger from CA
 - All we know it won't happen
- Good skill to have

Once you have Debugging Environment

- Easier source Code Object writing for Plex
 - you get intellisense and instant Syntax check.
- The Exe project is home of post generation
 - Build
 - Deployment

Simple Steps

- With Plex: Gen and Build your function with Plex or CodeLibWizard
- With Plex: Create EXE for your main function (Only Once)
- With VS: Open The EXE project
- With VS: Add function dll csproj (ones generated by Plex or CodeLib Wizard)
- With VS: Add references

Follow this rule

- Never Build EXE with Plex again (*1)
 - Plex ONLY preserve App.xaml file everything else will be replaced

(*1) Applying Plex PTF may requires App.xaml refresh

Clear Line is

- You manage EXE project
 - Let Plex deal rest
-
- Plex won't catch the speed of Microsoft VS development
 - Plex won't support every single feature VS provides
 - Customization and Generator are never be a good friend
 - You don't need to wait CA, you lose nothing and just do it.

Why Native Control?

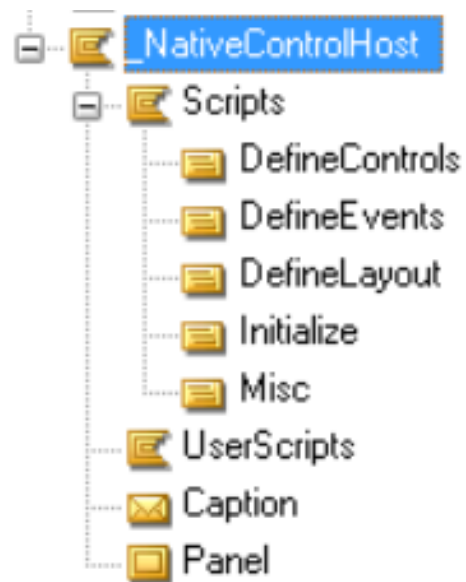
- Yes, you can use ActiveX with Plex .NET client but...
 - You sure, your ActiveX control will really work with .NET 4.x?
 - You know how old ActiveX control is?
 - You need to convert VB Script to C# anyway.
 - Google or Stack Overflow won't be your friend.

How to use Native Control?

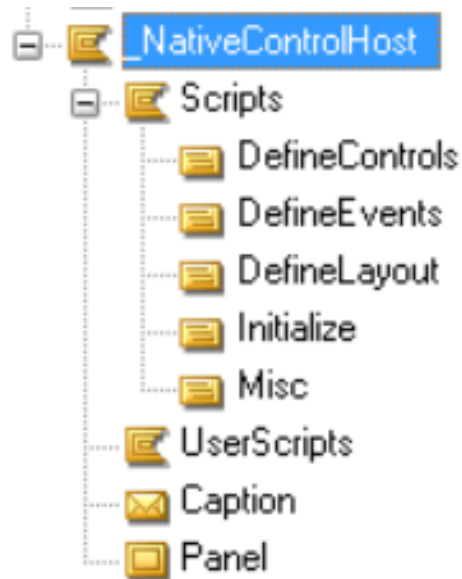
Simple Step...

- Get .NET Control
- Add Reference
- Define Controls
- Define Events
- Define Layout
- Initialize Controls

_NativeControlHost Function



_NativeControlHost Function



```
Source code: _NativeControlHost.Scripts.DefineControls
#panelmethods
DockPanel m_dp = new DockPanel ();
<ControlName> m_yourcontrol = new <ControlName> ();
#endpanelmethods
```

```
Source code: _NativeControlHost.Scripts.DefineEvents
#events
<ControlName>.<EventName> += new EventHandler (<ControlName>_<EventName>);
#endevents

#panelmethods
private void <ControlName>_<EventName>(object sender, EventArgs e)
{
    fnc.LogicalEvent("<LogicalEventName>");
}
#endpanelmethods
```

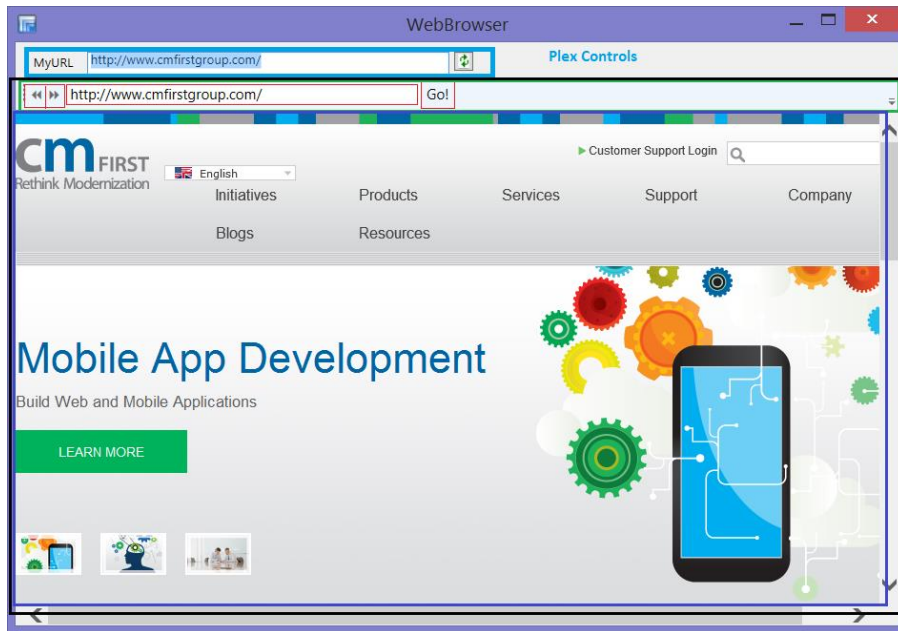
Demo



Web Browser



UI Control Design



- Plex Controls
 - Label
 - TextBox
 - Button
- DockPanel
 - ToolBar
 - Back Button
 - Forward Button
 - URL Text Box
 - Go Button
 - WebControl

Define Controls

- DockPanel
- ToolBar
- Back Button
- Forward Button
- URL Text Box
- Go Button
- WebControl

```
Source code: WebBrowser.Scri
#panelmethods
DockPanel m_dp = new DockPanel();
ToolBar m_toolbar = new ToolBar();
Button m_back = new Button();
Button m_forward = new Button();
TextBox m_urltext = new TextBox();
Button m_go = new Button();
WebBrowser m_wb = new WebBrowser();
#endpanelmethods
```

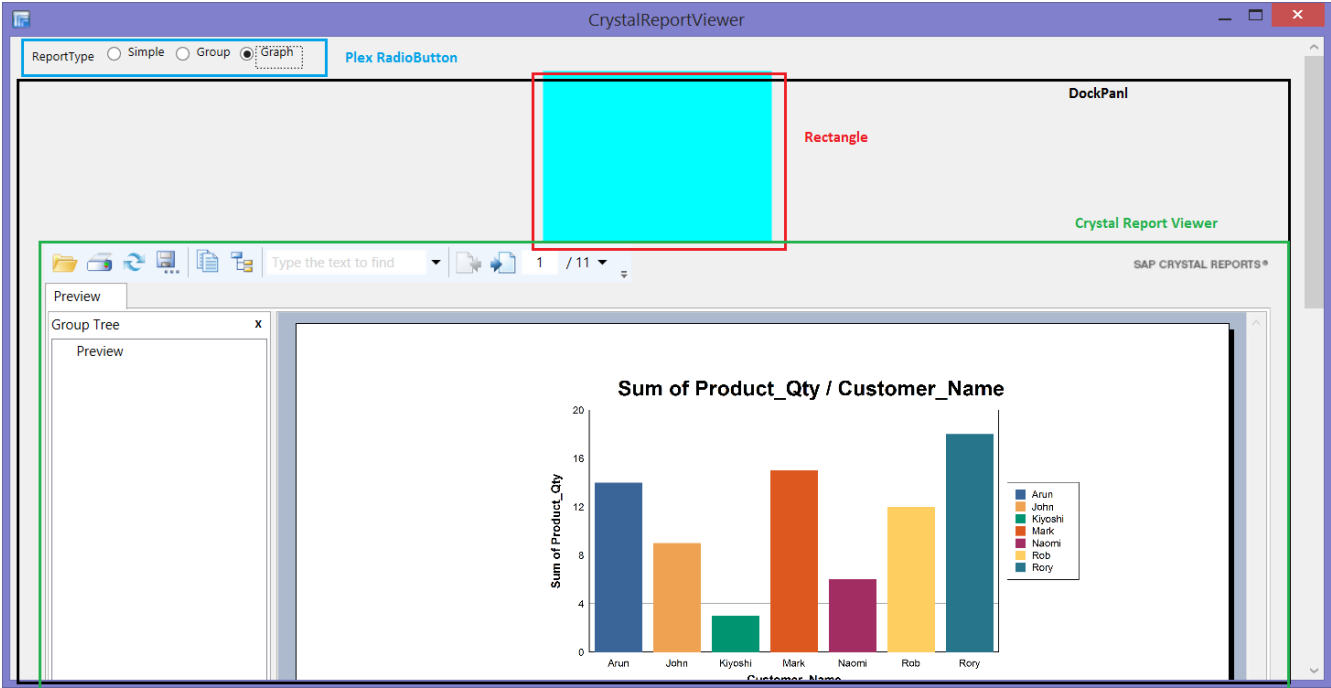
Define Layout

```
Source code: WebBrowser.Scripts.Definel
#events
m_back.Content = "\u23EA";
m_toolbar.Items.Add(m_back);
m_forward.Content = "\u23E9";
m_toolbar.Items.Add(m_forward);
m_urltext.MinWidth = 300;
m_toolbar.Items.Add(m_urltext);
m_go.Content = "Go!";
m_toolbar.Items.Add(m_go);
DockControl.SetDock(m_toolbar, Dock.Top);
m_dp.Children.Add(m_toolbar);
m_dp.Children.Add(m_wb);
m_dp.Margin = new Thickness(0, 30, 0, 0);
(ObservableContent as ObUIGrid).Children.Add(m_dp);
#endevents
```

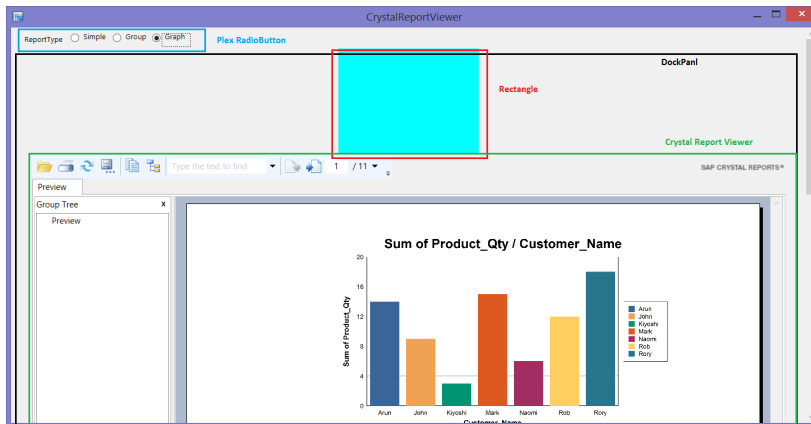
- DockPanel
 - Toolbar
 - Back Button
 - Forward Button
 - URL Text Box
 - Go Button
 - WebControl

Crystal Report Viewer

UI Control Design



UI Control Design



- Plex Controls
 - Radio Button
- DockPanel
 - Rectangle
 - Crystal Report Viewer

Define Controls

```
Source code: CrystalReportViewer.Scripts.Def
using SAPBusinessObjects.WPF.Viewer;

#panelmethods
DockPanel m_dp = new DockPanel();
CrystalReportsViewer m_crv = new CrystalReportsViewer();
Rectangle m_rect = new Rectangle();
#endpanelmethods
```

- DockPanel
- Rectangle
- Crystal Report Viewer

Define Layout

```
Source code: CrystalReportViewer.Scripts.DefineL
#events
m_rect.Width = 200;
m_rect.Height = 150;
m_rect.Fill = Brushes.Aqua;
DockPanel.SetDock(m_rect, Dock.Top);
m_dp.Children.Add(m_rect);
m_dp.Children.Add(m_crv);
m_dp.Margin = new Thickness(30, 30, 30, 0);

(ObservableContent as ObUIGrid).Children.Add(m_dp);
#endevents
```

- DockPanel
 - Rectangle
 - Crystal Report Viewer

Define Events

```
Source code: CrystalReportViewer.Scripts.DefineEvents
#events
m_rect.TouchUp += m_rect_GotTouchUp;
#endevents

#panelmethods
private void m_rect_GotTouchUp(object sender, TouchEventArgs e)
{
    TouchPoint point = e.GetTouchPoint(this);
    setBackgroundColor(point.Position.Y, point.Position.X);
    fnc.LogicalEvent("GotTouch");
}
#endpanelmethods
```

- Define TouchUp event to Rectangle
 - Changing Back Ground Color
 - Fire “GotTouch” Plex Logical Event

Define Events

```
Source code: CrystalReportViewer.Scripts.DefineEvents
#events
m_rect.TouchUp += m_rect_GotTouchUp;
#endevents

#panelmethods
private void m_rect_GotTouchUp(object sender, TouchEventArgs e)
{
    TouchPoint point = e.GetTouchPoint(this);
    setBackgroundColor(point.Position.Y, point.Position.X);
    fnc.LogicalEvent("GotTouch");
}
#endpanelmethods
```

- Define TouchUp event to Rectangle
 - Changing Back Ground Color
 - Fire “GotTouch” Plex Logical Event

You can add Any Control Events that Plex Panel Designer doesn't support.

- The report Engine is coded as .NET class Library
 - Loading Json data
 - Use Jason as DataSource of Crystal Report
 - Return Report Object

Summary

Contact Kiyoshi!

- To obtain Sample model and PDF documents
- This is too much for you but want to use them
- Want to have more deep information

terasawakiyoshi@gmail.com

www.linkedin.com/in/kiyoshiterasawa

I have One to share...

I hope you like this.



Time goes on...

Plex training..

In 2007, Passing around memory Sticks to install software

In 2009, Passing around Memory Sticks for VM ware image

In 2015, cloud base VM

But.. All wanted to do is provides Environment where CA Plex runs.

Plex is still 32 bit MFC application....

No core part has changed but you get benefit of time/cost...

API offers...

COM, (Jasmin ii), EJB, WCF, Rest API all good to **WRAP**
STORAGE/RelationalTable DB access function ;-)

All you want to do is expose Plex Functions for non Plex World.

But do you think those functions have a capability those outside world is doing?

API offers...

You have..

Read 1 row (SingleFetch) or 64 rows (BlockFetch) at a time.

To read 2 rows, you need two SingleFetch call OR

1 BlockFetch call by wasting 62 row slots.

To read 1000 rows? $1000 / 64 = 16$ times BlockFetch call?

You thought to read records as requested count?

To updates 100 records at a time...

You call UpdateRow 100 times.

You never imagine, sending 100 record images once then doing 100 Updates?

API offers...

1 extra call over the network is still a cost.

But your 1000 records of data size is larger than Youtube Video? Probably not.

Demo

