



# RELATIVITY FEST CHICAGO, IL

OCT 20-23, 2019

---

## Relativity Forms APIs

**Kevin  
Opperman**



**Adam  
Sorna**



**Keith  
Kaminski**



**Justin  
Dougherty**



# Agenda

---

- Relativity Forms Overview
- Relativity Forms Technology Changes
- Deeper Into Relativity Forms
- Example: RDO Conversion
- Q & A



## Product Overview

---

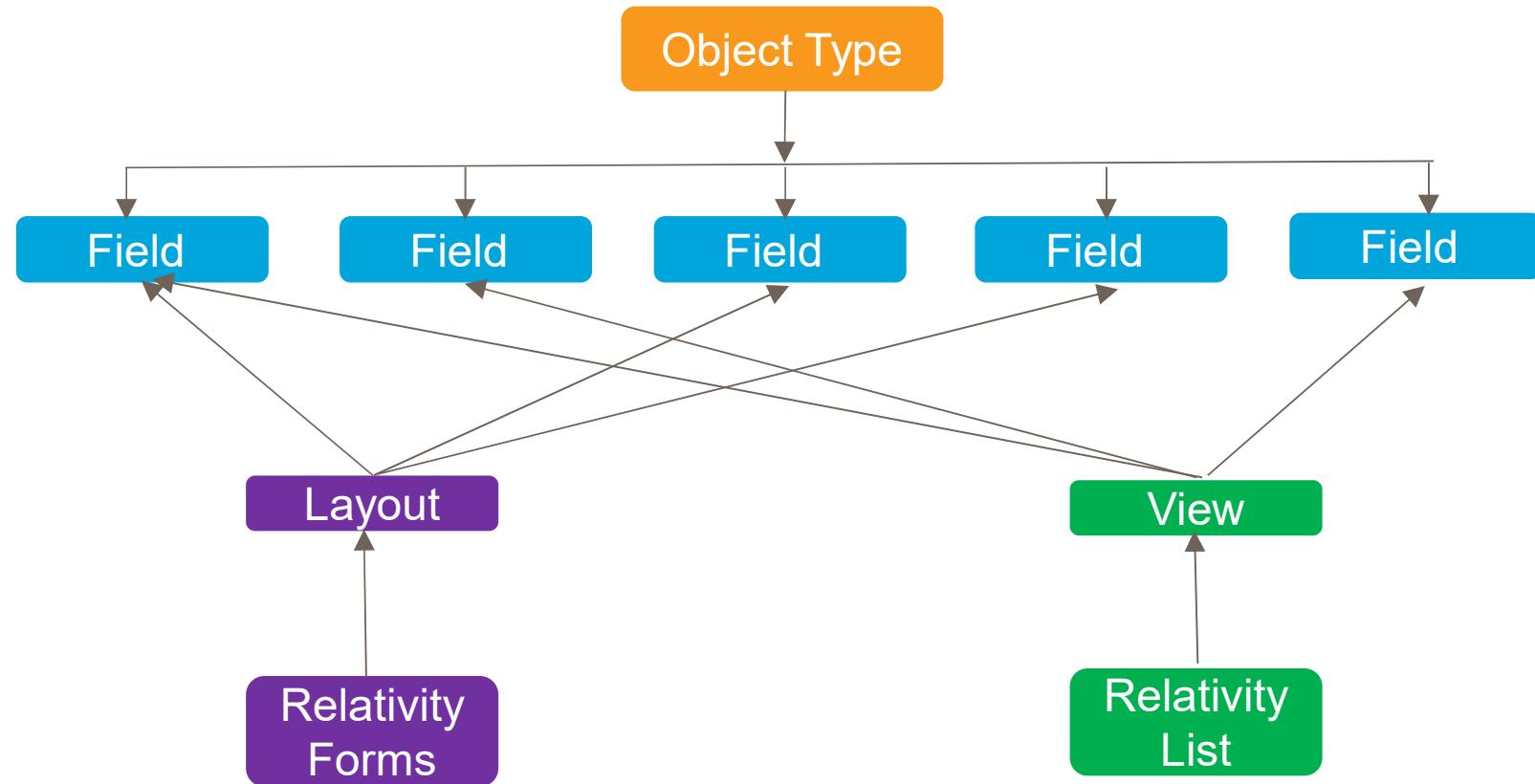
**Relativity Forms** is our new layout rendering engine with first class JavaScript APIs.

### Why is it important?

- Integration with our design system
- Part of our overall strategy to modernize our architecture
- Better development experience

# Relativity Object Model Fundamentals

---



# Relativity Forms

One | Justin Dougherty

Documents Workspace Admin Entities Justin External Tab Processing Indexing & Analytics Imaging & OCR Scripts Search Terms Reports Reporting Production Fest Object

Fest Object Layout Save and Previous Save Save and Next Save and New Save and Back Cancel Record 1 of 1 < > >>

**Object Information**

Name:	Case Overview	Artifact ID:	1054231
Date Field:	10/25/2019	Yes/No:	<input checked="" type="checkbox"/>
Users:	Dougherty, Justin	Long Text Field:	This is a long text field. This is a
File Field:	image (1).png	Fixed Length Field:	This is short text
Single Choice Field:	Responsive	Single Object Field:	Add
Multi Choice Field:	<input type="checkbox"/> Medical <input type="checkbox"/> Legal <input checked="" type="checkbox"/> HR <input checked="" type="checkbox"/> Benefits	Multi Object Field:	Select... Add

**Data**

Currency:	150.00	Whole Number Field:	14356
Decimal Value:	12.40	System Created By: Dougherty, Justin	
System Created On:	10/2/2019 9:32:27 AM CDT	System Last Modified By:	Dougherty, Justin
System Last Modified On:	10/2/2019 9:33:40 AM CDT	#RELATIVITYFEST	

## Toggling Relativity Forms

- New property on Object Type: Use Relativity Forms
- Default behavior controlled by DefaultUseRelativityFormsValue Instance Setting. Its default value is false.

Object Type Information

Name:	Foo
Parent Object Type:	Workspace
Dynamic:	Yes
Enable Snapshot Auditing On Delete:	Yes
Pivot:	Enabled
Sampling:	Enabled
Lists:	Disabled
Use Relativity Forms:	Yes
Copy Instances On Workspace Creation:	No
Copy Instances On Parent Copy:	No
Relativity Applications:	[...]

## What's different about Relativity Forms?

---

- Built with a Single Page Application architecture
  - More interactive user experience
- Introduction of Relativity Forms API
- Changes to Event Handlers that modify the UI
  - Page Interaction Event Handlers are created and are loaded differently
  - Console Event Handlers do not fire. Consoles are created from within PIEH.
  - Various other changes
- Many enhancements to improve the look-and-feel of the page

# Will you need to update your Application to support Relativity Forms?

---

## No

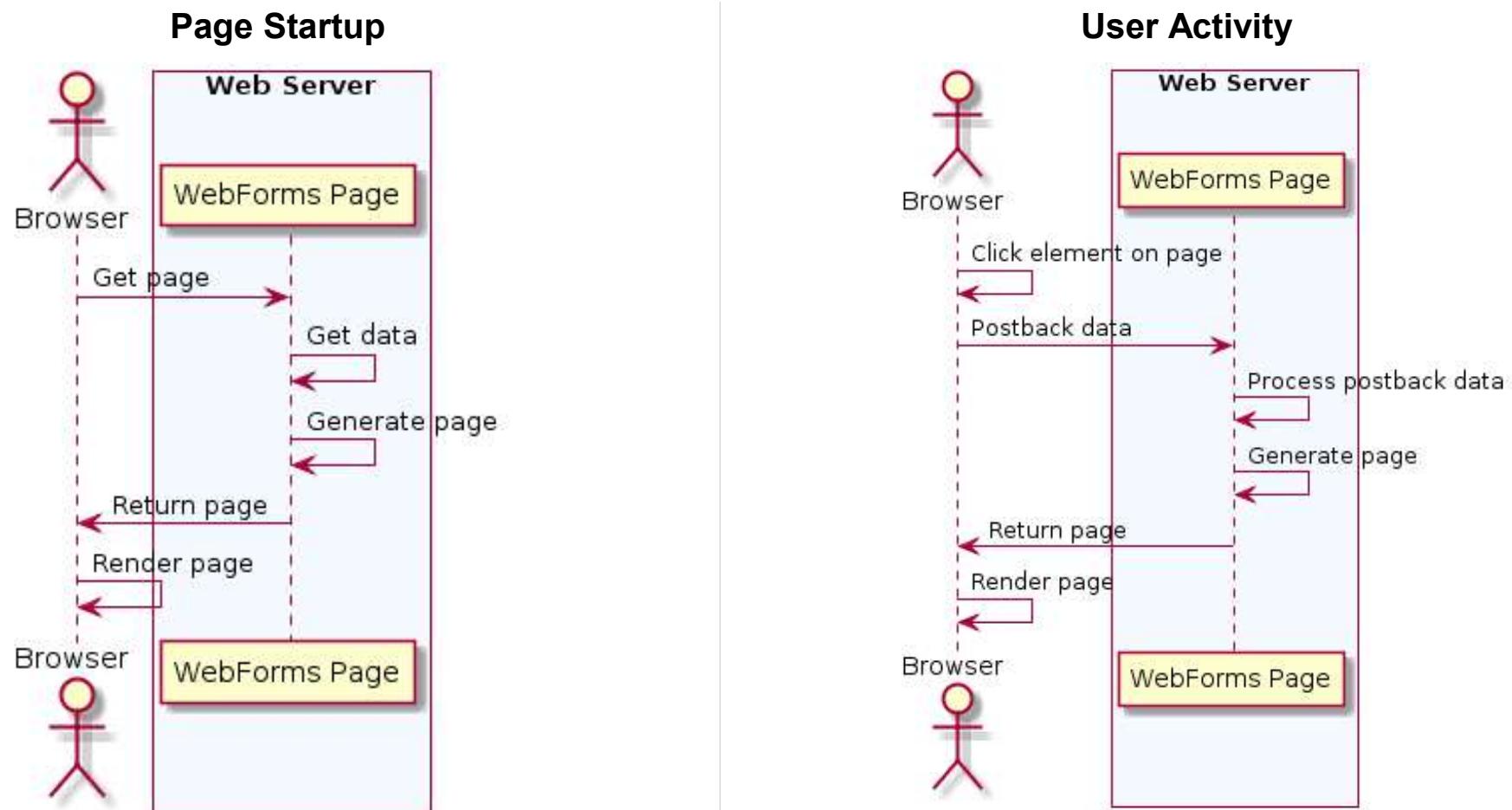
- Your application only uses built-in features available to a forms page
  - Layouts still drive the display of the page and all Layout elements are supported, including child lists
  - All action bar actions are fully functional out-of-the-box – buttons, changing layouts, etc...
  - Object Rules are respected
- Your application uses only the following Event Handlers on your Object Types:
  - PreLoad, PreSave, PostSave, PreDelete

## Yes

- Your application contains use Console or Page Interaction Event Handlers
- Your layouts include HTML-enabled or Layout Text fields and they modify content elsewhere on the page
- You use PreLoad Event Handlers to supply default values.
- You display messages with HTML or JavaScript content as part of PreLoad/PreSave/PostSave Event Handlers
- You use the RedirectToLayout PostExecuteAction.

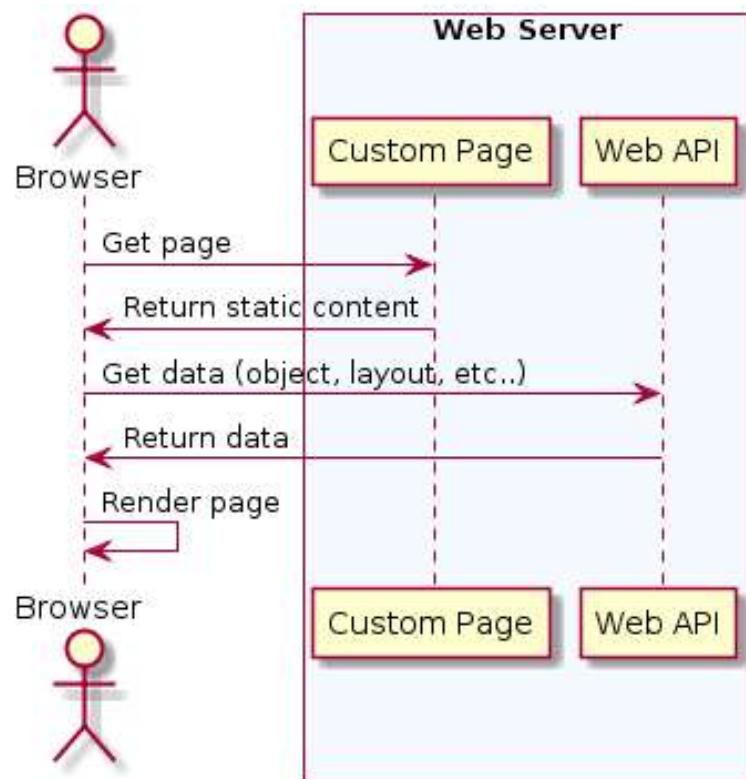
# Relativity Forms Technology Changes

# Classic Forms Architecture

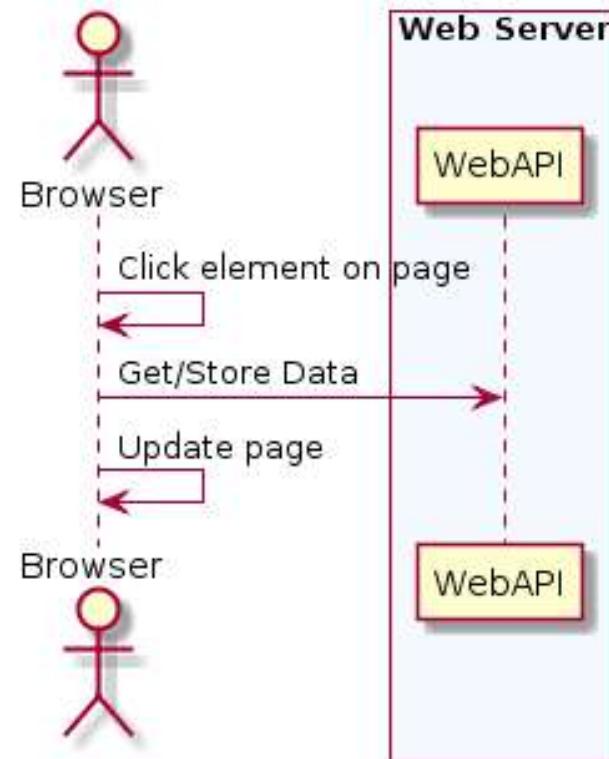


# Relativity Forms Architecture

## Page Startup



## User Activity



## A peek behind the curtain: Relativity Forms Technology Stack

---



Fuse.js



jspm.io



**CHOICES.JS**

# PageInteractionEventHandler Changes:

---

Classic:

```
0 references
public class MyPageInteractionEventhandler : PageInteractionEventHandler
{
    2 references
    public override Response PopulateScriptBlocks()
    {
        //Create a response object with default values.
        kCura.EventHandler.Response retVal = new kCura.EventHandler.Response() { Success = true };
        String alert = "<script type='text/javascript'> alert('My JavaScript fired!');</script>";
        this.RegisterStartupScriptBlock(new kCura.EventHandler.ScriptBlock() { Key = "alertKey", Script = alert });
        return retVal;
    }
}
```

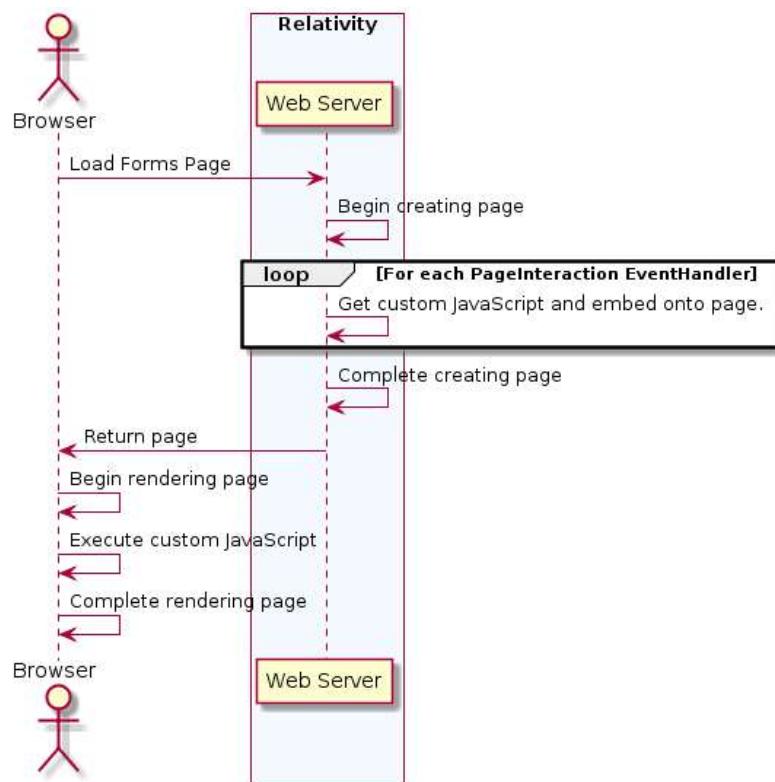
Relativity Forms:

```
0 references
public class MyPageInteractionEventhandler : PageInteractionEventHandler
{
    2 references
    public override Response PopulateScriptBlocks()
    {
        return new Response();
    }

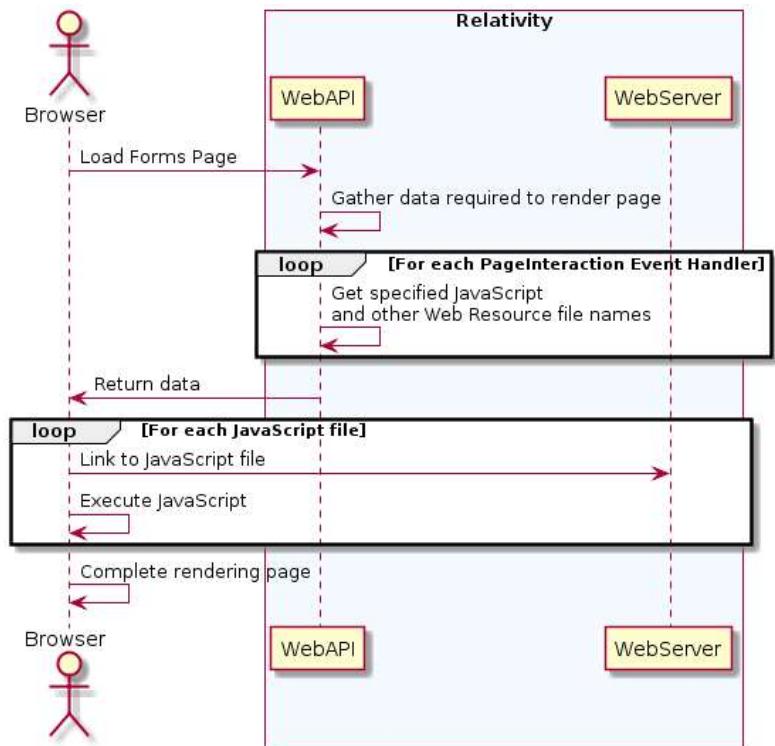
    1 reference
    public override string[] ScriptFileNames => new string[] { "fireAlert.js" };
}
```

# Changes to how JavaScript is loaded

## Classic

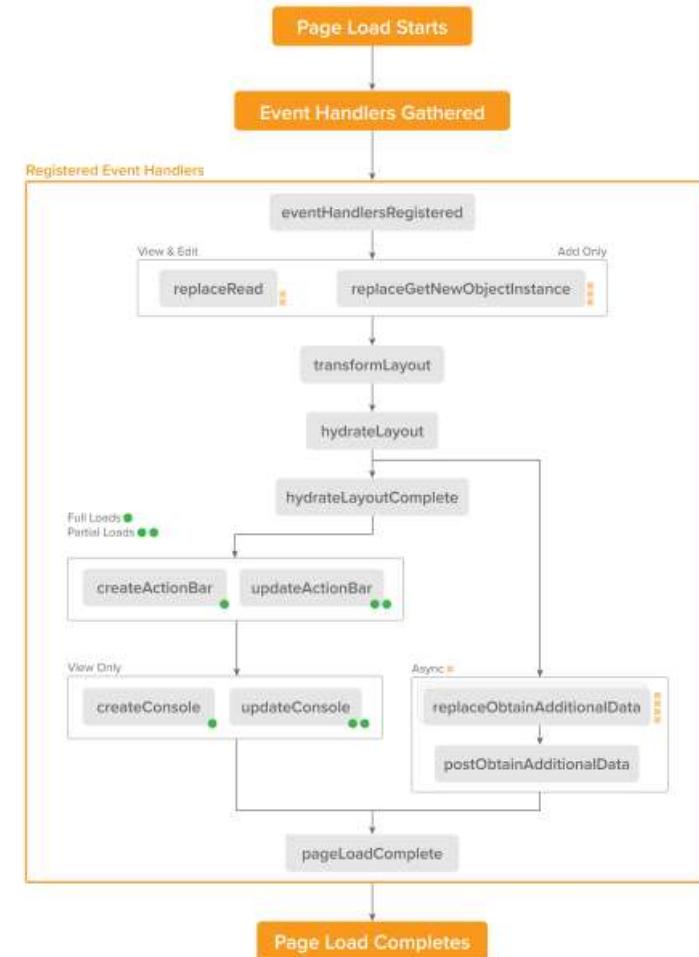


## Relativity Forms



# Relativity Forms API

- JavaScript API that allows customization of Forms page
  - Manipulate nearly all the content on the page – categories, fields, consoles, action bar buttons, lists, etc...
  - Create consoles
  - Implement common patterns and best practices like calling Kepler services and launching modals.
  - Well-defined pipelines with over two dozen page-level events that you can hook into

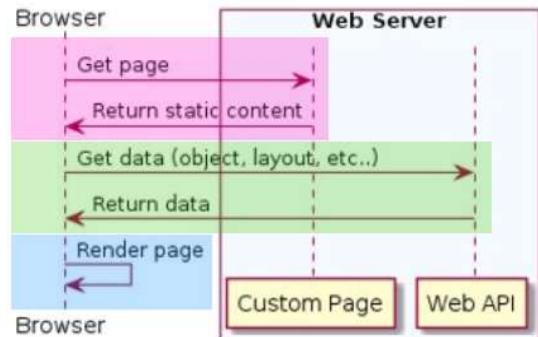


# Deeper Into Relativity Forms

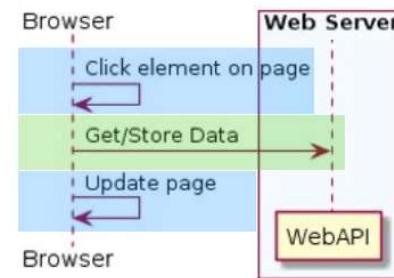
# Main Parts of Relativity Forms

- Relativity Forms Application
- FormViewModel
  - Metadata
  - State
  - Rendering
- PipelineManager
  - Supply data
  - Control life cycle

## Page Startup



## User Activity



Application      PipelineManager      FormViewModel

# Pipelines: The workflows you can tailor

- Load
  - Full or partial
- Change
  - Response to Field value change
- Submit
  - Create and update
- Delete
  - Current object and child object list items
- Item List
  - Setup columns and actions, handle reload

# Relativity Forms API

- Event-specific parameters
- The "this" binding
- eventNames
- convenienceApi

# Relativity Forms convenienceApi

- Field and static text manipulation
- Markup generators for action bars and consoles
- Http client
- i18n Internationalization
- Stock and custom modals
- Relativity Forms and general popup windows
- Validation Helpers
- Promise factory
- And more

# Example

---

Conversion of a Simple RDO to Relativity Forms

# Simple Object Type

- **PageInteractionEventHandler:**

One Field shown or hidden based upon another's value

JavaScript computing a status summary based upon field values, alters a static text in edit mode.

Summary generation shared on window for use in ConsoleEventHandler in View mode

- **ConsoleEventHandler:**

Shared summary generation used in an alert fired by the Console button

## View Mode:

Contain Yourself Application Admin Workspace Admin Test RF Features  
Contain Yourself Layout Edit Delete Back Edit Permissions View Audit Record 1 of 3

Contain yourself

Name: Calm Carla Contain Enthusiasm: Ok, I will.  
Current Enthusiasm: 100 Maximum Enthusiasm: 400

Contain yourself

p-dv-vm says  
Calm Carla is could still be more enthusiastic.

OK

CONTAIN YOURSELF CONSOLE  
Report Enthusiasm

## Edit Mode:

Contain Yourself Application Admin Workspace Admin Test RF Features  
Contain Yourself Layout Save Save and Next Save and New Save and Back Cancel Record 1 of 3

Contain yourself

Name: Calm Carla Contain Enthusiasm:  Ok, I will.  No, I cannot.  
Current Enthusiasm: 100 Maximum Enthusiasm: 400

Calm Carla is could still be more enthusiastic.

Contain Yourself Application Admin Workspace Admin Test RF Features  
Contain Yourself Layout Save Save and Next Save and New Save and Back Cancel Record 1 of 3

Contain yourself

Name: Calm Carla Contain Enthusiasm:  Ok, I will.  No, I cannot.  
Current Enthusiasm: 100

Calm Carla is brimming with unbridled enthusiasm!

# Page Interaction EventHandler

## Classic:

```
using System;
using kCura.EventHandler;

namespace Classic
{
    [kCura.EventHandler.CustomAttributes.Description("Classic Page Interaction EventHandler")]
    [System.Runtime.InteropServices.Guid("bbf7f065-92f4-4b93-9fcc-fcea596a950f")]
    public class PageInteraction : kCura.EventHandler.PageInteractionEventHandler
    {
        public static readonly Guid CURRENT_ENTHUSIASM_GUID = new Guid("692C7C4D-3C6B-4729-86E4-426A101E1479");
        public static readonly Guid MAXIMUM_ENTHUSIASM_GUID = new Guid("5426056E-C815-430C-93FB-57C979A88715");
        public static readonly Guid CONTAIN_ENTHUSIASM_GUID = new Guid("0EE79570-5540-49F9-8ED-BDE5840AF433");

        public override Response PopulateScriptBlocks()
        {
            string CONTAIN_ID = this.ActiveArtifact.Fields[CONTAIN_ENTHUSIASM_GUID.ToString()].ArtifactID.ToString();
            string MAXIMUM_ID = this.ActiveArtifact.Fields[MAXIMUM_ENTHUSIASM_GUID.ToString()].ArtifactID.ToString();
            string CURRENT_ID = this.ActiveArtifact.Fields[CURRENT_ENTHUSIASM_GUID.ToString()].ArtifactID.ToString();

            // controls are different in View mode than in Edit (which includes both Edit and Add)
            bool isViewMode = this.PageMode == kCura.EventHandler.Helper.PageMode.View;
            string jsIsViewMode = isViewMode ? "true" : "false"; // for use in Javascript

            // Create a response object with default values
            kCura.EventHandler.Response retVal = new kCura.EventHandler.Response();
            retVal.Success = true;
            retVal.Message = string.Empty;

            // Register script
            [JavaScript Region]

            this.RegisterStartupScriptBlock(new kCura.EventHandler.ScriptBlock() { Key = "startup", Script = startUp });

            return retVal;
        }
    }
}
```

## Relativity Forms:

```
using kCura.EventHandler;

namespace RelativityFormsConversion
{
    [kCura.EventHandler.CustomAttributes.Description("Relativity Forms Converted Page Interation EventHandler")]
    [System.Runtime.InteropServices.Guid("ba8ebea2-7228-455b-b13f-57d505b68d72")]
    public class PageInteractionEventHandler : kCura.EventHandler.PageInteractionEventHandler
    {
        public override Response PopulateScriptBlocks()
        {
            return new Response();
        }

        public override string[] ScriptFileNames
        {
            get
            {
                return new string[] { "relativityFormsConversion.js" };
            }
        }
    }
}
```

# Pain points in Classic:

- JavaScript as a String in .NET
- Fields in the JavaScript
- Sharing information between Page Interaction and Console Event Handlers

Classic:

```
29         // Register script
30         #region JavaScript Region
31
32         String startUp = String.Format(@"<script type=""text/javascript"">
33 $(document).ready(function() {{
34
35             // functions to get field Cells (parent element of the field you want
36             function getFieldCellByArtifactId(artifactId) {{
37                 return document.querySelector("[faartifactid = '" + artifactId + "']").parentElement;
38             }}
39
40             function getFieldCellByFriendlyName(friendlyName) {{
41                 return document.querySelector("[fafriendlyname=''" + friendlyName + "'']").parentElement;
42             }}
43
44             // functions to get values for fields regardless of PageMode (Input for text and number, Radio for yes/no* (in radio mode))
45             function getFieldInputValue(fieldCell) {{
46                 var value = fieldCell.childNodes[2].childNodes[0][isViewMode ? 'innerText' : 'value'];
47                 return value.trim();
48             }}
49
50             function getFieldRadioValue(fieldCell) {{
51                 var value;
52                 if (isViewMode) {{
53                     value = fieldCell.childNodes[2].textContent === yesValue;
54                 } else {{
55                     var radio = fieldCell.childNodes[2];
56                     var yesRadio = radio.querySelectorAll("[type = 'radio']")[0];
57
58                     if (yesRadio.checked) {
59                         value = true;
60                     } else {
61                         value = false;
62                     }
63                 }}
64
65                 return value;
66             }}
67
68             var isViewMode = {0};
69             var containEnthusiasmFieldId = '{1}';
70             var maximumEnthusiasmFieldId = '{2}';
71             var currentEnthusiasmFieldId = '{3}';
72             var yesValue = 'Ok, I will.';
73             var nameCell = getFieldCellByFriendlyName('Name');
74             var containEnthusiasmCell = getFieldCellByArtifactId(containEnthusiasmFieldId);
75             var maximumEnthusiasmCell = getFieldCellByArtifactId(maximumEnthusiasmFieldId);
76             var currentEnthusiasmCell = getFieldCellByArtifactId(currentEnthusiasmFieldId);
77             var staticText = $('.layoutText:contains("Contain yourself"))[0];
78
79             // Execute immediately upon document ready
80             toggleMaximumEnthusiasmByRadio();
81
82             if (isViewMode) {{
83                 window.getFlavorText = getFlavorText;
84             } else {{
85                 containEnthusiasmCell.onclick = toggleMaximumEnthusiasmByRadio;
86                 [nameCell, containEnthusiasmCell, maximumEnthusiasmCell, currentEnthusiasmCell].forEach(appendFunctionToInputBlur);
87             }}
88         }};
89     </script>", jsIsViewMode, CONTAIN_ID, MAXIMUM_ID, CURRENT_ID);
90
91         #endregion
92
93         this.RegisterStartupScriptBlock(new kCura.EventHandler.ScriptBlock() { Key = "startup", Script = startUp });
94     }
```

# Enable Relativity Forms

Object Type:

Object Type Information

Name:	Contain Yourself
Parent Object Type:	Workspace
Dynamic:	Yes
Enable Snapshot Auditing On	
Delete:	No
Pivot:	Disabled
Sampling:	Disabled
Lists:	Disabled
Use Relativity Forms:	Yes
Copy Instances On Workspace	
Creation:	No
Copy Instances On Parent	
Copy:	No
Relativity Applications:	Contain yourselves, people

Event Handlers:

DLL	Class Name	Company Name	Description	Execution Type	Application Name
(All)	(All)	(All)	(All)	(All)	(All)
FestClassicEventHandler	Classic.ConsoleEH	Classic Console EventHandler	Console	Contain yourselves, people	

Event Handlers  
Not Firing:

Contain Yourself Layout

Contain yourself	Name: Enthusiastic Edna	Contain Enthusiasm: No, I cannot.
	Current Enthusiasm: 100	Maximum Enthusiasm: 20
Contain yourself		

# Add Relativity Forms Scripts

- New or Updated Page Interaction EventHandler
- JavaScript file specified within the EventHandler (only a stub for now)

## PageInteractionEventHandler

```
using kCura.EventHandler;

namespace RelativityFormsConversion
{
    [kCura.EventHandler.CustomAttributes.Description("Relativity Forms Converted Page Interaction EventHandler")]
    [System.Runtime.InteropServices.Guid("ba8bea2-7228-455b-b13f-57d505b68d72")]
    public class PageInteractionEventhandler : kCura.EventHandler.PageInteractionEventHandler
    {
        public override Response PopulateScriptBlocks()
        {
            return new Response();
        }

        public override string [] ScriptFileNames
        {
            get
            {
                return new string[] { "relativityFormsConversion.js" };
            }
        }
    }
}
```

## relativityFormsConversion.js

```
(function (eventNames, convenienceApi) {
    var eventHandlers = {};

    // internal (or later, privileged) variables and functions will go here

    // eventHandlers[eventNames.HYDRATE_LAYOUT_COMPLETE] = function () {
    //     This will contain the initial state hide instruction
    // };

    // eventHandlers[eventNames.PAGE_INTERACTION] = function (modelData, event) {
    //     This will contain the change event handling
    // };

    eventHandlers[eventNames.CREATE_CONSOLE] = function () {
        // This will contain the console content code, but for
        // now does nothing. This is enough to make a console appear
    };

    return eventHandlers;
})(eventNames, convenienceApi));
```

# Results

The screenshot shows a web application interface with a form titled "Contain yourself". The form fields include "Name: Calm Carla", "Contain Enthusiasm: Ok, I will.", "Current Enthusiasm: 100", and "Maximum Enthusiasm: 400". Below the form is a developer tools Network tab showing a single request: "relativityFormsConversion.js" with a status of 200, type "fetch", initiator "app-build-5ee8864836.js:47203", size 1.0 KB, and time 1.36 s.

Name	Status	Type	Initiator	Size	Time	Waterfall
relativityFormsConversion.js	200	fetch	app-build-5ee8864836.js:47203	1.0 KB	1.36 s	

- Relativity Forms EventHandler script now being downloaded
- Console now appearing on the side

# Update the JavaScript File: Classic vs Relativity Forms

## First-class GUID support for Fields

- GUIDs can be used directly by Relativity Forms but must be ArtifactID or Friendly Name in Classic

Classic:

```
11 |     public static readonly Guid MAXIMUM_ENTHUSIASM_GUID = new Guid("5426056E-C815-430C-93FB-57C979A88715");
17 |     string MAXIMUM_ID = this.ActiveArtifact.Fields[MAXIMUM_ENTHUSIASM_GUID.ToString()].ArtifactID.ToString();
112|     var maximumEnthusiasmFieldId = '{2}';
```

Relativity Forms:

```
5 |     var MAXIMUM_ENTHUSIASM_GUID = "5426056E-C815-430C-93FB-57C979A88715".toLowerCase();
```

# Update the JavaScript File: Classic vs Relativity Forms

## Direct read and manipulation of Field and Static Text

- Fields can be read and manipulated without getting the DOM elements; values return in type appropriate to Field

Classic:

```
116 |     var containEnthusiasmCell = getFieldCellByArtifactId(containEnthusiasmFieldId);
117 |     var maximumEnthusiasmCell = getFieldCellByArtifactId(maximumEnthusiasmFieldId);
118 |
119 |     function getFieldCellByArtifactId(artifactId) {{
120 |         return document.querySelector("[faartifactid = '" + artifactId + "']");
121 |     }}
122 |
123 |     containEnthusiasmCell.onclick = toggleMaximumEnthusiasmByRadio;
124 |
125 |     function toggleMaximumEnthusiasm(show) {{
126 |         maximumEnthusiasmCell.hidden = !show;
127 |     }
128 |
129 |     function toggleMaximumEnthusiasmByRadio() {{
130 |         toggleMaximumEnthusiasm(getFieldRadioValue(containEnthusiasmCell));
131 |
132 |         function getFieldRadioValue(fieldCell) {{
133 |             var value;
134 |             if (isEditMode) {{
135 |                 value = fieldCell.childNodes[2].textContent === yesValue;
136 |             } else {{
137 |                 var radio = fieldCell.childNodes[2];
138 |                 var yesRadio = radio.querySelectorAll("[type = 'radio']")[0];
139 |                 value = yesRadio.checked;
140 |             }
141 |             return value;
142 |         }}
143 |     }
144 | }
```

Relativity Forms:

```
10  |     function updateMaximumEnthusiasm() {
11  |         var field = convenienceApi.fieldHelper;
12  |
13  |         return field.getValue(CONTAIN_ENTHUSIASM_GUID).then(function (containEnthusiasm) {
14  |             return field.setIsHidden(MAXIMUM_ENTHUSIASM_GUID, !containEnthusiasm);
15  |         });
16  |     };
17 | }
```

# Update the JavaScript File: Classic vs Relativity Forms

## Form metadata-aware JavaScript

- JavaScript handlers are aware of the form mode without injecting it from .NET handlers, or parsing Query String

Classic:

```
21 |     bool isViewMode = this.PageMode == kCura.EventHandler.Helper.PageMode.View;
22 |     string jsIsViewMode = isViewMode ? "true" : "false"; // for use in Javascript
110|     var isViewMode = {0};
```

Relativity Forms:

```
79 |     updateFlavorText(this.viewModelTypeId);
```

# Update the JavaScript File: Classic vs Relativity Forms

## Code Execution Timing

- Classic Page Interaction JavaScript fires easily either as the script is interpreted, or once the DOM is loaded; getting finer timing requires manually binding to DOM events
- Relativity Forms provides multiple specific extension points within the Load, Change, Submit, and Delete pipelines

```
function (eventNames, convenienceApi) {
    var eventHandlers = {};
    var CONTAIN_ENTHUSIASM_GUID = "0EE79570-5540-49F9-8DED-BDE5840AF433".toLowerCase();
    var containEnthusiasmId = null;
    ...
    eventHandlers[eventNames.HYDRATE_LAYOUT_COMPLETE] = function ()...;
    eventHandlers[eventNames.PAGE_INTERACTION] = function (modelData, event)...;
    eventHandlers[eventNames.CREATE_CONSOLE] = function ()...;
    return eventHandlers;
}(eventNames, convenienceApi));
```

# Update the JavaScript File: Classic vs Relativity Forms

## Scope-sheltered JavaScript code sharing by default

- Page Interaction and Console EventHandlers are one in the same in Relativity Forms, Console can access what the other handlers can access

### Classic:

```
$(document).ready(function() {{
    function getFlavorText() {{
        ...
    }}
    ...
    if (isViewMode) {{
        window.getFlavorText = getFlavorText;
    }}
    ...
}});
```

//Create the JavaScript for the button and set the button property.  
StringBuilder clickHandlerJavaScript = new StringBuilder("if (typeof window.getFlavorText === 'function') {}  
clickHandlerJavaScript.Append(" alert(window.getFlavorText()); } else {};  
clickHandlerJavaScript.Append(" alert('getFlavorText is unavailable on the window.');" );

### Relativity Forms:

```
(function (eventNames, convenienceApi) {
    var eventHandlers = {};
    ...
    function getFlavorText()...
    ...
    eventHandlers[eventNames.HYDRATE_LAYOUT_COMPLETE] = function ()...;

    eventHandlers[eventNames.PAGE_INTERACTION] = function (modelData, event)...;

    eventHandlers[eventNames.CREATE_CONSOLE] = function ()...;

    ...
    return eventHandlers;
})(eventNames, convenienceApi));
```

# Results

- Object Type

The screenshot shows the Relativity application's 'Object Type' configuration screen. At the top, there are buttons for Edit, Delete, Back, Edit Permissions, and View Audit. Below this is a section titled 'Object Type Information' containing various configuration settings:

- Name: Contain Yourself
- Parent Object Type: Workspace
- Dynamic: Yes
- Enable Snapshot Auditing On Delete: No
- Pivot: Disabled
- Sampling: Disabled
- Lists: Disabled
- Use Relativity Forms: Yes
- Copy Instances On Workspace Creation: No
- Copy Instances On Parent Copy: No
- Relativity Applications: Contain yourselves, people

Below this is a grid titled 'Event Handlers' with columns for DLL, Class Name, Comp..., Description, Execution Type, and Application Name. The grid contains the following data:

DLL	Class Name	Comp...	Description	Execution Type	Application Name
(All)	(All)	(All)	(All)	(All)	(All)
FestClassicEventHandlers	Classic.ConsoleEH		Classic Console EventHandler	Console	Contain yourselves, people
FestClassicEventHandlers	Classic.PageInteraction		Classic Page Interaction EventHandler	Page Interaction	Contain yourselves, people
RelativityFormsConversion	RelativityFormsConversion.PageInteractionEventhandler		Relativity Forms Converted Page Iteration EventHandler	Page Interaction	Contain yourselves, people

# Results

- Relativity Forms in action

The screenshot displays a sequence of five Relativity Form submissions for the 'Contain Yourself' layout. Each submission shows the following fields:

- Name:** Overzealous Oliver
- Current Enthusiasm:** 10000
- Contain yourself** (button)
- Contain Enthusiasm:** No, I cannot.

After each submission, a modal dialog box appears with the message "p-dv-vm says Overzealous Oliver is brimming with unbridled enthusiasm!" and an "OK" button.

The second submission includes a "Report Enthusiasm" button in the right sidebar.

The third submission includes a "Maximum Enthusiasm:" field set to 9000.

The fourth submission includes a "Contain Enthusiasm:" dropdown menu set to "No, I cannot."

The fifth submission includes a "Maximum Enthusiasm:" field set to 9000.

A black and white photograph of a woman with dark hair, wearing a dark blazer over a patterned top. She is looking down at a piece of paper she is holding in her hands. The background is dark and out of focus.

# Questions?

# Resources

---

- Platform Documentation: <https://platform.relativity.com/RelativityOne/>  
Relativity Forms:  
<https://platform.relativity.com/RelativityOne/Content/Relativity%20Forms/Relativity%20Forms%20API.htm>
- Relativity Forms Cheat Sheet: <https://github.com/relativitydev/relativity-forms-workshop-fest-2019/raw/master/docs/workshop/Relativity%20Forms%20Cheat%20Sheet.pdf>
- DevHelp: <https://devhelp.relativity.com/c/relativityui/relativityforms>  
Category: RelativityUI  
Sub-Category: RelativityForms
- Feedback: [RelativityFormsFeedback@relativity.com](mailto:RelativityFormsFeedback@relativity.com)