|  |  |  |  |
| --- | --- | --- | --- |
|  | Page Model - Dev Design.docx  [Status]  [Related Documents](file:///C:/Users/rukhan.REDMOND/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Content.Outlook/2UCU2MVS/TBD) | Owner | DevName |
| Dev Lead | LeadName |
| PM | PMName |
| Test | TestName |
| UE | WriterName |
| SE | SEName |
| UX | Usability |
|  |  |  |  |

Contents

[1 Introduction 2](#_Toc423620737)

[2 Goals 3](#_Toc423620738)

[3 Design 3](#_Toc423620739)

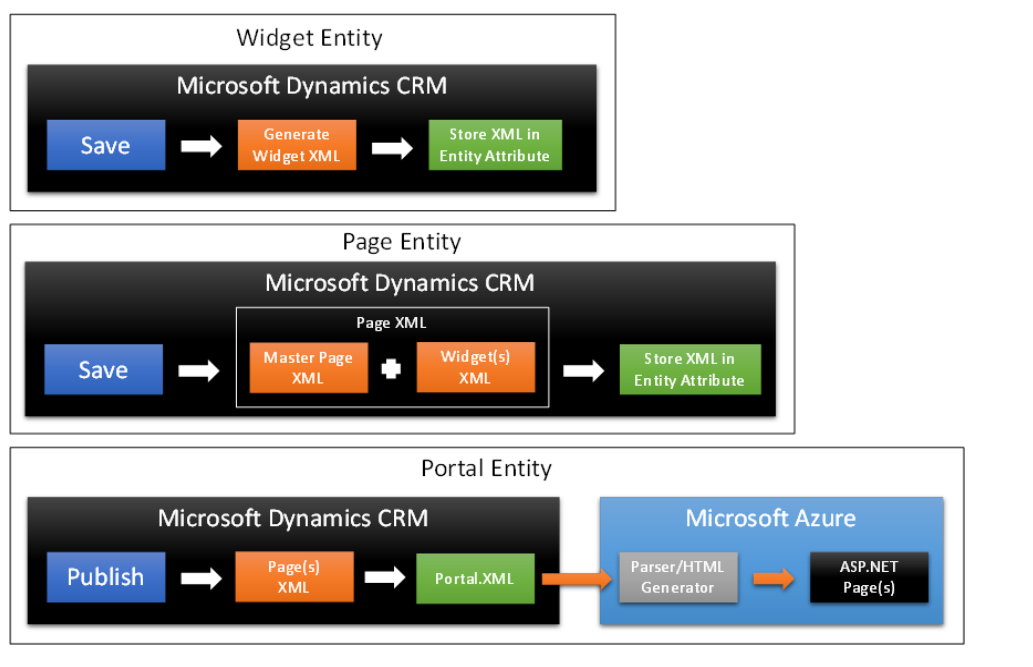
[4 Document History 14](#_Toc423620740)

# Introduction

## Architectural Overview

Architectural Overview

Description:   
  
1) The UI interface must contain the 'Save' button   
2) The button name must be translated and setted up by admin  
3) While saving, the plugin must analyze the CRM Data Stored in DB + Some MetaData and Generate Page/Widget XML   
4) When it is published it must be published at the Portal Entity Level to Azure as a package and sent to Azure, get directed to the Parser and generate the corresponding ASP.NET Page

The following architecture describes the main workflow of xml structure's generation and saves it to entity attribute  


## Goalsher all the necessary information of each server and display it.

1. Create common structure MasterPage and Page elements for Portal solution  
   - Create common structure for MasterPage to make sure the MasterPage structure is the same for all places.  
   - Create common structure for Page to make sure the Page structure is the same for all places.  
     
     
   Make sure the structures of MasterPage and Page are similar for CRM and Azure  
   -  
   -  
     
   Make sure the structures of MasterPage and Page are similar at the CRM's output and Azure's input  
     
   -  
   -  
     
     
   Make sure the structures of MasterPage and Page are valid and correct for all CRM and Microsoft politics

* Change the somketest executable so it can access a database and fill it with the results.
* Make a web server that can pull the information of the database and transform the data to the UI.

## Optimize all the program .

# Goals

## Placeholer

* Something important

## Placeholer

* Something important

## Placeholer

* Something important

# Design

## General Approach

ToDo: describe what we know

1. On the CRM Side we must generate the proper XML Structure
   1. We will be bringing specific data attributes from the Form XML generated by CRM Forms and merge those particular data points into the Control XML.
   2. Once the Control XML is finalized for the particular form, we will allow the user to add that particular form (Control XML) to a specific page.
   3. When the page is configured and published it will be pushed to Azure, Translation of the page particulars will occur at this point before reaching Azure & the HTML Parser/Generator.
2. Once the XML Page is complete, it is pushed to Azure where a Parser/Generator will take the XML and create a static ASP.NET Page in Azure.
   1. The logic within the Parser/Generator will be set to coincide with the overall standards & mapping between XML attributes/tags to corresponding HTML & Metadata.
   2. On the HTML side we will be assigning the appropriate ID’s/Classes/Tags/Labels/etc. based on the information sent over from CRM. Please see sections 3.3 & 3.4 for code samples & structure.

## Translation (TODO)

ToDo: describe what we know

## Master Page Attributes

Each Master Page has the following attributes which should be depicted in the generated XML. Please see the detailed description of CRM attributes in “FormWidget – Feature Spec” document, REQ 9 – REQ 20.

1. **Id**

The "Id" will be defined by the "Id" tag in XML. "Id" tag doesn't render in the HTML web page.

1. **Name**

The "Name" will be defined by the "Name" tag in XML. The "Name" will be defined by "title" tag in HTML

1. **LocalizedName**

The "LocalizedName" will be defined by the " LocalizedName " tag in XML. The " LocalizedName " will be defined by "title" tag in HTML.

1. **RequiredLevel**

The "RequiredLevel" will be defined by "RequiredLevel" tag in XML.

1. **Portal**

The Portal will be defined by  the "Portal" tag which may contain "Id" and "Name" tags in XML.

1. **Header**

The Header will be defined by the "Header" tag in XML. The "Header" also may contain <MasterPageHead > and <sections> tags. In HTML  The Header will be defined by the “div” tag with masterPageHeader id and masterPageHeader class.

1. **Body**

The "Body" will be defined by a "Body" tag which will be contain "MasterPageTop" and "MasterPageBottom" tags in XML. The "Body" will be defined by a "div" tag with "pageLayout" class in HTML.

## Page Attributes

Each Page has the following attributes which should be depicted in the generated XML. Please see the detailed description of CRM attributes in “FormWidget – Feature Spec” document, REQ 2 – REQ6.

1. **Id**

The "Id" will be defined by the "Id" tag in XML. "Id" tag doesn't render in the HTML web page.

1. **Name**

The "Name" will be defined by the "Name" tag in XML. The "Name" will be defined by "title" tag in HTML

1. **LocalizedName**

The "LocalizedName" will be defined by the "Name" tag in XML. The " LocalizedName " will be defined by "title" tag in HTML

1. **MasterPage**

The "MasterPage" will be defined by "MasterPage" tag wich will be contain "Id" and "Name" tags in XML.

1. **Type**

The "Type" will be defined by "Type" tag in XML.

1. **Entity**

The "Entity" will be defined by "Entities" tag which will be contain set of "Entity" tags in XML and will be defined as widgets in HTML

1. **RelativeUrl**

The "RelativeURL" will be defines by "RelativeURL" tag in XML.

1. **MainHeader**

The "MainHeader" will be defined by "MainHeader" tag in XML

1. **PageBody**

The "PageBody" will be defines by "PageBody" tag in XML and in HTML the "PageBody" will be defined by "div" tag with "pageLayout" class attribute.

## User Interface, Localization and Accessibility

The UI will be implemented in Javascript. The UI will show the status of GEO/SW/DC/POD/SG. This UI will be in the Support Dashboard. The accessibility will be for the CRM Product Group Members and CRM operation, site and support.

ToDo: write something about the HTML5 standards and Responsive UI as Nancy and Chris wrote us.

### Localization

### Accessibility

### Accessibility

|  |  |
| --- | --- |
| Work Items | Status |
| Review [Dev Guidelines](onenote:http://sharepoint/sites/crmhome/engineering/Engineering%20Notebooks/Accessibility/Guidelines.one#Dev%20Guidelines&section-id={FB59283C-D5D0-4A19-B618-24F58ACACE59}&page-id={4D1FD662-4F55-4FBA-B21C-F5C26043C110}&end)  ([Web view](http://sharepoint/sites/crmhome/engineering/_layouts/OneNote.aspx?id=%2fsites%2fcrmhome%2fengineering%2fEngineering%20Notebooks%2fAccessibility&wd=target%28Guidelines.one%7cFB59283C-D5D0-4A19-B618-24F58ACACE59%2fDev%20Guidelines%7c4D1FD662-4F55-4FBA-B21C-F5C26043C110%2f%29)) for accessibility |  |
| Is your UI keyboard accessible? |  |
| Is your UI screen reader accessible? |  |
| Does your UI meet the color contrast requirements? |  |
| Does your UI work well when zoomed in at most 200%? |  |
| Have you run the Accessibility Checker tool on your UI to ensure your HTML complies with accessibility standards? |  |

## Master Page XML Structure without widgets (Code)

<MasterPage>

<Id>**999**</Id>

<Name>**Master Page Template**</ Name>

<LocalizedName>**Master Page Template**</LocalizedName>

<RequiredLevel>none</RequiredLevel>

<Portal>

<Id>**1000**</Id>

<Name>**The outstanding CRM Portal**</Name>

</Portal>

<Header>

<MasterPageHead >

<sections>

<section Type=**"Metadata"** position=**"1"**/>

<section Type=**"File References"** position=**"2"**/>

<section Type=**"Non-File References"** position=**"3"**/>

</sections>

</MasterPageHead>

</Header>

<Body>

<MasterPageTop />

<MasterPageBottom/>

</Body>

</MasterPage>

## Master Page XML Structure with widgets (Code)

<MasterPage>

<Id>**999**</Id>

<Name>**Master Page Template**</ Name>

<LocalizedName>**Master Page Template**</LocalizedName>

<RequiredLevel>none</RequiredLevel>

<Portal>

<Id>**1000**</Id>

<Name>**The outstanding CRM Portal**</Name>

</Portal>

<Header>

<MasterPageHead >

<sections>

<section Type=**"Metadata"** position=**"1"**/>

<section Type=**"File References"** position=**"2"**/>

<section Type=**"Non-File References"** position=**"3"**/>

</sections>

</MasterPageHead>

</Header>

<Body>

<MasterPageTop >

<sections>

<section position=**"1"**>Placeholder for a Widget</section>

<section position=**"2"**> Placeholder for another Widget </section>

</sections>

</MasterPageTop >

<MasterPageBottom/>

</Body>

</MasterPage>

## Page XML Structure without widgets, not merged with MasterPage (Code)

<Page>

<Id></Id>

<Name></Name>

<LocalizedName></LocalizedName>

<MasterPage></MasterPage>

<Type></Type>

<Entities></Entities>

<RelativeURL></RelativeURL>

<Head>

<MainHeader></MainHeader>

</Head>

<Body>

<PageBody></PageBody>

</Body>

</Page>

## Page XML Structure merged with MasterPage (Code)

<Page>

<Id></Id>

<Name></Name>

<LocalizedName></LocalizedName>

<MasterPage>

<ID></ID>

<Name></ Name >

</MasterPage>

<Portal>

<Id></Id>

<Name> </Name>

</Portal>

<Type></Type>

<Entities></Entities>

<RelativeURL></RelativeURL>

<Head>

<MasterPageHeader></ MasterPageHeader >

<MainHeader></MainHeader>

</Head>

<Body>

<MasterPageTop></ MasterPageTop >

<PageBody></PageBody>

<MasterPageBottom></ MasterPageBottom >

</Body>

</Page>

## Elements XML Structure (Code)Elements XML Structure (Code)

### Metadata XML Structure

<section Type=**"Metadata"** position=**"1"**>

<Category>**HTML**</Category>

<Name>**Metadata**</Name>

<LocalizedName>**Metadata**</LocalizedName>

<RequiredLevel>**none**</RequiredLevel>

<Contents>**Some metadata of any kind**</Contents>

</section>

### File References XML Structure

<section Type=**"File References"** position=**"2"**>

<Category>**File References**</Category>

<Name>**File References**</Name>

<LocalizedName>**File References**</LocalizedName>

<RequiredLevel>**none**</RequiredLevel>

<sections>

<section position=**"1"**>

<Type>**CSS**</Type>

<Contents>**masterStylesheet.css**</Contents>

</section>

<section position=**"2"**>

<Type>**SCRIPT**</Type>

<Contents>**masterScripts.js**</Contents>

</section>

</sections>

</section>

### Non-File References XML Structure

<section Type=**"Non-File References"** position=**"3"**>

<Category>**Non-File References**</Category>

<Name>**Non-File References**</Name>

<LocalizedName>**Non-File References**</LocalizedName>

<RequiredLevel>**none**</RequiredLevel>

<sections>

<section position=**"1"**>

<Contents>**/some/reference/3**</Contents>

</section>

</sections>

</section>

### Body XML Structure

<PageBody>

<sections>

<section type="widget" position="1">

Place the Widget code here

</section>

</sections>

</PageBody>

### Entities XML Structure

<Entities>

<Entity>

Id>**1**</Id>

<Name>**Entity1**</Name>

<EntityType>**Case**</EntityType>

<EnableForExternalChannel>**1**</EnableForExternalChannel>

</Entity>

<Entity>

<Id>**2**</Id>

<Name>**Entity1**</Name>

<EntityType>**Case**</EntityType>

<EnableForExternalChannel>**1**</EnableForExternalChannel>

</Entity>

</Entities>

### RelativeURL XML Structure

<RelativeURL>

<Source>**/Unique/Page/URL**</Source>

</RelativeURL>

### Page Type XML Structure

<Type>

<TypeID>**123**</TypeID>

<TypeName>**Custom**</TypeName>

</Type>

### 

## Page HTML Structure (Code), with Master Page

ToDO: update me

<html lang=**"en-US"**>

<head>

<!—Metadata, File References, Non-File References from Master Page -->

<!—Metadata, File References, Non-File References from Page -->

</head>

<body>

<div class=**"masterPage"**>

<div class=**"masterPageTop"**>

<div class=**"masterPageHeaderImageWidgetDiv"**>

<section id=**"masterPageHeaderImageWidget"**>

<div id=**"masterPageHeader"** class=**"masterPageHeader"**>

<label for=**"masterPageHeader"**>**Here is the Master Page Header text**</label>

</div>

</section>

</div>

<div class=**"imageWidgetDiv"**>

<section id=**"imageWidget"**>

<div id=**"masterPageBanner"** class=**"masterPageBanner"**>

<label for=**"masterPageBanner"**>**Banner**</label>

</div>

</section>

</div>

</div>

<div class=**"pageLayout"**>

<div class=**"widgetNameDiv"**>**Some Widget**</div>

<div class=**"pageWidgetDiv"**>

<section id=**"pageWidget"** class=**"pageWidget"**>

</section>

</form>

</section>

</div>

<div class=**"someWidgetDiv"**>**Some Widget**</div>

</div>

<div class=**"masterPageFooter"**>

<div class=**"masterPageFooterWidgetDiv"**>

<section id=**"masterPageFooterWidget"**>

Placeholder for the Footer

</section>

</div>

</div>

</div>

</body>

</html>

## Elements HTML Structure (Code)

### Head elements HTML Structure

### Metadata HTML Structure

<meta charset=**"UTF-8"**>

<meta name=**"MasterPageMetadata"** content=**"Some MasterPage metadata of any kind"**>

<meta name=**"PageMetadata"** content=**"Some Page metadata of any kind"**>

### File References HTML Structure

<link rel=**"stylesheet"** type=**"text/css"** href=**"masterStylesheet.css"**>

<script src=**"masterScripts.js"**></script>

<link rel=**"stylesheet"** type=**"text/css"** href=**"pageStylesheet.css"**>

<script src=**"pageScripts.js"**></script>

<script src=**"someOtherPageScripts.js"**></script>

### Non-File References HTML Structure

<meta name=**"someReference"** content=**"/some/reference/3"**>

<meta name=**"someOtherReference"** content=**"/some/reference/4"**>

### Body HTML Structure

### Entities HTML Structure

The Entity element of the XML is not directly mapped to the HTML.

### Relative URL HTML Structure

The RelativeURL element of the XML is not directly mapped to the HTML. However, it should depict the Page’s URL.

### Page Type HTML Structure

The Type element of the XML is not directly mapped to the HTML.

# Document History

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Document State** | **Revision Details** | **Author** |
|  |  |  |  |
|  |  |  |  |