

CRM Migration Assistant

Speeding your migration to CRM 2011

Mitch Milam

CRM Accelerators

mitch@crmaccelerators.net



Revision 1.1.0

Table of Contents

CRM Migration Assistant.....	1
Requirements	1
Required Components:	1
Supported Operating Systems:	1
Trial Version	2
Supported Entities	2
Import JavaScript	2
Upload Capabilities	2
Converting to a Fully-Licensed Version	2
Overview	3
JavaScript Conversion	4
Converting directly from CRM 4.0	4
Converting from a exported CRM customization file.....	4
From a folder	4
From a single file	4
The Process	5
User Interface Functionality	6
User interface tips and tricks.....	8
Effective Use of the CRM Migration Assistant	9
Phase I: Pre-Upgrade	9
Phase II: Post-Upgrade.....	9
Uploading Converted JavaScript to CRM 2011	10
Submitting Issues or Feedback.....	13
Conversion Alerts.....	14
Conversion Alert List	14
Conversion Issues and Notes	16
crmForm.SubmitCrmForm	18
crmForm.ObjectTypeCode	18
JavaScript attachEvent and detachEvent	18
window.opener.parent.document.crmForm	18
Minified JavaScript	18
.src Property	19
Using Reference Variables	20
Tabs	21
The JavaScript event property	22
JavaScript Function Comparison	24
Technical notes and troubleshooting tips.....	28
Error: Timeouts while downloading customizations from CRM 4.0	28

Error: Unable to send feedback28

CRM Migration Assistant

Requirements

Version 1.1 of the CRM Migration Assistant communicates directly with CRM 2011 which introduces a new set of requirements:

Required Components:

- Windows Identity Foundation

Note: This may be downloaded from the following location:
<http://tinyurl.com/WindowsIdentity>

- Microsoft .NET Framework 4.0

Supported Operating Systems:

The Windows Identity Framework requires the one of following operating systems:

- Windows Vista Service Pack 2
- Windows 7
- Windows Server 2008 Service Pack 2
- Windows Server 2008 R2

Note: As you can see from the above list, Windows XP is not supported by the Windows Identity Framework so the CRM Migration Assistant will not properly function within that environment.

Trial Version

The trial version has a reduced feature set:

Supported Entities

Only the Account and Contact Entities will be converted. Additional entities will be ignored.

Import JavaScript

The ability to convert JavaScript files found in a folder is disabled.

Upload Capabilities

You cannot upload converted JavaScript into CRM 2011.

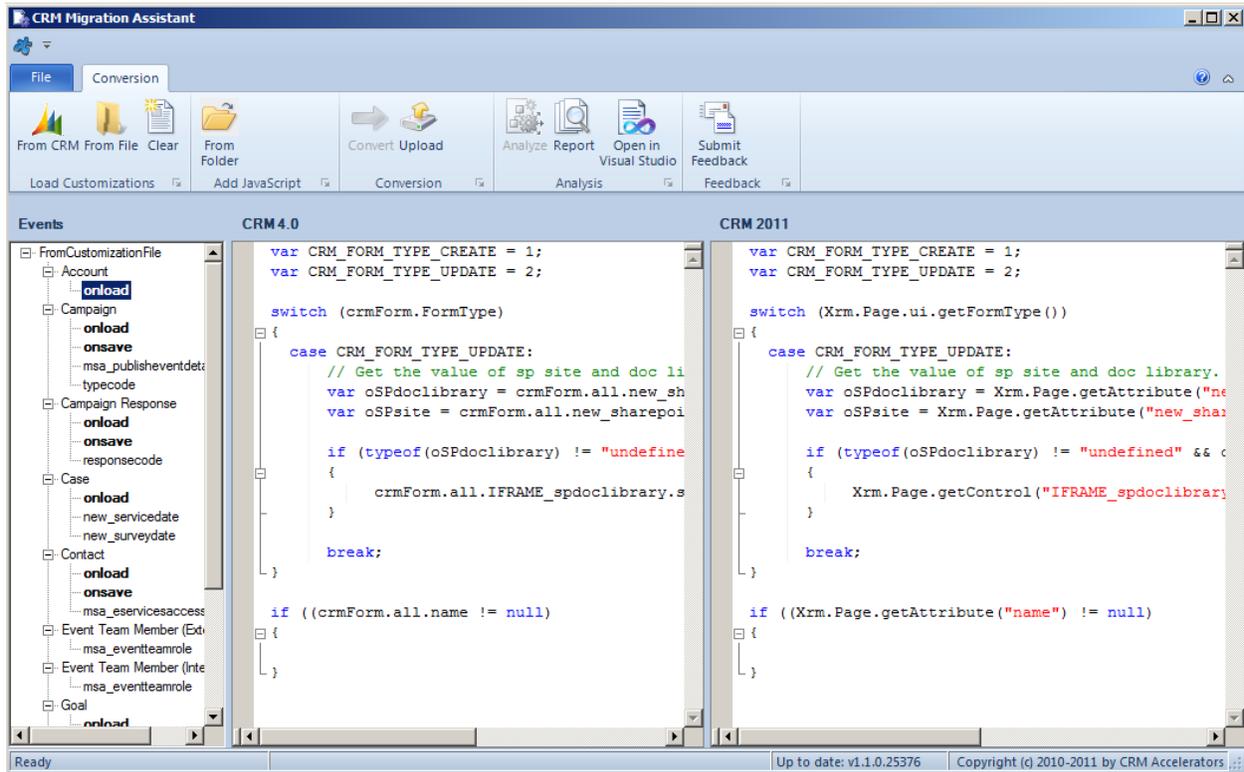
Converting to a Fully-Licensed Version

The conversion from the trial version into the full version is accomplished by installing a license file in the Licenses folder.

Overview

The CRM Migration Assistant will extract the JavaScript found within each CRM form and convert it from the CRM 4.0 object model to the CRM 2011 object model.

The user interface, shown in the following figure, will allow you to compare the CRM 4.0 version of each event with the converted, CRM 2011 version:



In addition to converting standard CRM 4.0 JavaScript, the CRM Migration Assistant also converts commonly used, but possibly unsupported JavaScript into the new CRM 2011 object model using supported functionality. These include:

- Hiding and showing fields, sections, tabs, and navigation items
- Using internal and/or undocumented CRM methods
- Using deprecated CRM methods or properties

Finally, a report is produced which outlines various points of interest regarding your conversion as well as lists any issues found during the conversion process which may not work correctly in CRM 2011.

JavaScript Conversion

There are four methods available for converting JavaScript from CRM 4.0 to CRM 2011

Converting directly from CRM 4.0

You can connect directly to CRM 4.0 using any of the following connection methods:

- On Premise
- CRM Online
- IFD

Customizations will be exported directly from CRM and the JavaScript contained within converted to CRM 2011.

Converting from a exported CRM customization file

You can convert a customization file that has been exported from CRM 4.0. The following file formats are supported:

- .ZIP file (this is assuming that the ZIP file is a standard CRM customization export file)
- .XML file

From a folder

This feature allows you to convert JavaScript files found within that folder.

Note: *Only files with a .js extension will be converted. Also, sub-folders will not be processed.*

From a single file

You may paste JavaScript into the CRM 4.0 edit box and click the Convert button to convert the code.

Note: *Since the additional information required to create a full-conversion project is unavailable, the additional files and functionality (Open in Visual Studio, Report, etc.) will not be available.*

The Process

Load your customizations from either the customization file or by connecting to CRM 4.0 directly.

Note: *The conversion process occurs immediately.*

A sub-folder will be created in the CRM Migration Assistant installation folder to contain the JavaScript and other related files produced by the conversion process. The name of the folder is dependent on the source of the CRM 4.0 customizations.

Directly from CRM 4.0: The folder name will begin with the organization name

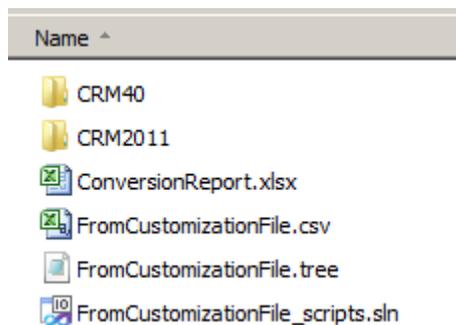
From a customization file: The folder name will begin with FromCustomizationFile

From a folder: The folder name will begin with FromFolder

Both folder names will end with the conversion date and time. Here are examples of both:

- Dynamics Tutorials-2010-10-13-1534
- FromCustomizationFile-2010-11-03-2348
- FromFolder-Samples-test-2011-06-12-1923

Within one of the above folders you will find a structure that looks like this:



CRM40 contains the extracted JavaScript in its original form

CRM2011 contains the converted JavaScript

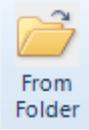
ConversionReport.xlsx is the Excel worksheet containing information about the conversion process.

FromCustomizationsFile.csv contains a list of JavaScript events associated with the Entities and Attributes within the import file.

FromCustomizationFile.tree is an internal file used by the CRM Migration Assistant to display data within the user interface.

FromCustomizationFile_Scripts.sln is a Visual Studio 2010 solution file that allows you to edit the extract CRM 4.0 and converted CRM 2011 JavaScript files within Visual Studio.

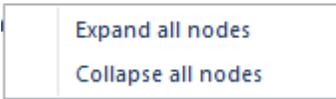
User Interface Functionality

Button	Functionality
	Connects to a CRM 4.0 organization and retrieves the JavaScript directly from CRM customizations.
	Extracts the JavaScript from an exported CRM 4.0 customizations file.
	Clears the Migration Assistant workspace.
	Converts the JavaScript files found in the selected folder.
	Converts the JavaScript found in the CRM 4.0 editor into CRM 2011 syntax. <i>Note: Use this option if you need convert a JavaScript fragment instead of the entire CRM organization or customization file.</i>
	Uploads your converted JavaScript into CRM 2011. You will be prompted for connection information and organization.

Button	Functionality																														
	<p>Opens the conversion report in Microsoft Excel.</p> <p>The report contains the following information:</p> <ul style="list-style-type: none"> • List of all attributes found in use during the conversion • List of all user-defined functions • List of all iFrames • List of instances where ActiveX controls are used • Any conversion issues • The location where the object was found <ul style="list-style-type: none"> ○ Entity ○ Attribute ○ Event ○ Tab ○ Section <p>The information is displayed much like this:</p> <table border="1" data-bbox="475 835 1429 974"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> </tr> </thead> <tbody> <tr> <td>Entity Name</td> <td>Event Name</td> <td>Attribute Name</td> <td>Object Name</td> <td>Type Name</td> </tr> <tr> <td>account</td> <td>onload</td> <td></td> <td>new_sharepointdocumentlibraryurl</td> <td>Attribute</td> </tr> <tr> <td>account</td> <td>onload</td> <td></td> <td>new_sharepointsiteurl</td> <td>Attribute</td> </tr> <tr> <td>account</td> <td>onload</td> <td></td> <td>IFRAME_spdoclibrary</td> <td>Attribute</td> </tr> <tr> <td>account</td> <td>onload</td> <td></td> <td>name</td> <td>Attribute</td> </tr> </tbody> </table>	A	B	C	D	E	Entity Name	Event Name	Attribute Name	Object Name	Type Name	account	onload		new_sharepointdocumentlibraryurl	Attribute	account	onload		new_sharepointsiteurl	Attribute	account	onload		IFRAME_spdoclibrary	Attribute	account	onload		name	Attribute
A	B	C	D	E																											
Entity Name	Event Name	Attribute Name	Object Name	Type Name																											
account	onload		new_sharepointdocumentlibraryurl	Attribute																											
account	onload		new_sharepointsiteurl	Attribute																											
account	onload		IFRAME_spdoclibrary	Attribute																											
account	onload		name	Attribute																											
	<p>Opens the generated solution within Visual Studio 2010</p>																														
	<p>Allows you to submit feedback on the CRM Migration Assistant or report an improper conversion.</p>																														
<p>Automatic Updates</p>	<p>An automated update mechanism is been built into the system. Shortly after the application launches, it will check the update server for updates and if found, they will be installed and CRM Migration Assistant will restart.</p> <p><i>Note: The current version is always listed on the toolbar:</i></p> <table border="1" data-bbox="407 1587 1187 1635"> <tr> <td>Up to date: v1.1.0.21161</td> <td>Copyright (c) 2010-2011 by CRM Accelerators</td> </tr> </table> <p><i>Clicking on the version number will start the update process manually.</i></p>	Up to date: v1.1.0.21161	Copyright (c) 2010-2011 by CRM Accelerators																												
Up to date: v1.1.0.21161	Copyright (c) 2010-2011 by CRM Accelerators																														

User interface tips and tricks

A few features have been built into the Migration Assistant that may not be quite so obvious:

Action	Result
Right-Click Events Treeview	Displays a pop-up menu that allows you to expand or collapse the treeview to show only the first-level of nodes (which is usually the Entity or file list: 
Hold down the Ctrl key while using scrollbar or mouse wheel on the editor windows	Causes both editor windows to synchronize their locations with the editor that currently has focus.

Effective Use of the CRM Migration Assistant

The migration process is generally broken down into two phases:

Phase I: Pre-Upgrade

In this phase, the CRM Migration Assistant can be used to gauge the amount of effort that will be required to fully upgrade the JavaScript found in your CRM 4.0 system.

Note: *It is very important that you make a backup of your customizations pre-upgrade. Just perform an Export All Customizations operation to make sure you have them on-disk.*

Once the system has been upgraded, the Migration Assistant can no longer connect to CRM directly to extract the customizations.

It typically works like this:

- 1) You run the Migration Assistant and convert your JavaScript.
- 2) You review the conversion report as well as the converted JavaScript to see how much effort will be required to upgrade your system.
- 3) At this point, you can actually begin the work of finalizing any conversion issues noted by the conversion process.

Phase II: Post-Upgrade

This phase occurs after your CRM 4.0 organization has been upgraded and you need to upload the JavaScript that was converted using the Migration Assistant or by hand.

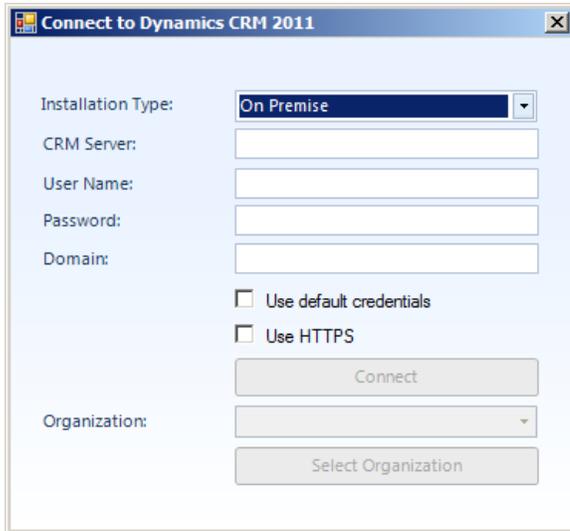
The following steps need to be performed:

- 1) From within the Migration Assistant, load your conversion project .
- 2) Click the Upload button to upload your converted JavaScript.
- 3) After the upload has completed, open CRM 2011.
- 4) Click Settings, Customization, Customize the System.
- 5) Click the Publish All Customizations button.
- 6) After the publish process has completed, review each of your Entities to make sure the migrated JavaScript is functioning as designed.

Uploading Converted JavaScript to CRM 2011

To upload your converted files to CRM 2011, perform the following steps:

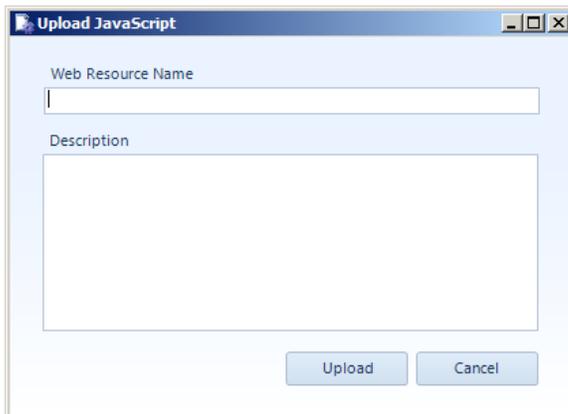
- 1) Click the Upload button.
- 2) Enter the CRM 2011 connection information:



The screenshot shows a dialog box titled "Connect to Dynamics CRM 2011". It contains the following fields and controls:

- Installation Type: A dropdown menu with "On Premise" selected.
- CRM Server: A text input field.
- User Name: A text input field.
- Password: A text input field.
- Domain: A text input field.
- Use default credentials: A checkbox.
- Use HTTPS: A checkbox.
- Connect: A button.
- Organization: A dropdown menu.
- Select Organization: A button.

- 3) Select the organization into which you wish to upload your converted files, then click the Select Organization button.
- 4) If you are uploading a single file, you will be prompted for the Web Resource Name and Description:



The screenshot shows a dialog box titled "Upload JavaScript". It contains the following fields and controls:

- Web Resource Name: A text input field.
- Description: A text area.
- Upload: A button.
- Cancel: A button.

Note: *Batch uploading of files from either CRM or from a folder will use the file name for both the web resource name and description.*

5) The following processes will occur:

- a. Any JavaScript web resources found with the CRM 2011 organization will be exported and placed into a folder within your Migration project folder system and will use the following naming convention:

CRM2011-WebResources-Backup-2011-06-09-2023

- b. A work folder will be created to hold the JavaScript files that will be uploaded to CRM and will use the following naming convention:

CRM2011-Upload-2011-06-09-2039

- c. The individual event files created during the Migration will be combined into a single file per CRM entity and will use the following naming convention:

account_migrated_main.js

As you can see, it begins with the name of the entity, followed by “_migrated_main.js” which will also be the name of the JavaScript Web Resources within CRM 2011.

- d. The files from the Upload working folder will be uploaded as JavaScript Web Resources.

Note: If a web resource of the same name already exists, its contents will be updated.

- e. The Main Form for each entity found within the Upload working folder will next be modified to use the newly uploaded and converted JavaScript.

Note 1: *There could be a situation where you have an entity in your migration folder that cannot be found with the CRM Online organization. In this case, a warning message will be displayed to inform you of the entities that do not exist:*

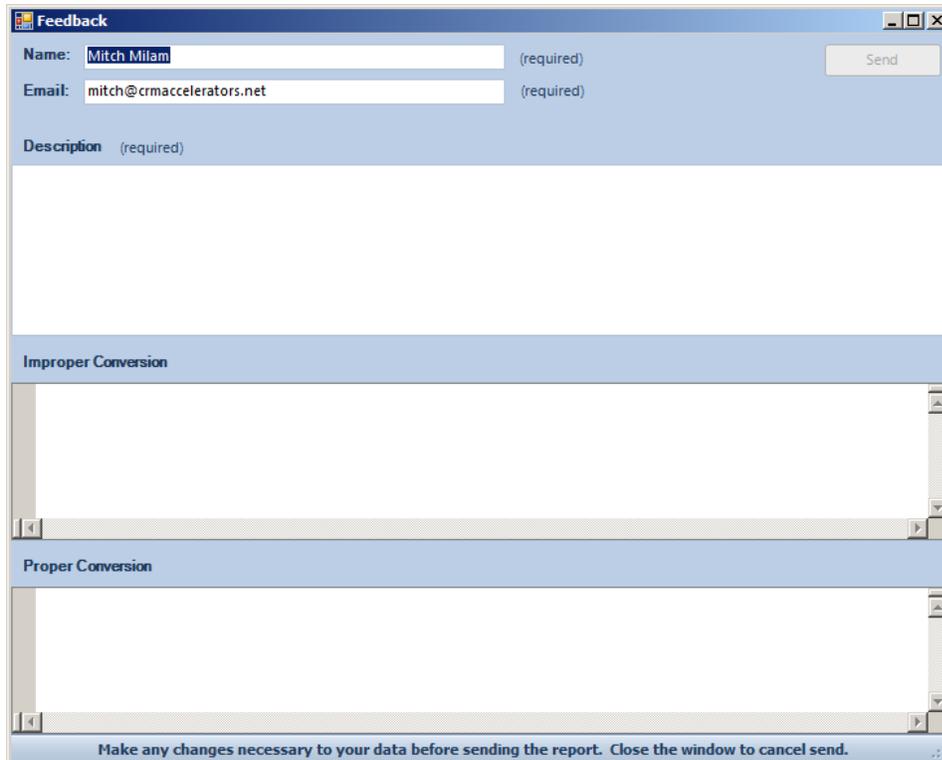


Note 2: *The Web Resources for those entities will still be created and populated with the migrated JavaScript. The note is merely to inform you that since the entity does not exist, the internal plumbing necessary to connect the web resource to the form will not be updated.*

- f. At this point, and assuming no error messages are produced, the migration to CRM 2011 is complete.
- g. Open CRM 2011 and review both the newly created JavaScript Web Resources and data entry forms to make sure that the migration was performed correctly.

Submitting Issues or Feedback

If you run into an issue with the conversion process you can submit feedback which will allow us to correct the conversion issue using the Submit Feedback button which will display the feedback form:



The screenshot shows a window titled "Feedback" with a blue header. It contains the following fields and elements:

- Name:** A text input field containing "Mitch Milam" with "(required)" to its right.
- Email:** A text input field containing "mitch@crmaccelerators.net" with "(required)" to its right.
- Send:** A button located to the right of the email field.
- Description:** A label "(required)" above a large, empty text area.
- Improper Conversion:** A section header above a large, empty text area with a scrollbar.
- Proper Conversion:** A section header above another large, empty text area with a scrollbar.
- Footer:** A blue bar at the bottom containing the text: "Make any changes necessary to your data before sending the report. Close the window to cancel send."

If you have purchased the product, your name and email address will be automatically populated. If you are working with the trial version, you'll need to supply this information.

Note: *This information is required because it is the only way we can contact you regarding your issue.*

Description allows you to specify the issue you found.

Improper Conversion should list the JavaScript as it was converted by the CRM Migration Assistant.

Proper Conversion is how you feel the JavaScript should have been converted. If you're not sure, then just leave this field blank.

Conversion Alerts

During the conversion process, we may encounter JavaScript that will need to be modified by hand after the conversion has completed. This is usually due to the use of unsupported code or techniques which will present problems within CRM 2011.

You can identify a conversion alert by searching for the following comment in your JavaScript:

```
/* CONVERSION ALERT */
```

This comment is placed immediately preceding the offending command. The issue, and its location, is also written to the conversion report worksheet.

Conversion Alert List

Here is a list of conversion alert conditions:

Alert Name	Cause
[command] is not supported in CRM 2011	A JavaScript function which was valid in CRM 4.0 will not work or will have issues in CRM 2011. The current offending command list is: <ul style="list-style-type: none">• attachEvent• detachEvent
[property] is an unsupported JavaScript property	A JavaScript property which was valid in CRM 4.0 is invalid in CRM 2011. The current offending property list is: <ul style="list-style-type: none">• elements
[function] is an unsupported internal function	An internal CRM function which was valid in CRM 4.0 is invalid in CRM 2011. The current offending property list is: <ul style="list-style-type: none">• crmForm.SubmitCrmForm• LookupControlItem
[property] is a deprecated property	A property which was valid in CRM 4.0 is not available in CRM 2011. The current offending property list is: <ul style="list-style-type: none">• crmForm.ObjectTypeCode
Explicit use of partial field name Partial field name	In CRM 4.0 it was possible to reference the field label and field input control by referencing the <code>_c</code> and <code>_d</code> fields, i.e.: <code>crmForm.all.name_c.style.visibility = "none";</code> While these fields are still valid, it is unnecessary to use them in CRM 2011 due to the enhanced object model.

Alert Name	Cause
Toolbar control manipulation	Many developers manipulated the CRM toolbars and menus to hide specific menus and buttons to prevent the users from performing unwanted functions. This code is no longer valid due to the change from a toolbar to a ribbon control in the user interface.

Conversion Issues and Notes

Showing and Hiding Elements

There are two general methods to set the visibility of an HTML DOM object. Both are converted to the native CRM 2011 methodology:

CRM 4.0	<code>{Element}.style.display {Element}.style.visibility</code>
CRM 2011	<code>Xrm.Page.getControl("new_field").setVisible();</code>

Hide a Field

CRM 4.0	<code>crmForm.all.new_field_c.style.display = 'none'; crmForm.all.new_field_d.style.display = 'none';</code>
CRM 2011	<code>Xrm.Page.getControl("new_field").setVisible(false);</code>

Show a Field

CRM 4.0	<code>crmForm.all.new_field_c.style.display = ''; crmForm.all.new_d.style.display = '';</code>
CRM 2011	<code>Xrm.Page.getControl("new_field").setVisible(true);</code>

Hide a Section

CRM 4.0	<code>oField.parentElement.parentElement.parentElement.style.display = "none";</code>
CRM 2011	<code>Xrm.Page.ui.tabs.get("tab_5").sections.get("section1").setVisible(false);</code>

Show a Section

CRM 4.0	<code>oField.parentElement.parentElement.parentElement.style.display = "";</code>
CRM 2011	<code>Xrm.Page.ui.tabs.get("tab_5").sections.get("section1").setVisible(true);</code>

Hide a Tab

CRM 4.0	<code>tab1Tab.style.display = "none";</code>
---------	--

CRM 2011	<code>Xrm.Page.ui.tabs.get("tab_5").setVisible(false);</code>
----------	---

Show a Tab

CRM 4.0	<code>tab1Tab.style.display = "";</code>
CRM 2011	<code>Xrm.Page.ui.tabs.get("tab_5").setVisible(true);</code>

Hide a Form Navigation Element

CRM 4.0	<code>var element = document.getElementById("navSubAct"); element.style.display = 'none';</code>
CRM 2011	<code>Xrm.Page.ui.navigation.items.get("navSubAct").setVisible(false);</code>

Show a Form Navigation Element

CRM 4.0	<code>var element = document.getElementById("navSubAct"); element.style.display = '';</code>
CRM 2011	<code>Xrm.Page.ui.navigation.items.get("navSubAct").setVisible(true);</code>

Dynamically Setting Field Requirement Levels

Technically, in CRM 4.0 this is an unsupported practice but that doesn't keep developers from using it.

CRM 4.0	<code>crmForm.all.new_field.req = 1; crmForm.SetFieldReqLevel("new_field", 1);</code>
CRM 2011	<code>Xrm.Page.getAttribute("new_field").setRequiredLevel("recommended"); Xrm.Page.getAttribute("new_field").setRequiredLevel("recommended");</code>

crmForm.SubmitCrmForm

The following internal crmForm method can be found in some solutions:

```
crmForm.SubmitCrmForm(crmForm.ObjectTypeCode, true, true, false);
```

This method is not supported in CRM 2011 and if found during the conversion process, a conversion alert will be inserted and it will be noted in the conversion report.

crmForm.ObjectTypeCode

There is no conversion provided for this crmForm property since it this property unavailable in CRM 2011 so in this particular case:

```
crmForm.ObjectTypeCode
```

Remains exactly the same. crmForm is **not** converted to Xrm.Page because Xrm.Page does not support this property.

JavaScript attachEvent and detachEvent

The JavaScript functions attachEvent and detachEvent are not supported in CRM 2011.

window.opener.parent.document.crmForm

This piece of JavaScript allows you to get a handle to the parent record of a record if it was opened from the context of that parent. For example, if you perform the following steps:

- 1) Open an Account
- 2) Click on Contacts
- 3) Open a Contact from the Associated Contacts grid

If you execute the above JavaScript, it will return a handle to the Account form. Once you have that handle, you can access the form properties and attributes as you normally would.

Usage	CRM 4.0	CRM 2011
Get	window.opener.parent.document.crmForm; window.opener.document.crmForm; window.opener.parent.crmForm;	window.top.opener.parent.Xrm.Page

Example:

```
window.top.opener.parent.Xrm.Page.getAttribute("name").getValue();
```

Minified JavaScript

Some developers pass their JavaScript through a minification process to reduce the size of the file that will be downloaded to the client machine. There are instances where the JavaScript produced will fail to convert because of the assumptions taken during the minification process.

.src Property

Some objects within the JavaScript document object model (DOM) have a .src property which allows you to specify the source location for various DOM elements. While there are legitimate uses for .src, such as the loading of external JavaScript files and the contents of iFrames, other usages of the border on the “very unsupported” side of CRM JavaScript development.

This creates a problem during the conversion process because the Migration Assistant has no method to determine which usage of .src has been encountered. That being the case, it was decided that all instances of .src will be converted to .getSrc and .setSrc, which are used to get and set the contents of an iFrame, respectively.

This means that any other usage of .src will be improperly converted and your converted JavaScript will not function as expected.

Here are examples of code that will be improperly converted:

CRM 4.0:

```
crmForm.all.new_learningplanid.src = '/_imgs/btn_off_lookup.gif';  
  
var script = document.createElement('script');  
script.language = 'javascript';  
script.src = '/ISV/scripts/date.js';
```

CRM 2011, post conversion:

```
Xrm.Page.getControl("new_learningplanid").setSrc('/_imgs/btn_off_lookup.gif');  
  
var script = document.createElement('script');  
script.language = 'javascript';  
script.setSrc('/ISV/scripts/date.js');
```

Using Reference Variables

It is a common practice with CRM 4.0 to assign a reference to a field to a variable then use that variable for further operations, like the following:

```
var oName = crmForm.all.name;
oName.Disabled = false;
oName.DataValue = "new name";
oName.Disabled = true;
oName.ForceSubmit = true;
```

The CRM Migration Assistant will properly convert this type of code, but a secondary issue is created because CRM 2011 separates the methods between attributes and form controls. With CRM 4.0, all methods were contained within the same `crmForm` object.

The two CRM 2011 objects are: **Xrm.Page.ui.controls** and **Xrm.Page.data.entity.attributes**.

The example above requires that the conversion process reference the control associated with the attribute for some of the methods to be valid. This is accomplished by referencing the attribute's first control using the command: **controls.get(0)**.

Here is an example of the converted code:

```
var oName = Xrm.Page.getAttribute("name");
oName.controls.get(0).setDisabled(false);
oName.setValue("new name");
oName.controls.get(0).setDisabled(true);
oName.setSubmitMode("always");
```

Since CRM 4.0 only allowed one instance of any specific attribute on a form at one time, this code will function as expected. However, if you add multiple instances of a field to a CRM 2011 form, this command may return improper or incorrect information so you will need to refactor your code in such a way as to prevent this issue in the future.

Tabs

In CRM 4.0 tabs were named tab0Tab – tab7Tabs with a hard-limit of 8 tabs. During the CRM 4.0 organization import, CRM will change the reference ID of each tab on a form. Instead of tab1Tab, you will see a GUID, which was the internal ID for the tab within the form.

During the conversion process, the Migration Assistant will create a list of tab IDs and perform a replacement for any occurrence of tabxTab with the ID associated with that tab.

This will cause your JavaScript to automatically match up the new tab naming system.

CRM 4.0:

```
crmForm.all.tab1Tab.style.display = "none";  
crmForm.all.tab2Tab.style.display = "none";  
crmForm.all.tab3Tab.style.display = "none";  
crmForm.all.tab4Tab.style.display = "none";
```

CRM 2011, post conversion:

```
Xrm.Page.ui.tabs.get("{9748ec98-3746-40cc-83bf-d15c7363166f}").setVisible(false);  
Xrm.Page.ui.tabs.get("{cafbf41a-42bf-4665-8562-1e37017eba69}").setVisible(false);  
Xrm.Page.ui.tabs.get("{0dd7233e-247b-4eef-a71b-0185da6d16ad}").setVisible(false);  
Xrm.Page.ui.tabs.get("{A6C643C6-E98F-4738-812F-0B7F7FFC80F7}").setVisible(false);
```

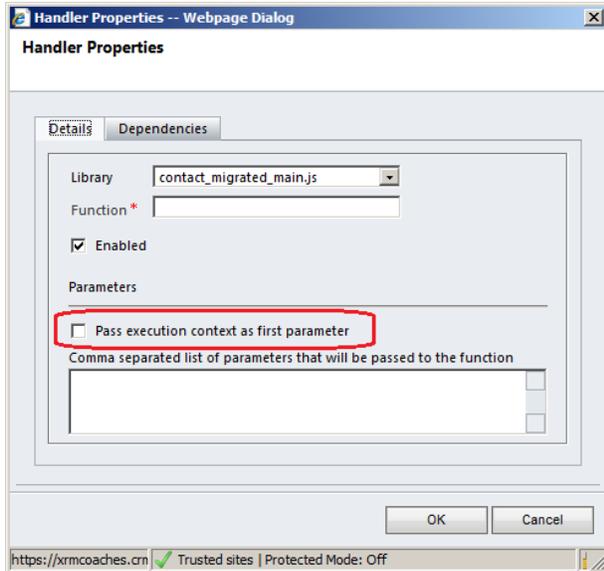
The JavaScript event property

In CRM 4.0, we use the JavaScript event property for two main purposes:

- 1) To write generic functions that will operate on data passed to it from a calling event.
- 2) To cancel a save operation should some criteria not match a pre-defined set of circumstances.

CRM 2011 has a very similar approach called the Execution Context, which requires a small amount of work on the part of the developer but which functions exactly the same.

Step one is to instruct CRM 2011 to pass the execution context as the first parameter of the event:



Step 2 is to add this parameter to the function you have declared to handle the event. This usually looks something like this:

```
function contact_onsave(executionObj)
{
}
```

Within the function, you access the execution object (executionObj) like any other Javascript variable.

Here are the comparisons between how CRM 4.0 and CRM 2011 handle the same event code:

event.returnValue

Usage	CRM 4.0	CRM 2011
Get	if (event.returnValue != false)	if (executionObj.getEventArgs().isDefaultPrevented() != false)
Set	event.returnValue = false;	executionObj.getEventArgs().preventDefault()

event.mode

Usage	CRM 4.0	CRM 2011
Get	var oField = event.Mode;	var oField = executionObj.getEventArgs().getSaveMode();

event.srcElement

Usage	CRM 4.0	CRM 2011
Get	var oField = event.srcElement;	var oField = executionObj.getEventSource();

During the conversion process, the parameter executionObj will be used to replace event.

During the upload process, if executionObj is detected within the converted JavaScript, then two things will occur:

- 1) When the function is created, the executionObj parameter is automatically added to the function declaration.
- 2) When the Entity Form is updated, any event function declaration containing executionObj will cause the Migration Assistant to automatically check the checkbox: ***Pass execution context as first parameter.***

No manual modification to the event declaration or the function associated with the event is required.

JavaScript Function Comparison

The following table lists each CRM 4.0 JavaScript function along with the equivalent CRM 2011 function, if available:

CRM 4.0	CRM 2011
GenerateAuthenticationHeader()	Xrm.Page.context.getAuthenticationHeader()
ORG_LANGUAGE_CODE	Xrm.Page.context.getOrgLcid()
ORG_UNIQUE_NAME	Xrm.Page.context.getOrgUniqueName()
-	Xrm.Page.context.getQueryStringParameters()
SERVER_URL	Xrm.Page.context.getServerUrl()
-	Xrm.Page.context.getUserId()
USER_LANGUAGE_CODE	Xrm.Page.context.getUserLcid()
-	Xrm.Page.context.getUserRoles()
IsOnline()	Xrm.Page.context.isOutlookClientOnline()
IsOutlookClient()	Xrm.Page.context.isOutlookClient()
IsOutlookLaptopClient	Xrm.Page.context.isOutlookClient()
IsOutlookWorkstationClient()	Xrm.Page.context.isOutlookClient()
-	Xrm.Page.data.entity.getDataXml()
ObjectTypeName	Xrm.Page.data.entity.getEntityName()
ObjectTypeCode	-
ObjectId	Xrm.Page.data.entity.getId()
IsDirty	Xrm.Page.data.entity.getIsDirty()
SaveAndClose()	Xrm.Page.data.entity.save("SaveAndClose")
-	Xrm.Page.data.entity.save("SaveAndNew")

CRM 4.0	CRM 2011
Save()	Xrm.Page.data.entity.save()
crmForm.all.[field]	Xrm.Page.getAttribute("field")
FireOnChange()	Xrm.Page.getAttribute("field").fireOnChange()
-	Xrm.Page.getAttribute("field").getAttributeType()
DefaultValue	Xrm.Page.getAttribute("field").getDefaultValue()
-	Xrm.Page.getAttribute("field").getFormat()
-	Xrm.Page.getAttribute("field").getInitialValue()
IsDirty	Xrm.Page.getAttribute("field").getIsDirty()
Max	Xrm.Page.getAttribute("field").getMax()
-	Xrm.Page.getAttribute("field").getMaxLength()
Min	Xrm.Page.getAttribute("field").getMin()
-	Xrm.Page.getAttribute("field").getName()
GetSelectedOption	Xrm.Page.getAttribute("field").getSelectedOption()
Options	Xrm.Page.getAttribute("field").getOptions()
-	Xrm.Page.getAttribute("field").getParent()
Precision	Xrm.Page.getAttribute("field").getPrecision()
RequiredLevel	Xrm.Page.getAttribute("field").getRequiredLevel()
GetSelectedOption	Xrm.Page.getAttribute("field").getSelectedOption()
-	Xrm.Page.getAttribute("field").getSubmitMode()
SelectedText	Xrm.Page.getAttribute("field").getText()
-	Xrm.Page.getAttribute("field").getUserPrivilege()
DataValue	Xrm.Page.getAttribute("field").getValue()

CRM 4.0	CRM 2011
<code>crmForm.SetFieldReqLevel("attribute", requirementLevel);</code>	<code>Xrm.Page.getAttribute("field").setRequiredLevel("recommended");</code>
<code>ForceSubmit</code>	<code>Xrm.Page.getAttribute("field").setSubmitMode()</code>
<code>DataValue = "test"</code>	<code>Xrm.Page.getAttribute("field").setValue("test")</code>
-	<code>Xrm.Page.ui.getCurrentControl()</code>
<code>FormType</code>	<code>Xrm.Page.ui.getFormType()</code>
-	<code>Xrm.Page.ui.getViewPortHeight()</code>
-	<code>Xrm.Page.ui.getViewPortWidth()</code>
-	<code>Xrm.Page.ui.refreshRibbon()</code>
<code>AddOption</code>	<code>Xrm.Page.getControl("field").addOption(option, [index])</code>
<code>Disabled</code>	<code>Xrm.Page.getControl("field").getDisabled()</code>
<code>Disabled = true;</code>	<code>Xrm.Page.getControl("field").setDisabled()</code>
<code>DeleteOption</code>	<code>Xrm.Page.getControl("field").removeOption(value)</code>
<code>SetFocus()</code>	<code>Xrm.Page.getControl("field").setFocus()</code>
<code>[IFRAME].src</code>	<code>Xrm.Page.getControl("IFRAME_one").getSrc()</code>
<code>[IFRAME].src = "http://newurl"</code>	<code>Xrm.Page.getControl("IFRAME_one").setSrc("http://newurl")</code>
-	<code>Xrm.Page.getControl("field").addCustomView</code>
-	<code>Xrm.Page.getControl("field").clearOptions</code>
-	<code>Xrm.Page.getControl("field").getAttribute</code>
-	<code>Xrm.Page.getControl("field").getControlType</code>
-	<code>Xrm.Page.getControl("field").getDefaultView</code>
-	<code>Xrm.Page.getControl("field").getLabel</code>

CRM 4.0	CRM 2011
-	Xrm.Page.getControl("field").getName
-	Xrm.Page.getControl("field").getParent
-	Xrm.Page.getControl("field").setVisible
-	Xrm.Page.getControl("field").refresh
-	Xrm.Page.getControl("field").setDefaultView
-	Xrm.Page.getControl("field").setLabel
-	Xrm.Page.getControl("field").setVisible

Technical notes and troubleshooting tips

Some of the settings within the CRM Migration Assistant configuration file may be changed to correct issues you may be having in your environment.

The file is called: **CRMMigrationAssistant.exe.config** and is located in the installation folder, along with the main executable: CRMMigrationAssistant.exe.

You may edit the .config file with Notepad or any other text editor.

Error: Timeouts while downloading customizations from CRM 4.0

Depending on the size of your customizations, you may encounter timeout issues when downloading customizations directly from CRM 4.0.

The default timeout value is 10 minutes, but it may be increased as necessary. Look for the following key within the .config file:

```
<setting name="Timeout" serializeAs="String">  
  <value>10</value>  
</setting>
```

Increase the value of 10 to a larger number.

Error: Unable to send feedback

If you attempt to send feedback using the feedback form and encounter the following error:



Then your network administrator may have blocked the default SMTP port that the CRM Migration Assistant uses to send email back to our feedback service.

Here is the key within the .config file:

```
<setting name="EmailPort" serializeAs="String">  
  <value>587</value>  
</setting>
```

Change that value from 587 to 25, which is the normal SMTP port for email communication.

Note: *Some ISPs block the normal port 25 to prevent users from sending email outside of the ISPs email servers. This is a SPAM prevention mechanism.*