XRM OBJECT MODEL

The Xrm.Page object serves as a namespace object to consolidate three objects on the form:

* [**Xrm.Page.context**](http://msdn.microsoft.com/en-us/library/gg328399.aspx)**:** Xrm.Page.context provides methods to retrieve information specific to an organization, a user, or parameters that were passed to the form in a query string.

[**Xrm.Page.data.**](http://msdn.microsoft.com/en-us/library/gg334720.aspx)Xrm.Page.data provides an entity object that provides methods to manage data within the entity form.

Xrm.Page.data.entity.attributes.get("new\_phonenumber").getValue();

Xrm.Page.getAttribute("new\_phonenumber").getValue();

* [**Xrm.Page.ui**](http://msdn.microsoft.com/en-us/library/gg327828.aspx)**:** Xrm.Page.ui provides collections and methods to manage the user interface of the form.
* Xrm.Page.ui.controls.get("new\_phonenumber ").setVisible(false);
* Xrm.Page.getControl("new\_phonenumber ").setFocus();

Some benefits of using the Xrm.Page model are:

1.Change the appearance or manipulate form objects like data controls (text boxes, checkboxes, etc.), tabs, sections, etc.

2.Identify elements to show/hide.

3.Specify names of objects from which to retrieve/store data values.

4.Provide namespaces to properties so the system allows multiple controls per field or multiple forms per entity.

### Xrm.Page.context

Xrm.Page.context provides methods to retrieve information specific to an organization, a user, or parameters that were passed to the form in a query string. Some of the common methods used are:

* [**getOrgUniqueName**](http://msdn.microsoft.com/en-us/library/d7d0b052-abca-4f81-9b86-0b9dc5e62a66#BKMK_getOrgUniqueName)**:** Returns the unique text value of the organizations name.
  + var OrganizationUniqueName = context.getContext().getOrgUniqueName();
* [**getClientUrl**](http://msdn.microsoft.com/en-us/library/d7d0b052-abca-4f81-9b86-0b9dc5e62a66#BKMK_getClientUrl)**:** Returns the base server URL. When a user is working offline with Microsoft Dynamics CRM for Microsoft Office Outlook, the URL is to the local Microsoft Dynamics CRM Web services.
* [**getUserId**](http://msdn.microsoft.com/en-us/library/d7d0b052-abca-4f81-9b86-0b9dc5e62a66#BKMK_getUserId)**:** Returns the GUID value of the SystemUser.id value for the current user.
  + var userGUID = context.getContext().getUserId();
* [**getUserRoles**](http://msdn.microsoft.com/en-us/library/d7d0b052-abca-4f81-9b86-0b9dc5e62a66#BKMK_getUserRoles)**:** Returns an array of strings representing the GUID values of each of the security roles that the user is associated with.
  + var currentUserRoles = Xrm.Page.context.getUserRoles();

### Xrm.Page.data.entity

The Xrm.Page.data.entity Attribute Methods are frequently used to get and set propertied of various attributes on the form.  Few examples of the methods commonly used are:

* Get the value from a CRM field
  + var varMyValue = Xrm.Page.getAttribute(“CRMFieldSchemaName”).getValue() ;
* Set the value of a CRM field
  + Xrm.Page.getAttribute(“po\_CRMFieldSchemaName”).setValue(‘My New Value’);
* Get the selected value of an optionset
  + Xrm.Page.getAttribute(“CRMFieldSchemaName”).getSelectedOption().text;
* Set the requirement level
  + Xrm.Page.getAttribute(“CRMFieldSchemaName”).setRequiredLevel(“none”);
  + Xrm.Page.getAttribute(“CRMFieldSchemaName”).setRequiredLevel(“required”);
  + Xrm.Page.getAttribute(“CRMFieldSchemaName”).setRequiredLevel(“recommended”);

### Xrm.Page.ui

Xrm.Page.ui contains methods to retrieve information about the user interface as well as collections for several subcomponents of the form. Xrm.Page.ui provides access to the following collections:

**Xrm.Page Namespace**

When writing form scripts, we will interact with objects from the Xrm.Page namespace to perform the following actions:

1. Get/Set attribute values.
2. Show/Hide user interface elements.
3. Reference multiple controls per attribute.
4. Access multiple forms per entity.
5. Manipulate form navigation items.

# Get Form types and modes in Dynamics CRM 2013

***getSaveMode()***, returns a value indicating how the save event was initiated by the user. The following table describes the supported values returned to detect different ways entity records may be saved by the user in CRM

|  |  |
| --- | --- |
| 1 | execObj.getEventArgs().getSaveMode(); |
| **Event Mode** | **Value** |
| Save | 1 |
| Save and Close | 2 |
| Deactivate | 5 |
| Reactivate | 6 |
| Send (Email) | 7 |
| Disqualify (Lead) | 15 |
| Qualify (Lead) | 16 |
| Assign (user or team owned entities) | 47 |
| Save as Completed (Activities) | 58 |
| Save and New | 59 |
| AutoSave | 70 |

**getFormType():** Method to get the form context for the record. The following table lists the form types that correspond to the return value.

|  |  |
| --- | --- |
| 1 | Xrm.Page.ui.getFormType(); |
| **Form Type** | **Value** |
| Undefined | 0 |
| Create | 1 |
| Update | 2 |
| Read Only | 3 |
| Disabled | 4 |
| Quick Create | 5 |
| Bulk Edit | 6 |
| Read Optimized | 11 |

var  formType= Xrm.Page.ui.getFormType();

getFormType() function returns a number

If the return value is 1 –  Form context is Create

If the return value is 2 –  Form context is Update