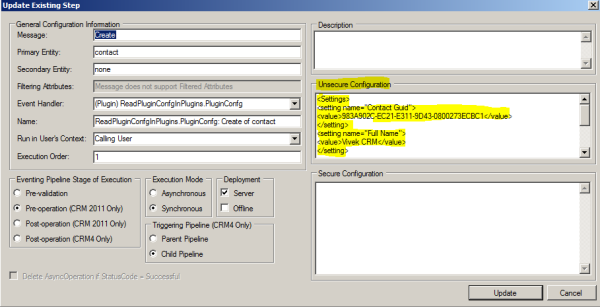
[Pass parameters to Plugin Using Secure or Unsecure Configuration Using Plugin Registration Tool in MS CRM 2011](http://srmscrm.wordpress.com/2013/10/22/pass-parameters-to-plugin-using-secure-or-unsecure-configuration-using-plugin-registration-tool-in-ms-crm-2011/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [October 22, 2013](http://srmscrm.wordpress.com/2013/10/22/pass-parameters-to-plugin-using-secure-or-unsecure-configuration-using-plugin-registration-tool-in-ms-crm-2011/) [Leave a comment](http://srmscrm.wordpress.com/2013/10/22/pass-parameters-to-plugin-using-secure-or-unsecure-configuration-using-plugin-registration-tool-in-ms-crm-2011/#respond)

**Pass parameters to Plugin Using Secure or Unsecure Configuration to Plugin:**

When you start developing plugins, you often need an input parameter or a configuration for the plugin execution which can be easily updated without having to re-compile and/or re-register the plugin.

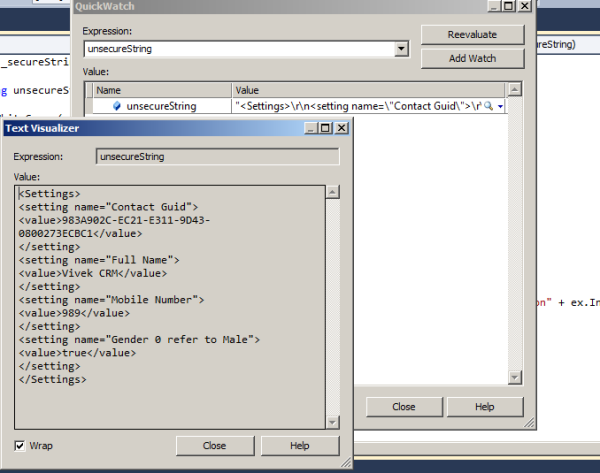
For example , i want to read some of the data which may change every time. Rather than creating the custom entity to read the Configuration kind of thing better to use this method to read the Parameters.

**Use the plugin step “Configuration”**  
When you register a plugin step, there is a field where you can specify some configuration parameters for the plugin execution as below:[](http://srmscrm.files.wordpress.com/2013/10/image.png)

Then in the Constructor of your plugin class you will get the configuration value which you can use later in the Execute method:

[](http://srmscrm.files.wordpress.com/2013/10/image2.png)

In the Quickwatch you can watch the total configuration as follows:

[](http://srmscrm.files.wordpress.com/2013/10/image1.png)

have a glance below for plugin code:

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using Microsoft.Xrm.Sdk;  
using System.Xml;

namespace ReadPluginConfgInPlugins  
{  
public class PluginConfg : IPlugin  
{  
private readonly string \_unsecureString;  
private readonly string \_secureString;

public PluginConfg(string unsecureString, string secureString)  
{  
if (String.IsNullOrWhiteSpace(unsecureString) || String.IsNullOrWhiteSpace(secureString))  
{  
try  
{  
XmlDocument doc = new XmlDocument();  
doc.LoadXml(unsecureString);  
Guid DefaultQueueGuid = PluginConfiguration.GetConfigDataGuid(doc, “Contact Guid”);  
string ContactFullName = PluginConfiguration.GetConfigDataString(doc, “Full Name”);  
int MobileNumber = PluginConfiguration.GetConfigDataInt(doc, “Mobile Number”);  
bool Gender = PluginConfiguration.GetConfigDataBool(doc, “Gender 0 refer to Male”);  
}  
catch (Exception ex)  
{  
throw new Exception(“SoapException” + ex.Message + “########” + ex.StackTrace + “$$$$Inner Exception” + ex.InnerException);  
}  
}   
}

public void Execute(IServiceProvider serviceProvider)  
{  
//Extract the tracing service for use in debugging sandboxed plug-ins.  
ITracingService tracingService =  
(ITracingService)serviceProvider.GetService(typeof(ITracingService));

// Obtain the execution context from the service provider.  
IPluginExecutionContext context = (IPluginExecutionContext)serviceProvider.GetService(typeof(IPluginExecutionContext));

// For this sample, execute the plug-in code only while the client is online.   
tracingService.Trace(“AdvancedPlugin: Verifying the client is not offline.”);  
if (context.IsExecutingOffline || context.IsOfflinePlayback)  
return;

// The InputParameters collection contains all the data passed   
// in the message request.  
if (context.InputParameters.Contains(“Target”) &&  
context.InputParameters["Target"] is Entity)  
{  
Entity entity = (Entity)context.InputParameters["Target"];

}

}  
}

}

For this approach also have a Pros and Cons.

PROS:

* The step configuration is solution-aware so it will be automatically transported with the plugin step.

CONS:

* You need to use the plugin registration tool or another application to update the step configuration.
* The configuration is step-specific so you have to provide it and/or update it for every step even if the value is the same for all the steps (the configuration is on each step instead of the assembly or plugin type).
* the configuration is just an attribute of the plugin step so you cannot control privileges on the configuration independently from privileges on plugin step entity.

**\*. Use the plugin step “Secure Configuration”**  
This is similar to the step Configuration except that the configuration data is stored in a separate entity which can be secured.  
PROS:

* The configuration data can be secured as any other entity using the CRM security model. This is useful when the configuration contains sensitive information such as passwords.

CONS:

* Secure configuration is not solution aware so you will need to configure it for each environment.
* You need to use the plugin registration tool or another application to update the step configuration.
* The configuration is step-specific so you have to provide it and/or update it for every step even if the value is the same for all the steps (the configuration is on each step instead of the assembly or plugin type).

**\*. Use a Web Resource**  
You can store configuration information in web resources, for example you might have some XML configuration stored in a web resource and have your plugin read the web resource each time it executes.

PROS:

* Web resources are solution aware and the GUID is preserved across environments so you can hardcode the web resource GUID in your plugin code. You can transport the web resource and the plugin in the same solution.
* Can be easily updated using the CRM UI.

CONS:

* You cannot secure the configuration since it depends on the web resource access privileges and most users will need at least read access to web resources.

Hope this may help you.

Regards,

[Update Ownerid in the CRM Form](http://srmscrm.wordpress.com/2013/10/03/update-ownerid-in-the-crm-form/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [October 3, 2013](http://srmscrm.wordpress.com/2013/10/03/update-ownerid-in-the-crm-form/) [Leave a comment](http://srmscrm.wordpress.com/2013/10/03/update-ownerid-in-the-crm-form/#respond)

In some times , we may need to update the Owner Lookup with the Current logged in user in MS CRM 2011.

Wrong Method :

**entity["ownerid"] = new EntityReference(“systemuser”,Guid);**

Preferred Method :

For this requirement we need to assign the Record to Current Logged in User in the Owner Look up.

Have a look the below code , it works fine for me.

**AssignRequest assign = new AssignRequest**  
**{**  
**Assignee = new EntityReference(“systemuser”,Guid),**  
**Target = new EntityReference(“<entity name>”, Guid)**  
**};**  
**// Execute the Request**  
**service.Execute(assign);**

Regards,

[Share a Record to The Specific Team In MS CRM 2011 using C# Code](http://srmscrm.wordpress.com/2013/07/04/share-a-record-to-the-specific-team-in-ms-crm-2011-using-c-code/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [July 4, 2013](http://srmscrm.wordpress.com/2013/07/04/share-a-record-to-the-specific-team-in-ms-crm-2011-using-c-code/) [Leave a comment](http://srmscrm.wordpress.com/2013/07/04/share-a-record-to-the-specific-team-in-ms-crm-2011-using-c-code/#respond)

Share a Record to The Specific Team In MS CRM 2011 using C# Code

For.eg. to share a incident record to a team

//Paste the Below code in your plugin or any custompage

private void ShareRecordToBuTeam(IOrganizationService service, string SharedRecordEntitySchemaName, Guid SharedRecordGuid , string TeamLogicalName,Guid TeamGuid)  
{  
GrantAccessRequest grantShareingRequest = new GrantAccessRequest()  
{  
Target = new EntityReference(SharedRecordEntitySchemaName,SharedRecordGuid),  
PrincipalAccess = new PrincipalAccess()  
{  
Principal = new EntityReference(TeamLogicalName,TeamGuid),  
AccessMask = AccessRights.ReadAccess  
}  
};  
GrantAccessResponse AccessResponse = (GrantAccessResponse)service.Execute(grantShareingRequest);  
}

//Pass the parameter to the Function as below

Need

ShareRecordToBuTeam(service,”<entity Logical Name>”,<SharedRecordGUID>, “<team logical name>”,<TeamGuid>);

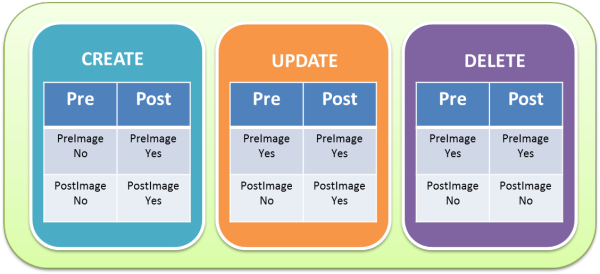
Cheers,

[Plugin Images Support – MS CRM 2011Plugins](http://srmscrm.wordpress.com/2013/05/30/plugin-images-support-ms-crm-2011plugins/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [May 30, 2013](http://srmscrm.wordpress.com/2013/05/30/plugin-images-support-ms-crm-2011plugins/) [Leave a comment](http://srmscrm.wordpress.com/2013/05/30/plugin-images-support-ms-crm-2011plugins/#respond)

Plugin Images

Have a glance below Image for quick reference .

[](http://srmscrm.files.wordpress.com/2013/05/entity-plugin-images.png)

Cheers,

[SetState Dynamic Entity – Using MS CRM 2011](http://srmscrm.wordpress.com/2013/05/29/setstate-dynamic-entity-using-ms-crm-2011/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [May 29, 2013](http://srmscrm.wordpress.com/2013/05/29/setstate-dynamic-entity-using-ms-crm-2011/) [Leave a comment](http://srmscrm.wordpress.com/2013/05/29/setstate-dynamic-entity-using-ms-crm-2011/#respond)

SetStae For Entity :

For e.g. While Reactivate the Incident Entity **“SetState”**

Register this Plugin in Both **“SetState”** and **“SetStateDynamicEntity”** on incident entity

Entity caseEntity;  
public void Execute(IServiceProvider serviceProvider)  
{  
IPluginExecutionContext context = (IPluginExecutionContext)serviceProvider.GetService(typeof(IPluginExecutionContext));  
IOrganizationServiceFactory factory = (IOrganizationServiceFactory)serviceProvider.GetService(typeof(IOrganizationServiceFactory));

try  
{  
if (context.InputParameters.Contains(“EntityMoniker”))  
{  
// Work with the Moniker  
EntityReference targetEntity = (EntityReference)context.InputParameters["EntityMoniker"];  
if (targetEntity.LogicalName != “incident”)  
{ return; }  
IOrganizationService service = factory.CreateOrganizationService(context.UserId);

switch (context.MessageName)  
{  
// Check if entity status is changed  
case “SetStateDynamicEntity”:  
if (context.InputParameters.Contains(“State”))  
{  
EntityReference CaseEntityRef = (EntityReference)context.InputParameters["EntityMoniker"];

if (((OptionSetValue)context.InputParameters["State"]).Value == 0)  
{

U can write any logic here

}  
}  
break;  
}  
service.Update(caseEntity);  
}  
}  
catch (Exception e)  
{  
throw new InvalidPluginExecutionException(“An error occured for SetState plugin ” + e.Message + e.InnerException);  
}

}

Cheers,

[Plugin Code to Deactivate Entity Record – MS CRM 2011](http://srmscrm.wordpress.com/2013/05/29/plugin-code-to-deactivate-entity-record-ms-crm-2011/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [May 29, 2013](http://srmscrm.wordpress.com/2013/05/29/plugin-code-to-deactivate-entity-record-ms-crm-2011/) [Leave a comment](http://srmscrm.wordpress.com/2013/05/29/plugin-code-to-deactivate-entity-record-ms-crm-2011/#respond)

**Deactivate Entity Record**

You can use below code at any Condition.

try

{

EntityReference MonikerEntityEntity = new EntityReference();  
MonikerEntityEntity.LogicalName = entity.LogicalName;  
MonikerEntity.Id = entity.Id;

OrganizationRequest request = new Microsoft.Xrm.Sdk.OrganizationRequest() { RequestName = “SetState” };  
request["EntityMonikerEntity"] = MonikerEntity;  
OptionSetValue state = new OptionSetValue(1);  
OptionSetValue status = new OptionSetValue(2);  
request["State"] = state;  
request["Status"] = status;

Service.Execute(request);  
}  
catch (Exception)  
{

throw;  
}

Cheers,

[Difference between Secure / Unsecure Configuration of Plugin Registration tool in CRM 2011](http://srmscrm.wordpress.com/2013/04/27/difference-between-secure-unsecure-configuration-of-plugin-registration-tool-in-crm-2011/)

[Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [April 27, 2013](http://srmscrm.wordpress.com/2013/04/27/difference-between-secure-unsecure-configuration-of-plugin-registration-tool-in-crm-2011/) [Leave a comment](http://srmscrm.wordpress.com/2013/04/27/difference-between-secure-unsecure-configuration-of-plugin-registration-tool-in-crm-2011/#respond)

Difference between Secure / Unsecure Configuration of Plugin Registration tool in CRM 2011

|  |  |
| --- | --- |
| Unsecure Configuration of  Plugin Registration tool in CRM 2011 | Secure Configuration of Plugin  Registration tool in CRM 2011 |
| Unsecure configuration information could be read by any user in CRM. Remember its *public* information (Eg: Parameter strings to be used in plugin could be supplied here) | The Secure Configuration information could be read only by CRM Administrators.(Eg: Restricted data from normal user could be supplied here) |
| Imagine that you include a plugin, plugin steps and activate them in a solution. Later solution was exported as Managed Solution to another environment. In this scenario, the supplied Unsecure configuration values would be available in the new environment. | Imagine that you include a plugin,plugin steps and activate them in asolution. Later solution was exportedas Managed Solution to anotherenvironment. In this scenario, thesupplied Secure configuration  information would *NOT*be available in the new environment. The simple  reason behind this is to provide more security to the contents of Secure Configuration. |

Note: If you supply parameter strings under secure configuration then the plugin will work fine only for the CRMIf you supply any parameter strings to Plugin and read by all of the users in CRM , use Unsecure Configuration. If

Consider a scenario that you have developed a plugin and certain parameter strings are designed to supply to the plugin in such a way that it is required to run the plugin smoothly.If you supply these parameter strings under secure configuration then the plugin will work fine only for the CRM Administrators. The simple reason is secure configuration can only be read by a CRM Administrator. So if the user is not a CRM administrator then the plugin would try to read  but it would fail just because its under the secure configuration.All public information should be supplied via Unsecure configuration section. So please remember these tips when you supply the secure and unsecure configuration via Plugin Registration tool.

Cheers,

[MS Dynamics CRM 2011 Entity Images](http://srmscrm.wordpress.com/2013/03/14/ms-dynamics-crm-2011-entity-images/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [March 14, 2013](http://srmscrm.wordpress.com/2013/03/14/ms-dynamics-crm-2011-entity-images/) [Leave a comment](http://srmscrm.wordpress.com/2013/03/14/ms-dynamics-crm-2011-entity-images/#respond)

**MS CRM 2011 Entity Images**

**PreEntityImages** :( Gets the properties of the primary entity before the core platform operation has begins)

**PreEntityImages** used to capture the data when the form loads. That is the data which is present by default when the form loads.  The syntax for using the PreEntityImages in CRM 2011 is changed as compared to CRM 4.0.

**Note:**The PreEntityImages cannot be registered for “create” operation.

Syntax Used in CRM 2011 :  
=======================

Suppose you registered the Plugin and added a Image with name “**PreImage** ”

**public EntityImageCollection PreEntityImages { get;}**  
**PostEntityImages** : (Gets the properties of the primary entity after the core platform operation has been completed. )

The **PostEntityImages** contains the attributes value which are finally changed. We can capture the changed data before the database operation takes place. And can do any kind of validation based on the changed data.

**Note:** **PostEntityImages** can only be registered  for **Update**message and cannot be registered on**Create** message.

Syntax Used in CRM 2011 :

Suppose you registered the Plugin and added a Image with name “**PostImage** ”

Syntax:  
=========  
**public EntityImageCollection PostEntityImages { get; }**

Cheers,

[CorrelationId Property in MS CRM](http://srmscrm.wordpress.com/2013/02/19/correlationid-property-in-ms-crm/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [February 19, 2013](http://srmscrm.wordpress.com/2013/02/19/correlationid-property-in-ms-crm/) [Leave a comment](http://srmscrm.wordpress.com/2013/02/19/correlationid-property-in-ms-crm/#respond)

**MS CRM 4.0**

The CorrelationId property identifies a relationship between multiple requests. For example, a request processed by a parent pipeline can cause the platform to create and process multiple child pipeline requests in order to complete the parent request. The parent and child pipeline requests have the same CorrelationId to establish a relationship among these requests.

public Guid CorrelationId {get; set;}

**Property Value**

The value of this property is a **Guid** type.

**Requirements**

**Web Service:** CrmService

**MS CRM 2011:**

Guid CorrelationId {get;}

Property Value

Type: [Guid](http://msdn.microsoft.com/en-us/library/system.guid.aspx)

The global unique identifier for tracking plug-in or custom workflow activity execution.

[Parent Vs Child Execution Pipeline in MS CRM 4.0](http://srmscrm.wordpress.com/2013/02/19/parent-vs-child-execution-pipeline-in-ms-crm-2011/)

[Ms Crm 2011](http://srmscrm.wordpress.com/category/ms-crm-2011/), [Ms Crm 2011 PlugIns](http://srmscrm.wordpress.com/category/ms-crm-2011/ms-crm-2011-plugins/) [February 19, 2013](http://srmscrm.wordpress.com/2013/02/19/parent-vs-child-execution-pipeline-in-ms-crm-2011/) [1 Comment](http://srmscrm.wordpress.com/2013/02/19/parent-vs-child-execution-pipeline-in-ms-crm-2011/#comments)

The event execution pipeline architecture provides the capability for a pipeline to initiate the execution of one or more subordinate pipelines in order to process a request. The originating pipeline is known as the “parent” while the subordinate pipeline is called the “child”. Web service method calls typically start a parent pipeline while (internal) platform calls start a child pipeline. Child pipelines only process **Create**, **Update**, or **Delete** messages as required by the platform to process the parent pipeline’s message. To determine which message is being processed by a child pipeline, you must enable [tracing](http://msdn.microsoft.com/en-us/library/cc151092.aspx) in Microsoft Dynamics CRM.

When a complex request is processed by a pipeline, the pipeline first executes any registered pre-event plug-ins and the core platform operation. Next, the pipeline starts a child pipeline and temporarily suspends processing until the child pipeline has finished processing its request. Once the child pipeline has completed, the parent pipeline can complete its post-event processing.

Parent and Child Context

Context plays an important role when it comes to parent-child pipeline execution. The [context](http://msdn.microsoft.com/en-us/library/bb959587.aspx) that is passed to plug-ins of the child pipeline will include the [parent context](http://msdn.microsoft.com/en-us/library/cc156401.aspx)that was executed immediately prior to the child pipeline’s invocation. This will enable you to track the parent and child pipeline executions and get more information about what is going on

Invocation Source

A plug-in can be registered to be executed from a parent pipeline or a child pipeline. At run time, a plug-in receives information as to how it was invoked through the[InvocationSource](http://msdn.microsoft.com/en-us/library/cc156398.aspx) property of the context that is passed as a parameter to the plug-in.

If you want your plug-in to run regardless of whether a Web service call or an (internal) platform call initiated the pipeline, register your plug-in in a child pipeline. However, do not use the [IPluginExecutionContext.CreateCrmService](http://msdn.microsoft.com/en-us/library/bb959583.aspx) or [IPluginExecutionContext.CreateMetadataService](http://msdn.microsoft.com/en-us/library/bb959584.aspx) methods if you are writing code for a plug-in in that is used in a child pipeline. In a child pipeline, you must instantiate the CrmService or MetadataService manually.

**Example**

The following code example shows how to create a CrmService proxy for plug-ins that execute in the child pipeline.

[C#]

/// <param name="context">The execution context that was passed to the plug-in's Execute method.</param>

/// <param name="flag">Set to True to use impersonation.</param>

/// <returns>A CrmServce instance.</returns>

private CrmService CreateCrmService(IPluginExecutionContext context, Boolean flag)

{

CrmAuthenticationToken authToken = new CrmAuthenticationToken();

authToken.AuthenticationType = 0;

authToken.OrganizationName = context.OrganizationName;

// Include support for impersonation.

if (flag)

authToken.CallerId = context.UserId;

else

authToken.CallerId = context.InitiatingUserId;

CrmService service = new CrmService();

service.CrmAuthenticationTokenValue = authToken;

service.UseDefaultCredentials = true;

// Include support for infinite loop detection.

CorrelationToken corToken = new CorrelationToken();

corToken.CorrelationId = context.CorrelationId;

corToken.CorrelationUpdatedTime = context.CorrelationUpdatedTime;

corToken.Depth = context.Depth;

RegistryKey regkey = Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\MSCRM");

service.Url = String.Concat(regkey.GetValue("ServerUrl").ToString(), "/2007/crmservice.asmx");

service.CorrelationTokenValue = corToken;

return service;

}