Note that not all requests contain a Target property that is of type Entity, so you have to look at each request or response. For example, Delete Request has a Target property, but its type is Entity Reference. The preceding code example would be changed as follows.

Similarly, the Output Parameters property contains the data that is in the response message, for example Create Response, currently being passed through the event execution pipeline.

**However, only synchronous post-event and asynchronous registered plug-ins have OutputParameters populated as the response is the result of the core platform operation.**

The property is of type ParameterCollection where the keys to access the response data are the names of the actual public properties in the response.

## [Pre and post entity images](javascript:void(0))

[PreEntityImages](https://msdn.microsoft.com/en-us/library/microsoft.xrm.sdk.iexecutioncontext.preentityimages.aspx) and [PostEntityImages](https://msdn.microsoft.com/en-us/library/microsoft.xrm.sdk.iexecutioncontext.postentityimages.aspx) contain snapshots of the primary entity's attributes before (pre) and after (post) the core platform operation.

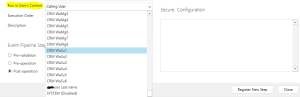
There are some events where images aren’t available. For example, only synchronous post-event and asynchronous registered plug-ins have[PostEntityImages](https://msdn.microsoft.com/en-us/library/microsoft.xrm.sdk.iexecutioncontext.postentityimages.aspx) populated. The create operation doesn’t support a pre-image and a delete operation doesn’t support a post-image.

In CRM plugin, “*IExecutionContext*” contains 2 properties

* UserId
  + *Gets the GUID of the user for whom the plug-in invokes “on behalf of”.*
* InitiatingUserId
  + *Gets the GUID of the user under which the current pipeline is executing.*

Consider a scenario

* You have a user “RAJ” with “Sales Person” role with only “User Level” “Read” privilege on ‘Contact’
* You have a plugin on Post Deletion of ‘Contact’ with name “PostContactDelete”
* Assume in one particular scenario user “RAJ” should be able to delete a ‘Contact’
* So you can run the “PostContactDelete” plugin in the user with “SystemAdministrator” role
  + (i.e., Set “Run in User’s Context” to User with admin role; In sample screen shot below I chosen my admin user whose name is  ‘CRM WaSu1)

[](https://rajeevpentyala.files.wordpress.com/2012/08/run-in-user-context.png)

Run in User Context

* When User “RAJ” logs in and try to delete ‘Contact’ the plug-in “PostContactDelete” fires. When you debug
  + IExecutionContext.UserId = GUID of  SystemAdministrator(i.e., OnBehalfOf User ‘RAJ’)
  + IExecutionContext. InitiatingUserId =GUID of  RAJ   (i.e., Actual User)

Plugins:

1. A realtime workflow is similar to a plugin
2. An asynchronous plugin is similar to a workflow

Take this particular scenario: You have a plugin that runs synchronously on the Update of a certain entity. Later down the track, you decide you want this plugin to run async, as the set of tasks performed by the plugin doesn’t really have to be realtime and you want to improve the performance of the core operation.

In this scenario, you have to be mindful of one particular plugin behaviour: transaction. When an exception is thrown inside a plugin, the core operation won’t succeed as the exception in the plugin will cause the transaction to rollback. When you change the *Execution Mode* to “Asynchronous”, you will still see that the plugin has failed in System Jobs area, but the transaction won’t be rolled back.

Register plugins to run sychronously, if you need the transaction to rollback on plugin exception and need to validate certain criteria for the core operation to succeed. If you don’t need these features, use a workflow.

**Isolation mode**

* Sandboxed
* none

If you choose Sandboxed the plugin code will be run in CRM Sandboxed Processing services, this service is sandboxed, if the plugin tries to modify or access external dll’s, certain system resources an error will be thrown.

if you choose none it gets run by the core CRM system in the event execution pipeline if it Synchronous, if it’s asynchronous it gets run by the CRM async service

[Plugin performance in Microsoft Dynamics CRM 2013/2015](http://stackoverflow.com/questions/30642382/plugin-performance-in-microsoft-dynamics-crm-2013-2015)**:**

Time to leave the shy mode behind and make my first post on stackoverflow. After doing loads of research (plugins, performance, indexes, types of update, friends) and after trying several approaches I was unable to find a proper answer/solution.   
  
So if possible I would like to get your feedback/help in a Microsoft Dynamics CRM 2013/2015 plugin performance issue (or coding technique)

**Scenario:**

Microsoft Dynamics CRM 2013/2015  
2 Entities with Relationship 1:N  
EntityA  
EntityB

EntityB has the following columns:  
Id | EntityAId | ColumnDemoX (decimal) | ColumnDemoY (currency)

Entity A has: 500 records  
Entity B has: 150 records per each Entity A record. So 500\*150 = 75000 records.

**Objective:**

Create a Post Entity A Plugin Update to "mimic" the following SQL command

Update EntityB

Set ColumnDemoX = (some quantity), ColumnDemoY = (some quantity) \* (some value)

Where EntityAId = (some id)

One approach could be:

using (varserviceContext = new XrmServiceContext(service))

{

var query = from a in serviceContext.EntityASet

where a.EntityAId.Equals(someId)

select a;

foreach (EntityAentA in query)

{

entA.ColumnDemoX = (some quantity);

serviceContext.UpdateObject(entA);

}

serviceContext.SaveChanges();

}

**Problem:**

The foreach for ***150 records*** in the post plugin update will take ***20 secs or more***.  
While the   
Update EntityB Set ColumnDemoX = (some quantity), ColumnDemoY = (some quantity) \* (some value) Where EntityAId = (some id)  
it will take ***0.00001*** secs

[How to Detect Duplicate record in CRM 2013 on Create (Pre-Validate ) Plugin](http://stackoverflow.com/questions/28628172/how-to-detect-duplicate-record-in-crm-2013-on-create-pre-validate-plugin)**:**

You need to retrieve the entity from the plugin context as described in the [documentation](https://msdn.microsoft.com/en-us/library/gg309673.aspx):

IPluginExecutionContext context = (IPluginExecutionContext)

serviceProvider.GetService(typeof(IPluginExecutionContext));

// The InputParameters collection contains all the data passed in the message request.

if (context.InputParameters.Contains("Target") &&

context.InputParameters["Target"] is Entity)

{

// Obtain the target entity from the input parameters.

Entity entity = (Entity)context.InputParameters["Target"];

// Your code here...

vardesiredValue = entity.GetAttributeValue<desiredtype>("desiredfield");

}

[How to get CRM 2013 Online Database schema](http://stackoverflow.com/questions/28272368/how-to-get-crm-2013-online-database-schema)**:**

A quicker option that doesn't require getting a backup of the database is to use the XRMtoolbox available at <http://xrmtoolbox.codeplex.com> and use the Metadata Doc Generator tool. That allows you to retrieve the Metadata and select all or some entities and various options for those entities and to save the output to a Word document or an Excel document

### ****CRM Online limitations****

I believe CRM online sandboxed plugins cannot use LINQ queries, which throws an error due to another transaction being created in the LINQ query.

### ****Correction – You can call WCF**** webservices

I originally put you couldn’t call webservices but I have been corrected thanks to a great comment from Andrii[Butenko](http://a33ik.blogspot.ca/)

### CRM 2015 Online allows custom workflows

Another great comment has caused me to update this post.

Custom workflow activities can now be registered in isolation mode in CRM 2015 online

Plug-ins may or may not execute within the database transaction of the Microsoft Dynamics CRM platform. Whether a plug-in is part of the transaction is dependent on how the message request is processed by the pipeline. You can check if the plug-in is executing in-transaction by reading the IsInTransaction property inherited by IPluginExecutionContext that is passed to the plug-in. If a plug-in is executing in the database transaction and allows an exception to be passed back to the platform, the entire transaction will be rolled back. Stages 20 and 40 are guaranteed to be part of the database transaction while stage 10 may be part of the transaction.

Any registered plug-in that executes during the database transaction and that passes an exception back to the platform cancels the core operation. This results in a rollback of the core operation. In addition, any pre-event or post event registered plug-ins that have not yet executed and any workflow that is triggered by the same event that the plug-in was registered for will not execute.

[CRM 2011 - Rollback mechanism for triggering a plugin within another plugin](http://stackoverflow.com/questions/21426976/crm-2011-rollback-mechanism-for-triggering-a-plugin-within-another-plugin)**:**

I have a question regarding the rollback mechanism in CRM 2011. I know that a synchronous plugin will rollback any changes it has performed if an error is thrown. But what if this plugin triggers another plugin during it's execution.

Consider the following plugins:

1. Plugin A: Triggers on pre-UPDATE of Task entity
2. Plugin B: Triggers on pre-UPDATE of Case(incident) entity

Scenario would be like this:

1. I update a case and Plugin B gets triggered.
2. Fields in the case are modified
3. One of the Tasks associated with the case is also updated with some information.
   * Plugin A is triggered
   * Fields related to the task are modified
4. Some other operations on the case record
5. Plugin B throws exception

My question is, would the operations performed in Plugin A get rolled back as well?

As long as the plugins are registered in the transaction (before or after, but still in), everything will get rolled back. If a plugin is registered for the Pre-Validation stage, it will not get rolled back.

This also assumes that you're retrieving the IOrganziationService from the PluginContext as well.

This is what happens on the server:

1. A request to update an entity comes in, are we currently in a transaction via the context? If not create a new Database Transaction and store it in the Plugin Context.
2. Request is made to update a different entity, transaction is passed to the new plugin's context, and step 1 is repeated for new plugin execution.

Handle exceptions in plug-ins:

For synchronous plug-ins, whether registered in the sandbox or not, the Microsoft Dynamics CRM platform handles exceptions passed back from a plug-in by displaying an error message in a dialog of the web application user interface. The exception message for asynchronous registered plug-ins is written to a System Job (**AsyncOperation**) record which can be viewed in the System Jobs area of the web application.

For synchronous plug-ins, you can optionally display a custom error message in the error dialog of the web application by having your plug-in throw an [InvalidPluginExecutionException](https://msdn.microsoft.com/en-in/library/microsoft.xrm.sdk.invalidpluginexecutionexception.aspx) exception with the custom message string as the exception **Message** property value. If you throw [InvalidPluginExecutionException](https://msdn.microsoft.com/en-in/library/microsoft.xrm.sdk.invalidpluginexecutionexception.aspx) and do not provide a custom message, a generic default message is displayed in the error dialog. It is recommended that plug-ins only pass an [InvalidPluginExecutionException](https://msdn.microsoft.com/en-in/library/microsoft.xrm.sdk.invalidpluginexecutionexception.aspx) back to the platform.

If a synchronous plug-in returns an exception other than [InvalidPluginExecutionException](https://msdn.microsoft.com/en-in/library/microsoft.xrm.sdk.invalidpluginexecutionexception.aspx) back to the platform, the error dialog is displayed to the user and the exception message ([System.Exception.Message](https://msdn.microsoft.com/library/system.exception.message.aspx)) with stack trace is also written to one of two places. For plug-ins not registered in the sandbox, the information is written to the Application event log on the server that runs the plug-in. The event log can be viewed by using the Event Viewer administrative tool. For plug-ins registered in the sandbox, the exception message and stack trace is written to the Microsoft Dynamics CRM platform trace. For more information about tracing, see the Logging and Tracing section of the [Debug a plug-In](https://msdn.microsoft.com/en-in/library/gg328574.aspx) topic.

Plug-in Tracing in Dynamics CRM 2015 Update 1

<http://inogic.com/blog/2015/04/plug-in-trace-log-in-crm-2015-update-1/>:

[Will an exception in an async plugin stop a creation of an instance?](http://stackoverflow.com/questions/18287401/will-an-exception-in-an-async-plugin-stop-a-creation-of-an-instance)

|  |  |  |
| --- | --- | --- |
| I'm talking to a customer who has re-registered one of their plugin to asynchronous position and they claim that their experience "weird behavior". According to the description, it seems that when an exception is thrown in the async plugin (message create), the process is not rolled back. The instance is created.  Is that an intended behavior? And if so, what can be done to stop it?  [c#](http://stackoverflow.com/questions/tagged/c%23)[plugins](http://stackoverflow.com/questions/tagged/plugins)[dynamics-crm-2011](http://stackoverflow.com/questions/tagged/dynamics-crm-2011)   |  |  | | --- | --- | | [share](http://stackoverflow.com/q/18287401)|[improve this question](http://stackoverflow.com/posts/18287401/edit) | asked Aug 17 '13 at 9:48  [[https://www.gravatar.com/avatar/e1d7083d88bd7d8eedea825a941dae57?s=32&d=identicon&r=PG](http://stackoverflow.com/users/1525840/konrad-viltersten)](http://stackoverflow.com/users/1525840/konrad-viltersten)  [Konrad Viltersten](http://stackoverflow.com/users/1525840/konrad-viltersten)  **7,319**1153113 | |
|  | |  |  | | --- | --- | |  |  |   add a comment | |

1 Answer 1

|  |  |
| --- | --- |
|  | Asynchronous plugins are not executed inside the main operation transaction, this means that when an exception is thrown the main operation is not rolled back.  The common example is the creation of a task inside a plugin when a new account is created:  If the plugin is synchronous and an InvalidPluginExecutionException is thrown in your plugin, both account and task will not be created and the error is prompted to the user.  If the plugin is asynchronous and an InvalidPluginExecutionException is thrown in your plugin, the account will be created, the task will not be created and no error is prompted to the user.  [**CRM 2011 Async Post Update Plugin**](http://stackoverflow.com/questions/10178824/crm-2011-async-post-update-plugin)**:**  **I have an Aynchronous Post-Operation plugin runs on an Update step. Now my challange is to find original values for the fields that the user is updating. Since it is post-operation step, is it possible to find the previous value fields that were updated?**  **Yes. Register a preimage snapshot of your entity against the step in your plugin**.  [Async& Sync plugins on the same event](http://stackoverflow.com/questions/33908001/async-sync-plugins-on-the-same-event)**:**  I have a sync plugin that triggers on Entity A, Update Post Operation event. I have another Async plugin that triggers on the same Entity A and on the Update Post Operation as well. These 2 plugins serve different purposes and they must be registered this way.  I have the execution order set as follows: Sync plugin 1 Async plugin 2  My question is if the Sync plugin triggers first and throw exception, does the Async plugin still triggers and what about if I switch them so the Async will have execution order 1 and the Sync execution order 2.  My goal is to rollback any update if any of these 2 plugins throw an exception.  Only when the synchronous plugin pipeline finishes without errors, asynchronous plugins are triggered.  So, the answer to your question is: no.  Should I register my plugins in Sandbox or no isolation?  http://gonzaloruizcrm.blogspot.in/2015/03/should-i-register-my-plugins-in-sandbox.html |

<http://www.magnetismsolutions.com/blog/ahmed-anwar's-blog/2015/08/07/executing-messages-in-a-single-transaction-in-dynamics-crm-2015>

<http://www.kingswaysoft.com/blog/2015/04/19/New-CRM-SDK-Feature---Transactional-Batching>

<http://chaitanyaprasadtk.blogspot.in/2013/01/execute-multiple-request-rollup-12-sdk.html>

Executing Messages in a Single Transaction in Dynamics CRM 2015

ExecuteMultipleRequest is used to batch multiple request at once,

http://mscrm-workarounds.blogspot.in/2011/07/override-created-on-or-created-by.html

Override Created on or Created by during Import

CRM 2015 – Understanding impersonation in plugins and knowing when to use it

https://crmbusiness.wordpress.com/2015/07/21/crm-2015-understanding-impersonation-in-plugins-and-knowing-when-to-use-it/

http://www.powerobjects.com/2012/08/06/how-to-impersonate-in-microsoft-dynamics-crm/

http://stackoverflow.com/questions/16723468/how-to-get-the-impersonated-user-id-in-a-plugin

To impersonate a user, set the CallerId property on an instance of OrganizationServiceProxy before calling the service’s Web methods.

Asynchronous plugins are not executed inside the main operation transaction, this means that when an exception is thrown the main operation is not rolled back.

The common example is the creation of a task inside a plugin when a new account is created:

If the plugin is synchronous and an InvalidPluginExecutionException is thrown in your plugin, both account and task will not be created and the error is prompted to the user.

If the plugin is asynchronous and an InvalidPluginExecutionException is thrown in your plugin, the account will be created, the task will not be created and no error is prompted to the user.