

Change Tracking Dynamics 365

A feature that provides the ability to track the changes on D365 made by System administrators/customizers. It includes change tracking for entities, JavaScript's, Assemblies and Processes.

1 Overview

The Change Tracking solution helps in tracking down the details of changes on who updated an entity, JavaScript, assemblies and processes along with the time of update. This solution is built on Dynamics 365 and as well works on Dynamics CRM 2016 (Online/ On premises).

2 How does Change Tracking Solution work?

The Change Tracking solution uses two plugins on “Publish” and “PublishAll” messages against “none”. The plugin on “Publish” message captures the user who published the component, name of the component and the time of publishing whereas the plugin of “PublishAll” message captures the user and time when a developer publishes a solution. It stores the details Change Tracking entity.

For Assemblies and workflows, a batch job which runs every hour (the time interval can be changed by the administrators) and collects information about the assemblies and workflows & stores it in Change Tracking entity.

Once downloaded from AppSource to your Dynamics 365 application, there are two steps to make the solution work, administrator would need to add a record in Batch process entity and create a bulk delete system batch job in order to remove the old records in Change Tracking entity. (the bulk delete system batch job is only to remove old data if you do not wish to maintain it)

Enlisted below are the different roles in Dynamics 365 who are involved in this process.

1. System Administrator –Add a record in Batch Process entity.



BATCH PROCESS : INFORMATION

Change Tracking Batch

Process* Change Tracking

Next Activation 8/4/2017 10:45 AM

Status Reason Scheduled

• General

Name* Change Tracking Batch

Process* Change Tracking Plugins and Workflows

Activation Frequency Hourly

Status Reason Scheduled

Last Activation 8/4/2017 4:15 AM

Target Records

```
<fetch version='1.0' output-format='xml-platform' mapping='logical' distinct='false'>
  <entity name='transactioncurrency'>
    <attribute name='transactioncurrencyid' />
    <attribute name='currencyname' />
    <attribute name='isocurrencycode' />
    <attribute name='currencysymbol' />
    <attribute name='exchangerate' />
    <attribute name='currencyprecision' />
    <order attribute='currencyname' descending='false' />
    <filter type='and'>
      <condition attribute='currencyname' operator='eq' value='US Dollar' />
    </filter>
  </entity>
</fetch>
```

Activation Counter 16

Hours Interval 1

Last Activation Compl Success

Next Activation 8/4/2017 10:45 AM

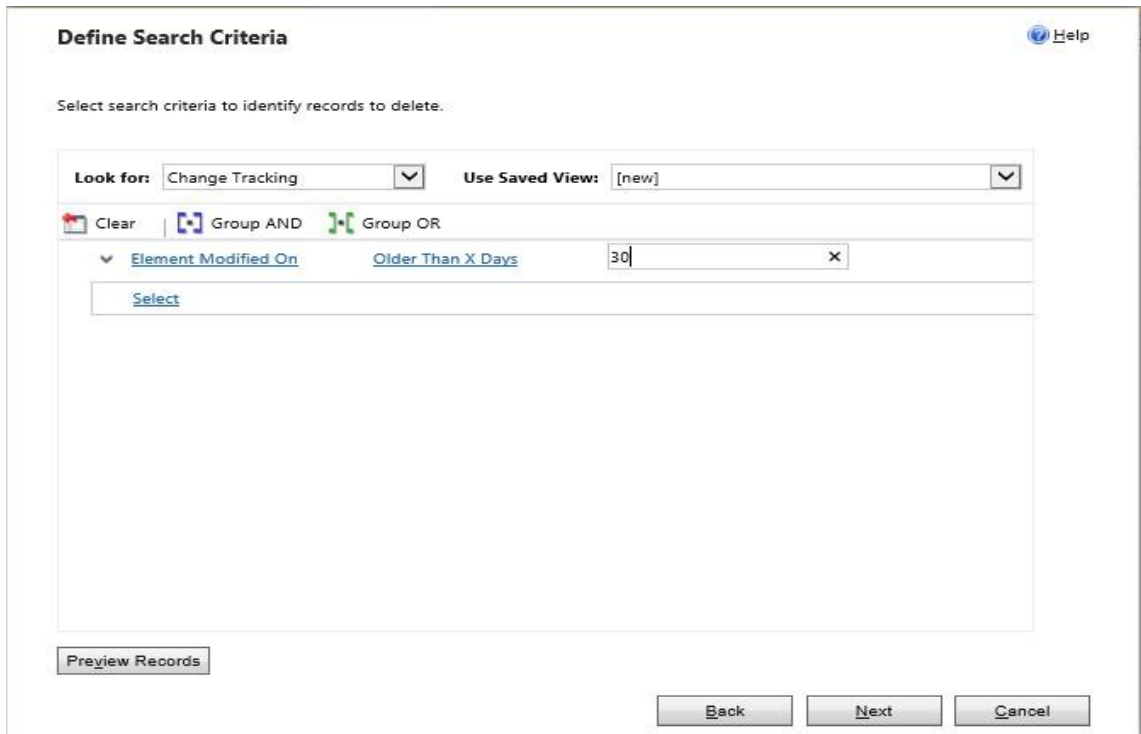
Last Activation Error --

The batch execution works in this way: It takes the fetchxml query from the Target records field and executes the code for each record found in the results of the query. As this effects the performance, we need to add a fetch query which results in only one record so that the code executes only once and the fetching, looping and creation of records is performed inside the code. So we are using a fetch on Currency entity where currency is US Dollar. By doing this the code executes only once and the fetching, looping and creating done at once.

Please add the below fetchxml in the Target Records field:

```
<fetch version='1.0' output-format='xml-platform' mapping='logical' distinct='false'>
  <entity name='transactioncurrency'>
    <attribute name='transactioncurrencyid' />
    <attribute name='currencyname' />
    <attribute name='isocurrencycode' />
    <attribute name='currencysymbol' />
    <attribute name='exchangerate' />
    <attribute name='currencyprecision' />
    <order attribute='currencyname' descending='false' />
    <filter type='and'>
      <condition attribute='currencyname' operator='eq' value='US Dollar' />
    </filter>
  </entity>
</fetch>
```

2. System administrator - Navigate to Setting → Data management → Bulk Record Deletion → Add new record.



Since the amount of data collected will be more, this system job helps in removing the records older than certain number of days. [It is also recommended to do this, in order to ensure proper performance for this change tracking].

3 End User Experience

Once the solution is installed and configured in the CRM organization, User can see the records getting added in the new entity as the changes happen in D365 instance.

In the following examples, components published or updated were captured and a new record is created with the necessary details:

Example records with data for a web resources:

General

Audit Information

| | | | |
|--|--|--|--|
| Message Name Publish | Published By Sravan Kumar Kancharla | Element Modified On 7/11/2017 6:13 AM | Published Element Name new_customizationaudittest |
| Published Element Type Web Resource | Web Resource Type Script (JScript) | | |

Published XML

```
<publish xmlns="http://schemas.microsoft.com/crm/2009/WebServices"> <webresources> <webresource> {B2418B9F-225B-E711-810E-3863BB36DD08} </webresource> </webresources> </publish>
```

General

Audit Information

| | | | |
|--|--|--|--|
| Message Name Publish | Published By Sravan Kumar Kancharla | Element Modified On 7/11/2017 6:14 AM | Published Element Name new_fetchcomponentdata |
| Published Element Type Web Resource | Web Resource Type Webpage (HTML) | | |

Published XML

```
<publish xmlns="http://schemas.microsoft.com/crm/2009/WebServices"> <webresources> <webresource> {C864A942-6C5D-E711-8110-3863BB36DD08} </webresource> </webresources> </publish>
```

Notice the change of "Web resource type" which can be captured for Web resources.

Example record with data for an Entity:

General

Audit Information

| | | | |
|----------------------------------|--|--|--|
| Message Name Publish | Published By Sravan Kumar Kancharla | Element Modified On 7/11/2017 6:16 AM | Published Element Name new_testentity |
| Published Element Type Entity | | | |

Published XML

```
<publish xmlns="http://schemas.microsoft.com/crm/2009/WebServices"> <entities> <entity>new_testentity</entity> </entities> </publish>
```

Example record with data for Publish All:

General

Audit Information

| | | | |
|---------------------------------------|--|--|------------------------------|
| Message Name PublishAll | Published By Sravan Kumar Kancharla | Element Modified On 7/11/2017 6:18 AM | Published Element Name -- |
| Published Element Type Publish All | | | |

Published XML

--

Example data for batch job:

Batch Job captures the information about the plugins and workflows:

| <input checked="" type="checkbox"/> | Message Name | Published Element Type | Web Resource T... | Published Element Name | Published By | Published XML | Element Modifie... | Created On ↓ |
|-------------------------------------|---------------------------------|------------------------|-------------------|---------------------------------|------------------------|---------------|--------------------|------------------|
| <input checked="" type="checkbox"/> | Workflow Update | Workflow | | Batch Process Scheduling | Sravan Kumar Kancharla | | 8/3/2017 11:59... | 8/3/2017 4:45 PM |
| | Assembly Update | Plugin | | Dynamize.AsyncBatchProc... | Sravan Kumar Kancharla | | 8/3/2017 11:58... | 8/3/2017 4:45 PM |
| | Workflow Update | Workflow | | Set Activation Counter initi... | Sravan Kumar Kancharla | | 8/3/2017 11:59... | 8/3/2017 4:45 PM |
| | Workflow Update | Workflow | | Set Hours Interval accordin... | Sravan Kumar Kancharla | | 8/3/2017 11:59... | 8/3/2017 4:45 PM |

4 Troubleshooting Steps

System Administrators can use the quick tips below for troubleshooting any reported issues:

1. Ensure plugins are enabled.
2. Ensure the workflow steps has been registered for the particular event i.e. on create, update.
3. Make sure the changes were published: navigate to 'Customize the System' and click 'Publish all Customizations'