**Processes/Workflow ownership FAQs**

I decided to post a blog article about this topic since I have been asked numerous times why solution import fails sometimes when trying to update an existing process. Before I get to the bottom of it, let me answer what are in my experience the most FAQ about process ownership:

**Q1: Under what user’s context does the workflow execute? (If the workflow creates a record, who will be the owner of that new record?)**

A1: It depends. Automatically triggered workflows (such as a workflow that triggers on account create) will execute in the context of the *owner* of the workflow. Therefore, if you have a send email step, the email will be by default sent from the e-mail account of the workflow owner. This is important to consider because the workflow owner might belong to a different business unit and have different privileges than the user who triggered the workflow (e.g. who created the account). Let’s say your workflow creates a task each time an account is created. Depending on the privileges of the user, the task might be in another business unit and not visible to the user, therefore you should consider adding an “assign step” that assigns the new task to the owner of the account. Now, if the workflow is executed on-demand, the workflow will then execute in the context of *the user who requests the workflow execution*. Because dialogs are always on-demand then they always execute in the context of the user who started the dialog.

**Q2: Why does the process execute under different users depending on how it was started?**

A2: This was a design decision based on security considerations. You don’t want to inadvertently be sending emails and executing actions without knowing it because some other user decided it. Therefore, by having this different behavior we can guarantee that the user under which the workflow executes is always aware that a workflow is performing some actions on his behalf. For the automatic workflow case, the owner of the workflow is also the person who activates it and who selects the trigger mechanism and the workflow steps so it is OK if the workflow executes under that user’s context. For the on-demand case, a user is specifically requesting some actions to be performed on his behalf by a workflow so the user is fully aware of the workflow definition and that it will execute; therefore it is safe to execute the workflow under that user’s context instead of the workflow owner (who might not be aware that a user requests an on-demand execution).

**Q3: Why can’t I activate/deactivate someone else’s workflow, even if I am the system administrator?**

A3: For the same security reason as explained above. You want the workflow owner to explicitly acknowledge that a workflow will be activated and will perform some actions on his behalf. You would not want to allow another user (even the system administrator) to decide that some process should be executed on another user’s behalf. If you want to activate/deactivate someone else’s process you must first assign it to yourself.

**Q4: If I assign an activated process to another user, why does the user have to re-activate it?**

A4: Active processes cannot be modified so the system automatically deactivates them before assigning it to the new user. As per Q3 above, only the new owner will be able to re-activate the process.

**Q5: I am importing a solution that contains processes and it fails with this error message “*The workflow cannot be published or unpublished by someone who is not its owner*”. What is wrong?**

A5: If your solution contains a process that already exists in the organization and is activated then solution import will attempt to update it. In order to do so, it must first deactivate it. However, if the owner of the activated process is not the same as the user who is importing the solution, then deactivating the process will fail (see Q3). Therefore you have a few options to fix this problem:

1.       Import the solution using the user who owns the activated process. This can be tricky, especially if there are multiple processes owned by different users which need to be updated by the solution import.

2.       Verify which processes are included in the solution, and then find them in the organization, if you can find them and they are not owned by you then you must assign them to yourself. You can reassign them to the original user after you import the solution; however, you will have to ask the process owners to activate it themselves.

Scheduling recurring workflows in CRM:

<http://gonzaloruizcrm.blogspot.in/2011/05/quite-often-we-have-business-process.html>

## [Security context of workflow processes](javascript:void(0))

**When a background workflow is configured as an on-demand process and is started by a user using the Run Workflow command, the actions that the workflow can perform are limited to those the user could perform based on the privileges and access levels defined by the security role(s) set for their user account.**

When a background workflow starts based on an event the workflow operates in the context of the person who owns it, usually the person who created the workflow.

For real-time workflows you have the **Execute As** option and you can choose whether the workflow should apply the security context of the owner of the workflow or the user who made changes to the record. If your workflow includes actions which all users would not be able to perform based on security constraints, you should choose to have the workflow run as the owner of the workflow.

**Form Assistant**:

On the right side of the dialog the **Form Assistant** gives you the ability to set or append dynamic values from the context of the current record. This includes values from related records that can be accessed from the N: 1 (many-to-one) relationships for the entity.

The options available in the **Form Assistant** depend on the field you have selected in the form. When you set a dynamic value, you will see a yellow placeholder known as a ‘slug’ that shows where the dynamic data will be included. If you want to remove the value, just select the slug and delete it. For text fields, you can use a combination of static and dynamic data.

With dynamic values you don’t know for certain that a field or related entity has the value you want to set. You can actually set a number of fields to try and set the value and sort them in order using the green arrows. If the first field doesn’t have data, the second field will be tried and so on. If none of the fields have data, you can specify a default value to be used.

Asynchronous Workflow fails what about start from 1st record onwards or?

If an asynchronous system job (workflow) fails several times consecutively, CRM starts to postpone execution of that job for longer and longer time intervals to allow the CRM administrator to investigate and resolve the issue. Once the job starts succeeding again, it will resume executing normally

### [Actions on running background workflows](javascript:void(0))

While a background workflow is running, you have options to **Cancel**, **Pause**, or **postpone** the workflow. If you have previously paused a workflow, you can **resume** it.

## [Monitoring background workflows](javascript:void(0))

Background workflows generate System Job records to track their status. You can access information about these system jobs in several places within the application:

Settings > System Jobs

This will include all types of system jobs. You will need to filter records to those where **System Job Type** is **Workflow**.

From the Workflow Process

Open the background workflow definition and go to the **Process Session** tab. This will show only the system jobs for this background workflow.

From the record

You can edit the entity form so that the navigation will include the **Background Processes** relationship. This will show all the system jobs that have been started in the context of the record.

## [Monitoring real-time workflows and actions](javascript:void(0))

Real-time workflows and actions do not use System Job records because they occur immediately. Any errors that occur will be displayed to the user in the application with the heading **Business Process Error**.

There is no log for successful operations. You can enable logging for errors by checking the **Keep Logs for workflow jobs that encountered errors** option in the **Workflow Log Retention** area at the bottom of the **Administration** tab for the process.

To view the log of errors for a specific process, open the real-time workflow or action definition and go to the **Process Session** tab. This will only show any errors logged for this process.

If you want a view of all the errors for any process, go to **Advanced Find** and create a view showing errors on the process session entity.

## [Monitoring dialogs](javascript:void(0))

Every dialog that is run will create a Process Session record. This record provides a summary of the interaction within the dialog. You can view the Process Sessions for a specific dialog by using the Process Sessions area for that dialog, or you can use **Advanced Find** to create a query where the related **Process** **Category** is **Dialog**.

Some key points about a real-time workflow include the following items:

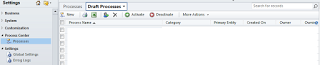
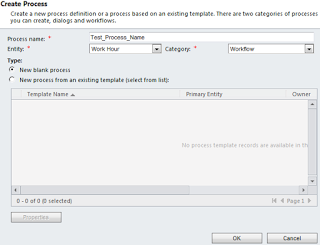
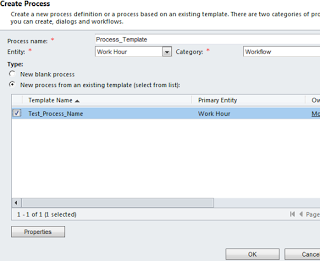
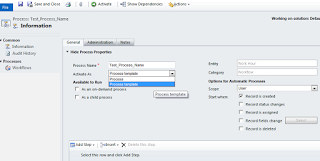
* Defined by using a **Workflow** entity record, similar to an asynchronous workflow.
* Executes in a stage of the event execution pipeline, similar to synchronous plug-ins. The real-time workflow can execute before (pre-operation), after (post-operation), or during the core operation. A real-time workflow that is executed during the core operation is the implementation of a custom action. Real-time workflows can be ranked within a stage just like you can do with plug-ins. More information: [Pipeline stages](https://msdn.microsoft.com/en-in/library/gg327941.aspx#bkmk_PipelineStages)
* Whether configured to run on-demand or in response to an event, a real-time workflow runs immediately rather than being queued to run at a later time.
* Can run in the security context of the logged on user or owner of the workflow. However, workflows set to run on-demand always run under the security context of the logged on user.
* Can’t contain any delay or wait activities.
* Only logs errors, and only when logging is enabled.
* Executes in the current transaction. All activities in the workflow and any child workflows, except asynchronous workflows, are part of a single transaction. Asynchronous child workflows are queued and execute in a separate transaction.
* Can be converted to asynchronous workflow and back to real-time.

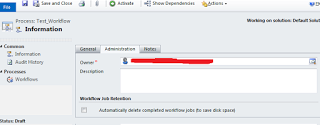
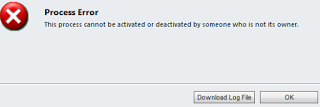
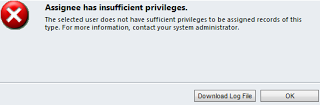
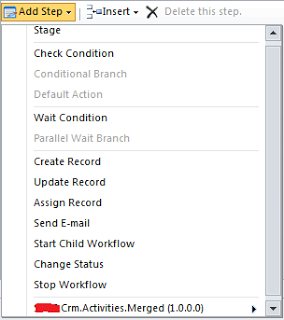
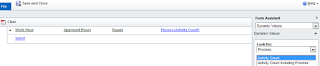
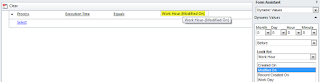
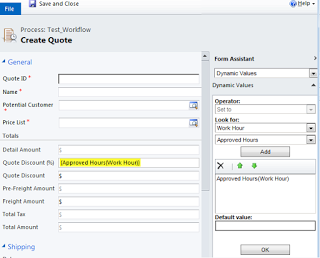
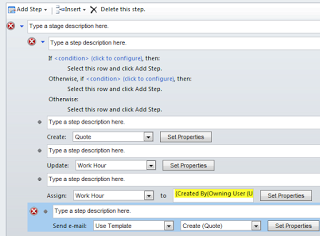
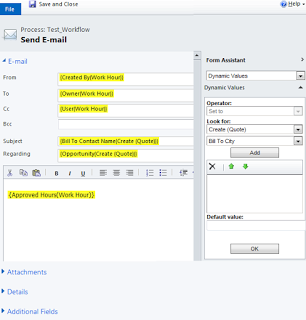
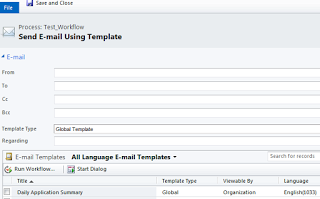
## [Error handling and rollback](javascript:void(0))

Any workflow activity 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 if it’s already occurred. In addition, any pre-event or post-event registered plug-ins or real-time workflows that haven’t executed yet. Any asynchronous workflows that are triggered by the same event won’t execute.

If you include a stop workflow ([TerminateWorkflow class](https://msdn.microsoft.com/library/system.activities.statements.terminateworkflow(v=vs.100).aspx)) activity with the cancelled option in your workflow, you can add a custom status message (**Reason** property) to that activity containing a single line of text to be displayed to the user in the **Business Process Error** dialog.

When **SyncWorkflowLogOnError** is set to **true**, real-time workflow errors are logged in **ProcessSession** entity records.

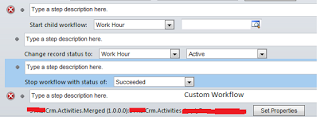
Dynamics CRM Workflow  
Typical workflow actions include sending an email message, creating a task, and updating a data field on a record.  
Dynamics CRM workflow uses the Windows Workflow Foundation framework for its core infrastructure.  
Dynamics CRM workflow is Asynchronous Processing Service.  
Dynamics CRM workflow processes in one of three ways: Manually by the user, Automatically from a trigger event & From another workflow  
Workflow Process Security - Creating and editing workflow processes & Running workflow processes (Manual or Automatic).  
Settings -> Process Center -> Processes  
[](http://2.bp.blogspot.com/-brHGmmxd3QQ/UYkZowi18OI/AAAAAAAABXY/Mi0svsMbop4/s1600/w1_1.png)  
**Create a workflow process**  
Enter a process name, select entity, select Workflow as the category and select new bank process, Press OK  
[](http://1.bp.blogspot.com/-_z6tQBtPWjw/UYkZohwsC4I/AAAAAAAABXE/MOWbiBNX_g0/s1600/w1.png)  
  
**Workflow Process Templates** - Save time to create new workflow, Select Activate As -> Process template, when creating the workflow. Press 'Activate'.  
[](http://2.bp.blogspot.com/-gNUyfC9S_Cs/UYkZpLHEjvI/AAAAAAAABXk/cWzWp5FfpzA/s1600/w3.png)  
**Create a Workflow Process Template**  
[](http://2.bp.blogspot.com/-iYKnOasDzJE/UYkZpHaHOTI/AAAAAAAABXc/xnKhUZ7Vvg4/s1600/w2.png)

**Administration tab** - update the owner, description, and check the workflow job retention   
automatically delete completed workflow jobs (to save disk space) - whether the completion of jobs is critical to maintain for auditing purposes. If not, check it.  
[](http://2.bp.blogspot.com/-Rd8DiIrStUM/UYkZpYz124I/AAAAAAAABXg/OhGtEmgdBSY/s1600/w4.png)  
**Notes tab** - to track notes about this workflow.  
Workflows can be activation and deactivation, by the workflow owner.   
[](http://4.bp.blogspot.com/-3saSUsPy-ZU/UYkZpozBhSI/AAAAAAAABXs/KthPuHxOnoc/s1600/w5.png)  
If user doesn't have permission to execute all of the steps in a workflow process, including any child workflow steps. It will fail.  
  
[](http://4.bp.blogspot.com/-Srj5sukZ8P0/UYkZpjyeSCI/AAAAAAAABXo/RHOj-3o3Jl8/s1600/w6.png)  
**Show Dependencies** - The dependent components are those that depend on this workflow and prevent you from deleting it while they depend on the workflow.  
  
**Available to Run (Workflow Process Execution option)**  
As an on-demand process - Execute a workflow process manually.  
As a child process - the workflow process as a child in a different workflow process.  
  
**Scope** - User, Business Unit, Parent: Child Business Units & Organization  
User Scope  - user-owned entities such as accounts, and contacts.   
Organization Scope - organization-owned entities such as addresses or products  
  
**An automatic workflow process** - CRM uses the combination of the workflow process owner’s privileges and the scope of the workflow process to determine the records affected by the workflow.  
  
For example, DNS user has Organization level update permission, but in the Workflow process scope is 'User' level, in this case DNS owns record will be updated by this process, because of limited by the scope of the workflow process.  
  
DNS user has User level update permission, but in the Workflow process scope is 'Organization' level, in this case DNS owns record will be updated by this process, because of limited by the scope of the workflow process.  
  
**Start When (Trigger Events or Actions)**  
Record is created - Creating a new record of an entity  
Record status changes - Changing the status (state - state code) of a record  
Record is assigned - Changing the owner of a record  
Record fields change  - Changing one or more values on a record  
Record is deleted  - Deleting a record  
  
**Step**  
[](http://2.bp.blogspot.com/-qg6ysRF6QC4/UYkZp_3Y2GI/AAAAAAAABXw/xgmhOTSYdns/s1600/w7.png)  
**Stages** - Stages act as groups for workflow steps. Wait conditions to each stage so that CRM won’t proceed to the next stage until the workflow process satisfies the wait conditions.  
  
**Check conditions**  
3 different check condition branches:  
Check Condition  -  the first if-then statement  
Conditional Branch - the Otherwise-if-then statement (else-if-then)  
Default Action  - the Otherwise statement (else)  
  
By clicking the <condition> (click to configure) link to open the Specify Condition dialog box  
  
**Dynamic Values**  - you can insert dynamic values in a check condition by placing your cursor in the appropriate field and selecting the dynamic value in the Form Assistant. For example, sends automatically email message to a employee every time logs a service request, here you can add dynamic employee's service details, phone number.   
[](http://3.bp.blogspot.com/-GD51Ks6Upv8/UYkZp35MrbI/AAAAAAAABX0/z7fBuRI1O4s/s1600/w8.png)  
[](http://4.bp.blogspot.com/-HqXFolPGn5o/UYkZqGydqCI/AAAAAAAABX4/-US9HpStaUs/s1600/w8_1.png)  
**operator** - Set to, Increment by, Decrement by, Multiply by, clear.  
**Look for options** - Select Process option -> Select any one - Activity Count (the current number of activities associated with the primary entity), Activity Count Including Workflow, Execution Time (the amount of time elapsed on the current workflow step, execution time value resets each time a step is taken.), Timeout (if wait condition selected, timeout will appear).  
  
**Wait Conditions** - enable you to incorporate time - or event-based dependencies to your workflow process.  Cases like - sending an email a specified amount of time before a service expires or just waiting a given amount of time before following up with a lead.  
  
**Step Actions**  
**Create Record** - activities and custom entity records.  
[](http://2.bp.blogspot.com/-fzzE1H_tzv8/UYkZqKOfI8I/AAAAAAAABX8/Iz8L6nh3yY8/s1600/w9.png)  
**Update Record** - update primary or related entities  
**Assign Record** - change the owner of the record referenced in the workflow process.  
Set Properties option  
[](http://1.bp.blogspot.com/-0aPcQ8pYJvM/UYkZo-D_vWI/AAAAAAAABXQ/D8YimTXRThE/s1600/w13.png)  
**Send E-mail**  
  
  
  
[](http://2.bp.blogspot.com/-ugzoA6jR9aU/UYkZoq0PIVI/AAAAAAAABXI/baxDBHz13qM/s1600/w11.png)  
**Email Templates** - Select 'Use Template' From Send e-mail  
  
[](http://4.bp.blogspot.com/-iYDeUxBBwvI/UYkZouz3xnI/AAAAAAAABXM/kxqSv9imVvw/s1600/w12.png)

**Start Child Workflow**

1. Execute an entirely separate workflow process as an action in the original workflow process.
2. Reference a workflow process as a child only if it has the As a child workflow option selected in the Available to Run section
3. Child workflow runs asynchronously - The parent workflow process will execute the child workflow process and then continue to the next step without waiting for the child workflow to finish its logic.
4. If you require a child workflow to execute synchronously, use a custom workflow activity in the child workflow.
5. When a workflow process executes a custom workflow activity, it does not proceed to the next step in the workflow process until Microsoft Dynamics CRM completes the entire process related to the activity.

**Change Status** - Change the status (state code) and status reason (status code) of a record.  
  
**Stop Workflow** - You want to stop a workflow somewhere in the middle of its process, typically when a condition is not met.  
Succeeded - Immediately stops the workflow process with a status of Succeeded.  
Canceled  - Immediately stops the workflow process with a status of Canceled.  
  
**Custom workflow actions** - it will appear after custom workflow activities are registered. If web interface workflow can't handle the situation,  use VS to create custom workflow.

[](http://2.bp.blogspot.com/-g9bghxII-78/UYkZoxa71VI/AAAAAAAABXU/Ml72diYV12Y/s1600/w14.png)

**Loop Detection**  
When you use child workflow processes, accidentally call the parent workflow process, in this situation workflow process can’t ever complete because it’s stuck in a loop. To avoid these situations, CRM includes loop detection logic to help minimize the possibility of infinite loops occurring in the system.

Depth counter - which increments each time it executes a process. By default, eight times before automatically halting the process.

CRM resets the depth counter to zero, if the workflow is still active and hasn’t been executed for a specified period of time, Time expiration boundary - For example, workflow updates a employee’s time card status (more than eight actions). CRM handles by using a concept known as the time expiration boundary.  
  
**System jobs status reasons** - Canceled, Canceling, Failed, In Progress, Pausing, Succeeded, Waiting, Waiting for Resources  
  
**Monitoring Workflow Jobs** - From entity record, Workflow Process Record, System Jobs (log details)

Q. Suppose there is a plug-in registered for account entity. When a user submits a request (e.g. account creation etc.) to the web-server then what will happen in the server? Ans

 –

 The plug-in will get loaded into the memory and will perform the operation it is needed to do. Q. Now what will happen if 100 users will submit the request to the web-server? The plug-in code will get loaded into the memory for 100 times? Ans

 –

 Answer is NO. Noticeable point over here is that the Microsoft CRM is a managed application and runs under .Net framework. So whenever the first request arrives at the web-server the plug-in code is loaded into the memory and will perform its operation and subsequently the same plug-in code will serve the process for other user as well. So this way it saves the amount of time required to load the plug-in into the memory. If the plug-in code is not being used for long then the Garbage collector will identify it and will sweep the plug-in out from the memory. Q. What is a Plug-in? Ans - A plug-in is custom business logic (code) that you can integrate with Microsoft Dynamics CRM 2011 and Microsoft Dynamics CRM Online to modify or augment the standard behavior of the platform. Another way to think about plug-ins is that they are handlers for events fired by Microsoft Dynamics CRM. You can subscribe, or register, a plug-in to a known set of events to have your code run when the event occurs. Q. What is a Workflow? Ans -

In Microsoft Dynamics CRM 2011, workflows are now called processes. In addition to this change, there are two categories of processes

**Real-time workflows :**

Similar to asynchronous workflows, real-time workflows can be used to model and automate real world business processes. They can optionally require user input, can start automatically based on specified event conditions, or can be started manually by a user. Real-time workflows are for business users, for example business analysts, to implement similar functionality to synchronous plug-ins without requiring .NET Framework programming experience. You can create asynchronous or real-time workflows in the Microsoft Dynamics CRM web application or in code.

Some key points about a real-time workflow include the following items:

* Defined by using a **Workflow** entity record, similar to an asynchronous workflow.
* Executes in a stage of the event execution pipeline, similar to synchronous plug-ins. The real-time workflow can execute before (pre-operation), after (post-operation), or during the core operation. A real-time workflow that is executed during the core operation is the implementation of a custom action. Real-time workflows can be ranked within a stage just like you can do with plug-ins. More information: [Pipeline Stages](ms-its:C:\Users\home\Desktop\2016\SDK\CrmSdk2016.chm::/html/1281920e-554b-458d-a48c-c5f6bb485062.htm#bkmk_PipelineStages)
* Whether configured to run on-demand or in response to an event, a real-time workflow runs immediately rather than being queued to run at a later time.
* **Can run in the security context of the logged on user or owner of the workflow. However, workflows set to run on-demand always run under the security context of the logged on user**.
* Can’t contain any delay or wait activities.
* Only logs errors, and only when logging is enabled.
* Executes in the current transaction. All activities in the workflow and any child workflows, except asynchronous workflows, are part of a single transaction. Asynchronous child workflows are queued and execute in a separate transaction.
* Can be converted to asynchronous workflow and back to real-time.