AUTODESK. UNIVERSITY

MFG226291

Escalating Your Workflow in a Timely Manner

Pete Markovic
IMAGINiT Technologies – Sr. Solutions Consultant

Learning Objectives

- Learn how to capitalize on the Escalation function in a workflow to monitor other records
- Understand how to control a workspaces workflow from a another workspace
- Demonstrate scripting techniques to keep records from different workspaces, 'In Sync'
- Learn how to apply the built-in Escalation functionality for processes that have a timed, monitored workflow

Description

A common business need is to monitor a workflow and take action when certain criteria has been met. With Fusion Lifecycle, we can capitalize on the Escalation functionality in a workflow to monitor a record and perform an action, when the criteria is met. Utilizing a "timer" workspace is an efficient way to accomplish this for types of records, such as supplier audits, calibration records or maintenance items.

Speaker(s)

Pete Markovic is a Sr. Solutions Consultant with IMAGINiT Technologies specializing in PLM/PDM systems. He has over 25 years of experience working in various industries, including Construction Equipment, Automotive, Aerospace, Consumer Products, High Tech Electronics and Medical Device Manufacturing. Pete has been working with CAD/PLM/PDM software since 1992. His first PDM tool being Unigraphics IMAN, managing Unigraphics CAD data. Throughout his career, Pete has been involved with and has managed many PLM/PDM software implementations, including Autodesk Fusion Lifecycle, Autodesk Vault, Teamcenter, Windchill PDMLink, Metaphase, and PTC Intralink.

Pete specializes in working across multiple business areas, analyzing and streamlining their processes to adapt them to a PLM/PDM environment. Pete especially enjoys developing efficient data management processes/workflows and solutions that address the challenges and business issues faced by manufacturing companies.

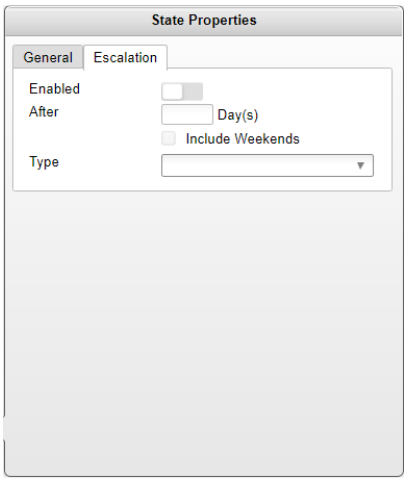
Timing is everything

A common part of automating workflows involves performing actions on a defined schedule. In Fusion Lifecycle we can execute scripts that do some sort of action when a user creates a record or edits a record. We can also do run an action script when a user executes a workflow transition. When we need to run an action script without user involvement, we can utilize the Escalation functionality in Fusion Lifecycle.

What is Escalation?

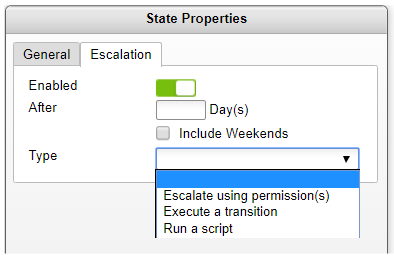
An often-overlooked piece of functionality in Fusion Lifecycle (FLC) is the escalation option we can apply at a workflow state. It appears simple enough, but is very powerful and it is not always understood how it can be applied.

When we place a new workflow state or double-click an existing workflow state, the State Properties dialog box will open. This dialog is where we find the Escalation tab.



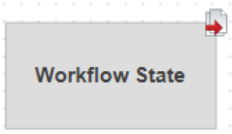
ESCALATION DIALOG

Once we enable Escalation, we can then select from three types. (Escalate using permissions(s), Execute a transition, Run a script)



ESCALATION TYPES

The escalation types are straightforward in that they do what their labels say. However, what we can do when we use the 'Execute a transition' type, on a looping transition, is not obvious. This technique is what enables us to repeat a script on a defined interval. The 'Run a script' may seem similar, but this will run a script one time and not repeat. Transitioning out of the workflow state and back to the same state is the trick (looping transition).



WORKFLOW STATE WITH ESCALATION ENABLED

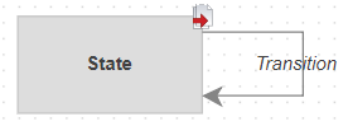
The looping transition

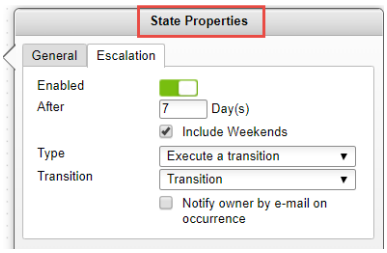
The looping transition should not be new to you as it is a technique commonly used in FLC, especially for Approvals. Often, it is shown as a single line pointing back to the workflow state. When this transition executes, it returns to the state it came from.

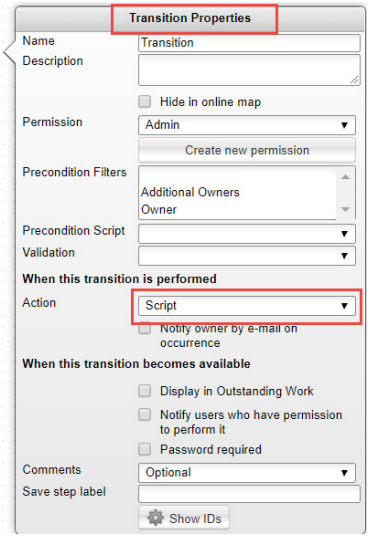
LOOPING APPROVAL TRANSITION

Combining the looping transition with escalation gives us the ability to repeat a script on a defined interval.

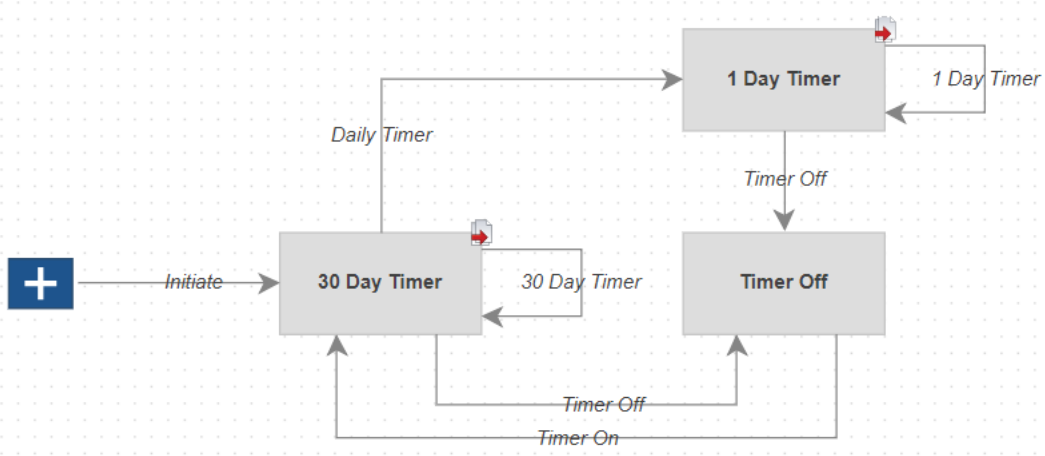
In this example, the transition will fire every 7 days. In the transition, we execute an Action script. Our Action script executes every 7 days until we transition out of the workflow state.







Now that we understand this concept, we can begin to apply it in our FLC environment. Depending on your business needs, the implementation of this technique will vary. Where there are long lead-time workflows (i.e. Supplier Audits, Scheduled Maintenance etc.) a stepped timer is a great approach.



MULTI-STEP TIMER WORKFLOW

For some processes, we may need to react to an event every 3 months, 6 month, a year or longer. Rather than having an escalation execute every day to evaluate our target, we can setup a multi-step timer approach.

In the above example, the escalation is set to run every 30 days and execute the 30 Day Timer transition. This transition runs an action script that evaluates a field in another record. The field in the other record is a 'number of days' field that decreases each day. Once the number is less than 31, the 30 Day Timer action script will perform the Daily Timer transition and move the workflow to the 1 Day Timer state. This workflow state has an escalation frequency of 1 day and executes the 1 Day Timer transition, which runs a 1 Day Timer action script. This new action script monitors the same value on the watched record. Once that value reaches less than one, the 1 Day Timer script transitions the workflow of the other record. The same action script transitions the Timer Off transition to move the workflow into the Timer Off state.

30 Day Timer script

```
//Check time to next Maintenance Due
var watchedAsset = item.ASSOCIATED_ASSET;
var daysOut = watchedAsset.TIME_TO_NEXT_MAINTENANCE;

if (daysOut !== null){

    if (daysOut < 31){
        item.performWorkflowTransition(776, 'Auto Transitioned by the System'); //Daily Timer transition
    }
```

1 Day Timer script

```
//Check time to next Maintenance Due
var watchedAsset = item.ASSOCIATED_ASSET;
var daysOut = watchedAsset.TIME_TO_NEXT_MAINTENANCE;

if (daysOut !== null){

    if (daysOut < 1){
        watchedAsset.performWorkflowTransition(787, 'Auto Transitioned by the System'); //Transition watched record
        item.performWorkflowTransition(781, 'Auto Transitioned by the System'); //Timer Off transition
    }
```

In this example, we used 30 and 1 day escalation intervals for our solution. A single timer of 1 day or additional timers at other increments can be used to accommodate different variations in workflows. Good planning will help you determine how many escalation states you will need and their increments to satisfy your requirements.

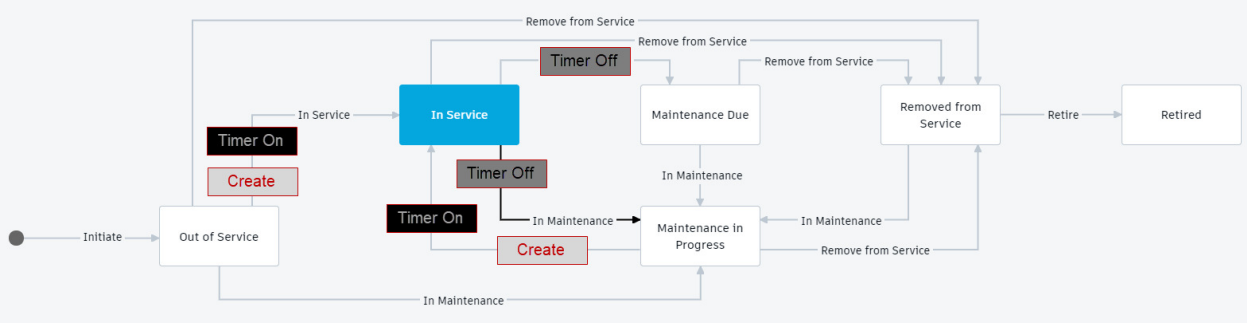
Workflow automation

Now that we understand how the Timers workspace works, we can put it into action. There is a 1:1 relationship of a timer to a watched record. The two records can be tied together by utilizing picklists of records referencing each other in their Item Details. The references should be created when the timer record is created by the watched record. Generally, this information will go into an Admin section on the watched record.

In the following example, we are going to use a Timers workspace to monitor records in an Assets workspace and determine when maintenance is due on our assets.

The first thing to do is create the asset workspace and develop its workflow. You will need to put fields in the Item Details of the asset that will allow you to capture the maintenance information. Fields like Maintenance Frequency, Last Maintenance, Next Maintenance and Time to Next Maintenance. The Time to Next Maintenance is the field we will be monitoring from the timer record.

Whenever our asset is not 'In Service', we do not need a timer monitoring the asset, so will turn it off by moving the timer in to Timer Off state in the timers' workflow. This diagram lays out when we need the timer and when it needs to be On or Off.



MARKED-UP WORKFLOW

In some cases where assets do not require maintenance, we do not need a timer record at all. Study your workflows and understand all of the potential paths to ensure you have everything covered. For example, any time we would put an asset in to 'In Service', we would check that a timer exists and if not, create one and turn it on.

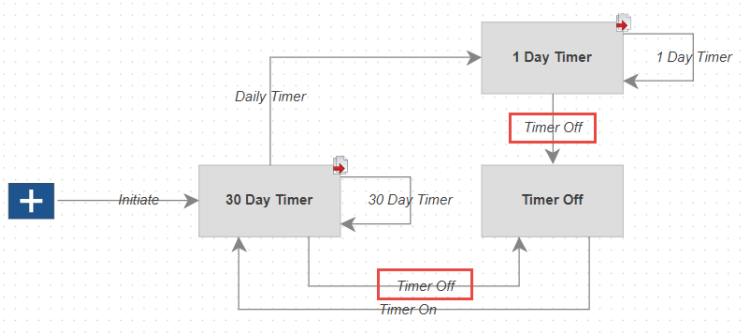
In our asset workflow, I have added a few functions to create a timer record, turn on the timer and turn off the timer.

```
function createTimer(){
    var newProperties = [];
    var timerNumber = item.NUMBER.slice(2);

    newProperties.NUMBER = 'Timer-'+timerNumber;
    newProperties.ASSET_NUMBER = item.NUMBER;
    newProperties.ASSET_NAME = item.DESCRIPTION;
    newProperties.ASSOCIATED_ASSET = item;

    var newItem = createNewItem('WS_ASSET_TIMERS', newProperties);

    if (newItem !== null){
        item.ASSOCIATED_TIMER = newItem;
    }
}
```

```
function timerOn(){
    if (assetTimer !== null){
        assetTimer.performWorkflowTransition(777, 'Auto transitioned by system');
    }
}
```

```
function timerOff(){
    if (assetTimer !== null){
        var timerWFState = assetTimer.descriptor.workflowState;
        if (timerWFState == '30 Day Timer'){
            assetTimer.performWorkflowTransition(779, 'Auto Transitioned by the System');
        }
        else if (timerWFState == '1 Day Timer'){
            assetTimer.performWorkflowTransition(782, 'Auto Transitioned by the System');
        }
    }
}
```

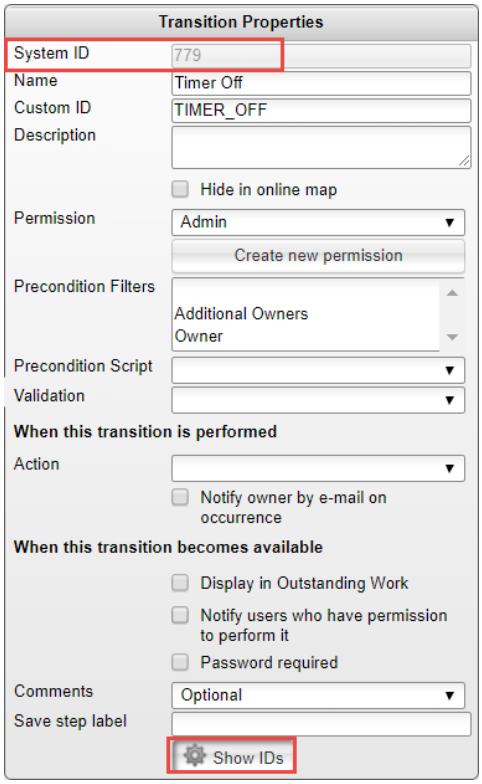
In our workflow, we have given the user an option to put an asset into maintenance before maintenance is due. Maybe a torque wrench was dropped and needs re-calibration. In this scenario, the timer could be in one of two states. Either the 30 day timer or 1 day timer, depending on how many days away it was from the next maintenance due. Since it could be in multiple states, we need to make sure our code can handle this scenario, so we are first looking up its current state to determine which transition to perform.



TIMER WORKFLOW

One of the biggest challenges with setting up the timers' workflow and the watched record workflow is ensuring you are in the correct state at the time you are moving its workflow. The transition has a System ID and this is hard coded in our scripts. If that transition is not available to the script, you will get an error. Diagraming out all the paths of the two workflows and visually seeing the coordination between two will help avoid these issue.

The item.performWorkflowTransition(779, 'Auto Transitioned by the System') is how we advance the workflow using a script. The transition number (779) is critical and can be found in the transition dialog. You have to turn on the 'Show IDs' in order to see it. Document the System IDs in your diagrams/documentation.

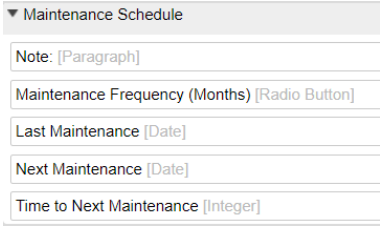


TRANSITION SYSTEM ID

Setting up the frequency

For the timers to function correctly, they will need to reference information in the watched record to determine if the defined criteria has been met.

In our asset workspace, we have a Maintenance Schedule section to reference from our timer records. The field 'Time to Next Maintenance' is evaluated in the action script on our escalation transition, on the associated timer. This field is an integer and is generated by a combination of the 'Maintenance Frequency' field and the 'Last Maintenance' field.

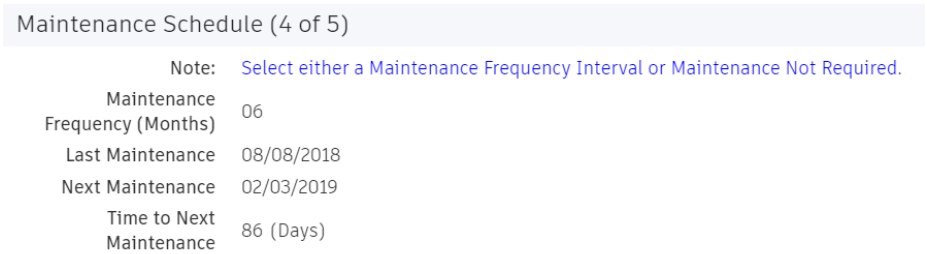


On the asset record, the user will select the appropriate Maintenance Frequency. An 'On Edit' script will calculate the 'Next Maintenance' date using the 'Maintenance Frequency' and 'Last Maintenance'. It is a good practice to make these fields required so the script can produce a result.

```
//Check for Maintenance Required
if (item.MAINTENANCE_FREQUENCY !== 'Scheduled Maintenance not Required'){
    if (item.MAINTENANCE_FREQUENCY !== null && item.LAST_MAINTENANCE !== null){
        var noMonths = item.MAINTENANCE_FREQUENCY;
        var noDays = noMonths*30;
        var newDate = DatePlusDays(item.LAST_MAINTENANCE, noDays);
        item.NEXT_MAINTENANCE = newDate;
    }
}
else if (item.MAINTENANCE_FREQUENCY === 'Scheduled Maintenance not Required'){
    item.NEXT_MAINTENANCE = null;
}
```

The 'Time to Next Maintenance' is calculated using a computed field (integer). The formula will determine how many days it is to the next maintenance due.

(DATEDIFF('DAY', LOCALTIMESTAMP, NEXT_MAINTENANCE))



RESULT

What to watch out for

One thing to watch out for is escalations can stop executing. This can happened when an escalation is scheduled to run while Fusion Lifecycle is down. If an escalation is scheduled to run and is missed, it will not restart on its own. An Admin can manually run the transition in the timer workspace to restart the escalation.

There are things we can do to minimize this risk. The first is to review when the escalations are running. What time of day do they run? If the last escalation ran at 2:30pm Monday and is scheduled to run every day, its next run will be at 2:30pm Tuesday. Scheduled system maintenance downtimes are generally in the late evenings, early morning hours. If you set your escalations to fire during business hours, it is unlikely they would try to run during downtime.

For some companies, it is critical that if the timers do miss an escalation, it is caught very soon after. An effective way of monitoring the timers and ensuring they are on schedule is to create an FLC report of the timers' workspace.

To do so, include the Current State and the Last Action Date.

In this example, the timers' workspace only has a 1 day escalation state (not multi-step), so the escalations fire every day. In the report, you will want to see the Last Action Date value be today or yesterday (in case it has not ran yet for the current day). The following results indicate the timers are all on schedule.

Report:Running Asset Timers

Report result : 101 rows

Number	Associated Asset	Last Action Date	Current State
000001	A-00019 /	11/09/2018	Timer On
000004	A-00006 /	11/09/2018	Timer On
000037	A-00066 /	11/09/2018	Timer On
000038	A-00022 /	11/09/2018	Timer On
000039	A-00023 /	11/09/2018	Timer On
000045	A-00007 /	11/09/2018	Timer On
000046	A-00015 /	11/09/2018	Timer On
000050	A-00013 /	11/09/2018	Timer On
000051	A-00014 /	11/09/2018	Timer On
000087	A-00026 /	11/09/2018	Timer On
000088	A-00030 /	11/09/2018	Timer On
000090	A-00106 /	11/09/2018	Timer On
000091	A-00024 /	11/09/2018	Timer On

TIMERS REPORT

Another technique for keeping the timers running is to have cascading escalation intervals. If a record does not have an action required for 6 months, running the escalation every day is not necessary. We can check every 30 days and transition to a 1 day interval, once we are within 30 days of action required. The less often the escalations run, the less chance of them trying to run when the system is down.

Conclusion

We are always looking for ways to be more efficient and ensure processes are followed and effective. When we have business processes that involve an action to occur at a certain time or frequency, they can difficult to manage. Utilizing the 'out of the box' functionality of escalations and combing them with transitions and action scripts, we can better manage these processes. This is one example of a process (Asset Maintenance) that can benefit from this technique. The process for applying this technique would be similar for other processes.