**Synchronize Equipment Counter with Line During End Job**

Taki Guan

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Changes |
| 1.0 | Taki Guan | 2019-4-15 | Create Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Version History 1](#_Toc4068331)

[Purpose 1](#_Toc4068332)

[Procedure 2](#_Toc4068333)

[Architecture 7](#_Toc4068334)

[Reference Link 7](#_Toc4068335)

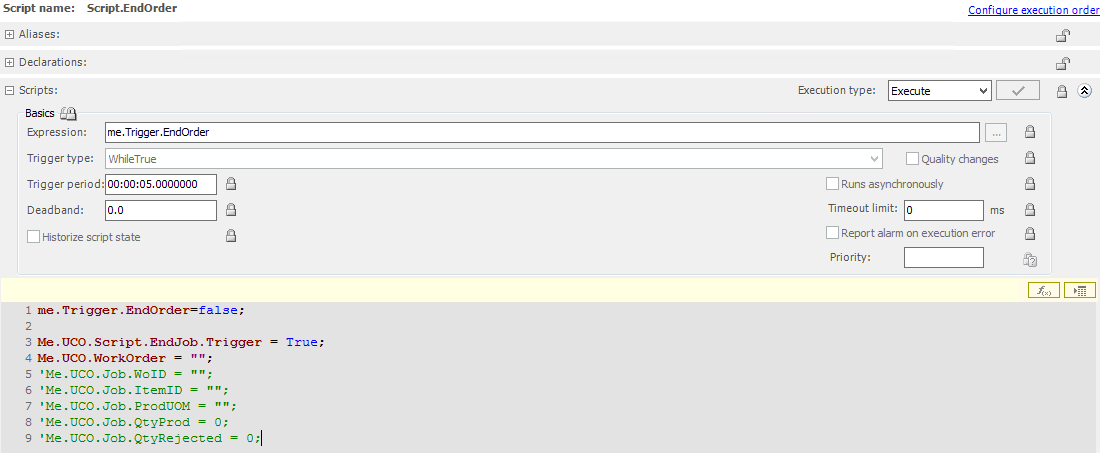
# Purpose

Sometimes we find the production equipment counter quantity is not synchronized with line quantity.

This scenario usually happens during ending order. So, we need to check what happens during ending order and what’s the problem.

# Procedure

**[$ngsfrEquipment]**



When [me.Trigger.EndOrder] this attribute is been set to true, this script [Script.EndOrder] will start to work.

me.Trigger.EndOrder = false;

Me.UCO.Script.EndJob.Trigger = True;

Me.UCO.WorkOrder = "";

*'Me.UCO.Job.WoID = "";*

*'Me.UCO.Job.ItemID = "";*

*'Me.UCO.Job.ProdUOM = "";*

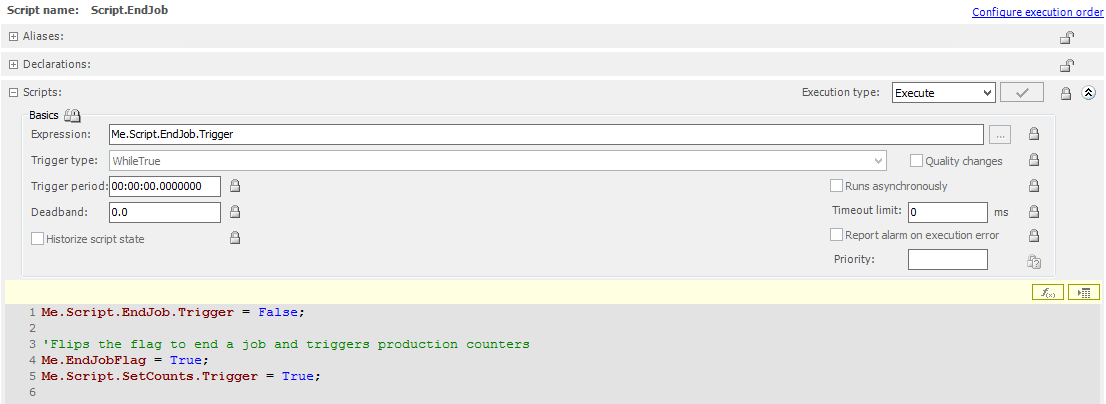
*'Me.UCO.Job.QtyProd = 0;*

*'Me.UCO.Job.QtyRejected = 0;*

You can see here I have commented some UCO attribute reset lines. I’ll explain why I make changes here and why this reset lines would make difference between line and equipment.

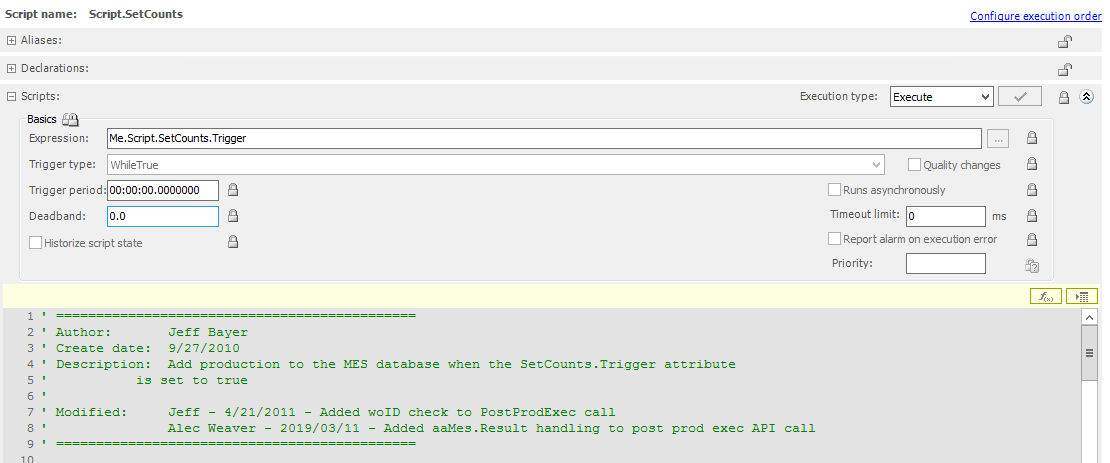
[Me.UCO.Script.EndJob.Trigger] is set to true here. Then we go to [ngsfrUtilizationCapability] to check what will happen.

**[$ngsfrUtilizationCapability]**



In [Script.EndJob], if [Me.UCO.Script.EndJob.Trigger] equals to true, [Me.EndJobFlag] and [Me.Script.SetCounts.Trigger] will be set to true.

Then we move to [Script.SetCounts], this script will work when [Me.Script.SetCounts.Trigger] is been set to true.



I capture some script out from the editor. Now I’ll explain what’s the meaning of below script.

*'Check to make sure the MES Session is valid before adding any production or clearing the interval counter*

If (result.Success == true) then

    For i = 1 To Me.Counters.Total

        x.BindTo("Me.Counter" + i);

        xabs.BindTo("Me.Counter" + i + ".AddProdQtyAbs");

        xtrg.BindTo("Me.Counter" + i + ".AddProdQtyTrg");

        xreas.BindTo("Me.Counter" + i + ".SetItemId");

        If Me.Counters.Interval[i] > 0 Then

            xabs = Round(Me.Counters.Interval[i], 1);

            xtrg = True;

        Else

        EndIf;

*'If the entity is the production entity for the line, trigger the counters on the OCO*

        If Me.Debug.Counters == True Then LogMessage("Conversion data = " + xprodconvert + "," + MyContainer.Cfg.RateConvert\_Type + "," + xSKUFactor +"," + xSKUFactorSecondary ); endif;

        If Me.Container == ProdEnt AND Me.Container <> MyContainer.LineName Then

            entID = MyContainer.LineEntId;

*'Added 2013-03-04 by Chris Pirrera*

            If xprodconvert == 1 THEN

                If MyContainer.Cfg.RateConvert\_Type == 1 THEN

                    qty = Round(xSKUFactor \* Me.Counters.Interval[i], 1);

                ELSEIF MyContainer.Cfg.RateConvert\_Type == 2 THEN

                    qty = Round(xSKUFactorSecondary \* Me.Counters.Interval[i], 1);

                ENDIF;

            ELSE

                qty = Round(Me.Counters.Interval[i], 1);

            Endif;

*'           qty = Round(Me.Counters.Interval[i], 1);*

*'End code added*

            woID = Me.Job.WoID;

            operID = Me.Job.LineOperId;

            seqNo = Me.Job.LineSeqNo;

*'shiftStart = Me.Shift.CurShiftStartTimeLocal;'could be UTC, validate*

*'shiftID = Me.Shift.ShiftID;*

            itemID = Me.Job.SKU;

            reasCd = xreas;

*'Added by yahua on 2018-11-01 to log job status of line when end order*

            if Me.EndJobFlag then

                LogMessage(System.String.Format("End order, try to set counts for line,wo:{0},oper:{1},seqNo:{2},item:{3},reasCd:{4},qty:{5}",woID,operID,seqNo,itemID,reasCd,qty));

            endif;

*'end code added*

            If woID == "" Then

*'Don't call PostProd if not running an order*

            Else

                If qty <> 0 Then

                    If Me.Debug.Counters == True Then LogMessage("ADD PROD-"+Me.TagName+ " Logging Prod for Line - "+MyContainer.LineName+" - Prod Qty = "+qty); EndIf;

                    dim curCrew as string;

                    dim xCrew as indirect;

                    xCrew.BindTo(System.String.Format("{0}.Shift.CurrentCrew",MyContainer.LineName));

                    if IsGood(xCrew) then

                        curCrew=xCrew;

                    else

                        curCrew="";

                    endif;

*'aaMES.Prod.aaJobExec.AddProdPostExec(sessionID, userId, entID, qty, woID, operID, seqNo, shiftStart, shiftID, itemID, reasCd, nullString, nullString, nullString, nullString, nullObject, nullObject, nullObject, nullString, nullObject, nullString, nullString, nullString, nullString, nullString, nullDBString, nullDBString, nullDBString, nullDBString, nullString);*

*' MES Result error handling added by Alec Weaver 2019/03/11*

                    mesResult = aaMES.Prod.aaJobExec.AddProdPostExec(sessionID,"Default Background User", entID, qty, woID, operID, seqNo, shiftStart, shiftID, itemID, reasCd, nullString, nullString, nullString, nullString, nullObject, nullObject, nullObject, nullString, nullObject, nullString, nullString, nullString, nullString, nullString, nullDBString, curCrew, nullDBString, nullDBString, nullString);

                    if mesResult.Success then

                        If Me.Debug.Counters == True Then LogMessage("Add Prod for Line Completed."); EndIf;

                    else

                        LogError("Add Prod for Line Failed: " + mesResult.Exception.Message);

                    endif;

                EndIf;

            EndIf;

        EndIf;

        Me.Counters.Interval[i] = 0;

        If Me.EndJobFlag == True or me.Util.Runtime == True Then

*'If the job is ending, reset the no job counts*

            Me.Counters.NoJobCount[i] = 0;

        EndIf;

    Next;

    If Me.EndJobFlag == True Then

*'If the job is ending, trigger a job end on the UCO and clear the end job flag*

        Me.EndJobFlag = False;

        Me.ProdAttrs.EndJobCmd = True;

    EndIf;

Else

    LogMessage("MES session not found!");

EndIf;

This script will use store procedure [ngsfr\_S\_UCO\_Get\_Line\_Prod\_Data] to get data from [job\_exec] table. If this result is true, then set counter for equipment. And if the equipment is production equipment, then set counter for the line at the same time.

        If Me.Container == ProdEnt AND Me.Container <> MyContainer.LineName Then

            entID = MyContainer.LineEntId;

The most important assignment is as below. The script assign UCO’s [job.WoID] to variable [woID].

            woID = Me.Job.WoID;

            operID = Me.Job.LineOperId;

            seqNo = Me.Job.LineSeqNo;

*'shiftStart = Me.Shift.CurShiftStartTimeLocal;'could be UTC, validate*

*'shiftID = Me.Shift.ShiftID;*

            itemID = Me.Job.SKU;

            reasCd = xreas;

If we reset UCO’s [job.WoID] to empty in script [Script.EndOrder]. Then the line quantity can’t be correctly set here.

            If woID == "" Then

*'Don't call PostProd if not running an order*

            Else

This is why I comment below code in script [Script.EndOrder].

*'Me.UCO.Job.WoID = "";*

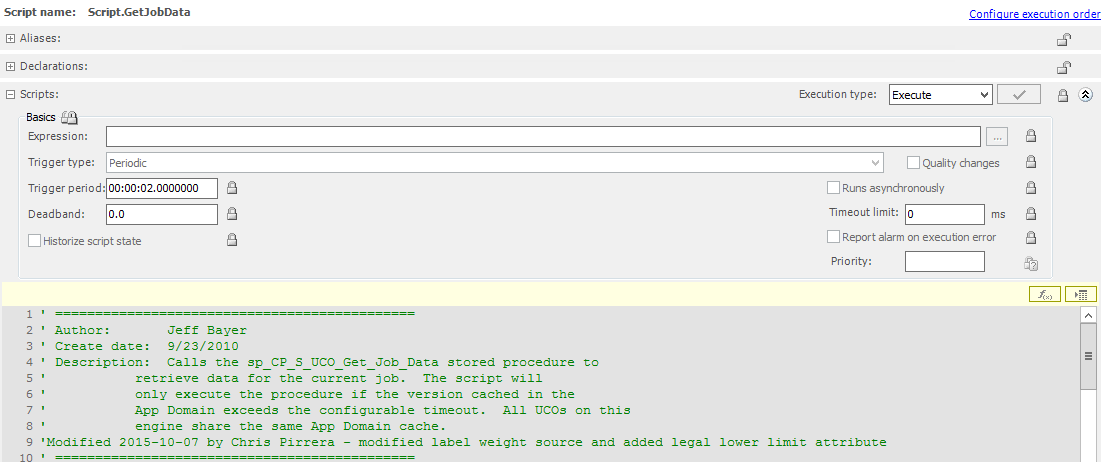
*'Me.UCO.Job.ItemID = "";*

*'Me.UCO.Job.ProdUOM = "";*

*'Me.UCO.Job.QtyProd = 0;*

*'Me.UCO.Job.QtyRejected = 0;*

And we don’t need to worry about when these values been reset. You can reference [Script.GetJobData] to find the reset logic. It run every 2 seconds.



The condition based on if there are records added to [job\_exec] table for specific entity. If there is no record added in specific timestep, these attributes will be reset.

If dataTable.Rows.Count > 0 Then

    If Me.Debug == True Then LogMessage("DataTable is not empty"); EndIf;

*'Select the row entity name from the table*

    selectRow = dataTable.Select("ent\_name='"+Me.Container+"'");

    If selectRow.Length > 0 Then

        If Me.Debug == True Then LogMessage("Entity "+Me.Container+" found in DataTable"); EndIf;

*'Populate Job Attributes*

        Me.Job.WoID = selectRow[1]["wo\_id"];

        Me.Job.OperID = selectRow[1]["oper\_id"];

        Me.Job.SeqNo = selectRow[1]["seq\_no"];

        Me.Job.ItemID = selectRow[1]["item\_id"];

        Me.Job.ItemDisplay = selectRow[1]["item\_display"];

        Me.Job.ItemDesc = selectRow[1]["item\_desc"];

        Me.Job.QtyReqd = selectRow[1]["qty\_reqd"];

        Me.Job.QtyProd = selectRow[1]["qty\_prod"];

        Me.Job.QtyRejected = selectRow[1]["qty\_rejected"];

        Me.Job.ProdUOM = selectRow[1]["prod\_uom"];

        Me.Job.EstProdRate = selectRow[1]["est\_prod\_rate"];

        Me.Job.BatchSize = selectRow[1]["batch\_size"];

        Me.Job.SKU = selectRow[1]["sku"];

        Me.Job.SKUDisplay = selectRow[1]["sku\_display"];

        Me.Job.SKUDesc = selectRow[1]["sku\_desc"];

        Me.Job.LineOperId = selectRow[1]["line\_oper\_id"];

        Me.Job.LineSeqNo = selectRow[1]["line\_seq\_no"];

        Me.Job.LineQtyProd = selectRow[1]["line\_qty\_prod"];

        Me.Job.LineQtyRejected = selectRow[1]["line\_qty\_rejected"];

        Me.Job.LineQtyReqd = selectRow[1]["line\_qty\_reqd"];

        Me.Job.LabelWeight = selectRow[1]["labelweight"];

        Me.Job.LegalLowerLimit = selectRow[1]["legalLL"];

        Me.Job.Running = True;

    Else

        If Me.Debug == True Then LogMessage("Entity "+Me.Container+" not found in DataTable"); EndIf;

*'Clear Job Attributes*

        Me.Job.WoID = "";

        Me.Job.OperID = "";

        Me.Job.SeqNo = "";

        Me.Job.ItemID = "";

        Me.Job.ItemDisplay = "";

        Me.Job.ItemDesc = "";

        Me.Job.QtyReqd = 0;

        Me.Job.QtyProd = 0;

        Me.Job.QtyRejected = 0;

        Me.Job.ProdUOM = "";

        Me.Job.EstProdRate = 0;

        Me.Job.BatchSize = 0;

        Me.Job.SKU = "";

        Me.Job.SKUDisplay = "";

        Me.Job.SKUDesc = "";

        Me.Job.LineOperId = "";

        Me.Job.LineSeqNo = "";

        Me.Job.LineQtyProd = 0;

        Me.Job.LineQtyRejected = 0;

        Me.Job.LineQtyReqd = 0;

        Me.Job.LabelWeight = 0;

        Me.Job.LegalLowerLimit = 0;

        Me.Job.Running = False;

    EndIf;

Else

*'No Job found*

    Me.Job.WoID = "";

    Me.Job.OperID = "";

    Me.Job.SeqNo = "";

    Me.Job.ItemID = "";

    Me.Job.ItemDisplay = "";

    Me.Job.ItemDesc = "";

    Me.Job.QtyReqd = 0;

    Me.Job.QtyProd = 0;

    Me.Job.QtyRejected = 0;

    Me.Job.ProdUOM = "";

    Me.Job.EstProdRate = 0;

    Me.Job.BatchSize = 0;

    Me.Job.SKU = "";

    Me.Job.SKUDisplay = "";

    Me.Job.SKUDesc = "";

    Me.Job.LineOperId = "";

    Me.Job.LineSeqNo = "";

    Me.Job.LineQtyProd = 0;

    Me.Job.LineQtyRejected = 0;

    Me.Job.LineQtyReqd = 0;

    Me.Job.LabelWeight = 0;

    Me.Job.LegalLowerLimit = 0;

    Me.Job.Running = False;

EndIf;

# Architecture

# Reference Link