Automatically Start or End WO on Equipment

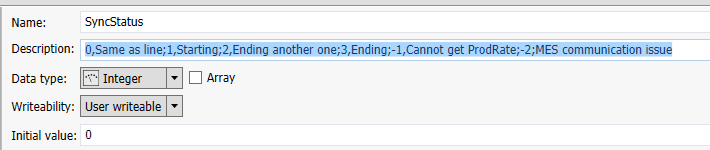
2019-01-16 By Yahua Chen

For Colgate Sanxiao Plant

## For OCO of line only have one job position

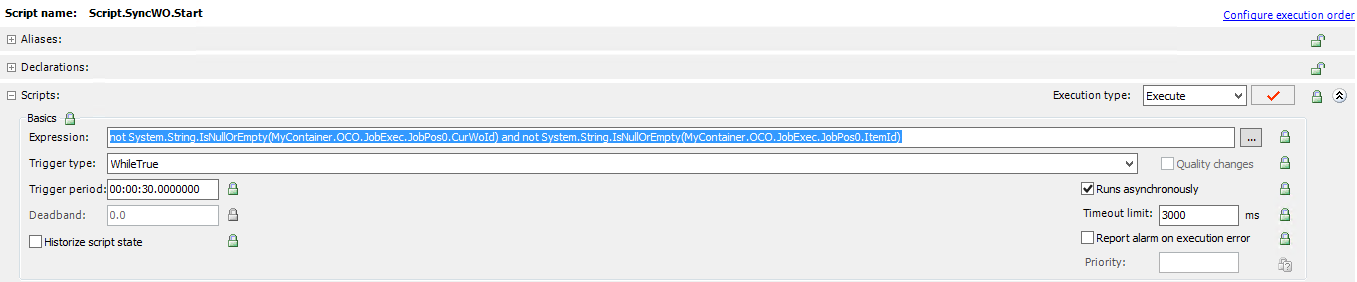
### 1.1 template $SXTufting

* Add attribute SyncStatus



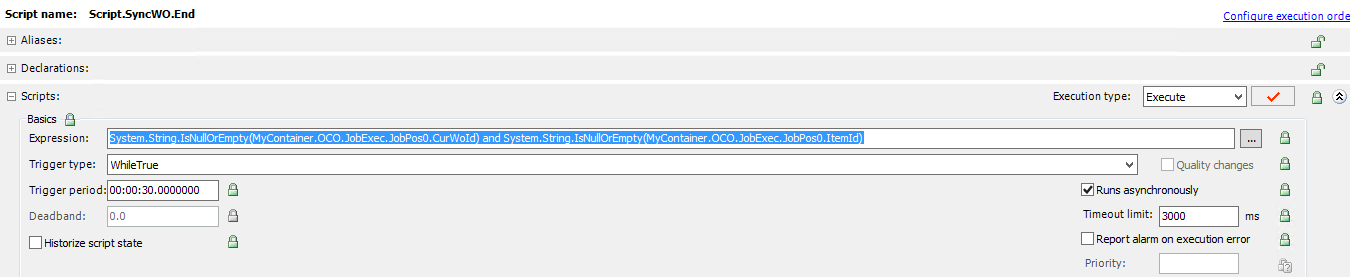
|  |  |
| --- | --- |
| Name | SyncStatus |
| Description | 0,Same as line;1,Starting;2,Ending another one;3,Ending;-1,Cannot get ProdRate;-2;MES communication issue |
| Data Type | Integer |
| Initial Value | 0 |

* New script “Script.SyncWO.Start”



|  |  |
| --- | --- |
| Name | Script.SyncWO.Start |
| Expression | not System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.CurWoId) and not System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.ItemId) |
| Trigger Type | While True |
| Trigger Period | 30 Seconds |
| Run as async | Checked, Time out: 3000 |
| Script | dim curTime as time;  dim lineWOStartTime as time;  dim lineWO as string;  dim lineItem as string;  curTime=now();  lineWOStartTime=MyContainer.OCO.JobExec.JobPos0.CurWoId.Time;  lineWO=MyContainer.OCO.JobExec.JobPos0.CurWoId;  lineItem=MyContainer.OCO.JobExec.JobPos0.ItemId;  '  if curTime>lineWOStartTime.AddSeconds(40) then  if Me.UCO.Job.WoID<> lineWO then  if System.String.IsNullOrEmpty(Me.UCO.Job.WoID) then  'ready to start job  Dim clientSession As aaFactMES.aaClientSession;  Dim result As aaFactMES.Result;  result = aaFactMES.aaClientSession.GetInstance();  if result.Success then  clientSession = result.Value;  result = clientSession.GetDSBySQL(System.String.Format("select dbo.fn\_GetProdRate({0},'{1}') as ProdRate,dbo.fn\_GetItemSpare('{1}',1) as SKUFactor1,dbo.fn\_GetItemSpare('{1}',2) as SKUFactor2",Me.EntId,lineItem));  if result.Success then  dim ds as System.Data.DataSet;  ds=result.DataSet\_Value;  if ds<>null and ds.Tables(0).Rows.Count>0 then  dim prodRate as float;  dim skuFactor1 as float;  dim skuFactor2 as float;  prodRate=ds.Tables(0).Rows(0).Item("ProdRate");  skuFactor1=ds.Tables(0).Rows(0).Item("SKUFactor1");  skuFactor2=ds.Tables(0).Rows(0).Item("SKUFactor2");  if prodRate>0 then  dim batchSize as float;  if not Me.Cfg.RateConversion then  batchSize=Round(prodRate,0.01);  else  If Me.Cfg.RateConvert\_Type == 1 Then  batchSize=Round(prodRate/skuFactor1,0.01);  ElseIf Me.Cfg.RateConvert\_Type == 2 Then  batchSize=Round((prodRate/skuFactor1)/skuFactor2,0.01);  endif;  endif;  LogMessage(System.String.Format("Line WO ID:{0},try to start job on equipment...",MyContainer.OCO.JobExec.JobPos0.CurWoId));  Me.SyncStatus=1;  Me.UCO.WorkOrder=lineWO;  Me.UCO.Item=lineItem;  Me.UCO.Operation=MyContainer.OCO.JobExec.JobPos0.CurOperId;  Me.UCO.BatchSize=batchSize;  Me.UCO.RequiredQuantity=(batchSize/ MyContainer.OCO.JobExec.CreateJobAttrs.BatchSize)\* MyContainer.OCO.JobExec.JobPos0.QtyReqd;  Me.UCO.StartQuantity=(batchSize/ MyContainer.OCO.JobExec.CreateJobAttrs.BatchSize)\* MyContainer.OCO.JobExec.JobPos0.QtyReqd;  Me.UCO.ProdAttrs.StartJobCmd=true;  else  Me.SyncStatus=-1;  endif;  else  Me.SyncStatus=-1;  endif;  else  Me.SyncStatus=-2;  endif;  else  Me.SyncStatus=-2;  endif;  else  'another wo is running, need to stop it first  Me.SyncStatus=2;  LogMessage(System.String.Format("Line WO ID:{0},current equipment WO ID:{1}, try to stop job on equipment...",MyContainer.OCO.JobExec.JobPos0.CurWoId,Me.UCO.Job.WoID));  Me.UCO.ProdAttrs.EndJobCmd=true;  endif;  else  Me.SyncStatus=0;  endif;  endif; |

* New script “Script.SyncWO.End”

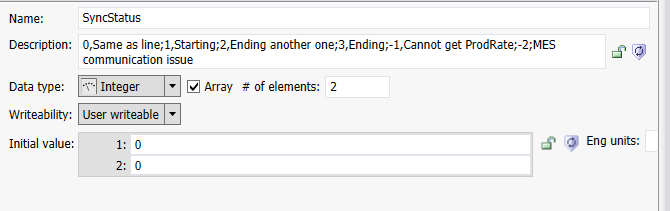


|  |  |
| --- | --- |
| Name | Script.SyncWO.End |
| Expression | System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.CurWoId) and System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.ItemId) |
| Trigger Type | While True |
| Trigger Period | 30 Seconds |
| Run as async | Checked, Time out: 3000 |
| Script | dim curTime as time;  dim lineWOStartTime as time;  dim lineWO as string;  dim lineItem as string;  curTime=now();  lineWOStartTime=MyContainer.OCO.JobExec.JobPos0.CurWoId.Time;  lineWO=MyContainer.OCO.JobExec.JobPos0.CurWoId;  lineItem=MyContainer.OCO.JobExec.JobPos0.ItemId;  '  if curTime>lineWOStartTime.AddSeconds(40) then  if not System.String.IsNullOrEmpty(Me.UCO.Job.WoID) then  LogMessage(System.String.Format("Line WO ID is empty,try to stop wo:{0} on equipment...",Me.UCO.Job.WoID));  Me.SyncStatus=3;  Me.UCO.ProdAttrs.EndJobCmd=true;  else  Me.SyncStatus=0;  endif;  endif; |

## For OCO of line have 2 job positions

### 2.1 template $ngsfrLineExt\_FCS.IM

* Add attribute SyncStatus



|  |  |
| --- | --- |
| Name | SyncStatus |
| Description | 0,Same as line;1,Starting;2,Ending another one;3,Ending;-1,Cannot get ProdRate;-2;MES communication issue |
| Data Type | Integer[2] |
| Initial Value | 0 |

* New script “Script.SyncWO.StartPos0”

|  |  |
| --- | --- |
| Name | Script.SyncWO.StartPos0 |
| Expression | not System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.CurWoId) and not System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.ItemId) |
| Trigger Type | While True |
| Trigger Period | 30 Seconds |
| Run as async | Checked, Time out: 3000 |
| Script | dim curTime as time;  dim lineWOStartTime as time;  dim lineWO as string;  dim lineItem as string;  dim lineOper as string;  dim lineReq as float;  dim curPos as integer;  curPos=0;  curTime=now();  lineWOStartTime=MyContainer.OCO.JobExec.JobPos0.CurWoId.Time;  lineWO=MyContainer.OCO.JobExec.JobPos0.CurWoId;  lineItem=MyContainer.OCO.JobExec.JobPos0.ItemId;  lineOper=MyContainer.OCO.JobExec.JobPos0.CurOperId;  lineReq=MyContainer.OCO.JobExec.JobPos0.QtyReqd;  '  if curTime>lineWOStartTime.AddSeconds(40) then  dim xmlJob as string;  xmlJob=System.String.Format("{0}{1}",Me.UCO.XML.JobInfo,Me.UCO.XML.JobInfo1);  dim xDoc as System.Xml.XmlDocument;  xDoc=new System.Xml.XmlDocument;  try  xDoc.LoadXml(xmlJob);  catch  xDoc=new System.Xml.XmlDocument;  endtry;  dim curEqWO as string;  dim xmlNode as System.Xml.XmlNode;  xmlNode=xDoc.SelectSingleNode(System.String.Format("/Ent/Job[@JobPos={0}]",curPos));  if xmlNode<>null then  curEqWO=xmlNode.Attributes.GetNamedItem("WoID").Value;  endif;  Dim clientSession As aaFactMES.aaClientSession;  Dim result As aaFactMES.Result;  result = aaFactMES.aaClientSession.GetInstance();  If result.Success then  clientSession = result.Value;  dim curSessId as integer;  dim curUser as string;  curSessId=clientSession.SessionId;  curUser="Default Background User";  if Me.Debug then LogMessage(System.String.Format("Session ID:{0},User:{1}",curSessId,curUser)); endif;  dim xmlExec as Fact.Common.xmlExecBuilder;  dim xmlParser as Fact.Common.XMLParser;  if curEqWO<> lineWO then  if System.String.IsNullOrEmpty(curEqWO) then  'ready to start job  dim sqlString as string;  sqlString=System.String.Format("select dbo.fn\_GetProdRate({0},'{1}') as ProdRate,dbo.fn\_GetItemSpare('{1}',1) as SKUFactor1,dbo.fn\_GetItemSpare('{1}',2) as SKUFactor2,dbo.fn\_GetJobBatchSize('{2}',0) as lineBatchSize",Me.EntId,lineItem,lineWO);  if Me.Debug then LogMessage(sqlString); endif;  result = clientSession.GetDSBySQL(sqlString);  if result.Success then  dim ds as System.Data.DataSet;  ds=result.DataSet\_Value;  if ds<>null and ds.Tables(0).Rows.Count>0 then  dim prodRate as float;  dim skuFactor1 as float;  dim skuFactor2 as float;  dim lineBatchSize as float;  prodRate=ds.Tables(0).Rows(0).Item("ProdRate");  skuFactor1=ds.Tables(0).Rows(0).Item("SKUFactor1");  skuFactor2=ds.Tables(0).Rows(0).Item("SKUFactor2");  lineBatchSize=ds.Tables(0).Rows(0).Item("lineBatchSize");    if prodRate>0 then  dim batchSize as float;  dim reqQty as float;  dim nextSeqNo as integer;  if not Me.Cfg.RateConversion then  batchSize=Round(prodRate,0.01);  else  If Me.Cfg.RateConvert\_Type == 1 Then  batchSize=Round(prodRate/skuFactor1,0.01);  ElseIf Me.Cfg.RateConvert\_Type == 2 Then  batchSize=Round((prodRate/skuFactor1)/skuFactor2,0.01);  endif;  endif;    Me.SyncStatus[curPos+1]=1;  reqQty=(batchSize/lineBatchSize)\*lineReq;  LogMessage(System.String.Format("Line WO ID:{0},try to start job on equipment...,reqQty:{1}",lineWO,reqQty));  Dim aaResult As aaMES.Result;  aaResult=aaMES.Prod.aaJob.GetNextSeqNo(curSessId,lineWO,lineOper,nextSeqNo);  if aaResult.Success then  if Me.Debug then LogMessage(nextSeqNo); endif;  else  LogMessage(aaResult.Exception.Message);  endif;  dim uomID as integer;  if Me.UCO.TargetJobProdRateUoM=="hours/batch" then  uomID=0;  elseif Me.UCO.TargetJobProdRateUoM=="minutes/batch" then  uomID=1;  elseif Me.UCO.TargetJobProdRateUoM=="seconds/batch" then  uomID=2;  elseif Me.UCO.TargetJobProdRateUoM=="batches/hour" then  uomID=3;  elseif Me.UCO.TargetJobProdRateUoM=="batches/minute" then  uomID=4;  elseif Me.UCO.TargetJobProdRateUoM=="batches/second" then  uomID=5;  else  uomID=3;  endif;    xmlExec = New Fact.Common.xmlExecBuilder(clientSession.SessionId, "OPERA\_StartJobOnEquip", Fact.Common.StdWriteCommands.Add);  xmlExec.AddParam("varEntID",Me.EntId);  xmlExec.AddParam("varPos",curPos);  xmlExec.AddParam("varWO",lineWO);  xmlExec.AddParam("varOper",lineOper);  xmlExec.AddParam("varSeqNo",nextSeqNo);  xmlExec.AddParam("varItem",lineItem);  xmlExec.AddParam("varQtyReqd",reqQty);  xmlExec.AddParam("varBatchSize",batchSize);  xmlExec.AddParam("varProdRate",prodRate);  xmlExec.AddParam("varProdUom",uomID);  xmlExec.AddParam("varUser",curUser);  'If Me.Debug == True Then LogMessage(xmlExec.XML); EndIf;  LogMessage(xmlExec.XML);  clientSession.SendXMLCmd(xmlExec.XML, xmlParser);  else  Me.SyncStatus[curPos+1]=-1;  endif;  else  Me.SyncStatus[curPos+1]=-1;  endif;  else  Me.SyncStatus[curPos+1]=-2;  endif;  else  'another wo is running, need to stop it first  Me.SyncStatus[curPos+1]=2;  LogMessage(System.String.Format("Pos:{0},Line WO ID:{1},current equipment WO ID:{2}, try to stop job on equipment...",curPos,lineWO,curEqWO));  xmlExec = New Fact.Common.xmlExecBuilder(clientSession.SessionId, "OPERA\_EndJobOnEquip", Fact.Common.StdWriteCommands.Add);  xmlExec.AddParam("varEntID",Me.EntId);  xmlExec.AddParam("varPos",curPos);  If Me.Debug == True Then LogMessage(xmlExec.XML); EndIf;  clientSession.SendXMLCmd(xmlExec.XML, xmlParser);  endif;  else  Me.SyncStatus[curPos+1]=0;  endif;  else  Me.SyncStatus[curPos+1]=-2;  endif;  endif; |

* New Script “Script.SyncWO.StartPos1”

|  |  |
| --- | --- |
| Name | Script.SyncWO.StartPos1 |
| Expression | not System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos1.CurWoId) and not System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos1.ItemId) |
| Trigger Type | While True |
| Trigger Period | 30 Seconds |
| Run as async | Checked, Time out: 3000 |
| Script | dim curTime as time;  dim lineWOStartTime as time;  dim lineWO as string;  dim lineItem as string;  dim lineOper as string;  dim lineReq as float;  dim curPos as integer;  curPos=1;  curTime=now();  lineWOStartTime=MyContainer.OCO.JobExec.JobPos1.CurWoId.Time;  lineWO=MyContainer.OCO.JobExec.JobPos1.CurWoId;  lineItem=MyContainer.OCO.JobExec.JobPos1.ItemId;  lineOper=MyContainer.OCO.JobExec.JobPos1.CurOperId;  lineReq=MyContainer.OCO.JobExec.JobPos1.QtyReqd;  '  if curTime>lineWOStartTime.AddSeconds(40) then  dim xmlJob as string;  xmlJob=System.String.Format("{0}{1}",Me.UCO.XML.JobInfo,Me.UCO.XML.JobInfo1);  dim xDoc as System.Xml.XmlDocument;  xDoc=new System.Xml.XmlDocument;  try  xDoc.LoadXml(xmlJob);  catch  xDoc=new System.Xml.XmlDocument;  endtry;  dim curEqWO as string;  dim xmlNode as System.Xml.XmlNode;  xmlNode=xDoc.SelectSingleNode(System.String.Format("/Ent/Job[@JobPos={0}]",curPos));  if xmlNode<>null then  curEqWO=xmlNode.Attributes.GetNamedItem("WoID").Value;  endif;  Dim clientSession As aaFactMES.aaClientSession;  Dim result As aaFactMES.Result;  result = aaFactMES.aaClientSession.GetInstance();  If result.Success then  clientSession = result.Value;  dim curSessId as integer;  dim curUser as string;  curSessId=clientSession.SessionId;  curUser="Default Background User";  if Me.Debug then LogMessage(System.String.Format("Session ID:{0},User:{1}",curSessId,curUser)); endif;  dim xmlExec as Fact.Common.xmlExecBuilder;  dim xmlParser as Fact.Common.XMLParser;  if curEqWO<> lineWO then  if System.String.IsNullOrEmpty(curEqWO) then  'ready to start job  dim sqlString as string;  sqlString=System.String.Format("select dbo.fn\_GetProdRate({0},'{1}') as ProdRate,dbo.fn\_GetItemSpare('{1}',1) as SKUFactor1,dbo.fn\_GetItemSpare('{1}',2) as SKUFactor2,dbo.fn\_GetJobBatchSize('{2}',0) as lineBatchSize",Me.EntId,lineItem,lineWO);  if Me.Debug then LogMessage(sqlString); endif;  result = clientSession.GetDSBySQL(sqlString);  if result.Success then  dim ds as System.Data.DataSet;  ds=result.DataSet\_Value;  if ds<>null and ds.Tables(0).Rows.Count>0 then  dim prodRate as float;  dim skuFactor1 as float;  dim skuFactor2 as float;  dim lineBatchSize as float;  prodRate=ds.Tables(0).Rows(0).Item("ProdRate");  skuFactor1=ds.Tables(0).Rows(0).Item("SKUFactor1");  skuFactor2=ds.Tables(0).Rows(0).Item("SKUFactor2");  lineBatchSize=ds.Tables(0).Rows(0).Item("lineBatchSize");    if prodRate>0 then  dim batchSize as float;  dim reqQty as float;  dim nextSeqNo as integer;  if not Me.Cfg.RateConversion then  batchSize=Round(prodRate,0.01);  else  If Me.Cfg.RateConvert\_Type == 1 Then  batchSize=Round(prodRate/skuFactor1,0.01);  ElseIf Me.Cfg.RateConvert\_Type == 2 Then  batchSize=Round((prodRate/skuFactor1)/skuFactor2,0.01);  endif;  endif;    Me.SyncStatus[curPos+1]=1;  reqQty=(batchSize/lineBatchSize)\*lineReq;  LogMessage(System.String.Format("Line WO ID:{0},try to start job on equipment...,reqQty:{1}",lineWO,reqQty));  Dim aaResult As aaMES.Result;  aaResult=aaMES.Prod.aaJob.GetNextSeqNo(curSessId,lineWO,lineOper,nextSeqNo);  if aaResult.Success then  if Me.Debug then LogMessage(nextSeqNo); endif;  else  LogMessage(aaResult.Exception.Message);  endif;  dim uomID as integer;  if Me.UCO.TargetJobProdRateUoM=="hours/batch" then  uomID=0;  elseif Me.UCO.TargetJobProdRateUoM=="minutes/batch" then  uomID=1;  elseif Me.UCO.TargetJobProdRateUoM=="seconds/batch" then  uomID=2;  elseif Me.UCO.TargetJobProdRateUoM=="batches/hour" then  uomID=3;  elseif Me.UCO.TargetJobProdRateUoM=="batches/minute" then  uomID=4;  elseif Me.UCO.TargetJobProdRateUoM=="batches/second" then  uomID=5;  else  uomID=3;  endif;    xmlExec = New Fact.Common.xmlExecBuilder(clientSession.SessionId, "OPERA\_StartJobOnEquip", Fact.Common.StdWriteCommands.Add);  xmlExec.AddParam("varEntID",Me.EntId);  xmlExec.AddParam("varPos",curPos);  xmlExec.AddParam("varWO",lineWO);  xmlExec.AddParam("varOper",lineOper);  xmlExec.AddParam("varSeqNo",nextSeqNo);  xmlExec.AddParam("varItem",lineItem);  xmlExec.AddParam("varQtyReqd",reqQty);  xmlExec.AddParam("varBatchSize",batchSize);  xmlExec.AddParam("varProdRate",prodRate);  xmlExec.AddParam("varProdUom",uomID);  xmlExec.AddParam("varUser",curUser);  'If Me.Debug == True Then LogMessage(xmlExec.XML); EndIf;  LogMessage(xmlExec.XML);  clientSession.SendXMLCmd(xmlExec.XML, xmlParser);  else  Me.SyncStatus[curPos+1]=-1;  endif;  else  Me.SyncStatus[curPos+1]=-1;  endif;  else  Me.SyncStatus[curPos+1]=-2;  endif;  else  'another wo is running, need to stop it first  Me.SyncStatus[curPos+1]=2;  LogMessage(System.String.Format("Pos:{0},Line WO ID:{1},current equipment WO ID:{2}, try to stop job on equipment...",curPos,lineWO,curEqWO));  xmlExec = New Fact.Common.xmlExecBuilder(clientSession.SessionId, "OPERA\_EndJobOnEquip", Fact.Common.StdWriteCommands.Add);  xmlExec.AddParam("varEntID",Me.EntId);  xmlExec.AddParam("varPos",curPos);  If Me.Debug == True Then LogMessage(xmlExec.XML); EndIf;  clientSession.SendXMLCmd(xmlExec.XML, xmlParser);  endif;  else  Me.SyncStatus[curPos+1]=0;  endif;  else  Me.SyncStatus[curPos+1]=-2;  endif;  endif; |

* New Script “Script.SyncWO.EndPos0”

|  |  |
| --- | --- |
| Name | Script.SyncWO.EndPos0 |
| Expression | System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.CurWoId) and System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos0.ItemId) |
| Trigger Type | While True |
| Trigger Period | 30 Seconds |
| Run as async | Checked, Time out: 3000 |
| Script | dim curTime as time;  dim lineWOStartTime as time;  dim lineWO as string;  dim lineItem as string;  dim curPos as integer;  curPos=0;  curTime=now();  lineWOStartTime=MyContainer.OCO.JobExec.JobPos0.CurWoId.Time;  lineWO=MyContainer.OCO.JobExec.JobPos0.CurWoId;  lineItem=MyContainer.OCO.JobExec.JobPos0.ItemId;  '  if curTime>lineWOStartTime.AddSeconds(40) then  dim xmlJob as string;  xmlJob=System.String.Format("{0}{1}",Me.UCO.XML.JobInfo,Me.UCO.XML.JobInfo1);  dim xDoc as System.Xml.XmlDocument;  xDoc=new System.Xml.XmlDocument;  try  xDoc.LoadXml(xmlJob);  catch  xDoc=new System.Xml.XmlDocument;  endtry;  dim curEqWO as string;  dim xmlNode as System.Xml.XmlNode;  xmlNode=xDoc.SelectSingleNode(System.String.Format("/Ent/Job[@JobPos={0}]",curPos));  if xmlNode<>null then  curEqWO=xmlNode.Attributes.GetNamedItem("WoID").Value;  endif;  if not System.String.IsNullOrEmpty(curEqWO) then  Dim clientSession As aaFactMES.aaClientSession;  Dim result As aaFactMES.Result;  result = aaFactMES.aaClientSession.GetInstance();  If result.Success then  clientSession = result.Value;  dim xmlExec as Fact.Common.xmlExecBuilder;  dim xmlParser as Fact.Common.XMLParser;  LogMessage(System.String.Format("Pos:{0},Line WO ID is empty,try to stop wo:{0} on equipment...",curPos,curEqWO));  Me.SyncStatus[curPos+1]=3;  xmlExec = New Fact.Common.xmlExecBuilder(clientSession.SessionId, "OPERA\_EndJobOnEquip", Fact.Common.StdWriteCommands.Add);  xmlExec.AddParam("varEntID",Me.EntId);  xmlExec.AddParam("varPos",curPos);  If Me.Debug == True Then LogMessage(xmlExec.XML); EndIf;  clientSession.SendXMLCmd(xmlExec.XML,xmlParser);  else  Me.SyncStatus[curPos+1]=-2;  endif;  else  Me.SyncStatus[curPos+1]=0;  endif;  endif; |

* New Script “Script.SyncWO.EndPos1”

|  |  |
| --- | --- |
| Name | Script.SyncWO.EndPos1 |
| Expression | System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos1.CurWoId) and System.String.IsNullOrEmpty(MyContainer.OCO.JobExec.JobPos1.ItemId) |
| Trigger Type | While True |
| Trigger Period | 30 Seconds |
| Run as async | Checked, Time out: 3000 |
| Script | dim curTime as time;  dim lineWOStartTime as time;  dim lineWO as string;  dim lineItem as string;  dim curPos as integer;  curPos=1;  curTime=now();  lineWOStartTime=MyContainer.OCO.JobExec.JobPos1.CurWoId.Time;  lineWO=MyContainer.OCO.JobExec.JobPos1.CurWoId;  lineItem=MyContainer.OCO.JobExec.JobPos1.ItemId;  '  if curTime>lineWOStartTime.AddSeconds(40) then  dim xmlJob as string;  xmlJob=System.String.Format("{0}{1}",Me.UCO.XML.JobInfo,Me.UCO.XML.JobInfo1);  dim xDoc as System.Xml.XmlDocument;  xDoc=new System.Xml.XmlDocument;  try  xDoc.LoadXml(xmlJob);  catch  xDoc=new System.Xml.XmlDocument;  endtry;  dim curEqWO as string;  dim xmlNode as System.Xml.XmlNode;  xmlNode=xDoc.SelectSingleNode(System.String.Format("/Ent/Job[@JobPos={0}]",curPos));  if xmlNode<>null then  curEqWO=xmlNode.Attributes.GetNamedItem("WoID").Value;  endif;  if not System.String.IsNullOrEmpty(curEqWO) then  Dim clientSession As aaFactMES.aaClientSession;  Dim result As aaFactMES.Result;  result = aaFactMES.aaClientSession.GetInstance();  If result.Success then  clientSession = result.Value;  dim xmlExec as Fact.Common.xmlExecBuilder;  dim xmlParser as Fact.Common.XMLParser;  LogMessage(System.String.Format("Pos:{0},Line WO ID is empty,try to stop wo:{0} on equipment...",curPos,curEqWO));  Me.SyncStatus[curPos+1]=3;  xmlExec = New Fact.Common.xmlExecBuilder(clientSession.SessionId, "OPERA\_EndJobOnEquip", Fact.Common.StdWriteCommands.Add);  xmlExec.AddParam("varEntID",Me.EntId);  xmlExec.AddParam("varPos",curPos);  If Me.Debug == True Then LogMessage(xmlExec.XML); EndIf;  clientSession.SendXMLCmd(xmlExec.XML,xmlParser);  else  Me.SyncStatus[curPos+1]=-2;  endif;  else  Me.SyncStatus[curPos+1]=0;  endif;  endif; |

## 3. Store procedure

* Function fn\_GetProdRate

|  |
| --- |
| USE [MESDB]  GO  /\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[fn\_GetProdRate] Script Date: 3/14/2019 10:25:52 AM \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  --- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Module Name - fn\_GetProdRate  -- Description - get prodRate of given ent,item  -- Author - Yahua Chen  --  -- Date Created - 2018-10-29  -- Version - 1.0  ----------------------------------------------------------------------------------------------------------------------------  -- Date\_Modified Modified\_By Ver Changes/Comments  -- 2018-10-29 Yahua 1.0 created  ----------------------------------------------------------------------------------------------------------------------------  ALTER FUNCTION [dbo].[fn\_GetProdRate](@varEntID int,@varItemID nvarchar(40)) returns float  AS  BEGIN  declare @rltVal float  if @varEntID is not null and @varItemID is not null  begin  select top 1 @rltVal= case when ISNUMERIC(ea.attr\_value)=1 then cast(ea.attr\_value as float) else null end  from item\_category\_link icl,category cg,ent\_attr ea,attr  where icl.item\_id=@varItemID and icl.category\_id=cg.category\_id and cg.category\_name like '%size'  and ea.ent\_id=@varEntID and ea.attr\_id=attr.attr\_id  and attr.attr\_desc='Prod\_Rate\_'+cg.category\_name  end  return isnull(@rltVal,0)  END |

* Function fn\_GetItemSpare

|  |
| --- |
| USE [MESDB]  GO  /\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[fn\_GetItemSpare] Script Date: 3/14/2019 10:27:00 AM \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  --- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Module Name - [fn\_GetItemSpare]  -- Description - get spare1,2,3,4 coulumn for given item  -- Author - Yahua Chen  --  -- Date Created - 2018-10-29  -- Version - 1.0  ----------------------------------------------------------------------------------------------------------------------------  -- Date\_Modified Modified\_By Ver Changes/Comments  -- 2018-10-29 Yahua 1.0 created  ----------------------------------------------------------------------------------------------------------------------------  ALTER FUNCTION [dbo].[fn\_GetItemSpare](@item\_id nvarchar(40),@spareId int) returns float  AS  BEGIN  declare @spareStr nvarchar(80),@rltVal float  select top 1 @spareStr=case @spareId when 1 then spare1 when 2 then spare2 when 3 then spare3 when 4 then spare4 else null end  from item where item\_id=@item\_id  set @rltVal=case when ISNUMERIC(@spareStr)=1 then cast(@spareStr as float) else null end  return isnull(@rltVal,1)  END |