**Quick Selection of Downtime**

Taki Guan

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Changes |
| 1.0 | Taki Guan | 2019-4-12 | Create Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Version History 1](#_Toc4068331)

[Purpose 1](#_Toc4068332)

[Procedure 2](#_Toc4068333)

[Architecture 2](#_Toc4068334)

[Reference Link 16](#_Toc4068335)

# Purpose

This file is to describe how to configure these two functions as below:

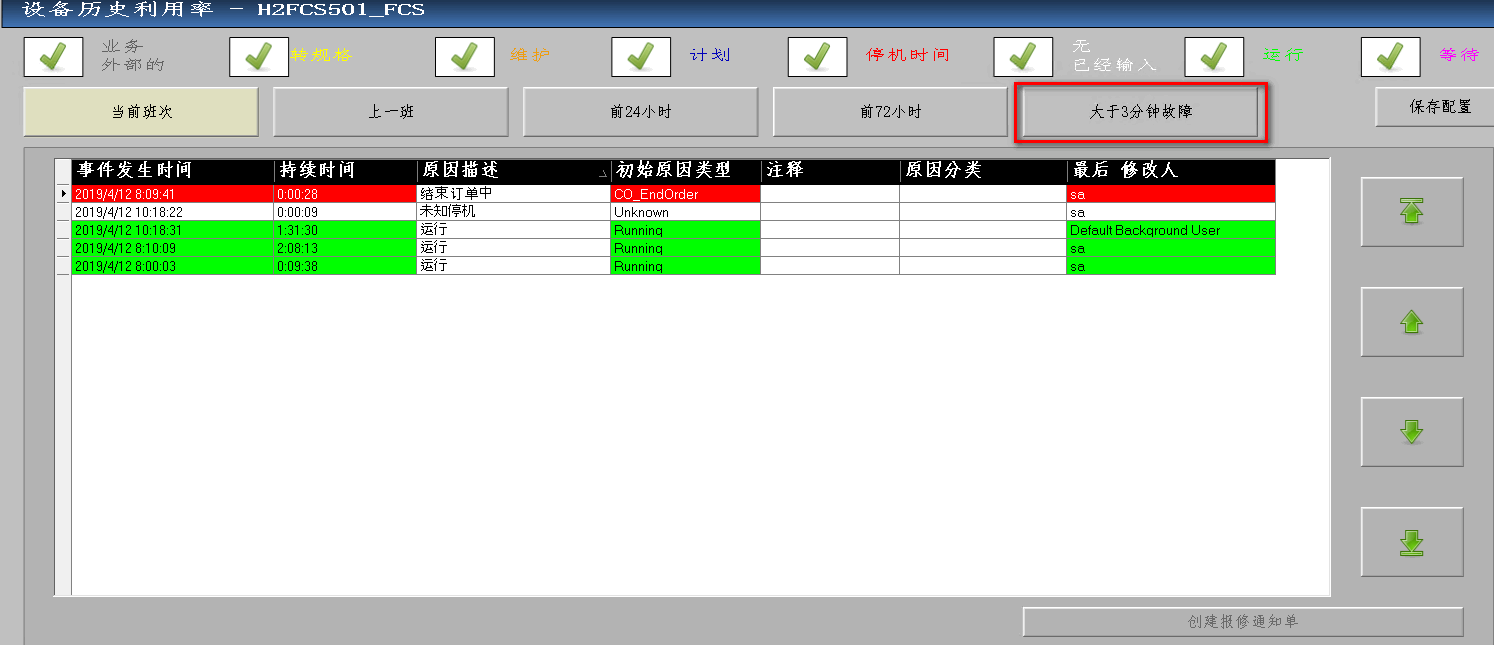
|  |
| --- |
| 1. Quick select unknown downtime reason over 3 min;  2. Show warning message if unknown downtime reason exists 30 min before shift change. |

This file will help other developers to find the store procedure and batch code generation logic.

# Procedure

**[operaDowntime\_History]**

In this graphic I added one button to quick select downtime over 3 min to help operator quick select all unknown downtime records to specific exact downtime reasons.



**[operaBtn\_Open\_StateHistory]**

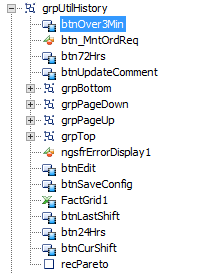
If there are unknown downtime records over 3 min, this button will blink. Operator can directly click on this button to quickly access all records over 3 min.

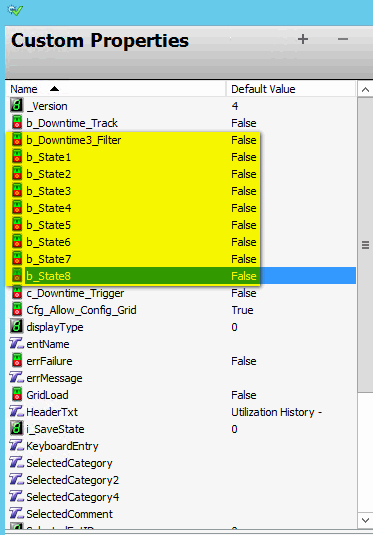


# Architecture

**[operaDowntime\_History]**

I added one button [btnOver3Min] to click to filter result.





We need to create custom properties:

[b\_Downtime3\_Filter]

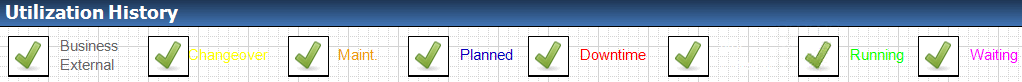
This is to track is the button is clicked or not.

[i\_SaveState]

This is to save the filter type before click this button.

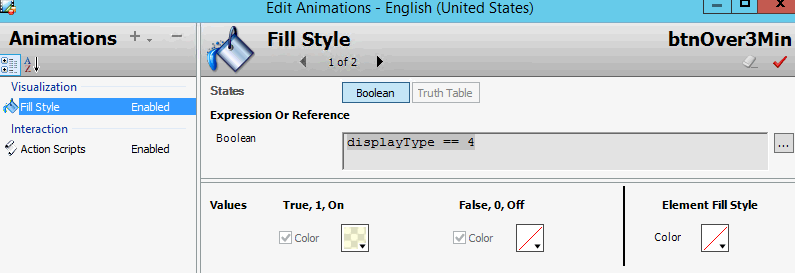
[b\_State1] – [b\_State8]

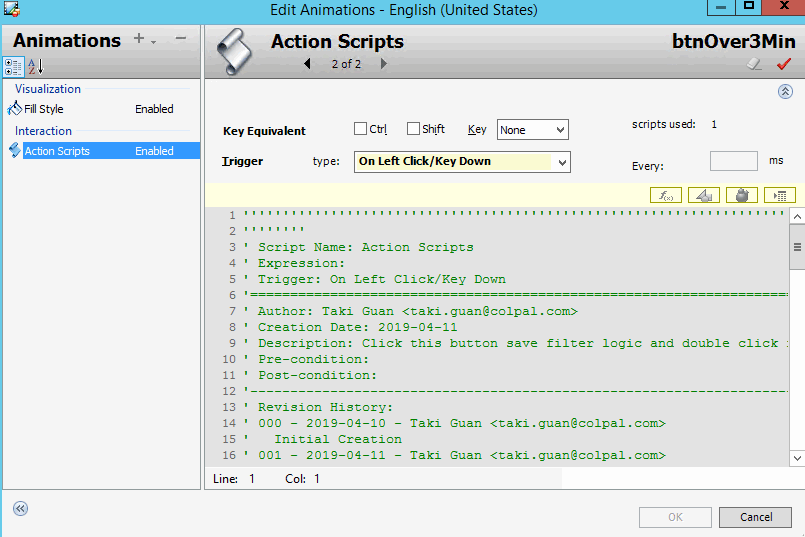
This is to save the state if below check box if checked or not before click the button.



After we create the graphic we need to design below logic to initialize the graphic.

I add one [Fill Style] animation for [btnOver3Min] button, when the button has been clicked, [displayType] will be set to 4 in [Action Script].





**[Action Scripts]**

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

*''''''''*

*' Script Name: Action Scripts*

*' Expression:*

*' Trigger: On Left Click/Key Down*

*'=======================================================================================*

*' Author: Taki Guan <taki.guan@colpal.com>*

*' Creation Date: 2019-04-11*

*' Description: Click this button save filter logic and double click reset filter logic*

*' Pre-condition:*

*' Post-condition:*

*'---------------------------------------------------------------------------------------*

*' Revision History:*

*' 000 - 2019-04-10 - Taki Guan <taki.guan@colpal.com>*

*' Initial Creation*

*' 001 - 2019-04-11 - Taki Guan <taki.guan@colpal.com>*

*' Add script description*

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

if(b\_Downtime3\_Filter == false) then

*' save configuration*

    i\_SaveState = displayType;

    b\_State1 = Filter\_Window.Wait\_Filter.Wait\_Checked.Visible;

    b\_State2 = Filter\_Window.Run\_Filter.Run\_Checked.Visible;

    b\_State3 = Filter\_Window.NE\_Filter.NE\_Checked.Visible;

    b\_State4 = Filter\_Window.DT\_Filter.DT\_Checked.Visible;

    b\_State5 = Filter\_Window.Planned\_Filter.Planned\_Checked.Visible;

    b\_State6 = Filter\_Window.Maint\_Filter.Maint\_Checked.Visible;

    b\_State7 = Filter\_Window.CO\_Filter.CO\_Checked.Visible;

    b\_State8 = Filter\_Window.BE\_Filter.BE\_Checked.Visible;

*' set configuration*

    displayType = 4;

    Filter\_Window.Wait\_Filter.Wait\_Checked.Visible = false;

    Filter\_Window.Run\_Filter.Run\_Checked.Visible = false;

    Filter\_Window.NE\_Filter.NE\_Checked.Visible = true;

    Filter\_Window.DT\_Filter.DT\_Checked.Visible = false;

    Filter\_Window.Planned\_Filter.Planned\_Checked.Visible = false;

    Filter\_Window.Maint\_Filter.Maint\_Checked.Visible = false;

    Filter\_Window.CO\_Filter.CO\_Checked.Visible = false;

    Filter\_Window.BE\_Filter.BE\_Checked.Visible = false;

    Filter\_Window.Wait\_Filter.Wait\_NotCheck.Visible = true;

    Filter\_Window.Run\_Filter.Run\_NotCheck.Visible = true;

    Filter\_Window.NE\_Filter.NE\_NotCheck.Visible = false;

    Filter\_Window.DT\_Filter.DT\_NotCheck.Visible = true;

    Filter\_Window.Planned\_Filter.Planned\_NotCheck.Visible = true;

    Filter\_Window.Maint\_Filter.Maint\_NotCheck.Visible = true;

    Filter\_Window.CO\_Filter.CO\_NotCheck.Visible = true;

    Filter\_Window.BE\_Filter.BE\_NotCheck.Visible = true;

*' set filter to true*

    b\_Downtime3\_Filter = true;

    trgRefreshData = True;

elseif b\_Downtime3\_Filter == true then

*'load configration*

    displayType = i\_SaveState;

    Filter\_Window.Wait\_Filter.Wait\_Checked.Visible = b\_State1;

    Filter\_Window.Run\_Filter.Run\_Checked.Visible = b\_State2;

    Filter\_Window.NE\_Filter.NE\_Checked.Visible = b\_State3;

    Filter\_Window.DT\_Filter.DT\_Checked.Visible = b\_State4;

    Filter\_Window.Planned\_Filter.Planned\_Checked.Visible = b\_State5;

    Filter\_Window.Maint\_Filter.Maint\_Checked.Visible = b\_State6;

    Filter\_Window.CO\_Filter.CO\_Checked.Visible = b\_State7;

    Filter\_Window.BE\_Filter.BE\_Checked.Visible = b\_State8;

    Filter\_Window.Wait\_Filter.Wait\_NotCheck.Visible = b\_State1 ^ 1;

    Filter\_Window.Run\_Filter.Run\_NotCheck.Visible = b\_State2 ^ 1;

    Filter\_Window.NE\_Filter.NE\_NotCheck.Visible = b\_State3 ^ 1;

    Filter\_Window.DT\_Filter.DT\_NotCheck.Visible = b\_State4 ^ 1;

    Filter\_Window.Planned\_Filter.Planned\_NotCheck.Visible = b\_State5 ^ 1;

    Filter\_Window.Maint\_Filter.Maint\_NotCheck.Visible = b\_State6 ^ 1;

    Filter\_Window.CO\_Filter.CO\_NotCheck.Visible = b\_State7 ^ 1;

    Filter\_Window.BE\_Filter.BE\_NotCheck.Visible =b\_State8 ^ 1;

*' reset filter*

    b\_Downtime3\_Filter = false;

    trgRefreshData = true;

endif;

If the button is been clicked once [trgRefreshData] is set to true, then will triger script [RefreshData]. It will use store procedure [OPERA\_UI\_GetUtilHistory] to retrieve data.

[OPERA\_UI\_GetUtilHistory]

I make some changes for this store precedure:

* Add IF statement for filter type 4
* Add WHERE statement for filter type 4

*-- Added 2019-04-12 by Taki Guan*

*-- If @type = 4, still use current shift start date*

    IF @type = 4 *-- Current Shift*

    BEGIN

        SELECT @start = cur\_shift\_start\_time\_local

        FROM ent WITH (NOLOCK)

        WHERE ent\_id = @ent\_id;

        SET @end = GETDATE();

    END

*-- End added 2019-04-12 by Taki Guan*

*-- Modified 2019-04-12 by Taki Guan*

*-- If @type = 4 then select the result of "Not Entered" Downtime and Make Duration > 180 s*

*-- Else follow original logic*

    IF @type = 4

        BEGIN

            SELECT util\_log.log\_id,

                    util\_log.ent\_id,

                    ent.ent\_name,

                    shift.shift\_desc,

                    util\_state.state\_desc,

                    util\_state.color,

                    util\_reas.reas\_desc,

                    dbo.fnDurationToHHMMSS(duration) As duration,

                    util\_log.reas\_cd,

                    util\_log.event\_time\_local,

                    util\_log.reas\_pending,

                    util\_log.raw\_reas\_cd,

                    util\_log.runtime,

                    util\_log.downtime,

                    util\_log.fixedtime,

                    util\_log.vartime,

                    COALESCE(util\_log.comments,'') As comments,

                    util\_log.shift\_start\_local,

                    util\_log.last\_edit\_by,

                    util\_log.last\_edit\_at,

*-- added by Chris Pirrera on 2013-03-20*

                    util\_log.category3,

                    util\_log.category2,

*-- end code added*

*-- added by Greg McGraw on 2016-12-06*

                    util\_log.category4

*-- end code added*

            FROM util\_log WITH (NOLOCK)

                JOIN ent WITH (NOLOCK) ON ent.ent\_id = util\_log.ent\_id

                JOIN shift WITH (NOLOCK) ON shift.shift\_id = util\_log.shift\_id

                JOIN util\_state WITH (NOLOCK) ON util\_state.state\_cd = util\_log.state\_cd

                JOIN util\_reas WITH (NOLOCK) ON util\_reas.reas\_cd = util\_log.reas\_cd

            WHERE util\_log.ent\_id = @ent\_id

                AND event\_time\_local >= @start

                AND event\_time\_local <= @end

*-- added 2013-10-31 by chris pirrera*

                AND (util\_state.state\_desc IN (SELECT items FROM dbo.ngsfr\_fn\_S\_Split\_List(@filter)) OR @filter IS NULL)

*-- added 2019-04-10 by Taki Guan*

                AND duration >= 180

*-- end code added*

            ORDER BY event\_time\_local DESC;

        END

    ELSE

        BEGIN

            SELECT util\_log.log\_id,

                    util\_log.ent\_id,

                    ent.ent\_name,

                    shift.shift\_desc,

                    util\_state.state\_desc,

                    util\_state.color,

                    util\_reas.reas\_desc,

                    dbo.fnDurationToHHMMSS(duration) As duration,

                    util\_log.reas\_cd,

                    util\_log.event\_time\_local,

                    util\_log.reas\_pending,

                    util\_log.raw\_reas\_cd,

                    util\_log.runtime,

                    util\_log.downtime,

                    util\_log.fixedtime,

                    util\_log.vartime,

                    COALESCE(util\_log.comments,'') As comments,

                    util\_log.shift\_start\_local,

                    util\_log.last\_edit\_by,

                    util\_log.last\_edit\_at,

*-- added by Chris Pirrera on 2013-03-20*

                    util\_log.category3,

                    util\_log.category2,

*-- end code added*

*-- added by Greg McGraw on 2016-12-06*

                    util\_log.category4

*-- end code added*

            FROM util\_log WITH (NOLOCK)

                JOIN ent WITH (NOLOCK) ON ent.ent\_id = util\_log.ent\_id

                JOIN shift WITH (NOLOCK) ON shift.shift\_id = util\_log.shift\_id

                JOIN util\_state WITH (NOLOCK) ON util\_state.state\_cd = util\_log.state\_cd

                JOIN util\_reas WITH (NOLOCK) ON util\_reas.reas\_cd = util\_log.reas\_cd

            WHERE util\_log.ent\_id = @ent\_id

                AND event\_time\_local >= @start

                AND event\_time\_local <= @end

*-- added 2013-10-31 by chris pirrera*

                AND (util\_state.state\_desc IN (SELECT items FROM dbo.ngsfr\_fn\_S\_Split\_List(@filter)) OR @filter IS NULL)

*-- end code added*

            ORDER BY event\_time\_local DESC;

        END

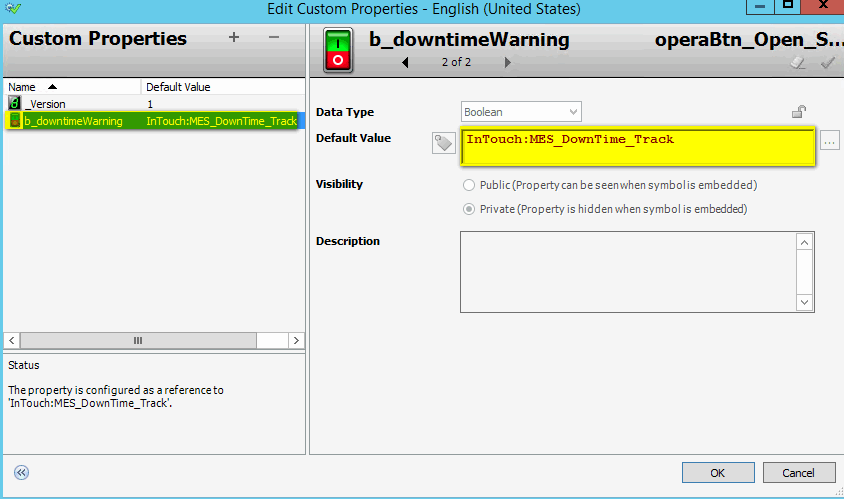
*-- End Modified 2019-04-12 by Taki Guan*

Then I finished the 1st requriement, when this button been clicked. All unknow downtime records over 3 min will be retrived. If it is been clicked again, it will restore all selection.

**[operaBtn\_Open\_StateHistory]**

Now we start to complete 2nd requirement. We should have warning when there are unknown downtime records.

I create a custom properties [b\_downtimeWarning] to link with a global InTouch tag [MES\_DownTime\_Track], they are all boolean value.



We created a script to run when current line, current entity changes or every minute to check if there are unknown downtime records.

[Script\_Check\_Downtime]

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

*''''''''*

*' Script Name: Script\_Check\_DownTime*

*' Expression: InTouch:MES\_Cur\_Ent + InTouch:MES\_Cur\_Line + InTouch:$Minute*

*' Trigger: DataChange*

*'=======================================================================================*

*' Author: Taki Guan <taki.guan@colpal.com>*

*' Creation Date: 2019-04-11*

*' Description: Check if there are unknown downtime records over 3 min of current entity*

*' Pre-condition:*

*' Post-condition:*

*'---------------------------------------------------------------------------------------*

*' Revision History:*

*' 000 - 2019-04-10 - Taki Guan <taki.guan@colpal.com>*

*' Initial Creation*

*' 000 - 2019-04-11 - Taki Guan <taki.guan@colpal.com>*

*' Comment some debug log messages*

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

dim entName as string;

If IsGood(InTouch:MES\_Cur\_Ent) Then

    entName = InTouch:MES\_Cur\_Ent;

*'LogMessage("entName = " + entName);*

    Dim clientSession As aaFactMES.aaClientSession;

    Dim result As aaFactMES.Result;

    Dim xmlBuilder As Fact.Common.XMLReadBuilder;

    result = aaFactMES.aaClientSession.GetInstance();

    If (result.Success == true) then

        clientSession = result.Value;

        xmlBuilder = Fact.Common.XMLReadBuilder("OPERA\_UI\_GetUtilHistory", "GetByKey");

        xmlBuilder.AddFilter("ent\_name", entName);

        xmlBuilder.AddFilter("type", 4);

        xmlBuilder.AddFilter("filter", "Not Entered,");

        result = clientSession.GetDS(xmlBuilder.Xml);

        If (result.Success == true) then

*'LogMessage("result.DataSet\_Value.Tables(1).Rows.Count: " + result.DataSet\_Value.Tables(1).Rows.Count);*

            if(result.DataSet\_Value.Tables(1).Rows.Count > 0) then

                b\_downtimeWarning = true;

            else

                b\_downtimeWarning = false;

*'LogMessage("b\_downtimeWarning = " + b\_downtimeWarning + " @ " + System.DateTime.Now());*

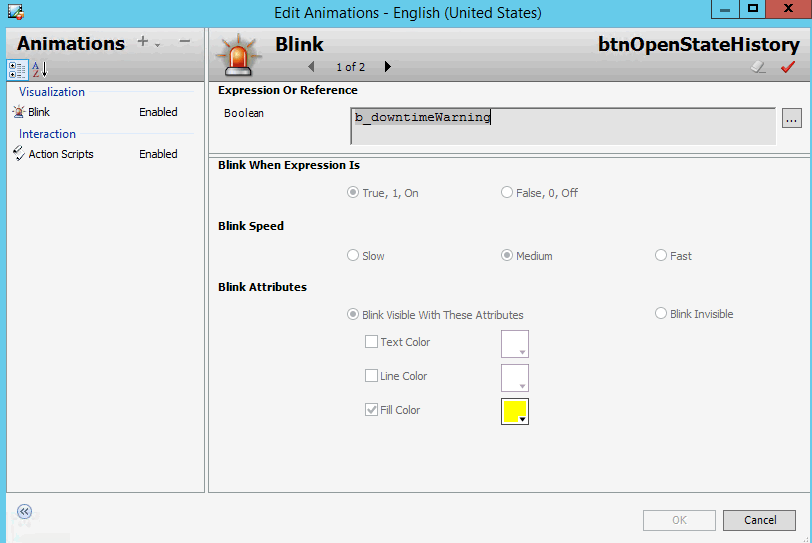
            endif;

        Endif;

    EndIf;

EndIf;

If [b\_downtimeWarning] is true, the equipment history button will blink. I added a blink animation.



Then If operator click the equipment history button, they want to see flited result. We need to add some script in graphic [operaDowntime\_History].

[Predefined Scripts]

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

*''''''''*

*' Script Name: Predefined Scripts*

*' Expression:*

*' Trigger: On Show*

*'=======================================================================================*

*' Author: Taki Guan <taki.guan@colpal.com>*

*' Creation Date: 2019-04-11*

*' Description: Set custom property for "b\_Downtime\_Track"*

*' Pre-condition:*

*' Post-condition:*

*'---------------------------------------------------------------------------------------*

*' Revision History:*

*' 000 - 2019-04-10 - Taki Guan <taki.guan@colpal.com>*

*' Initial Creation*

*' 001 - 2019-04-11 - Taki Guan <taki.guan@colpal.com>*

*' Comment some debug log messages*

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

entName = InTouch:MES\_Cur\_Ent;

SetCustomPropertyValue("b\_Downtime\_Track","InTouch:MES\_DownTime\_Track",false);

if b\_Downtime\_Track then

    c\_Downtime\_Trigger = true;

endif;

grpUtilHistory.Visible = True;

grpUtilHistory.Enabled = True;

grpUpdateComment.Visible = False;

grpUpdateComment.Enabled = False;

trgRefreshData = True;

grpUtilHistory.FactGrid1.AllowConfigure=Cfg\_Allow\_Config\_Grid;

grpUtilHistory.btnSaveConfig.Visible=Cfg\_Allow\_Config\_Grid;

[DowntimeTrigger]

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

*''''''''*

*' Script Name: DowntimeTrigger*

*' Expression: b\_Downtime\_Track*

*' Trigger: DataChange*

*'=======================================================================================*

*' Author: Taki Guan <taki.guan@colpal.com>*

*' Creation Date: 2019-04-10*

*' Description: If b\_Downtime\_Track data changes then assign value to c\_Downtime\_Trigger*

*' Pre-condition:*

*' Post-condition:*

*'---------------------------------------------------------------------------------------*

*' Revision History:*

*' 000 - 2019-04-10 - Taki Guan <taki.guan@colpal.com>*

*' Initial Creation*

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

if b\_Downtime\_Track then

    c\_Downtime\_Trigger = true;

endif;

[DowntimeOver3]

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

*''''''''*

*' Script Name: DowntimeOver3*

*' Expression: c\_Downtime\_Trigger*

*' Trigger: OnTrue*

*'=======================================================================================*

*' Author: Taki Guan <taki.guan@colpal.com>*

*' Creation Date: 2019-04-10*

*' Description: If the equipment history button blink and clicked then filter unknown downtime reason for operators*

*' Pre-condition:*

*' Post-condition:*

*'---------------------------------------------------------------------------------------*

*' Revision History:*

*' 000 - 2019-04-10 - Taki Guan <taki.guan@colpal.com>*

*' Initial Creation*

*' 001 - 2019-04-11 - Taki Guan <taki.guan@colpal.com>*

*' Comment some debug log messages*

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

c\_Downtime\_Trigger = false;

if not b\_Downtime3\_Filter then

*' save configuration*

    i\_SaveState = displayType;

    b\_State1 = Filter\_Window.Wait\_Filter.Wait\_Checked.Visible;

    b\_State2 = Filter\_Window.Run\_Filter.Run\_Checked.Visible;

    b\_State3 = Filter\_Window.NE\_Filter.NE\_Checked.Visible;

    b\_State4 = Filter\_Window.DT\_Filter.DT\_Checked.Visible;

    b\_State5 = Filter\_Window.Planned\_Filter.Planned\_Checked.Visible;

    b\_State6 = Filter\_Window.Maint\_Filter.Maint\_Checked.Visible;

    b\_State7 = Filter\_Window.CO\_Filter.CO\_Checked.Visible;

    b\_State8 = Filter\_Window.BE\_Filter.BE\_Checked.Visible;

*' set configuration*

    displayType = 4;

    Filter\_Window.Wait\_Filter.Wait\_Checked.Visible = false;

    Filter\_Window.Run\_Filter.Run\_Checked.Visible = false;

    Filter\_Window.NE\_Filter.NE\_Checked.Visible = true;

    Filter\_Window.DT\_Filter.DT\_Checked.Visible = false;

    Filter\_Window.Planned\_Filter.Planned\_Checked.Visible = false;

    Filter\_Window.Maint\_Filter.Maint\_Checked.Visible = false;

    Filter\_Window.CO\_Filter.CO\_Checked.Visible = false;

    Filter\_Window.BE\_Filter.BE\_Checked.Visible = false;

    Filter\_Window.Wait\_Filter.Wait\_NotCheck.Visible = true;

    Filter\_Window.Run\_Filter.Run\_NotCheck.Visible = true;

    Filter\_Window.NE\_Filter.NE\_NotCheck.Visible = false;

    Filter\_Window.DT\_Filter.DT\_NotCheck.Visible = true;

    Filter\_Window.Planned\_Filter.Planned\_NotCheck.Visible = true;

    Filter\_Window.Maint\_Filter.Maint\_NotCheck.Visible = true;

    Filter\_Window.CO\_Filter.CO\_NotCheck.Visible = true;

    Filter\_Window.BE\_Filter.BE\_NotCheck.Visible = true;

*' set filter to true*

    b\_Downtime3\_Filter = true;

    trgRefreshData = True;

endif;

[RefreshData]

Here I add one script for if there no unknown downtime records over 3 min. Assign false value to MES tag [MES\_Downtime\_Track] . Here I have some idea if you want there is no records the window returns to original display. If you want to use it. You can uncomment the if statement.

*' Added on 2019-04-10 by Taki Guan<taki\_guan@colpal.com> to trigger windows refresh if no unknown downtime reason over 3 min*

*' if dispalyType = 4 and no result then set the type to current shift*

            if(displayType == 4 and result.DataSet\_Value.Tables(1).Rows.Count == 0) then

*' Reset MES\_DownTime\_Track*

                InTouch:MES\_DownTime\_Track = false;

                LogMessage("InTouch:MES\_DownTime\_Track = " + InTouch:MES\_DownTime\_Track + " @ " + System.DateTime.Now());

*'if b\_Downtime3\_Filter then*

                 if false then

*'load configration*

                    displayType = i\_SaveState;

                    Filter\_Window.Wait\_Filter.Wait\_Checked.Visible = b\_State1;

                    Filter\_Window.Run\_Filter.Run\_Checked.Visible = b\_State2;

                    Filter\_Window.NE\_Filter.NE\_Checked.Visible = b\_State3;

                    Filter\_Window.DT\_Filter.DT\_Checked.Visible = b\_State4;

                    Filter\_Window.Planned\_Filter.Planned\_Checked.Visible = b\_State5;

                    Filter\_Window.Maint\_Filter.Maint\_Checked.Visible = b\_State6;

                    Filter\_Window.CO\_Filter.CO\_Checked.Visible = b\_State7;

                    Filter\_Window.BE\_Filter.BE\_Checked.Visible = b\_State8;

                    Filter\_Window.Wait\_Filter.Wait\_NotCheck.Visible = b\_State1 ^ 1;

                    Filter\_Window.Run\_Filter.Run\_NotCheck.Visible = b\_State2 ^ 1;

                    Filter\_Window.NE\_Filter.NE\_NotCheck.Visible = b\_State3 ^ 1;

                    Filter\_Window.DT\_Filter.DT\_NotCheck.Visible = b\_State4 ^ 1;

                    Filter\_Window.Planned\_Filter.Planned\_NotCheck.Visible = b\_State5 ^ 1;

                    Filter\_Window.Maint\_Filter.Maint\_NotCheck.Visible = b\_State6 ^ 1;

                    Filter\_Window.CO\_Filter.CO\_NotCheck.Visible = b\_State7 ^ 1;

                    Filter\_Window.BE\_Filter.BE\_NotCheck.Visible =b\_State8 ^ 1;

*' reset filter*

                    b\_Downtime3\_Filter = false;

                    trgRefreshData = true;

                endif;

            endif;

*' End added on 2019-04-10 by Taki Guan<taki\_guan@colpal.com>*

# Reference Link