






FVL21941	OPERATOR	MAN	PV	OP	TEXT	CLR
SOL. A FEED VALVE			CLOSED	OPEN	ACTIVE	

Change Zone

TIC21741	NORM	PROG	MAN	AUTO	CAS	PV	SP	OP%	Close
STM TEMP						33.4	33.4	22.3	
STEAM TEMP CONTROL									

◆ Objectives

-  **Explain what is meant by Change Zone**
-  **Understand the use of the Change Zone**
-  **Understand the Change Zone behavior**
-  **How to Implement the Change Zone**
-  **Examine the standard Change Zone**
-  **Design and build a custom Change Zone**

Main Idea

The topics covered in this module help you understand what are change zones and how to implement them. It also teaches you how to build custom change zones that will enhance your operator input to your process.

Objective

At the end of this module, you will be able to script performant change zones. To accomplish that goal, the following objectives are covered:

- Explain what is meant by Change zone
- Understand the use of the Change zone
- Understand the Change zone behavior
- How to Implement the Change zone
- Examine the standard Change zone
- Design and build a custom Change zone

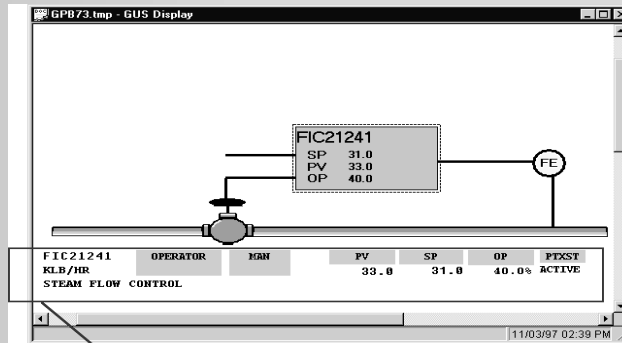
What Is A Change Zone

- The Change Zone displays a subset of a point's parameters and enables the operator to change values for those parameters which may be changed by an operator.
- The subset of parameters displayed in a Change Zone depends on the point type and other variables such as algorithm type.

What Is a Change Zone

Main Idea

Change zones are GUS display regions that enable your plant operation's personnel to make process control changes. For the display designer, a change zone represents an embedded display that shows a subset of a control point's parameters.



From this change zone, an operator can make control loop changes to the mode, setpoint, output, and other control parameters.

For performance, it may be better to have the change zone in a separate display.

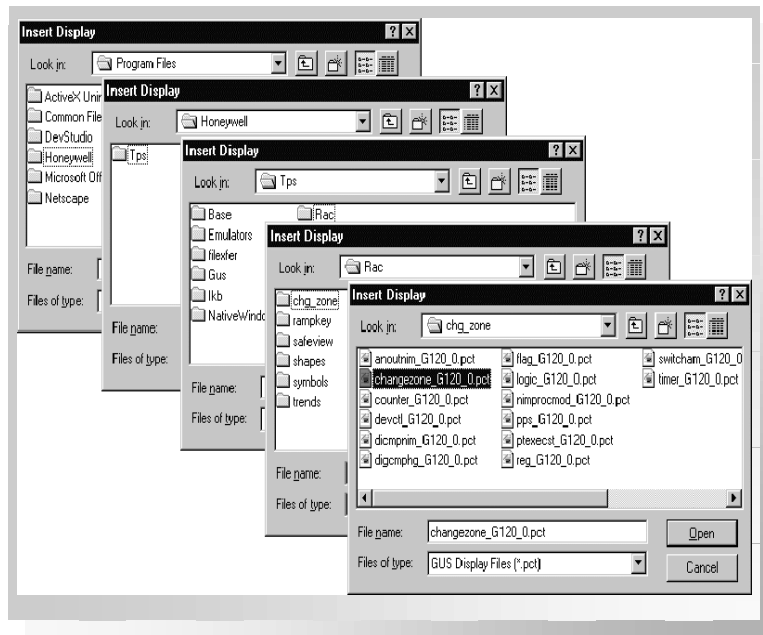
Standard Change Zone Concepts

- The overall procedure is to insert what is considered to be a “standard change zone.”
- As you become more experienced in display scripting, you may decide to create your own *custom change zone*.
- For now, review several concepts introduced in the overall standard change zone implementation procedure.

Standard Change Zone Picture in RAC Folder

- Honeywell provides a set of pre-built change zone pictures in the GUS RAC folder. The acronym RAC means Reusable Application Components.
- The standard change zone accommodates more than regulatory control points. In fact, the chg_zone folder has several pictures to support other point types such as timers, digital points, and logic points.
- The only file you need to select from the chg_zone folder is the change zone.pct file. At display run time, if the change zone needs to display a particular point type (e.g., a timer), then it invokes the needed file (e.g., timer.pct file).

Path to RAC Folder



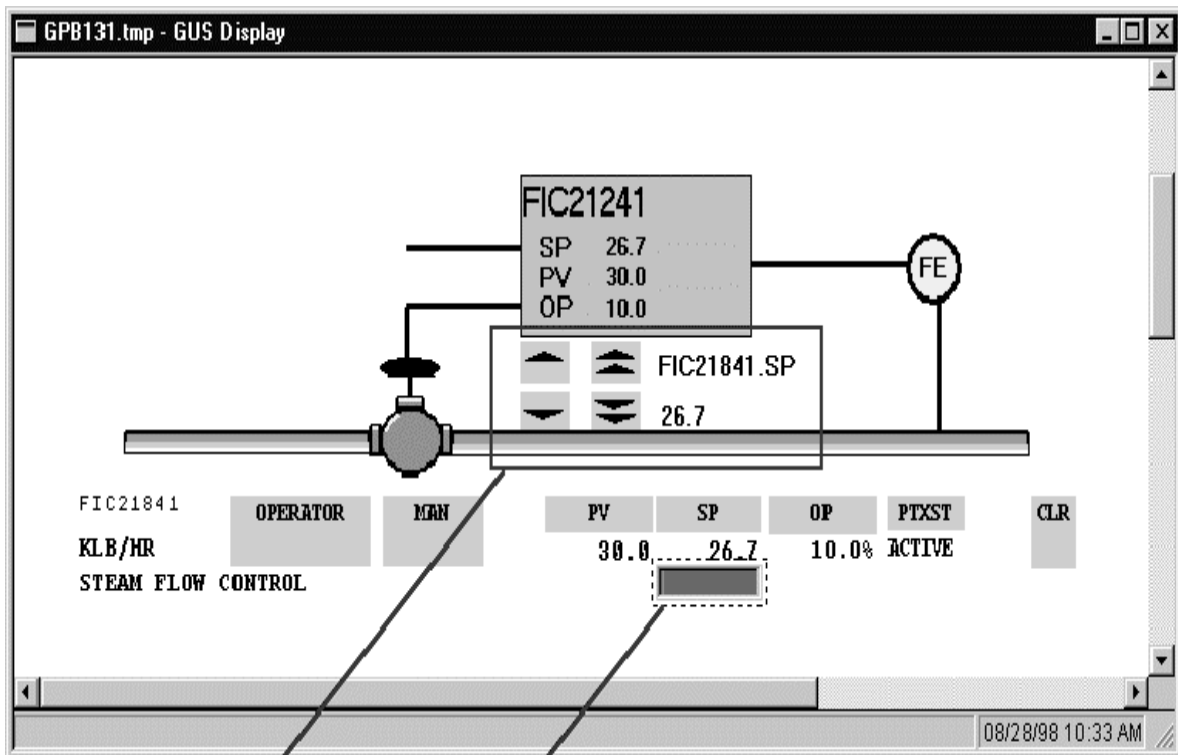
Filename in RAC Folder

The change zone filenames may indicate the system release. For example, the filename shown above includes "_G120_0" to indicate Release 120. To see how other standard displays appear, refer to your Display Builder User's Guide (either on-line help or paper manual) and look up "Standard Change Zone Displays" under the "Change Zones" topic.

Ramp Key Option

- When implementing a standard change zone, you have the option of inserting a ramp key function.
- The ramp keys provide a display selectable setpoint and output ramping function, which acts in a similar fashion to the Integrated Keyboard ramping keys.
- At display build time, you decide where to position the rampkey picture.

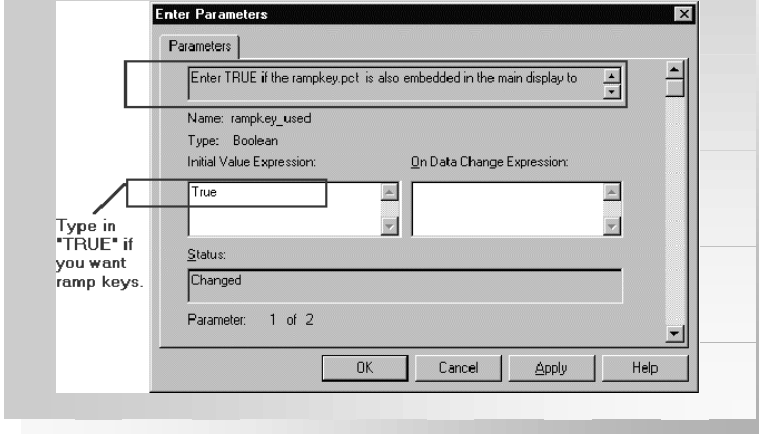
Ramp Keys



The ramping keys appear in the display whenever the setpoint or output buttons are selected from the changezone. The operator can then select the displayed up down arrows to raise or lower the setpoint or output.

Ramp Key Dialog Set "True" to Embed Display

The decision whether to implement the ramping key function occurs when you are inserting the standard changezone.pct display. At that time, dialogs appear requesting data for the rampkey picture.



Ramp Key Option

If you want the ramping function to appear when the change zone is invoked, type TRUE in the Initial Value Expression field. Then, at runtime, the rampkey display appears when an operator selects the setpoint or output function from the changezone.

The operator can then click on the display's arrow keys to ramp the setpoint or output. For the ramp keys to appear at display buildtime requires an extra step of inserting the rampkey display.

Ramp Key Dialog to Specify Data Collection Rate

- The next dialog that appears is the data collection rate for the ramp key function. You specify how often the data is updated in the ramp key display.
By pressing Enter without making a numeric entry, the update rate defaults to 4 seconds.

The screenshot shows a dialog box titled "Display Object Properties - EmbeddedPicture1" with a close button (X) in the top right corner. It has several tabs: "General", "Text", "Line", "Fill", "Transformation", "Property Inheritance", and "Parameters". The "Parameters" tab is selected. Inside the dialog, there is a text field with the prompt "Enter the number of the data collection group that dispcb.\$guscz01 is in." Below this is a "Name:" dropdown menu showing "group_for_guscz01", a "Type:" label with the value "Integer", and an "Initial Value Expression:" label with a text field containing "245". To the right of the "Initial Value Expression" field is a "Data Change Expression:" label with an empty text field. Below these fields is a "Status:" label with a text field containing "Validated". At the bottom left, it says "Parameter: 2 of 3". At the bottom right, there are "Previous" and "Next" buttons. At the very bottom of the dialog, there are "OK", "Cancel", "Apply", and "Help" buttons.

Data Collection Group

Configuration of a data collection group and an update rate for the data collection group is encountered in situations where periodic data polling may be desired. The data collection group implementation determines whether entities, such as tagnames or display entities (e.g., alarm color system entities) are assigned to data collection groups, and the user-configurable update rate is assigned to the data collection group.

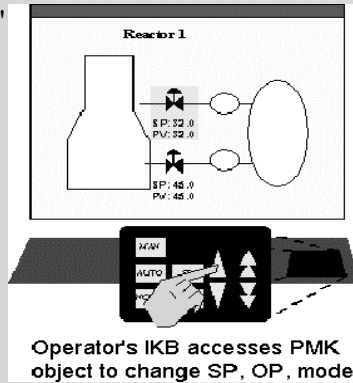
Point Manipulation Keys

- On your Integrated Keyboard are keys that are called Point Manipulation Keys (PMK). The Point Manipulation Keys provide mode change and setpoint and output ramp functionality for a standard change zone.



How Keys Work

- There is no extra coding on your part to make these keys work within a standard change zone because the Honeywell change zone does this for you through an invisible display object called the "PMK Object."

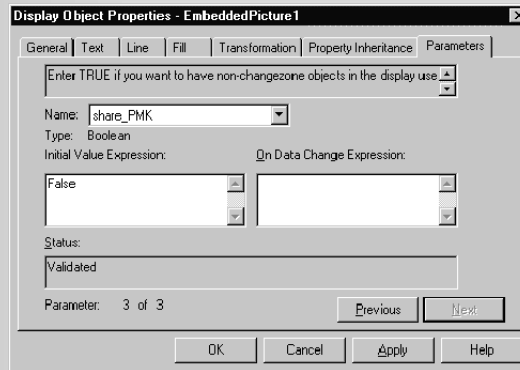


Operator IKB Access

At display runtime, the operator can then use the Point manipulation Keys from the Integrated Keyboard after selecting the desired change zone parameters.

PMK Dialog Specifies PMK Sharing

- The third dialog that appears during change zone insertion asks you whether you want to have PMK functionality available for any other selectable objects in this display.



PMK Object

Because there is only one global PMK object available for a display, the dialog asks whether you want to share that functionality with non-change zone objects.

For typical displays, you could enter "false" for the PMK share object. Remember that "false" means the Point Manipulation Keys still work for change zone selected SP, OP, and mode parameters.

If you decide to enter "true" for the PMK share dialog, then you have to add the necessary PMK coding on any selectable object in your display in order for the PMK object to work properly. This includes code that ensures that you ramp the desired control point. If you do not enter the PMK related script on selectable non-change zone objects, then PMK runtime errors are encountered when you select the non-changezone display object.

The type of PMK scripting required to support a "true" entry is beyond the scope of this introduction but will be covered later in the course.

After completing the necessary parameter entry dialogs, you then click on a display position for the standard change zone to appear. After inserting the standard change zone display, you insert the ramp key display, the topic of our next discussion. Inserting the ramp key picture provides the operator display selectable ramp keys, while the change zone itself automatically provides Integrated Keyboard selectable ramp keys. The end result of these steps is a highly interactive operator interface.

Display Building Tip

- When your change zone and ramp key displays are inserted into the display, they are invisible.
- Choose View>Invisible Objects so that you can view, size, and position your change zone and ramp key displays

Assign Point to Appear in Change Zone

- After inserting the change zone and the optional ramp key pictures, the next implementation step is to assign a point to appear in the change zone.
- There are several ways to do this. For a control point to appear in the change zone requires that you code script assigning the point to the change zone entity, \$CZ_ENTY.

Interpret Change Zone syntax,
DispDB.[*\$CZ_ENTY*]

Honeywell System Entity

Regardless of the scripting approach you decide to use, to make a point appear in the standard change zone requires the use of the Honeywell system entity *\$CZ_ENTY*.

For example, the following statement causes the point FC_00251 to appear in the change zone:

```
DispDB.[$CZ_ENTY] = "FC_00251"
```

Where:

DispDB represents the Display Database object,
\$CZ_ENTY represents the change zone variable of the display database object,
FC_00251 represents a control point.

The example statement, `DispDB.[$CZ_ENTY] = "FC_00251"`, shows that *\$CZ_ENTY* must be enclosed with brackets. Brackets [] are needed because the *\$CZ_ENTY* entity begins with a non-text \$ character. The \$ indicates that *CZ_ENTY* is a "system" entity. Additionally, DispDB precedes [*\$CZ_ENTY*]. DispDB represents the display database of your GUS display, or more accurately, the display database object.

The display database object can be thought of as an object in your GUS display in the same way that you think of a text object or rectangle object. That is, the display database object, just like a rectangle or text object, has its own unique set of properties. The display database object's properties include items such as containers or arrays that store variables such as integers, real numbers, strings, and system entities such as *\$CZ_ENTY*.

CONTRUCTION of CUSTOM CHANGE ZONE

The Change Zone must allow an operator to observe and change the following for any points that have these parameters:

- **Set Point (SP)**
- **Process Variable (PV)**
- **Output (OP)**
- **Mode (Mode)**
- **Any other parameter** that may be necessary for the operator usage.

Operation of the Change Zone

- The Operator selects a point by touching a target on a display.
- The Change Zone is presented at the bottom of the display or any other position that may be appropriate.
- The Change Zone shows the name of the point, the description of the point and the values for each of the above-mentioned parameters.

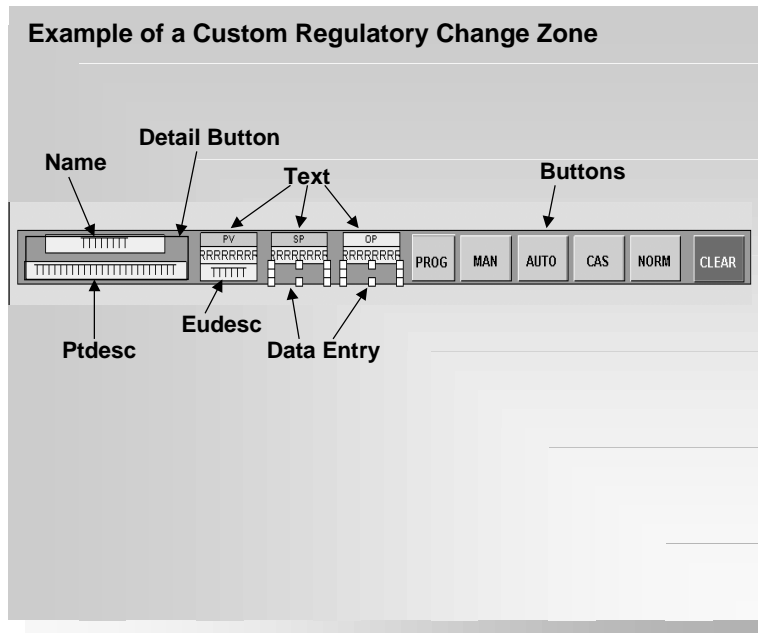
Operation of the Change Zone

- To change any of the numeric parameters (e.g., SP, PV, or OP), the operator uses a touch target to select the parameter. After selection, he/she types the new value in and, when the "Enter" key is pressed, the parameter value is changed. For the Mode parameter, the operator selects the new value by first touching one of three targets. Each of these targets represents a value of the Mode parameter (i.e. Manual, Auto, or Cascade). Following selection of the value, an EXECUTE target appears. When this target is touched, the selected value is stored into the Mode parameter.

Operation of the Change Zone

- The Change Zone is to contain a target that will allow the operator to call up the Detail Display for the selected point.
- Once a point has been selected and the Change Zone for that point displayed, the Change Zone is to remain on the screen. If a subsequent point is selected, the Change Zone is updated to reflect the new point.
- The Change Zone is to contain a target that will allow the operator to remove it from display.

Change Zone Example



Example of Values and Script for change zone

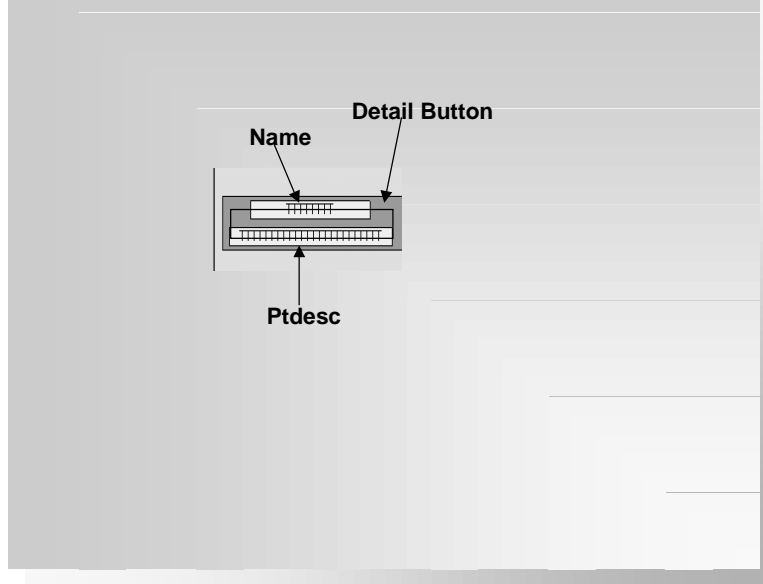
Values

Name: dispdb.ent100.[name]
Ptdesc: dispdb.ent100.ptdesc
Eudesc: dispdb.ent100.eudesc
PV: dispdb.ent100.pv
SP: dispdb.ent100.sp
OP: dispdb.ent100.op

Note: You could also use dispdb.[\$cz_enty] for access to the change zone entity values

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

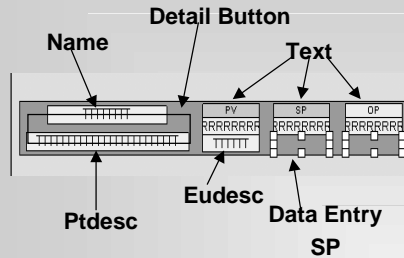
Script

Detail Button

```
Sub OnLButtonUp()  
  
on error goto badstatus  
  
detail dispdb.ent100  
  
exit sub  
  
badstatus:  
    'Do nothing - end subroutine  
  
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

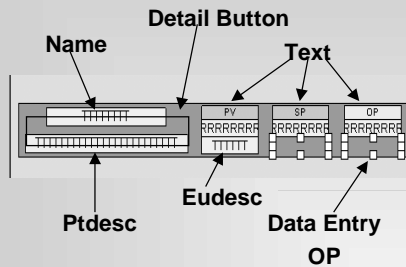
Script

Data Entry for SP (these code examples are for the Honeywell OLE “inserted” Data Entry Box)

```
Private entry as long
Sub GoodEntry(sData As String)
ON ERROR GOTO BADSTATUS
entry = sData
if dispdb.ent100.modattr = "PROGRAM" or dispdb.ent100.mode <> "AUTO" then
    Begin Dialog MessageDialog „192,109,"Mode error"
        CancelButton 124,68,64,36
        Text 8,8,176,52,"Point not in correct mode for setpoint change",.Text1,"Arial",18
    End Dialog
    Dim MsgDialog As MessageDialog
    Dialog MsgDialog,-1,10000
elseif entry >= dispdb.ent100.splofm and entry <= dispdb.ent100.sphilm then
    dispdb.ent100.sp = entry
    me.selstart = 0
    me.slength = 7
    me.seltext = ""
    me.text = ""
    me.visible = false
    me.enabled = false
else
    Begin Dialog Message2Dialog „192,109,"Entry error"
        CancelButton 124,68,64,36
        Text 8,8,176,52,"Limit or range error on setpoint entry",.Text1,"Arial",18
    End Dialog
    Dim Msg2Dialog As Message2Dialog
    Dialog Msg2Dialog,-1,10000
end if
exit sub
BADSTATUS:
    if err.number <> 1052 and err.number <> 1054 then
        BEEP
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"
    end if
End Sub
Sub BadEntry(sData As String)
    Begin Dialog MessageDialog „192,109,"Mode error"
        CancelButton 124,68,64,36
        Text 8,8,176,52,"Invalid setpoint entry",.Text1,"Arial",18
    End Dialog
    Dim MsgDialog As MessageDialog
    Dialog MsgDialog,-1,10000
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

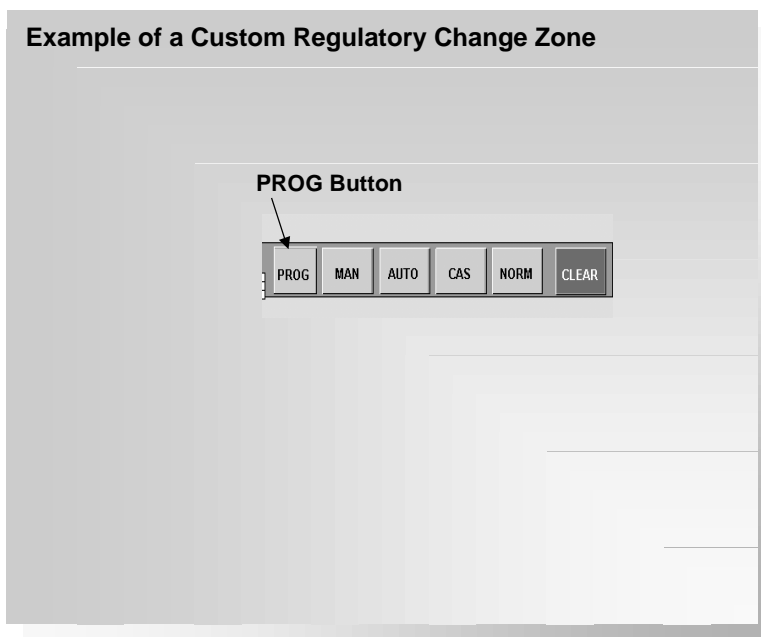
Script

Data Entry for OP

```
Private entry as long
Sub GoodEntry(sData As String)
    ON ERROR GOTO BADSTATUS
    entry = sData
    if dispdb.ent100.modattr = "PROGRAM" or dispdb.ent100.mode <> "MAN" then
        Begin Dialog MessageDialog ,192,109,"Mode error"
            CancelButton 124,68,64,36
            Text 8,8,176,52,"Point not in correct mode for output change",.Text1,"Arial",18
        End Dialog
        Dim MsgDialog As MessageDialog
        Dialog MsgDialog,-1,10000
    elseif entry >= dispdb.ent100.oploim and entry <= dispdb.ent100.ophilm then
        dispdb.ent100.op = entry
        me.selstart = 0
        me.sellength = 7
        me.seltext = ""
        me.text = ""
        me.visible = false
        me.enabled = false
    else
        Begin Dialog Message2Dialog ,192,109,"Entry error"
            CancelButton 124,68,64,36
            Text 8,8,176,52,"Limit or range error on output entry",.Text1,"Arial",18
        End Dialog
        Dim Msg2Dialog As Message2Dialog
        Dialog Msg2Dialog,-1,10000
    end if
    exit sub
BADSTATUS:
    if err.number <> 1052 and err.number <> 1054 then
        BEEP
        MSGBOX ERROR$ & " : " & err.number,cbExclamation,"Error"
    end if
End Sub
Sub BadEntry(sData As String)
    Begin Dialog MessageDialog ,192,109,"Entry error"
        CancelButton 124,68,64,36
        Text 8,8,176,52,"Invalid output entry",.Text1,"Arial",18
    End Dialog
    Dim MsgDialog As MessageDialog
    Dialog MsgDialog,-1,10000
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

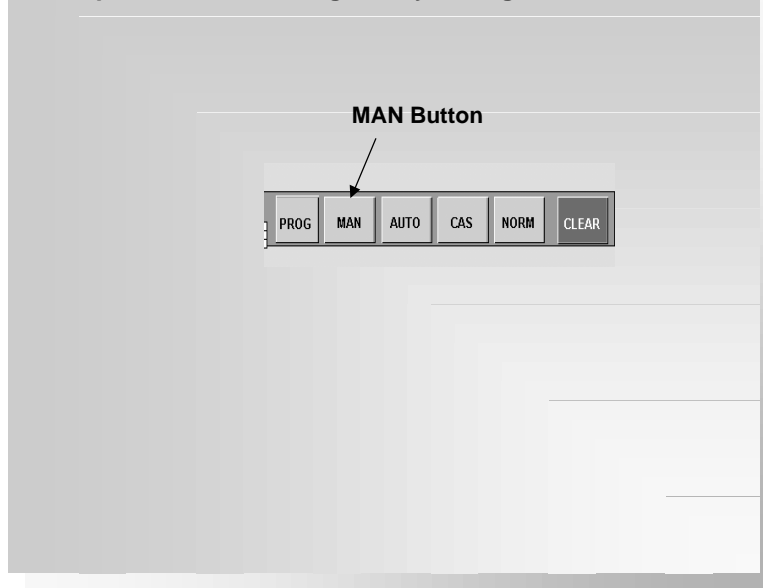
Script

Prog Button

```
Sub OnDataChange()  
ON ERROR GOTO BADSTATUS  
if dispdb.ent100.modattr = "PROGRAM" then  
    me.fillcolor = TDC_GREEN  
else  
    me.fillcolor = makecolor(204,204,204)  
end if  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number, vbExclamation, "Error"  
    end if  
End Sub  
Sub OnLButtonUp()  
ON ERROR GOTO BADSTATUS  
    dispdb.ent100.modattr = "PROGRAM"  
    me.fillcolor = TDC_GREEN  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number, vbExclamation, "Error"  
    end if  
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

Script

Man Button

```
Sub onDataChange()  
ON ERROR GOTO BADSTATUS  
if dispdb.ent100.mode = "MAN" then  
    me.fillcolor = TDC_YELLOW  
else  
    me.fillcolor = makecolor(204,204,204)  
end if  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number, vbExclamation, "Error"  
    end if  
End Sub  
Sub OnLButtonUp()  
ON ERROR GOTO BADSTATUS  
    dispdb.ent100.mode = "MAN"  
    dispdb.ent100.modattr = "OPERATOR"  
    me.fillcolor = TDC_YELLOW  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number, vbExclamation, "Error"  
    end if  
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

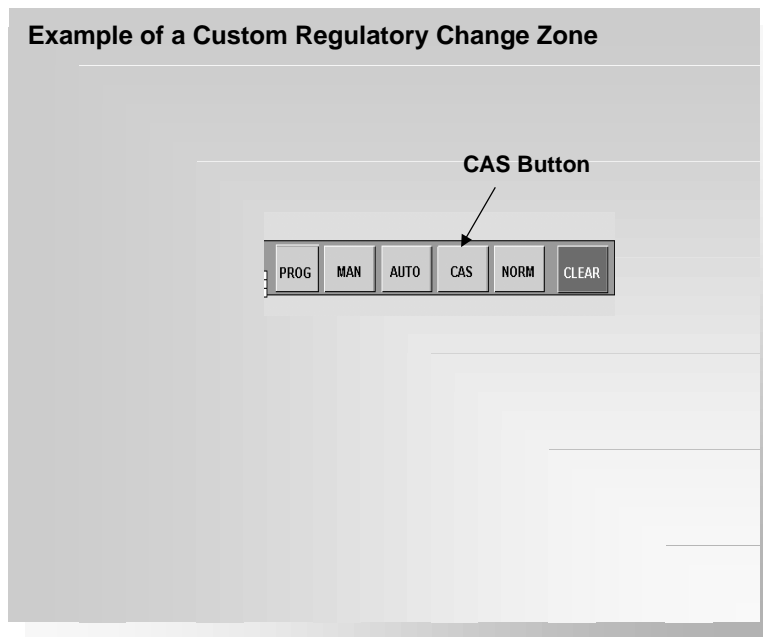
Script

Auto Button

```
Sub OnDataChange()  
ON ERROR GOTO BADSTATUS  
if dispdb.ent100.mode = "AUTO" then  
    me.fillcolor = TDC_GREEN  
else  
    me.fillcolor = makecolor(204,204,204)  
end if  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"  
    end if  
End Sub  
Sub OnLButtonUp()  
ON ERROR GOTO BADSTATUS  
dispdb.ent100.mode = "AUTO"  
dispdb.ent100.modattr = "OPERATOR"  
me.fillcolor = TDC_GREEN  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"  
    end if  
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

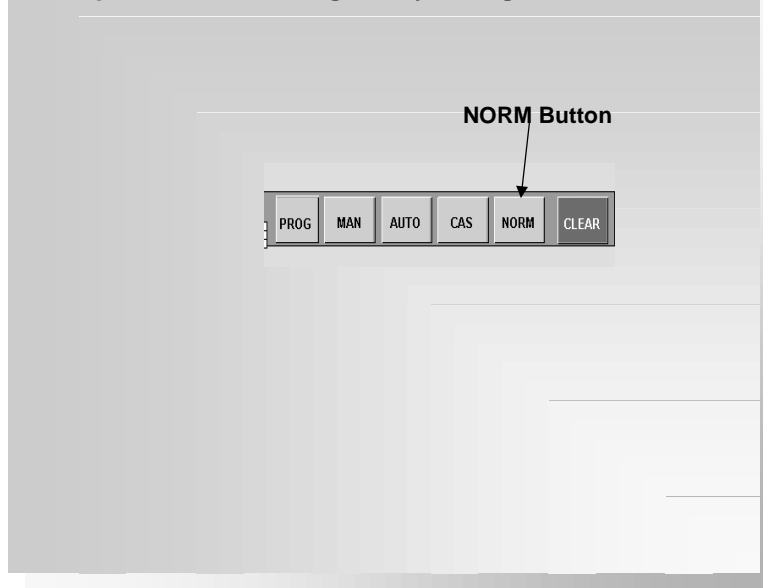
Script

Cas Button

```
Sub OnDataChange()  
ON ERROR GOTO BADSTATUS  
if dispdb.ent100.mode = "CAS" then  
    me.fillcolor = TDC_GREEN  
else  
    me.fillcolor = makecolor(204,204,204)  
end if  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"  
    end if  
End Sub  
Sub OnLButtonUp()  
ON ERROR GOTO BADSTATUS  
    dispdb.ent100.mode = "CAS"  
    dispdb.ent100.modattr = "OPERATOR"  
    me.fillcolor = TDC_GREEN  
exit sub  
BADSTATUS:  
    if err.number <> 1052 and err.number <> 1054 then  
        BEEP  
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"  
    end if  
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

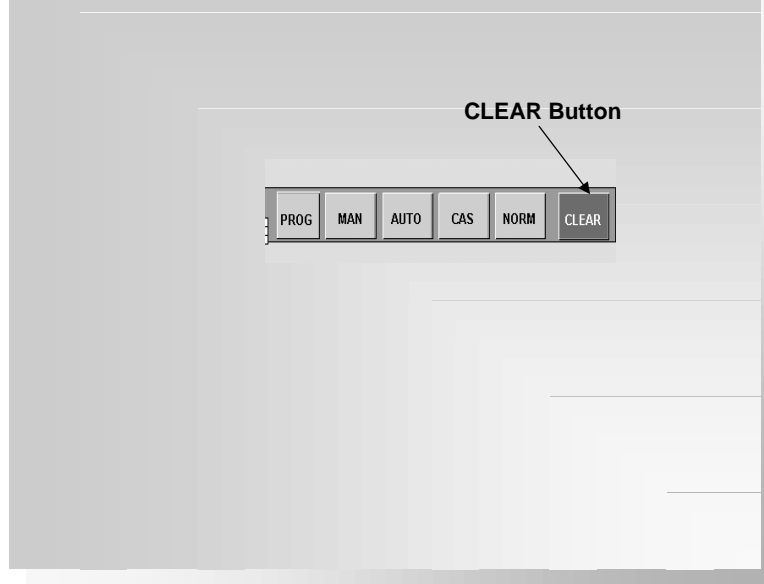
Script

Norm Button

```
Sub OnDataChange()
ON ERROR GOTO BADSTATUS
if dispdb.ent100.mode = dispdb.ent100.nmode and dispdb.ent100.modattr = dispdb.ent100.nmodattr then
    me.fillcolor = TDC_CYAN
else
    me.fillcolor = makecolor(204,204,204)
end if
exit sub
BADSTATUS:
    if err.number <> 1052 and err.number <> 1054 then
        BEEP
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"
    end if
End Sub
Sub OnLButtonUp()
ON ERROR GOTO BADSTATUS
    dispdb.ent100.mode = dispdb.ent100.nmode
    dispdb.ent100.modattr = dispdb.ent100.nmodattr
    me.fillcolor = TDC_CYAN
exit sub
BADSTATUS:
    if err.number <> 1052 and err.number <> 1054 then
        BEEP
        MSGBOX ERROR$ & " : " & err.number,ebExclamation,"Error"
    end if
End Sub
```

Change Zone Example

Example of a Custom Regulatory Change Zone



Example of Values and Script for change zone

Script

Clear Button

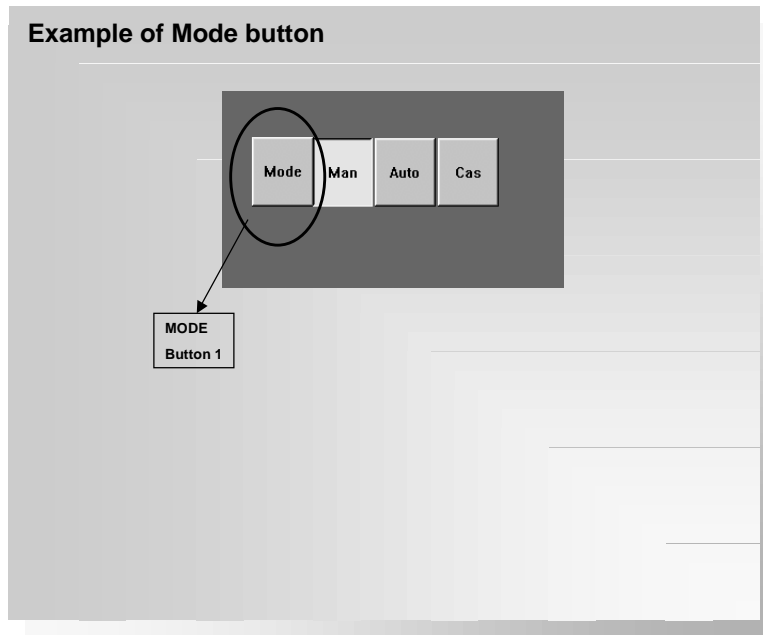
No code provided. You will design the script for the clear button.

Final Note

- ✱ The previous example represents a regulatory change zone; you can use this template to build other change zones for different point types.
- ✱ Consult with your users (operators); what do they expect from the change zone and build accordingly.
- ✱ Remember (in most cases) that they are the ones using the display and not you, the developer.

Change Zone Example

Example of Mode button



Sample script for MODE

Sub OnDataChange()

```
on error goto trap
modes3.visible = false
modes3.selectable = false
me.text = mid$(cstr(dispdb.[%cz_enty].mode),1,4)
me.visible = true
if mid$(DISPDB.[%CZ_ENTY].MODE,1,3) = "MAN" then MODES3.buttonstate(0) = true
if mid$(DISPDB.[%CZ_ENTY].MODE,1,4) = "AUTO" then MODES3.buttonstate(1) = true
if mid$(DISPDB.[%CZ_ENTY].MODE,1,3) = "CAS" then MODES3.buttonstate(2) = true
exit sub
trap:
me.visible = false
```

End Sub

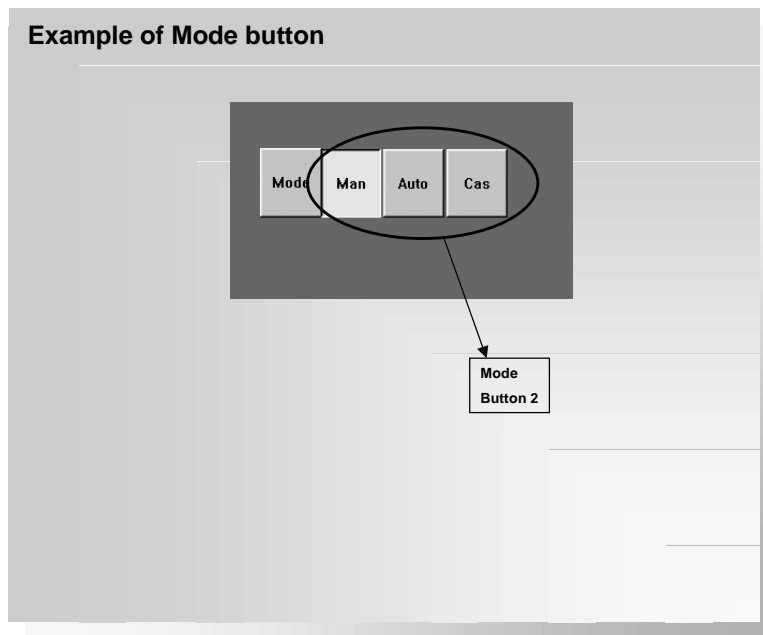
Sub OnLButtonUp()

```
me.visible = false
modes3.visible = true
me.selectable = false
modes3.selectable = true
```

End Sub

Change Zone Example

Example of Mode button



Sample script for MODES3

private sel as long

Sub OnLButtonup()

 sel = me.buttonindex

 select Case sel

 case 0

 dispdb.[\${cz_enty}].mode = "Man"

 Case 1

 dispdb.[\${cz_enty}].mode = "Auto"

 Case 2

 dispdb.[\${cz_enty}].mode = "Cas"

 case else

 msgbox "Problem"

 end select

 mode.visible = true

 me.visible = false

 mode.selectable = true

 me.selectable = false

End Sub