



◆ Objectives



**To Understand Honeywell's
Procedure Actions**



What Are: Actors

Targets



**Using different types of
Actors**



**Relationship between TDC
Actors and Display Builder
Scripts**

Objectives

At the end of this module you will be able to do the following:

- Understand the Honeywell Procedure actions.
- Understand the terms of Actors, Targets.
- Learn how to use the different types of actors.
- Understand the relationship between TDC Actors and Display Building Scripts.

Actors & Scripts

Action Procedure - Actors

- An action procedure is one or more actors that specify a desired action.
- Action procedures are also called actors.
- Actors are triggered by touching a target or pressing on the configurable buttons on the Operator's keyboard.
- Action procedures are defined and associated with targets and buttons by users.

Actors

- A function, implemented by Honeywell, that is called an action procedure.
- Create predefined actions by using the actors.
- There are grouping of actors which are used to execute different functions.
- Actors are triggered by activating (e.g., touching) a target or pressing one of the configurable buttons on the Operator's Keyboard.

Actors & Scripts

Target

- Rectangular areas in the picture that initiate some specified action when selected by the operator.
- The action is defined by using an action procedure (actors).

Target

- Selectable by user either by mouse click or touch screen.
- User should be able to recognize target areas - designer should have cursor change or provide a tooltip help.

Actors & Scripts

Types of Actors

- There are different groupings of actors
 - **Display actors** call up various displays and provide screen-management functions.
 - **Keyboard actors** simulate keyboard buttons.
 - **Operator input actors** allow input from the operator.

Types of Actors

Common types of LCN Actors

<u><i>Display actors</i></u>	<u><i>Keyboard actors</i></u>	<u><i>Operator Input actors</i></u>
Schem	Enter	R_Bool
Group		R_Date
Detail		R_Dur
Alarmsum		R_Time
Chgzone		R_Ent
Mult_ov		R_Ent_N
Msgsum		R_Enm
Clear_cz		R_Int
Clr_scrn		R_Par
Prompt_c		R_Real
Key		R_Senm
Move		R_Senm_N
Update		R_Str
Que_key		R_Var
Update		

Actors & Scripts

Types of Actors cont.

- **Store actors** store values in either the local Display Database or the system database.
- **Read and Store actors** are a combination of the two previous types.
- **Get actors** retrieve data from a specified database.
- **General actors** are miscellaneous functions that do not fit in other categories.

Types of Actors

Common types of LCN Actors

- | <u>Store actors</u> | <u>Read & Store actors</u> | <u>Get actors</u> | <u>General actors</u> |
|---------------------|--------------------------------|-------------------|-----------------------|
| S_Bool
Group | RS_Loc | | R_Bool |

Actors & Scripts

Types of Actors cont.

- **Range Checking actors** check to see if input values fall within specified limits.
- **Trend actors** are used to manipulate trend lines in custom displays.
- **Arithmetic, Logical, Compare, and Conditional actors** are used to perform mathematical, conditional, and comparative operations.

Types of Actors

Common types of Actors

Range actors

S_Bool

Trend actors

RS_Loc

Arithmetic actors

R_Bool

Actors & Scripts

Use of Actors

- Each actor has a unique calling sequence, typically of the following form:

ACTOR NAME(parameters)

E.g.. GROUP(1, 6)

This actor is used to call up a specific group display with a specific point selected (group 1, point 6 in this case).

Notes

Actors & Scripts

Use of Actors Cont..

- Some actors have no parameters others have six or more parameters. An actor can return a value that is then used as a parameter for another actor. In such cases, the form is typically

ACTOR1 NAME(ACTOR2 NAME(parameters))

- ACTOR2 executes first to provide the parameter for ACTOR1.

Notes

Actors & Scripts

Use of Actors Cont.

- When separated by semicolons, a series of LCN actors can be executed in sequence from left to right, for example,

`ACTOR1(parameters);ACTOR2(parameters);...ACTORn(parameters)`

- An actor that is used to return a parameter executes before the actor that uses that parameter, as explained above.

Notes:

BasicScript only allows one command per line. What might appear in the LCN Picture Editor as

DELAY(0,1,1); GROUP(100,1)

would now, in the GUS Display Builder, appear as

**DELAY 0, 1, 1,
GROUP 100, 1**

Actors & Scripts

Relationship Between Actors and Scripts

- TDC Actors will only run in scripts handling USER INTERFACE Events
- These events can be
 - OnLButtonClick
 - OnDbClick etc.*and any other events that will require user interaction*

Notes

If a TDC Actor is in a script like OnPeriodicUpdate or OnDataChange, you will get a runtime error.

Actors & Scripts

Relationship Between Actors and Scripts

- **Built-in Functions**

- The built-in functions of BasicScript will provide the equivalent capabilities to the built-in functions of the TDC Picture Editor.
- The following material will map the usage of BasicScript to the Picture Editor.
 - **Tag names and DBB syntax**
 - **X,Y Coordinates**
 - **ON, OFF**
 - **Enumerations**
 - **Date, Time, Duration**
 - **BasicScript extensions**
 - **BasicScript Line Restrictions.**

Notes

Actors & Scripts

Relationship Between Actors and Scripts

Tag names and DBB syntax

- Tagnames in BasicScript are properties of the built-in **LCN** and **DISPDB** object.

– For example:

Fic21941.PV becomes

LCN.Fic21941.PV

INT01 becomes

DISPDB.INT01

Access to Process Data from Scripts

Process data access in GUS follows an object paradigm. This means that tags are treated like objects and their related parameters are handled as properties.

Built-in Objects

GUS Basic provides the following built-in objects which provides access to Local Control Network(LCN) data:

LCN - This object represents the LCN. Points can be accessed only as properties of the LCN Data object

DISPDB - This object represents the display data base.

COLLECTOR - This object represents the functionality of the TDC Picture Editor's collectors which provides access to certain system data.

Actors & Scripts

Properties of the LCN Object

- LCN Points(tags) are accessible as properties of the LCN object.
 - Eg. LCN.FIC21941
- Parameters are accessible as properties of a point object.
 - Eg. LCN.FIC21941.PV

Properties of the LCN Object

- A point is an object in GUS Basic; it can have properties. The properties of a point are its parameters. These parameters are also treated as objects in GUS Basic. The most commonly accessed property of a parameter will be its value. Other values are status, type, and various enumeration type values.
 - Eg. To assign a PV value to a text object named Text1 use
Text1.text = LCN.FIC21941.PV
- The default property is the value. For example
LCN.FIC21941.PV will return the value of the PV.
This reference is the shortcut for LCN.FIC21941.PV.VALUE

Actors & Scripts

Accessing LCN Names

- Point and Parameter names may contain characters that are not valid in GUS Basic.
- Escape Mechanism.....
 - Enclose the invalid name in square brackets.
 - Eg. Me.text = LCN.[\$A100].[!param1]
- Reserved words will be treated in the same manner.
 - Name is a reserved word
 - Eg. Me.text = LCN.FIC21941.[name]
- Evaluate Operator - accepts one string argument
 - Eg. Me.text = LCN.FIC21941.evaluate("name")

Accessing LCN Names

- When using the evaluate operator, enclose only one piece of the name form in the operator at a time.
Eg. LCN.[\$MN11B30].fl(34) *Correct*
 LCN.[\$MN11B30.fl(34)] *Incorrect*

Actors & Scripts

LCN Data Types vs. Basic Data Types

<u>LCN Data Type</u>	<u>Basic Data Type</u>
Integer	Integer
Real	Single or Double
Boolean	Boolean
String	String
Enumeration	String
SD Enumeration	String
Date/Time	Date
Unknown	String

Notes

Actors & Scripts

– How to Use Actors in Scripts

- X,Y coordinates
 - In GUS are expressed in pixels, with the left top as origin.
- ON, OFF
 - In Basic scripts use True and False

Notes

Actors & Scripts

– How to Use Actors in Scripts

- Enumerations
 - Enum built-in function
 - To use `QUE_KEY(ENTER)` in BasicScript
use `QUE_KEY(enum("$buttons:enter"))`
- Date, Time, Duration
 - the # sign must be in front of any date or time when used in basicscript.

Notes

Actors & Scripts

– How to Use Actors in Scripts

- BasicScript Extensions
- BasicScript Line Restrictions
 - allows only one command per line

Notes

BasicScript extensions can be implemented either as a function or a command. A function returns a value and has a parenthesized argument list. For example, **Msgbox**, if used with a parenthesized argument list, will return an integer based on which button was selected. An **Inputbox** is also a BasicScript function that returns a value. Therefore, any actor functions that do not return a value have been implemented in GUS BasicScript as commands with no parenthesis. For example,

Group(100,1) in the Native Window Picture Editor
becomes **Group 100, 1** in GUS BasicScript.

Basic Script allows only one command per line. For example,

Delay(0,1,1); Group(100,1) in the Picture Editor

now requires two separate lines in GUS BasicScript

Delay 0,1,1
Group 100,1

Actors & Scripts

– How to Use Display Actors in Scripts

- To use display actors, check the Appendix at the end of this module
 - Most of the display actors are listed as available as scripting extension
 - eg. To call up the Alarm Summary use the following syntax

```
Sub OnLButtonClick()  
  Dim x as single  
  x = LCN.FIC21941.PV  
  Alarmsum(0)  
End Sub
```

Display Actors

```
Sub OnLButtonClick()  
  Dim x as single  
  x = LCN.FIC21941.PV ' this assignment statement  
  Alarmsum(0)         ' makes a link to the LCN  
End Sub               ' object
```

This example works because a reference to the LCN is made with the assignment statement. Without this statement an error will occur.

The alarm summary will appear in the Native Window only if an operating display is currently displayed in the Native Window.