Honeywell

Experion PKS
GUS Display Translator User's Guide

EPDOC-XX47-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX47-en-431A	431	0	February 2015

Disclaimer

Contents

1 About This Document

This document describes the steps to run the Display Translator from an EST node, begin diagnosis of errors that may have occurred as a result of the translation, and test the displays (pictures) to ensure they function as they did prior to the translation.

Revision history

Revision	Date	Description
A	December 2013	Initial release of document.

2 References

The following list identifies all documents that may be sources of reference for material discussed in this publication.

Document Name	Document ID
TPS File Transfer Installation and User Guide	EPDOC-X144-en-430A
GUS Display Authoring Tutorial	EPDOC-XX43-en-430A
GUS Display Builder User's Guide	EPDOC-XX44-en-430A
GUS Display Scripting User's Guide	EPDOC-XX45-en-430A
Picture Editor Reference Manual	SW09-650
Actors Manual	SW09-655
Picture Editor Data entry	SW11-650
Picture Editor Form Instructions	SW12-650

3 Introduction

3.1 About Display Translator

The Display Translator runs on the Experion Station-TPS (ES-T) nodes connected to a Local Control Network (LCN). Another software package called File Transfer allows you to access LCN History Module files (such as .DS files when using Display Translator) from ES-T and Personal Computer (PC) on the network.

TPN (LCN) graphics (.DS files) can run in Native Window as files on the LCN, but otherwise are not usable on an ES-T.

Display Translator can translate .DS graphics that have been created and displayed on Universal Station, Universal Station^{x}, or Universal Work Station into GUS object files.

Attention

To transfer and translate .DS files from the History Module to ES-T, the optional File Transfer software must be loaded on your ES-T.

Generally, this document contains information only pertinent to the use of GUS Display Translator. Successful use of the Display Translator depends very much on the correct setup of the File Transfer software.

For information and procedures regarding File Transfer, refer to the *File Translator Installation and User Guide*.

3.1.1 Minimum software revision

To translate existing .DS graphics into GUS-usable graphics, the existing .DS graphics must be at LCN software release R500 or greater.

3.2 Intended Audience

This document is intended for those who are trained and/or experienced with the following:

- Microsoft Windows Explorer
- operation and navigation of a ES-T
- accessing the LCN system History Module (HM) using File Transfer software
- building/modifying LCN .DS graphics using the LCN Picture Editor
- building/modifying GUS .PCT graphics using GUS Display Builder

3.3 What You Will Have When You Are Done

The Display Translator transforms .DS source files into GUS display files (.PCT files) by reading each record, modifying the contents as needed, and converting the resulting information into corresponding GUS objects or script.

The following table lists the .DS elements (as they exist prior to translation) and the corresponding GUS object(s). The variations shown in the GUS objects depend on certain attributes of the original .DS elements.

.DS Element	GUS Object
Bar	Rectangle
Circle phantom	Circular ellipse
Conditional Behavior	'On Data Change' script on Object(s) to which the condition was applied
Line	Line, Polyline, Rectangle, or Closed Polyline
Picture Comment	Description field of picture properties
Quarter phantom	Arc
Solid	Rectangle or Closed Polyline, Polyline
Subpicture	Embedded display (Has an option to 'extract' a copy of the embedded display and save it as a separate .PCT file)
Target	Rectangle with OnLeftButtonUp script
Text	Text
Value	Text
Variant	A group of text or embedded display objects with a group 'On Data Change' script to make the components either visible or invisible based on the variant logic.

4 Considerations for Display Translator

4.1 Areas Requiring Special Consideration

Specific components of .DS graphic files require special handling and/or modifications prior to successful translations of subject graphics files.

It is important that you are aware of certain special considerations regarding how Display Translator handles certain forms of data.

The special considerations are as follows:

- Variants or Conditions that do not reference LCN data or display database variables will not force 'On Data Change' script events, and therefore will not execute at runtime. These scripts must be changed to 'On Display Startup' scripts.
- Priorities of objects are translated, but it is not a clean translation.
- When a log file is designated in the translator syntax, it will overwrite an existing log file with the same name in the same location (same directory), without warning.
- Set collection rates and groups are not carried over by the translation.
- Do not translate a .DS file for a local language at another local language's ES-T. For example, do not translate a .DS file that includes Russian characters on a Chinese version ES-T.
- The Reverse and Half Intensity properties are not accessible through the object property sheets. They can only be changed through the use of scripts. The behavior at runtime of these properties can be unpredictable if used in a subpicture with inheritance.
- Objects located above the TDC display region, or that straddle the display/non-display region will be moved to the region under the display in GUS Display Builder. The objects are accessible by setting zoom to 25% or 50%. If an object located at the top of the display is missing after translation, verify it does not cross the boundary.
- Default group settings differ between TPS Picture Editor and GUS Display Builder. Group 0 is the default group setting in the TPS Picture Editor, and Group 1 is the default group setting in the GUS Display Builder.
- GS_VAR actor functionality is not supported in GUS displays. Scripts using this actor will generate runtime errors and must be commented out.

4.2 What is Translated and What is Not?

Display Translator does not translate some graphics components. The translator generates a warning message when it encounters an object it cannot translate.

For details, review the following three lists, which contain functions that do not translate, that partially translate, and that should fully translate.

4.2.1 Functions that do not translate

Function	Does It Translate?	Comments
Backplane Functions (Radar, XY_Plot, Line graph, Ring, Keylock)	no	
Color Palettes	no	
Overlays	no	
Pie Chart	no	
Set collection	no	Set Collection values are not being carried over by translation. Must be manually re-entered after display validation.
Standard Change Zone	no	Use the GUS change zone supplied in the C:\Program Files\Honeywell \TPS\RAC\chg_zone directory.
Trend	no	Trends are not supported at this time. GUS Display Builder OCXTRENDs do not support all of the same default conventions used when creating the .PCT files.

4.2.2 Functions that partially translate

Function	Does It Translate?	Comments
Actors (General support)	partially	The following actors are not supported, but are 'passed through' the translator to the GUS file (and commented out of the script): Mult_Ov, Overlay, Back_Ov, Clr_Scrn, Palette, User_Cz, Eq_List, Trend actors and collectors, Move, Cursor, Tab, Select, GS_var
Priorities (z-order) of objects	partially	Rules for GUS Display Builder z-order manipulation do not exactly match the TPS schematic object and FGB priority rules. In some cases, manual reordering of display objects in the GUS display will be required. See 'Verify Correct Operation of Displays' in this document for further information.
Reverse Text / Values	partially	In the .PCT files, users may have graphic objects inserted between the text and its background. Since GUS Display Builder text and its background have the same ‘z' order, there may be some schematics that will display these objects differently (graphic object will be in the background).

4.2.3 Functions that successfully translate

Function	Does It Translate?	Comments
Actor (immediate) data references - Direct	yes	
Actor (immediate) data references - Indexed	yes	

Function	Does It Translate?	Comments
Actor (immediate) data references - indirect (2-level)	yes	
Bars	yes	Black background is not provided by GUS bar.
Conditionals	yes	
Custom Change Zone	yes	If the Custom Change Zone was used in an overlay, refer to section 'Cleaning Up Translated Displays.'
General target support	yes	
Quarter Phantom	yes	Older .PCT schematics may report a severe translator error against a ‘Phantom EOL'. To correct, rewrite the .TPS schematic in the Picture Editor (no need to compile).
Scanned data references - Direct	yes	
Scanned data references - Indexed	yes	
Scanned data references - Indirect (2-level)	yes	
Static Graphics	yes	
Static text	yes	
Subpictures	yes	Parameterized variables have been implemented as inline parameters.
Translation of Enumerations	yes	Display Translator can not resolve data types for enumeration references in conditionals and variants of the form 'IF &tag.op = stop then…'. Schematics with these forms will have to be translated in the interactive mode (no log file specified), with the user being prompted to input the enumeration type. If this form is encountered during a non-interactive translation (error output to a log file), a note to translate interactively is included in the log file.
User-defined DDB	yes	
Values	yes	
Variants	yes	

4.3 Data Access Considerations

The following list of DDBs and Data Access topics associated with .DS pictures are not supported by GUS displays.

DDB/Topic	Description	Results
$CZ_VID1 $CZ_VID2 $CZ_VID3 $CZ_VID4 $CZ_VID5 $CZ_ENM1 $CZ_ENM2 $CZ_ENM3 $CZ_INFO $GRPBASE	Unsupported DDBs	Translator reports a warning message if used in a value. Data access to these DDBs will not work in scripting.
Indexed entities	Data Access Topic	Not supported in GUS Display Builder
CZ group	Unsupported	

5 Prior to Running the Display Translator

5.1 Summary of Steps Required to Translate Displays

Because the graphics available on ES-T use methods such as OLE (Object Linking and Embedding), the form graphics take after translation from .DS to .PCT is much different than the original .DS form. Some aspects of existing .DS graphics do not translate properly, without your intervention.

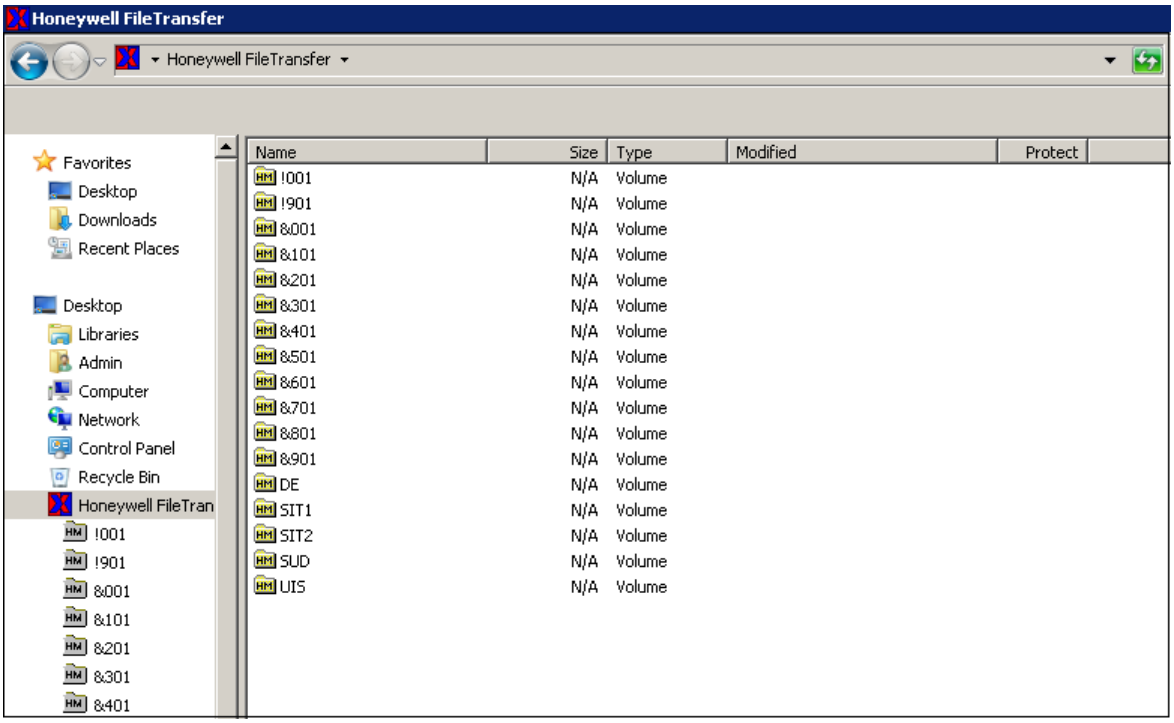
With that in mind, the general steps you will likely have to go through to successfully complete the translation are as follows:

1. Ensure the .DS graphics to be translated are compatible with LCN R500 or later software.
2. Ensure the File Transfer software is loaded and its service is running.
3. Ensure the Global User Station is operational and, through the use of File Transfer software, is linked to the HM containing the .DS picture elements.
4. Verify that the GUS Display Translator software is loaded.
5. Run the translation.
6. Check for errors.
7. Correct the errors and rerun the translator.
8. Test and correct display components.
9. Validate the .PCT file.
10. Run the translated displays to verify correct operation.

5.2 Ensure ES-T Has Access to HM

For the Display Translator to locate the existing .DS graphic files to be translated, it must access them from the ES-T, over the LCN, to the appropriate History Module (HM). File Transfer software makes this possible. You may install File Transfer on the local ES-T or on another ES-T on the network.

File Transfer shows the HM as a workspace in Windows Explorer, as shown by the following figure.



5.3 Ensure All Required Software is Installed and Running

5.3.1 Display Translator

Display Translator software automatically installs with the licensed GUS Display Builder software package.

Prior to using GUS Display Translator, ensure that it is installed and operational.

5.3.2 File Transfer

Creating a File Transfer link between your ES-T and an HM on an Experion system requires the installation and use of the File Transfer software.

For information and procedures regarding installation and use of File Transfer, refer to the *File Transfer Installation and User Guide* (FE05).

6 Run GUS Display Translator

6.1 Overview of Performing Display Translation

6.1.1 Prerequisites

Prior to running the Display Translator, ensure that all affected components of the translation are ready and accessible. If you have not already done so, review and complete the following steps in the previous section before proceeding:

- Ensure ES-T Has Access To HM
- Ensure Display Translator and File Transfer software is installed and running

Also, if you have not already done so, create a new directory on (or accessible by) the ES-T that will contain the translated display files.

A naming convention that includes the current revision of GUS Display Translator being used is useful later, when migrating to newer software.

6.1.2 Translation options

GUS Display Translator includes the following options:

Option	Description
Interactive Mode or Batch Mode	If a log file is specified, all messages are written to the file in batch mode. If a log file is not specified, the user must interactively acknowledge dialog boxes during translation.
Directory Translation mode or Individual File Translation mode	The files to be translated can be selected on a directory basis, individually, or in groups of files.
Embedded Display Translation	If this option is set, the 'main' .PCT file is stored under the target directory. A separate subdirectory is created, titled with the main display's name. This subdirectory holds the extracted embedded display .PCT file(s).

Regardless of which option(s) you choose, the software performs checks to ensure the following:

- the input and output directories, and the input files exist,
- the output directory has WRITE permissions,
- the user has permissions to generate a file, and
- the .DS files are true LCN display source files, and they are in the correct minimum revision of LCN software. (Display Translator ignores non .DS files.)

6.1.3 Restrictions

Do not translate displays on a node that is required for control. Data transfer through the GUS interface during translation will be restricted, and could affect view of your process.

Due to performance restrictions, embedded display extraction of an entire directory is not recommended; instead, extract on a file-by-file basis.

If .PCT files with the same name already exist in the same output directory, they will be overwritten, without a warning message.

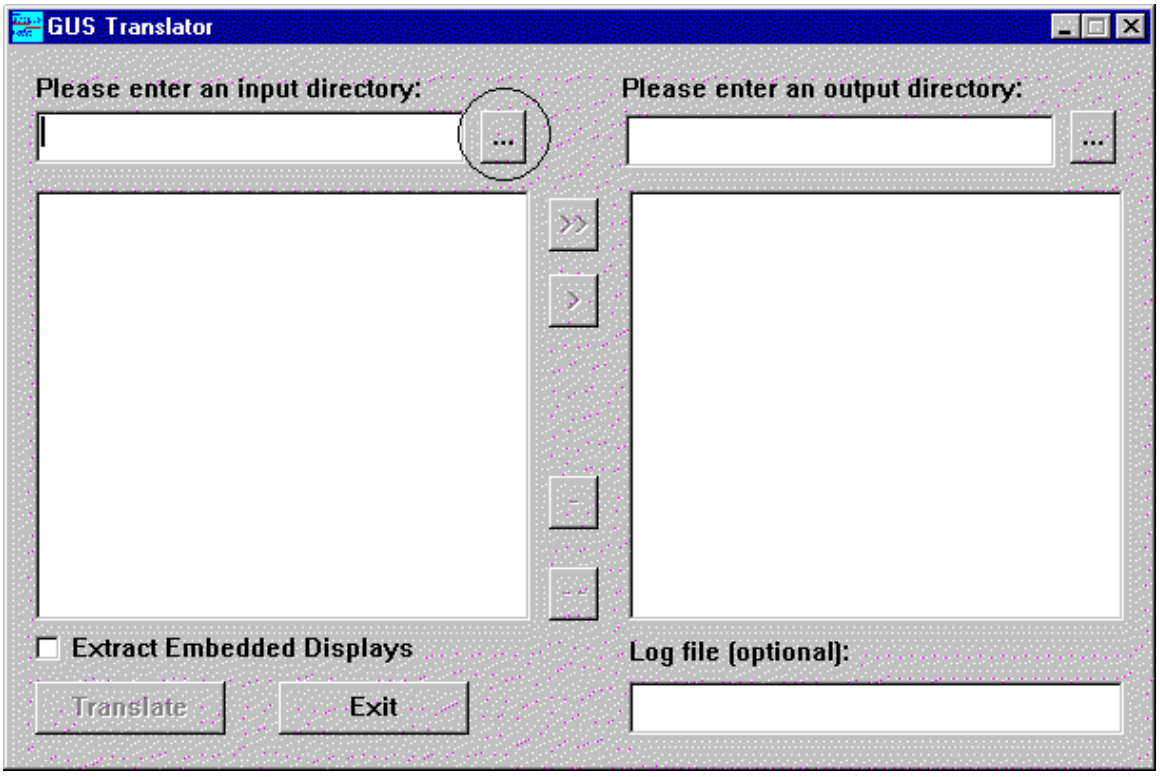
6.2 Running GUS Display Translator

Perform the following steps to execute the GUS Display Translator program.

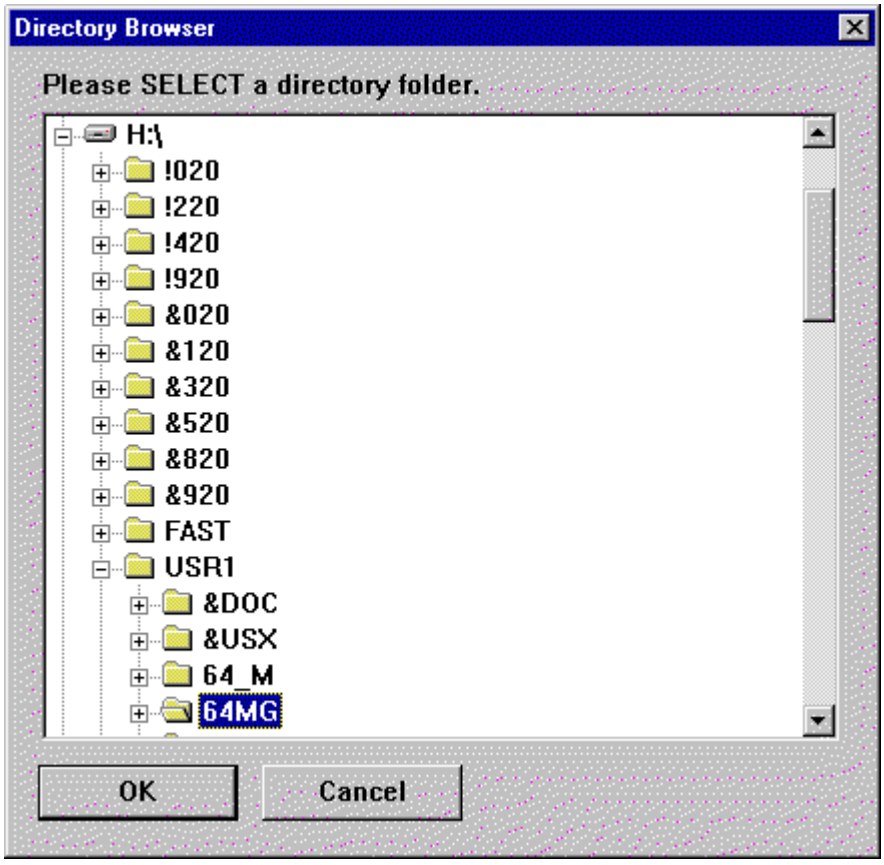
1 Select **Start Menu** > **All Programs** > **Honeywell Experion PKS** > **TPS Applications** > **GUS Display Builder Tools** > **Display Translator**.

2 On the **Please enter an Input Directory** line, enter the path to the input directory containing the picture (.DS) files to be translated.

Enter the path, if known, or use click the Browse button. 



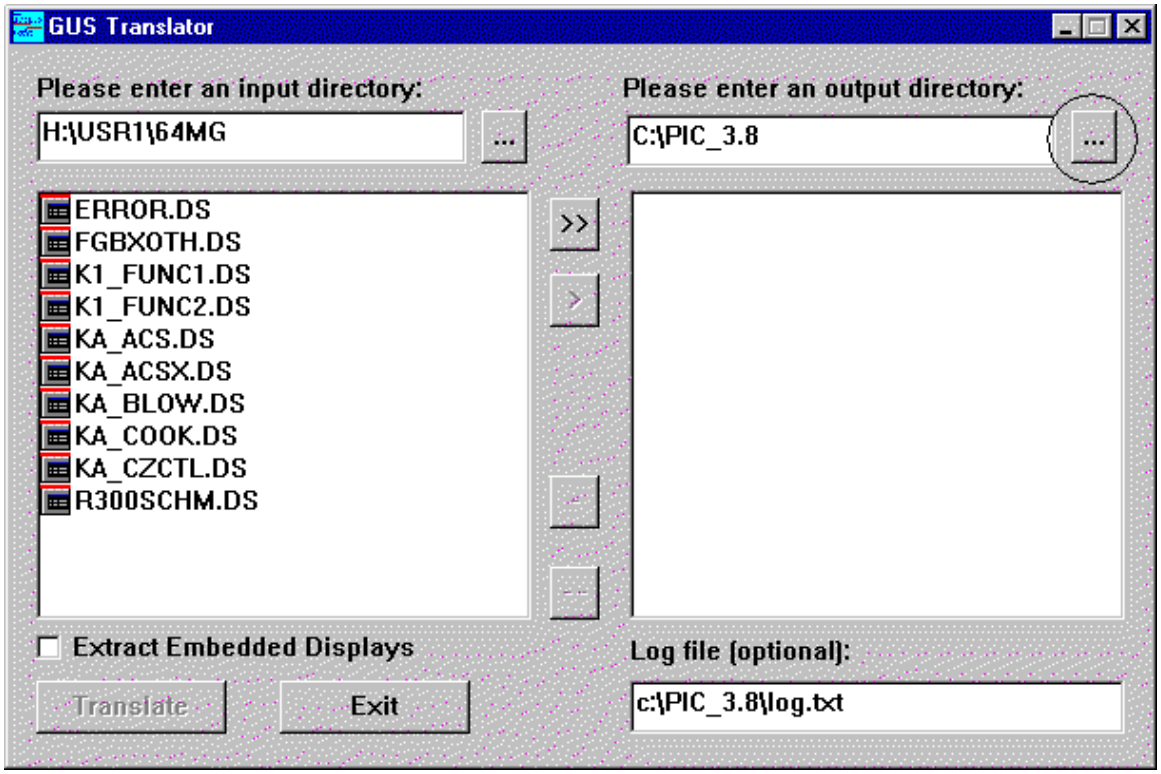
3 If you click the Browse button  , navigate to the network drive and lowest level HM directory containing your input files, and then click **OK**.



4 The .DS files contained in the directory appear in the left 'input directory' pane of the dialog box.

On the **Please enter an Output Directory** line, enter the path to the output directory that will contain the display files (.PCT) that have been translated.

Enter the path, if known, or click the Browse button. ...



If you click the Browse button,

navigate to the network drive and the lowest level directory where you want to store the translated files, and then click **OK.**

If you the results of your translation output to a log file, then enter the full path and log file name on the **Log File (optional)** line.

If you choose not to designate a log file, the error and status messages will appear on the screen during translation, and will require your interactive acknowledgment of error messages.

5 To select all files in a directory, select the >> button.

To select individual files, click on the filename(s) in the input list. Use the Ctrl key while selecting non-contiguous files, and shift+space to select contiguous files from the list.

Then click the > button to create a list of output file names (with a .PCT suffix) in the output column.

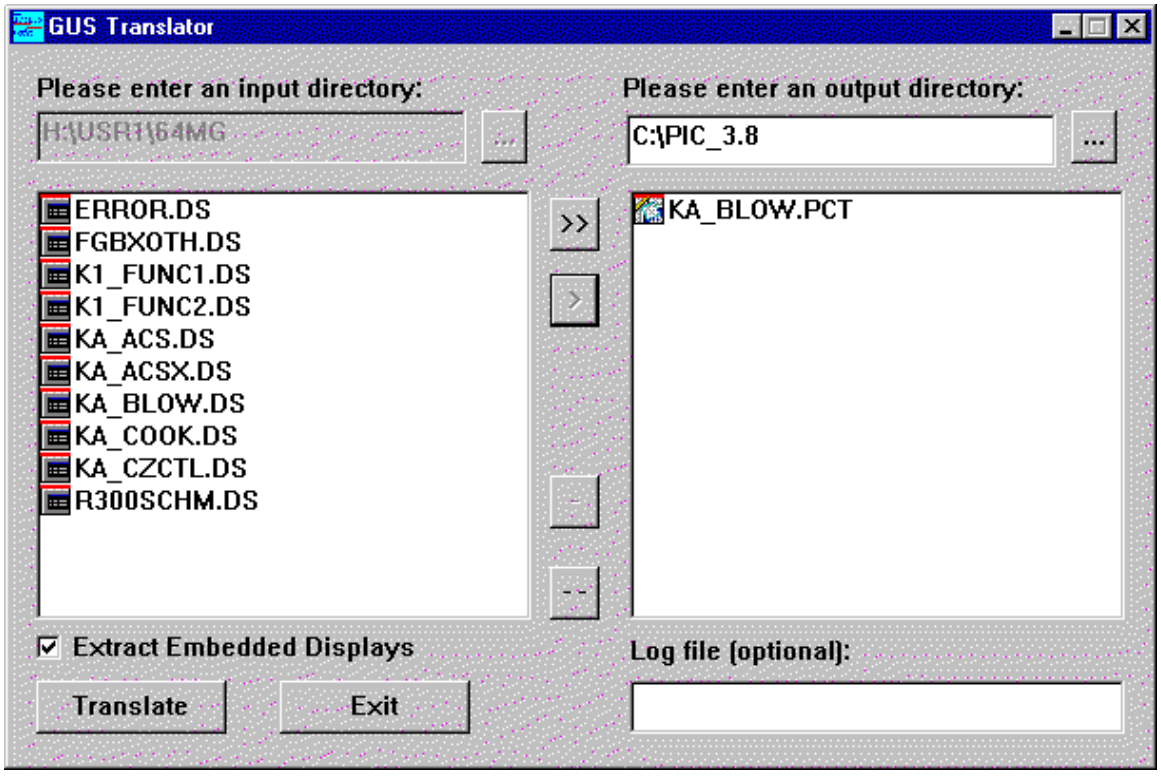
To clear all output file names, click on the -- button.

To remove individually selected (highlighted) files, click on the - button.

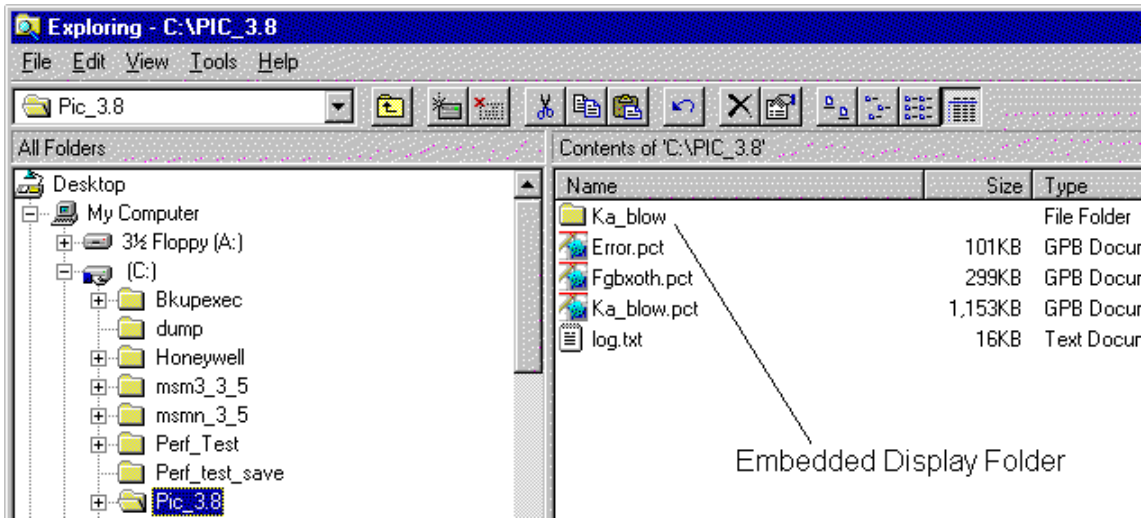
6 **Skip this step if you are not creating embedded subpicture files.**

NOTE: Single file embedded display extraction is recommended.

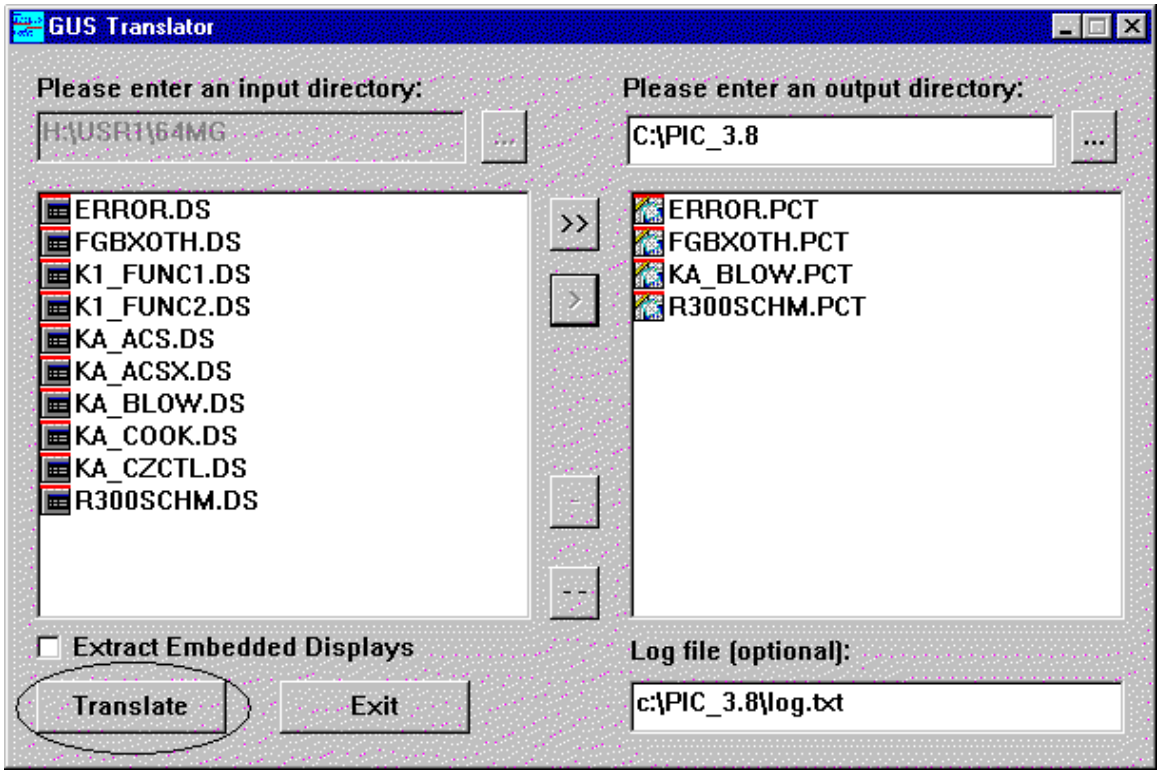
To create subpicture .PCT files, select the 'Extract Embedded Displays' button. When this option is used, copies of the embedded displays in the schematic being translated are created in a separate folder within the target folder. That folder within a folder will be given the same name as the translated file.



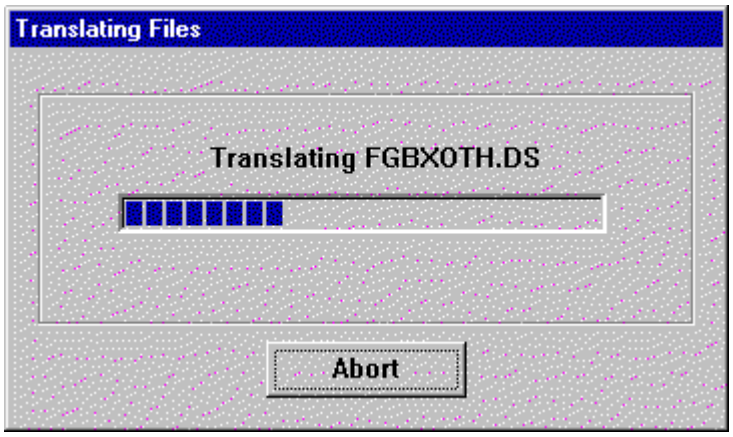
The resulting embedded folder for this example is shown below:



7 Once all files you wish to translate appear in the output column, Click the Translate button. The translation will begin.



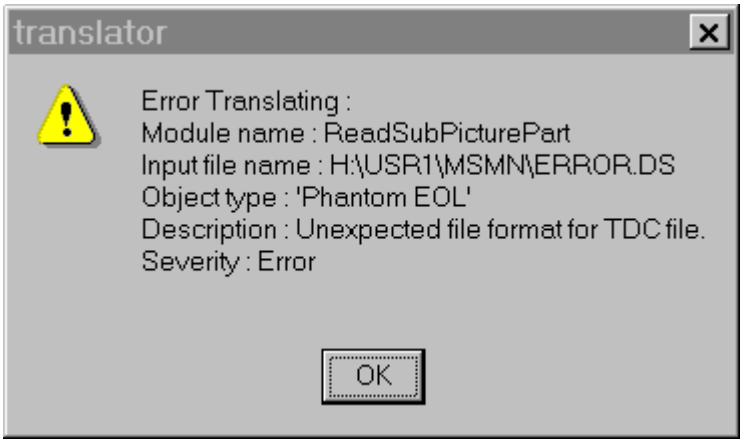
When translating, a progress bar, similar to the example below will appear.



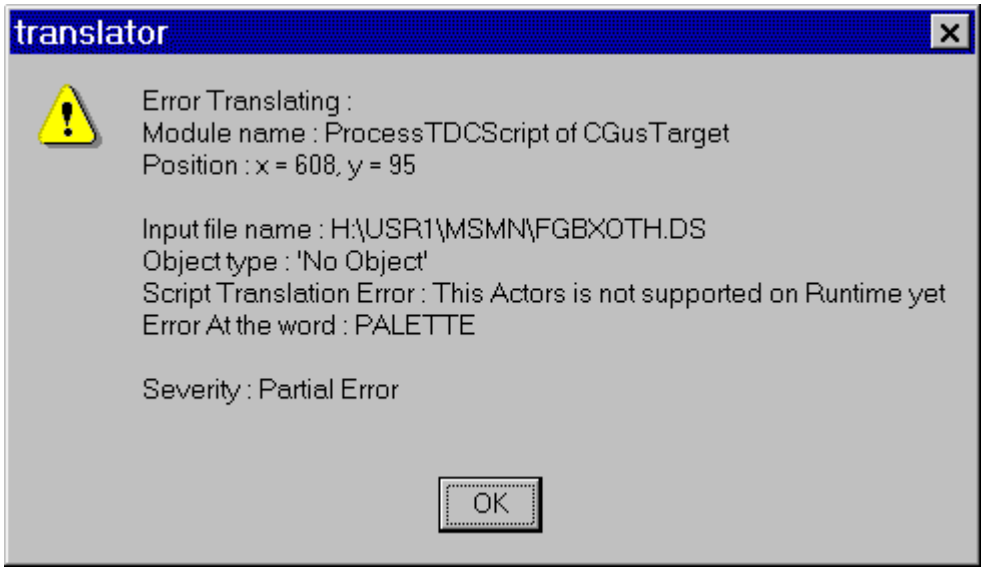
8 If you have not defined a log file, and an object is encountered that cannot be translated, a warning dialog box is displayed. Examples of warning dialog boxes are shown below.

Select **OK** to continue.

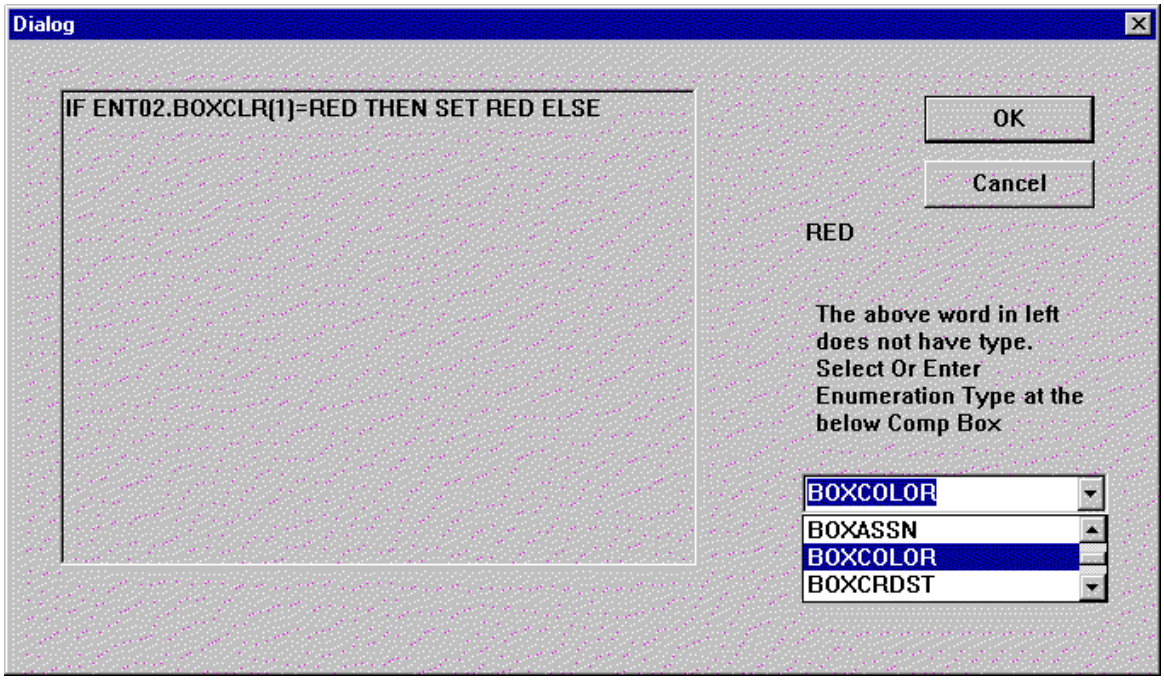
In this example, the input file name was ERROR.DS It failed with a severity of 'Error,' which indicates the file was not translated.



The example below shows the input file name as FGBXOTH.DS. It failed with a Severity of 'Partial Error,' which indicates the file translated with at least one non-fatal error.



9 If the translator encounters an enumeration or data type it cannot interpret, and you have not designated a log file, a dialog box similar to the example shown below is displayed.



Select the appropriate type, by scrolling through the menu options, or by typing the entry, if known.

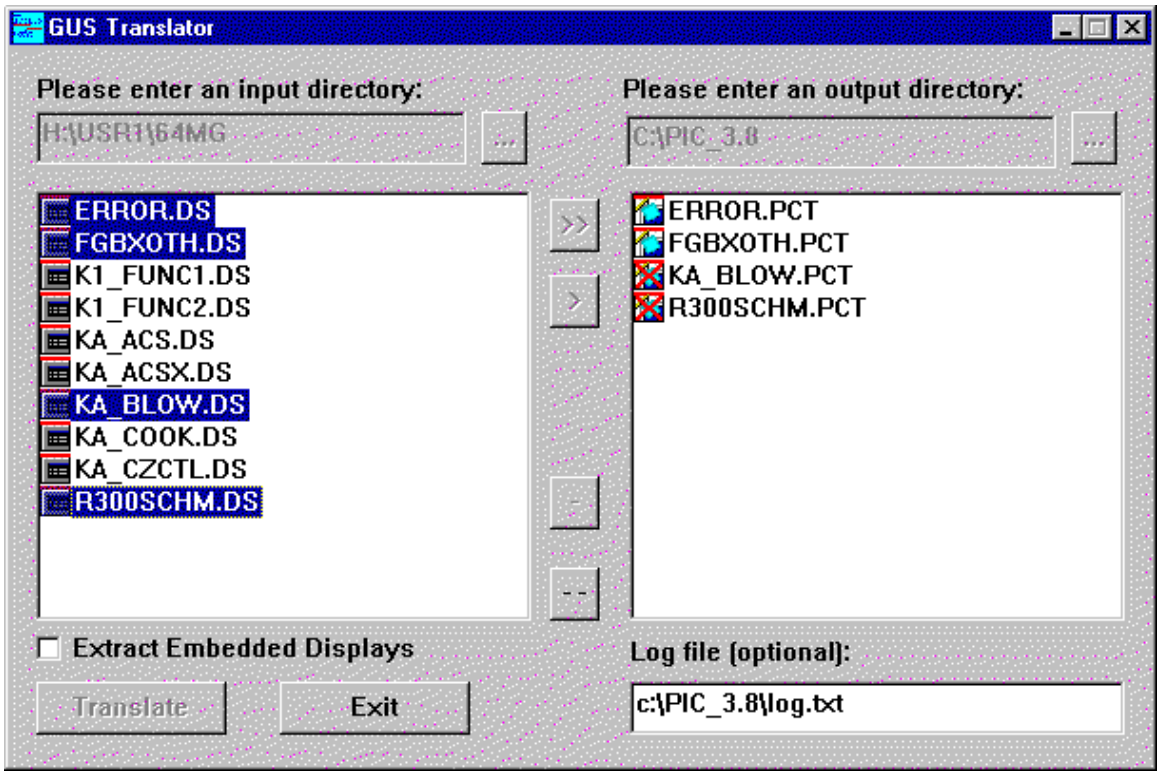
After entering the correct type, select OK to continue.

NOTE: The translator attempts to determine what enumeration or data type it is attempting to interpret, and displays that results as the first entry in the list.

6.3 Debugging Translator Errors

The following steps describe how to tell if a translation has failed and actions you can take to troubleshoot translation failures.

1 If for some reason, the translation failed on any of the files, a red 'X' will appear over the icon next to the translated file name in the output file column. Examine the log file (or the error message report) to determine what failed in the translation.



2 If the translation fails, here are some suggested troubleshooting steps:

- Re-translate display interactively, if indicated in the Log file. (Do not specify a log file.)
- If Phantom EOL is indicated, rewrite the .DS schematic in the TPS Picture Editor (no need to compile).
- Attempt to 'Verify' the .DS schematic in the TPS Picture Editor.
- Modify conditions or variants with ‘ifelse,' ‘thensub,' etc. to ‘if else' and ‘then sub,' etc.
- Delete offending object in the .DS file and retranslate. (Be sure to recreate the object in the GUS display.)

7 Validate and Save .PCT Files

7.1 Validation and Save Procedure

7.1.1 Steps to validate and save new .PCT files

Perform the following steps to validate and save each newly translated .PCT file.

Note: If the validation is unsuccessful, the save will not occur.

If you need additional details regarding the validation process, refer to the *GUS Display Builder User's Guide* (GU23).

Attention

The validation procedure compiles the script and verifies that the LCN references are properly defined on the TPN system to which your ES-T is connected.

Therefore, the validate function must be run before a display can be run.

1 Open GUS Display Builder:
Start Menu > **All Programs** > **Honeywell Experion PKS** > **TPS Applications** > **GUS Display Builder**.

2 Select the File menu, and then OPEN a translated .PCT file.

3 VALIDATE the .PCT file by selecting the File menu, then selecting
Save With Validation. If the validation is successful, you will see a message similar to this (It could say On-line or Off-line):

If validation fails, the file will not be saved. A failed validation would be indicated by:

Fatal error occurred during display validation

You can save a file that did not validate by selecting the File menu, then selecting Save As (or Save).

4 REPEAT steps 2 and 3 for all remaining .PCT files.

7.1.2 What is it validating?

The display validate function can be run either off-line or on-line. It

- compiles all scripts (object scripts, embedded display scripts, display scripts, etc.),
- checks syntax of the variable expressions on the object's dynamics (e.g., bar, rotate), and
- if the validation is performed on-line, validates external data references in all scripts (references of the form 'lcn' And 'dispdb').

7.1.3 What is not validated?

Data references passed to actor functions are not validated. If improper data references are passed into function calls, errors will be detected and annunciated at runtime.

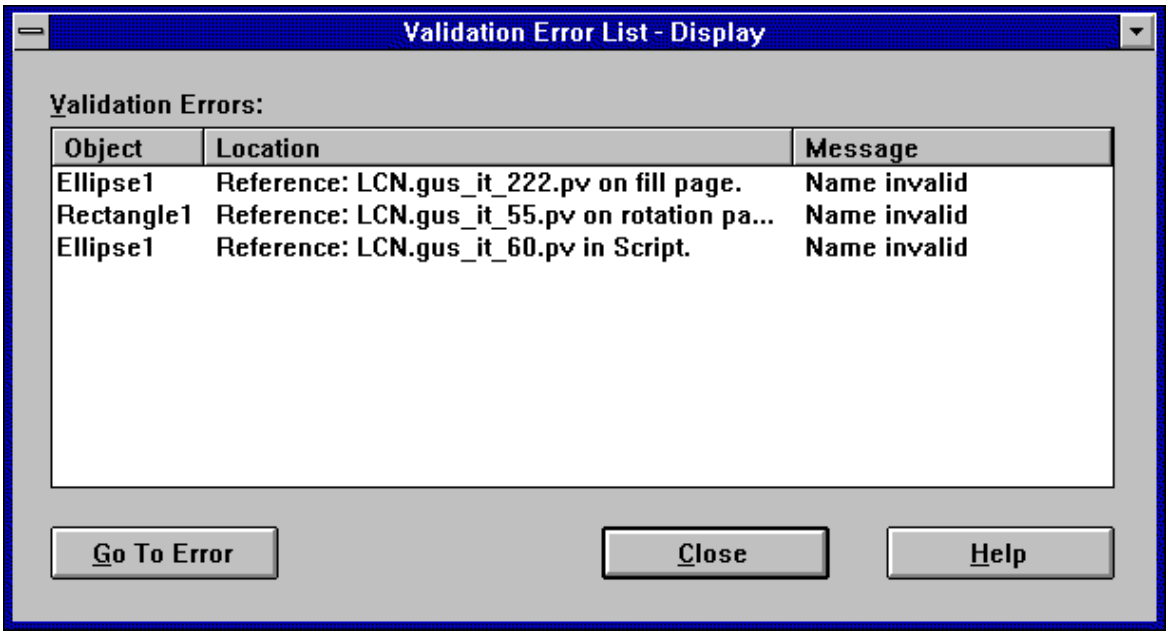
For example, DispDB.ENT01='JUNK' will validate, but will generate a runtime error.

7.1.4 Validation messages

When the display validation process has completed, one of these actions will occur:

If There Were…	You Will See This…	And the Status Bar Indicates…
Errors found during display validation	'Fatal error occurred during display validation'	Not Validated (see example below)
No errors found during display validation	'On-line [names resolved] validation'	Validated

If errors occur the **Validation Error List** displays automatically (see example in the following figure). To get more information on an error, highlight the error line, and click *G*o **To Error**. This should provide enough information so you can go back to the translated display and correct the error-causing conditions.



7.1.5 Validation Error List - data description

Below are the descriptions of the data in each column.

Column	Description
Object	The name of the object whose script and/or dynamic has the error.
Location	Identifies the data reference in error and the location of the error (either the script or which object dynamics).
Message	Contains a description of the error.

If you need additional details regarding the Validation Error List, refer to the *GUS Display Builder User's Guide* (GU23).

8 Verify Display Operation

8.1 Steps To Verify Correct Operation

The translated display and its components (embedded displays, variants and conditionals) must be validated and thoroughly tested before commissioning the display. The following **two tables** list the general steps.

8.1.1 To validate display subpictures or components

1. Translate the main display and all of its subpictures.
2. Open each subpicture in the builder.
3. Check for correct drawing and z-order of the graphics and text. Correct z-order by using the send backward/ send forward commands.
4. Open scripts for targets, conditionals and variants. Check for correct logic and data forms.
5. Modify the scripts for functions not supported by GUS displays (refer to the section on Cleaning Up Translated Displays).
6. Build a new GUS display with the modified subpicture and 'dummy' inputs.
   Validate and run this display, then thoroughly test the subpicture for correct target actions, and conditional/ variant response.
7. After validating the subpicture, replace the subpicture on the main display, if it was modified.
8. Verify the subpicture fits correctly in the main display, and in the correct z-order. Correct if needed.
   Depending on the results, a redesign of the subpicture components may be required. (Refer to the section on Object Priorities in Cleaning Up Translated Displays).
9. Repeat steps 3 through 8 for all subpictures.
10. Repeat steps 3 through 5 for all other components of the main display.

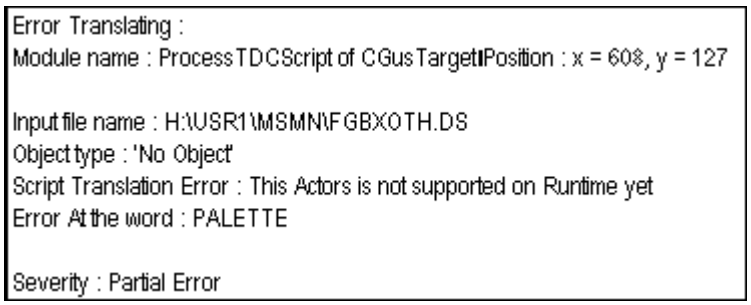
8.1.2 To validate the main display and test for correct functionality

1. Validate the main display, and correct any reported errors.
2. If it is not possible to test the display in an ‘off-line mode,' then open the display on a native window, and run the GUS display.
   Thoroughly check that the display is correctly reporting the plant conditions, as viewed by the native window.
3. If all looks well in the previous step, make adjustments, if possible, to the process via the native window, and check for the correct response from the GUS display.
4. Now execute the same change from the GUS display target.
   Verify the correct control action has taken place by examining the native window.
   Verify the GUS display also displays the correct information.

9 Interpreting Log File and Interactive Messages

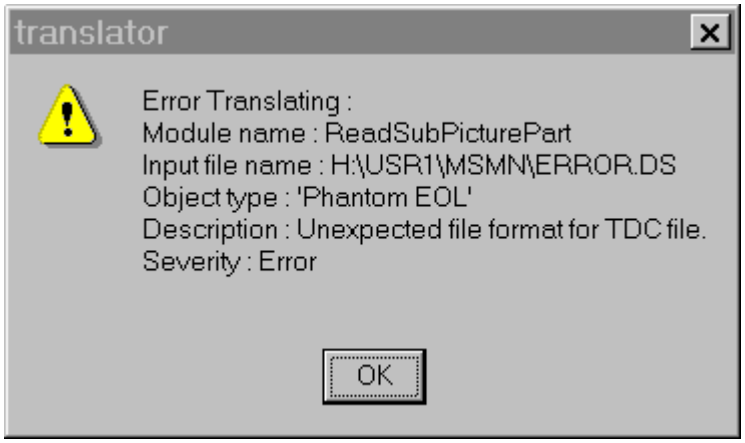
9.1 Overview of Log File Log File

If a log file has been designated, the Display Translator creates a text file that contains pertinent information about warnings and errors produced when attempting to translate a .DS file, and places that information in a log file.



9.2 Overview of Interactive Messages

If a log file is not specified, a dialog box containing information similar to that found in the log file is presented upon a failure during a file translation. The user must acknowledge the message before the translation can proceed.



9.3 Content of Error and Warning Messages

The log file and the interactive message dialog box contain the following:

- the input file name and path
- the type of object that was involved in the error, warning, or exception
- the GUS Display Builder x and y axis locations
- other information, depending on the type of failure (see the following table)

Note: If the severity for an individual file translation is **Partial**, the problem will not cause a translation failure. If the severity is **Error**, this problem caused the translation of that .DS file to fail.

9.3.1 Description of fields in log file and interactive messages

Field	Description
Error Translating :	File name
Module name :	Describes which portion of the translation process failed
Input file name :	Pathname and file name that has a problem
Object type :	The type of object being translated when the problem occurred
Position : x = y =	The .PCT x and y axis coordinates of the object being translated. To convert: x (in .DS file) = GUS x /8 y (in .DS file) = 28 - GUS y /16
Severity :	Could be one of two types: Partial Error (will not cause a translation failure) Error (will cause a translation failure)
Line number :	Not implemented. Currently always '0.'
Description :	A brief description what may have caused the failure. This field sometimes includes a suggested corrective action.

9.4 Log File Examples

The following are examples of log files and their contents.

Note: Not all fields are included in the example of each failure type. The example includes only the fields that contain information pertinent to that type of failure.

Example 1

```
Error Translating :
Module name : ReadSubPicturePart
Input file name : H:\USR1\MSMN\ERROR.DS
Object type : 'Phantom EOL'
Description : Unexpected file format for TDC file.
Severity : Error
```

Example 2 (Severity: ‘Partial Error')

```
Error Translating :
Module name : ProcessTDCScript of CGusTargetIPosition : x = 608, y = 95

Input file name : H:\USR1\MSMN\FGBXOTH.DS
Object type : 'No Object'
Script Translation Error : This Actors is not supported on Runtime yet
Error At the word : PALETTE

Severity : Partial Error
```

Example 3 (Severity: ‘Partial Error')

Error Translating :

Module name : ReadSubRef

Input file name : H:\USR1\MSM2\DT_ACK.DS

Object type : ‘SubRef'

The Trend Phantom is not yet supported, Object skipped.

Sub-Picture Name : DT_ACK

Position : x = 223, y = 130

Severity : Partial Error

Example 4 (Change Zone Subpicture)

Error Translating :

Module name : GetChangeZonePath

Input file name : G:\USR1\MSMN\CZ_COMT.DS

Object type : ‘No Object'

Description : A valid Change Zone sub picture could not be found. Please edit after translation.

Severity : Partial Error

Example 5 (Objects outside of display region)

Error Translating :

Module name : RepositionObject of CGusPoly

Position : x = 536, y = 47

Input file name : H:\USR1\MSM2\CEMBIG.DS

Object type : ‘No Object'

Description : This object spans the GUS valid region and invalid region. It will be translated to span the inactive region.

Severity : Partial Error

10 Cleaning Up Translated Displays

10.1 Methods For Clean Up

The methods you chose to correct or work around current translation functions will depend on your own particular displays and requirements.

The methods described here are suggestions for workarounds for items not currently supported by the translator.

10.2 Cleaning Up Object Priority Level

GUS Display Builder 'z-order layering' does not treat object priorities in the same way as the TDC 3000^X plane priority. TDC object priorities are determined by a hierarchy of object (valve, bar, variant) color and FGB priorities. In GUS displays, an object has only one z-order, therefore all of its components will exist on the same plane.

Subpictures and main display objects may require manipulation (using Send To Back/Front menu options) to correct priority errors. Priority problems will most likely occur in subpictures or variants.

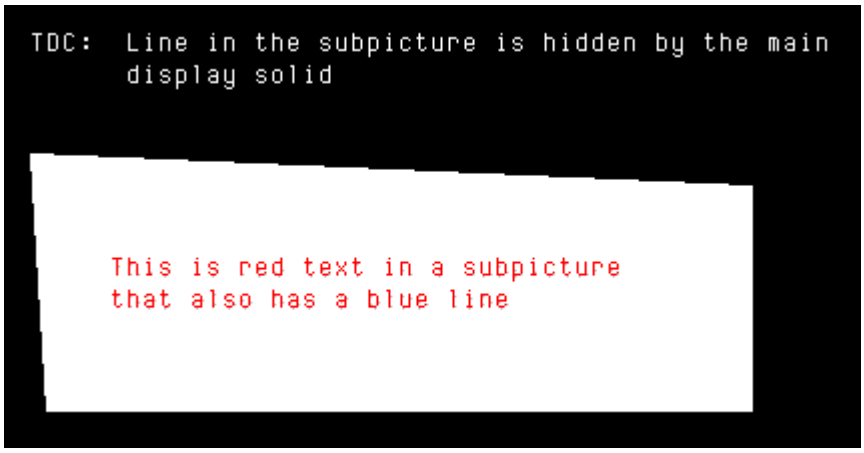
Following are two examples that compare the plane priorities with the object priorities.

10.2.1 Subpictures and embedded displays

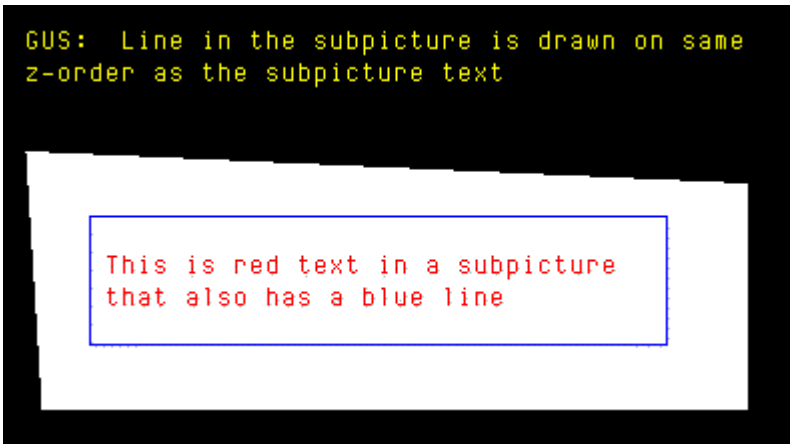
Some objects in a TDC subpicture appear above, and other objects in the subpicture are drawn behind main display objects, depending on the factors above. In a GUS display, all objects in the embedded display exist on the same z-order, therefore all objects in an embedded display will be drawn in front of (or behind) other objects in the display. (See the following example.)

Example:

A TDC 3000^X subpicture consists of a blue line and a red text value. The color priority for the whole subpicture is yellow (setting at the time subpicture was added). This subpicture overlaps a solid of color priority white. At design and run time, it will appear that the line in the subpicture is 'behind' the solid in the main picture, while the text in the subpicture is 'on top' of the solid in the main picture. (See below)



When translated, the new GUS display will give the entire embedded display the same Z-order. The embedded display's Z-order will be based on the highest object priority level for the TDC 3000^X subpicture (the text). So in the GUS display, both the text and line in the subpicture will appear 'on top' of the main picture's solid.

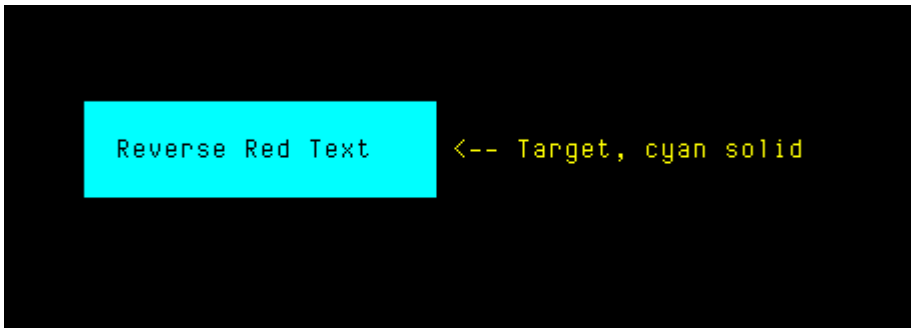


10.2.2 Reverse Video Text

Reverse video text objects are subject to the single 'z-order' rule as well. Background color and foreground text in GUS display are drawn on the same z-order. Main display items that would be inserted between the text and its background in TDC, will be drawn completely behind or completely in front of the text object (text and its background color) in a GUS display.

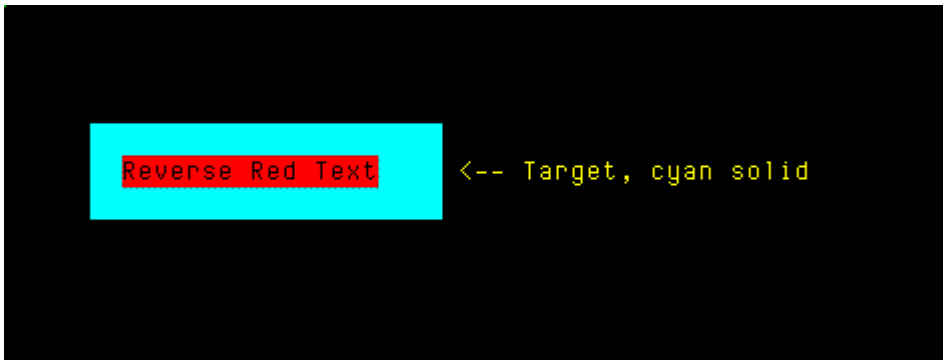
Example:

Reverse video red text is added over a solid target The TDC 3000^{X} hierarchy is first text (black, since it is reverse), then the solid target, then the reverse background (which is hidden by the target solid).



In a GUS display, the text background will be drawn as a red solid over the target. In this case, one option would be to delete the text (or value), and add it again as black text only. (NOTE: The reverse property is not currently accessible from the property sheets, and can only be set dynamically from a script.)

Another option that would improve performance by reducing the number of display objects is to save the target (rectangle) script, delete the rectangle and text object, and add a new text object with the appropriate colors. Then open a script for the text and paste in the original script.


Reverse Red Text
<-- Target, cyan solid

10.3 Cleaning Up Overlay, Mult_Ov, and Back_Ov

The overlay functionality is not supported.

Consider adding the overlay displays as embedded displays. The logic to make the correct embedded display invisible/visible and selectable/unselectable will need to be added to the calling scripts.

CAUTION

In the TDC 3000^{X}, the overlay and main display have separate local display databases (DDBs). Because the overlay is now treated as an embedded display, it will use the same local DDB variables as the main display, which can lead to potential conflicts.

The overlays can optionally be accessed through a separate GUS display (if multiple displays are configured). The logic to call the overlay (i.e. using InvokeDisplay) must be added to the main display. The overlay display will be initialized to the global DDB values of the calling display.

CAUTION

This is a one-time global DDB exchange only. Any subsequent changes to the global DDBs in one display will not be reflected in the other.

10.4 Cleaning Up Change Zone Subpicture

The standard TDC 3000^X change zone is not translated. If the translator encounters the change zone subpicture, a note is made in the log file that the change zone was not located.

A GUS change zone is provided under the C:\Program Files\Honeywell\TPS\RAC\chg_zone. The GUS change zone is the same 3 x 80 character size as the original (TDC 3000^X) change zone. If the change zone was used as a subpicture, or an overlay, insert the GUS change zone in its place.

Attention

The change zone uses black as the text or value color. If the display background is to be kept black, insert a solid (gray) rectangle, the same size as the change zone, behind the change zone. Add an 'OnDataChange' script to the rectangle of:

```
SubOnDataChange
If DispDB.[$cz_enty] = '' then
me.visible = false
Else
me.visible = true
End If
End Sub
```

Be sure to send the solid to the back (behind the change zone embedded display).

10.5 Cleaning Up Change Zone Actors

The User_cz actor is not supported in GUS displays. The chg_zone actor is translated as:

```
OnLeftButtonUp DispDB. [$CZ_enty]= 'Pointname' End Sub
```

The store to DispDB. $cz_enty makes the change zone visible.

10.6 Cleaning Up QUE_KEY

Currently, there is no good workaround for the Que_Key functions. If relying on Que_Key for pre-filled Text Input Ports (TIPs), the operator will now need to fill in the entire input. See the example below.

Example: The TDC 3000^X target opens a TIP of 8 characters and fills in $PRSTS and prompts for a 2 character node number, *nn*. In GUS Display Builder, the operator would have to type $PRSTSnn.

10.7 Cleaning Up Cursor, Tab, and Select Actors

Cursor, Tab, and Select functionality is not supported in GUS Display Builder. If cursor or tab with Select actors were used in a display (Example: to invoke a new overlay or start a trend), the target actions could be combined into a single script.

10.8 Cleaning Up Conditions and Variants Without LCN or DDB Access

Access to at least one LCN or DDB variable is required in an OnDataChange script for the script to be run, even on display callup. In this situation, these scripts should be changed to OnDisplayStartup. See the following two examples.

Example 1: A condition or variant using the code 'IF ON THEN SET BLACK' translates to 'IF TRUE = TRUE THEN SET BLACK' in the GUS Display Builder. This conditional sequence was generally used to hide an object at runtime that needed to be visible in the Picture Editor. It will not be executed in the GUS display.

As an alternative in a GUS display, add an OnDisplayStartup script that sets the object INVISIBLE and remove the OnDataChange conditional or variant code.

Example 2: A subpicture has an object with a condition of 'If &INT=1 then set red else if &INT=2 then set blue else…'. When added to the main display, the user enters an integer (constant) value. Since there are not any LCN or DDB variable references, the embedded display object's initial conditions will be drawn.

10.9 Cleaning Up Back Plane Functions

There is no workaround for backplane functions with the exception of the Keylock function, which has been implemented in standard TDC 3000^X actors (refer to the *ACTORS manual*).

10.10 Cleaning Up Trends

Build new OLE Trend Controls to replace the TDC trends.

10.11 Operational Differences between Displays

10.11.1 Guideline

Translated displays adhere to the fundamental rules for developing performant displays. If a translated display is running slowly in GUS display, refer to the Display Authoring Tutorial for techniques to improve performance.

10.11.2 Operational Differences between TDC 3000X and GUS Displays

There are several interface differences that operators need to beware of when using a translated display on a ES-T node.

The differences are described in the following table.

TDC 3000^{X} and GUS Display Operational Differences
Select key on the keyboard will not activate a script under the current cursor position.
Messages for Text Input Port (TIP) entries are displayed at the lower left 'bar' in the GUS Window. The TIP will be located in the upper left corner.
There are some differences in error message reporting.
Buttons use a dialog ('Windows') interface, not the TIP interface.
TIPs do not time out.
Pre-filled prompts will now have to be completely typed in (QUE_KEY not supported)
Bars accessing bad values do not show an X through the bars, but now have a wide red box surrounding the bar.
Fast update is once per second instead of two times per second.

11 TPS Actor Translation

11.1 Actors and Other Functions How They Are Translated

The following TPS actors, actions, commands, etc. are translated into a GUS script, or in some cases, are translated into a GUS-usable function, such as a script property access, script standard function, or a GUS recognizable character or actor.

Original Display Actor Name	Is Translated into…
DMD_UPD	Script Property Access (LCN.UPDATE 0)
UPDATE*	Script Property Access (LCN.UPDATE.....)

* Regions do not exist in GUS displays

Original Keyboard Event Action	Is Translated into…
ENTER	'QUE_KEY ENUM('$BUTTONS:ENTER')

Original Operator Input Actor	Is Translated into…
R_ENT	R_ENT_N

Original Store To DDB	Is Translated into…
S_BOOL	BasicScript assignment statement
S_DATE	BasicScript assignment statement
S_DUR	BasicScript assignment statement
S_TIME	BasicScript assignment statement
S_ENT	BasicScript assignment statement
S_INT	BasicScript assignment statement
S_REAL	BasicScript assignment statement
S_SENM	BasicScript assignment statement
S_STR	BasicScript assignment statement
S_VAR	BasicScript assignment statement

Original Read Data And Store into DDB	Is Translated into…
RS_LOC	each DDB type of R_xxx Actors

Original Read From DDB	Is Translated into…
G_BOOL	DispDB built-in object
G_DATIME	DispDB built-in object
G_ENT	DispDB built-in object
G_INT	DispDB built-in object
G_REAL	DispDB built-in object
G_SENM	DispDB built-in object
G_STR	DispDB built-in object
G_VAR	DispDB built-in object

Original Store To DDB	Is Translated into…
SS_BOOL	DispDB built-in object
SS_DATE	DispDB built-in object
SS_DUR	DispDB built-in object
SS_TIME	DispDB built-in object
SS_ENT	DispDB built-in object
SS_INT	DispDB built-in object
SS_REAL	DispDB built-in object
SS_STR	DispDB built-in object
SS_VAR	DispDB built-in object

Original Read From DDB	Is Translated into…
GS_BOOL	DispDB built-in object
GS_DATIME	DispDB built-in object
GS_ENT	DispDB built-in object
GS_INT	DispDB built-in object
GS_REAL	DispDB built-in object
GS_SENM	DispDB built-in object
GS_STR	DispDB built-in object
GS_VAR	GETVAR when translated - Remove from script
GS_VAR_S	GETVARID - Remove from script

Original General Actor	Is Translated into…
C_DATTIM	C_DATTIM
CONCAT	BasicScript ‘+' operator

Original Arithmetic Function	Is Translated into…
ADD_I	BasicScript ‘+' operator
SUB_I	BasicScript ‘-' operator
MUL_I	BasicScript ‘*' operator
DIV_I *	BasicScript ‘/' operator
MOD_I	BasicScript ‘mod' operator
NEG_I	BasicScript ‘(- ....)' operator
ADD_R	BasicScript ‘+' operator
SUB_R	BasicScript ‘-' operator
MUL_R	BasicScript ‘*' operator
DIV_R	BasicScript ‘/' operator
NEG_R	BasicScript ‘(- ....)' operator

* For the DIV_I, 0.5 is subtracted to round down.

Original Logical Function	Is Translated into…
AND	BasicScript AND operator

Original Logical Function	Is Translated into…
OR	BasicScript OR operator
XOR	BasicScript XOR operator
NOT	BasicScript NOT operator

Original Comparative Function	Is Translated into…
CMP_I	BasicScript comparison operator
CMP_R	BasicScript comparison operator
CMP_S	BasicScript comparison operator
CMP_E	BasicScript comparison operator

Original Conditional Function	Is Translated into…
IF	BasicScript 'IF' Statement
ELSE	BasicScript Statement
ENDIF	BasicScript Statement

Original Conversion Function	Is Translated into…
FLOAT	BasicScript
IE_ENT	External property of an entity DDB item
IE_VAR_E	External property of an entity DDB item
IE_VAR_P	Property access + EXTR_PAR
IE_VAR	External property of an ENT DDB item
TRUNC	BasicScript ‘FIX'
EI_ENT	External property of an ENT DDB item

Original KeyLock Function	Is Translated into…
$KEYRST	KEY_RST
$KEYCHG	KEY_CHG ENUM('KEYLEVL:ENGR')

Original Intrinsic Function	Is Translated into…
EXTERNAL	EXTERNAL property of a parameter object
INTERNAL	INTERNAL property of a parameter object
STATUS	STATUS property of a point object

12 Notices

Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.

12.1 Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

12.2 How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.

  or

- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

12.3 Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

12.4 Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.