

LASTIC USER INTERFACE

FORM BUILDER GUIDE

COPYRIGHT NOTICE

This document in whole or in part may not be reproduced or copied to any medium visual or non-visual without the express written permission of Lastic Limited.

Copyright 2001-2013 Lastic Limited.

CONTENTS

- 1. INTRODUCTION**
- 2. FORM EDITING**
 - 2.1 GUI Builder**
 - 2.2 Terminal Builder**
 - 2.3 Form details**
 - 2.4 Menus**
- 3. MAINTENANCE**
 - 3.1 Messages**
 - 3.2 Utility menu**
 - 3.3 Data types**
 - 3.4 Function table**
 - 3.5 Help**
 - 3.6 Look-up table**
 - 3.7 Button library**
 - 3.8 Validation codes**
 - 3.9 Validate Program Entrypoints**
- 4. CONTROL PROPERTIES AND EVENTS**
 - 4.1 Properties**
 - 4.2 Events**
 - 4.3 Event properties**

1.

INTRODUCTION

This manual describes the procedures for constructing and editing forms, including the various associated table maintenance facilities.

The Application program interface is described in the programmer's guide.

This manual describes both the character terminal and GUI versions of the form editors.

Environmental details maintenance, including highlighting, terminal types and character terminal presentation are described in the System Manager's Guide along with installation and software release facilities.

2. FORM EDITING

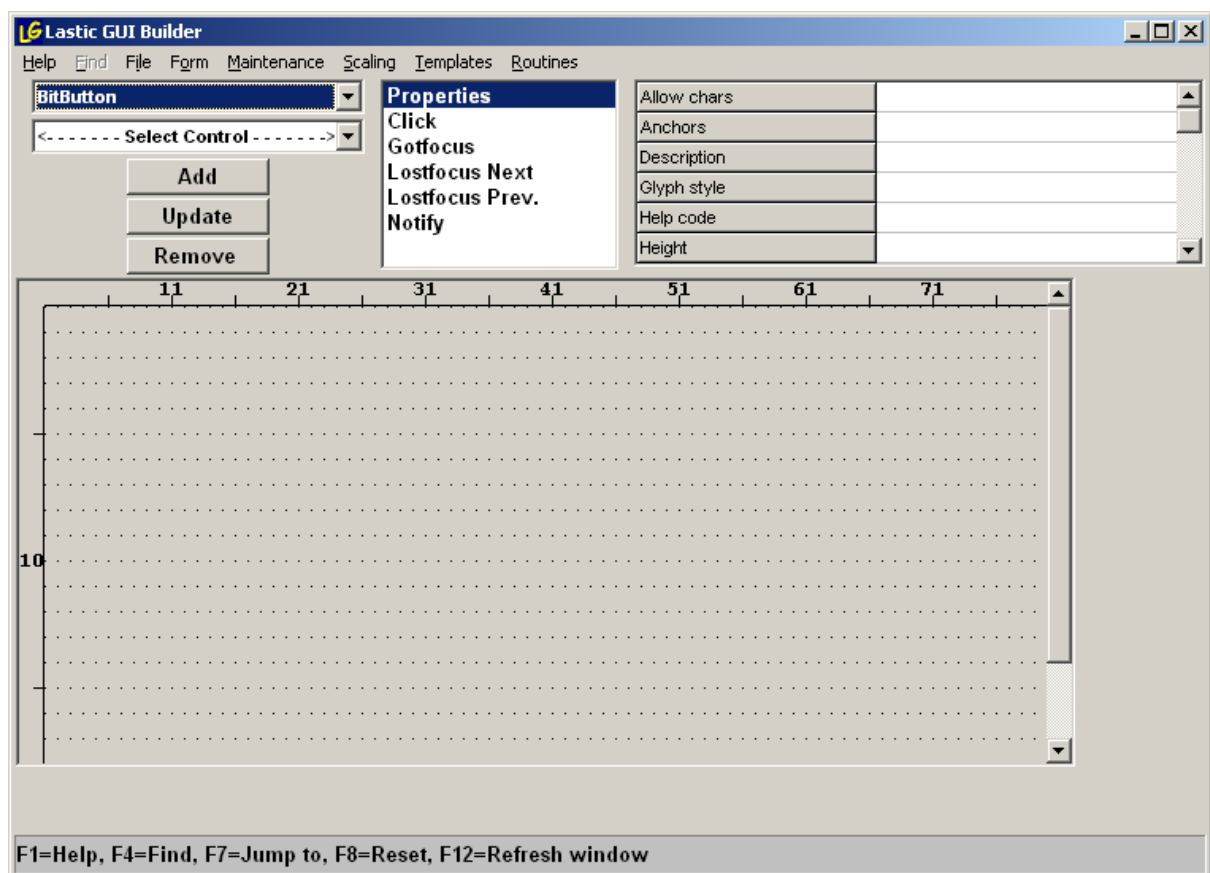
The system provides two form editors in order to utilise the benefits of the corresponding user interfaces though forms built within one environment can be used in the other with the exceptions of those controls which are peculiar to the GUI environment, namely, the Picture and Workbench controls.

2.1 GUI BUILDER

The GUI Builder is available for users with a GUI development licence and provides a WYSIWYG form development environment.

2.1.1 Starting the GUI Builder

To start the Lastic GUI Builder type **d ^l zgm** from within the emulator environment of Lastic GUI. The form builder is invoked:



2.1.2 Creating a new form.

Select "New" from the "File" menu. This will initialise the GUI Builder. You can then commence creating controls etc. as required.

2.1.3 Editing an existing form

Select "Open" from the file menu to obtain the form requesting the system and form to be loaded. Look-ups are available on these fields. Having located the required form press the "OK" button. The selected form will be displayed in the GUI Builder.

2.1.4 Editing form level information.

To change such as the form boundaries, form level events etc. select "Form details" from the Form menu. The details can be edited as required. If the form positions (top/left and bottom/right) are changed the program checks that sufficient room is available to hold the controls as designed. If not, the change is denied.

2.1.5 Updating the Form Database

Until explicitly saved, the changes you have made are not put back into the form database. To save a form, use the "Save" or "Save as" options from the File menu.

Save For existing forms only, selecting this option saves the edited form back against its original name.

Save as For new and existing forms, the "Save as" option requires definition of the system and form name against which it is to be saved. This also provides a means of replicating the form by saving it as a new name. (This does not remove the form under the previous name). If a new system is required this first must be created using the system maintenance described in chapter 3.

Delete To remove a form from the form database, select "Delete" from the File menu. Confirmation of the deletion will be requested.

2.1.6 Creating a new control

To enable access to the controls combobox and the "Add" button, ensure no controls are currently selected. To do this, click on an unused area of the workbench. Now select the type of control required from the combobox. This will cause the applicable options within the listbox to be displayed and also the properties within the grid.

The Lastic GUI builder provides many default values for a control to reduce the amount of editing required. To generate defaults, after selecting the required control type, click on the workbench to identify where the control is to be positioned and then click the "Add" button. This will result in the grid being populated with default values though if one or more mandatory values is missing an error will be displayed and focus set to the grid cell requiring action. You may now enter or edit the details required in the grid (including selecting the appropriate events and entering the information against those properties). When satisfied, press the "Add" button again and, if all is well, the control will be created and positioned in the workbench.

2.1.7 Changing a control's details

Selection of a control to amend can be performed by either selecting it using the mouse (left button click) or from the drop down list of controls (← Select Control → in the above snapshot)

An existing control may be changed using either the mouse or the property grid (depending upon the properties to be changed). The mouse can be used to change position and size though for all other properties the property grid must be used. Where controls overlap selection may be easier to locate the control via the drop down list – this is particularly relevant where splitter controls are, effectively hidden behind the foreground control.

Editing/Resizing using the Mouse

To select a single control, click the mouse on that control or select it from the drop down list. The details of that control will be shown above in the control combobox, the control list, property/event listbox and the property grid.

The control may be moved by holding the mouse left button down on that control and moving the displayed rectangle to the required position. The control may be resized by positioning the mouse around the border of the control, pressing and holding the mouse left button down and then dragging the displayed rectangle to obtain the required size. Some controls are not resizable (i.e. single line controls cannot be resized vertically, radiobuttons, checkboxes etc. are fixed according to the size of the prompt text).

When changing size or position using the mouse, the "Update" button is not required - the changes are made automatically.

Editing using the properties grid.

Having selected a single control using the mouse the properties/events may be selected as required from the list box causing the corresponding properties to be displayed in the grid. These properties may be edited as required and the 'Update' button pressed for those changes to take effect.

If any property detail is in error the changes will be rejected, a message displayed for that error and the

grid cell given focus to enable the correction to be made.

For properties with "codes" such as datatype, help, look-up, highlight, validation and function codes a search facility is available using the Lastic GUI standard look-up mechanism.

Properties requiring single key entries such as Y/N/L/R/C etc. may be entered in either upper or lowercase.

Many properties having multiple options can be viewed as a list using the look up facility. Use F4 or the Find menu.

2.1.8 Removing a control

Click on the required control and then press the remove button. This will produce a confirmation request messagebox. Confirming the removal will cause the control to be removed from the workbench.

NOTE: There is no undo facility.

2.1.9 Selecting multiple controls

It is possible to select more than one control at a time for moving or removal. To select more than one control press the mouse left button down above the "top/left" position of the first control (in an unused area of the form) and, holding the button down, drag the mouse to another unused area of the workbench such that the displayed rectangle encompasses all of the controls requiring selection. Upon release of the mouse the selected controls will be highlighted. To remove the selected controls merely press the remove button. To cancel the selection, click the mouse on an unused area of the workbench. To move the selected controls (you cannot resize controls when more than one is selected), press the mouse left button down while positioned on one of the selected controls, hold the button down and move the mouse such that the displayed rectangle is positioned where the controls are to be placed and then release the mouse button.

2.1.10 Menus

Creating/editing menus

Select the "Menus" option from the Form menu. This shows the first level form (i.e. the menubar entries). The order of menubar entries depends upon the collating sequence of the menu codes. To create a menu you must first create the menubar level. You may then select that level and press the "Edit" button to obtain the form for creating/editing the individual menu entries. Again the order of the menu entries are dependent upon the collating sequence of the menu codes. Details of the menu information is described in the next chapter.

Merging menus

It is possible to use the menus from another form. To do this, the menubar entry code created must correspond to the code used on the original form. check the MERGE FROM checkbox and then press the edit button. The form will then ask for the original system and form name. You can use a different menubar description with the merged menu though the contents will be identical.

2.1.11 Scaling and resizing

If the form being designed is too large for the workbench you can extend the size of the workbench using the normal Windows resize mechanism. This will automatically cause the workbench to resize. If there is insufficient room on your screen you can, alternatively, reduce the scale of the workbench using the "scaling" menu.

2.1.12 Templates

The template facility provides a very powerful means of rapid development. In many cases, a control is to appear on a number of forms (e.g. a customer number request). This control will have standard prompt, help, look-up facilities etc.

To simplify and standardise the approach, once such a control has been created you can select that control, select "Create" from the template menu and save all the details for that control as a template (like a finished control library) with a name and description to suit.

When a control is required which was saved as a template, select "Load" from the Template menu and locate the required template (a look-up facility is provided). When located, the full details of that control is populated into the property grid (including event processing). All you have to do is click on the workbench to identify where it is to be placed then press the "Add" button.

2.1.13 Routines

When developing event driven applications care should be taken to ensure variable usage is encapsulated within each separate routine. Work variables should be 'newed' to ensure there is no overwriting of comparable variables and also that the newly created variables are removed upon completion of that routine.

Lastic GUI and Lastic Terminal Builder now provide facilities to analyse code to determine where variables are set for this purpose.

The code checker enables code to be pasted into a document control which will then show all variables (names only) which are being set within that code. It excludes variables commencing with '%lz'.

The Scan routine enables selected routines (including wild cards (e.g. abc*) to be selected along with a list of variable names. This checker will then scan all those routines and will list all occurrences of where any of the variables are set within those routines. It uses the routine global as defined within the system set up (as used for distribution) to identify M routines.

2.2 TERMINAL BUILDER

The Terminal Builder is available for users with a character terminal development licence and provides a WYSIWYG form development environment. The nature of character terminals means that there are no mouse style facilities

2.2.1 Starting the Terminal Builder

To start the Lastic Terminal Builder type `d ^lztm` from the Mumps programmer prompt. This displays the initial form which allows creation and amendment of existing controls on a form.

2.2.2 Creating a new form.

Select "New" from the "File" menu. This will initialise the Terminal Builder. You can then commence creating controls etc. as required.

2.2.3 Editing an existing form

Select "Open" from the file menu to obtain the form requesting the system and form to be loaded. Look-ups are available on these fields. Having located the required form press the "OK" button.

Controls can be created, amended or deleted from the base form as displayed. To operate the PAINT edit feature, select "Paint" from the menu. This will show the existing form enabling controls to be created, amended, removed etc.

2.2.4 Editing form level information.

To change such as the form boundaries, form level events etc. select "Form details" from the Form menu. The details can be edited as required. If the form positions (top/left and bottom/right) are changed the program checks that sufficient room is available to hold the controls as designed. If not, the change is denied.

2.2.5 Updating the Form Database

Until explicitly saved, the changes you have made are not put back into the form database. To save a form, use the "Save" or "Save as" options from the File menu.

Save For existing forms only, selecting this option saves the edited form back against it's original name.

Save as For new and existing forms, the "Save as" option requires definition of the system and form name against which it is to be saved. This also provides a means of replicating the form by saving it as a new name. (This does not remove the form under the previous name). If a new system is required this first must be created using the system maintenance facility described in chapter 3 below.

Delete To remove a form from the form database, select "Delete" from the File menu. Confirmation of the deletion will be requested.

2.2.6 Creating a new control

If using the Paint facility, press the "application" function key to obtain the options form. Check the "controls" radiobutton then press the 'OK' button, This will invoke the controls form. If not using the paint facility, leave the control name blank on the base form and move to the control type combobox.

Select the control type from the combobox. The listbox and grid will change to reflect the facilities available for the control type you have selected.

You can generate defaults for the properties by pressing the "Add" button. This will fill in all defaults in the property grid and (for character terminal builder) will also attempt to find a position in the form which can hold the control. Note, the control will not be added until all mandatory properties have been defined.

You can then return to the grid to change or fill-in any information required.

2.2.7 Changing a control's details

Facilities are provided within the "Paint" screen to both move one or more controls or to edit the properties of a selected control.

Moving the control

To select a single control, use the cursor keys to position on that control, press the "application" function key and obtain the options form. Press <CR> on the "Move" radiobutton and again on the "OK" button. The options form disappears and the paint screen is in "drag/drop" mode with the cursor positioned at the top/left of the selection.. Move the cursor to the new "top/left" position and then press <CR>. This will cause the control to be moved to the new position.

Editing using the properties grid.

Select a single control and invoke the options form as described above. Now check the "Edit" radiobutton and press the "OK" button, The control details form will be displayed for this control. The properties/events may be selected as required from the list box causing the corresponding properties to be displayed in the grid. These properties may be edited as required and the 'Update' button pressed for those changes to take effect.

If any property detail is in error the changes will be rejected, a message displayed for that error and the grid cell given focus to enable the correction to be made.

For properties with "codes" such as datatype, help, look-up, highlight, validation and function codes a search facility is available using the Lastic standard look-up mechanism.

Properties requiring single key entries such as Y/N/L/R/C etc. may be entered in either upper or lowercase.

To obtain details of the value of a property where that property is a "code", press the <CR> key when that cell has focus. The details of that "code" will also be displayed in the messagebar at the foot of the main form.

2.2.8 Removing a control

Select a single control and invoke the options form as described above. Check the "Remove" radiobutton then press the "OK" button. This will produce a confirmation request messagebox. Confirming the removal will cause the control to be removed.

NOTE: There is no undo facility.

2.2.9 Selecting multiple controls

It is possible to select more than one control at a time for moving or removal. To select more than one control press the "application" function key to invoke the options form. Check the "Change selection" radio button which enables the "Mode" controls. Now check the "Clear" selection checkbox and "Select group" radiobutton then press the "OK" button. The Paint operation is now in group selection mode. To select members of a "group", position on a control and press <CR> (if you move to another control; you will see that the control has changed highlighting to indicate that it has been selected). A control can be deselected by pressing <CR> a second time on that control.

You can move to which ever controls are to be included in the group pressing <CR> on those to be included.

When selection is complete, press the "application" function key again to invoke the options form. You can now select either the move or remove options and press the "OK" button.

If moving, the editor considers the groups of controls as a logical rectangle and positions the cursor at the top/left of that rectangle. moving the cursor to the new position, press <CR> again to re-position those controls. Remember, after moving, the group is still selected thus, unless a second move is required, you should then invoke the options form again, check the "change selection" radio button, check the "clear selection" checkbox and check the "select current" radiobutton. This will reset the selection mode to the current control and will also remove the group of controls from the selection. If deleting, confirmation of the deletion will be requested. If confirmed, the group of controls will be removed. If not confirmed, the group remains selected and you may need to clear the selection as described above.

2.2.10 Menus

Creating/editing menus

Select the "Menus" option from the Form menu. This shows the first level form (i.e. the menubar entries). The order of menubar entries depends upon the collating sequence of the menu codes. To create a menu you must first create the menubar level. You may then select that level and press the "Edit" button to obtain the form for creating/editing the individual menu entries. Again the order of the menu entries are dependent upon the collating sequence of the menu codes. Details of the menu information is described in the next chapter.

Merging menus

It is possible to use the menus from another form. To do this, the menubar entry code created must correspond to the code used on the original form. check the MERGE FROM checkbox and then press the edit button. The form will then ask for the original system and form name. You can use a different menubar description with the merged menu though the contents will be identical.

2.2.11 Templates

The template facility provides a very powerful means of rapid development. In many cases, a control is to appear on a number of forms (e.g. a customer number request). This control will have standard prompt, help, look-up facilities etc.

To simplify and standardise the approach, once such a control has been created you can select that control, select "Create" from the template menu and save all the details for that control as a template (like a finished control library) with a name and description to suit.

When a control is required which was saved as a template, select "Load" from the Template menu and locate the required template (a look-up facility is provided). When located, the full details of that control is populated into the property grid (including event processing). All you have to do is click on the workbench to identify where it is to be placed then press the "Add" button.

2.2.12 Routines

When developing event driven applications care should be taken to ensure variable usage is encapsulated within each separate routine. Work variables should be 'newed' to ensure there is no overwriting of comparable variables and also that the newly created variables are removed upon completion of that routine.

Lastic GUI and Lastic Terminal Builder now provide facilities to analyse code to determine where variables are set for this purpose.

The code checker enables code to be pasted into a document control which will then show all variables (names only) which are being set within that code. It excludes variables commencing with '%lz'.

The Scan routine enables selected routines (including wild cards (e.g. abc*)) to be selected along with a list of variable names. This checker will then scan all those routines and will list all occurrences of where any of the variables are set within those routines. It uses the routine global as defined within the system set up (as used for distribution) to identify M routines.

2.3 FORM DETAILS

To change form level information, select the "Form Details" from the "Form" menu in the editor. Details of the form's fields are given below and are also provided within the help facility.

Form details

Help Find

Top left 5;10 Bottom right 23;80 Frame Y Sequencing M

Version no. 2.60 Timeout override Resize ctrl type

Help code %gblzgf ☐ Hide help for invisible fields

Event entrypoints

Pre-form

Application

☐ Use RightMouseClicked

Pre-exit

Menu details

Disable utilities Y

Inherit Options

Show Menubar

Text form details

Variable

Chars

☐ Word-wrap ☐ Read only

Character input fields

Start End

Extent Look-up

Highlight codes

Form 0 Focus 10 Skip 16

OK Cancel

F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window

- Top left** The location of the top left hand corner of the form within the overall screen, in the format of line;column (\$y;\$x in Mumps terminology). The line number cannot exceed the maximum number of lines viewable on this screen nor the width exceed the maximum number of columns viewable on this screen. The GUI form builder obtains the metrics of the screen and also of the line and column size parameters from the set up file and determines the line and column maximums from these. The character terminal form builder obtains the information from the general set up.
- Bottom right** The location of the bottom right hand corner of the form within the overall screen in the same format as the start position. Range values apply as for the start position.
- Frame** 'B' = Banner form – No frame, No caption, No system menus, No application Menus
 'F', 'D' = fixed / dialog box frame, Caption, System menus, No Application menus
 'N' = Single line frame (None in char terminal), Caption, System menus, application menus
 'Y', 'R' Resizeable frame, Caption, system and application menus
- Sequencing** This may be either 'A' for automatic or 'M' for manual. When set to automatic, interactive controls are automatically sequenced, in the order of line/column. Thus focus will be given to all controls with their top/left position on line 1 before line 2 etc. and, in start column number sequence within each line. If set to 'M' the sequence number will be requested when defining the individual controls. It is suggested that a form is always set to 'A' initially and, if any changes are then required, after creating the form and it's control details, this is changed to 'M' and the required changes made to the respective controls. If this field is set back to 'A', then upon amending any control, all sequences will be regenerated back to the default sequence for that form.

Version no A free-format coding to permit a version number of this form to be held. It is intended that such a version should be maintained comparable to the associated application program version.

Timeout override An optional field providing an override of the system standard. Time out values, if specified may be in the range 0 to 99 seconds - a much lower range. The primary objective of this override is to enable forms to be periodically refreshed or to be used to monitor keyboard process abort requests.

Resize ctrl type May be set to D (Document), G(Grid), N(Outline) or W(Workbench) to indicate the control type to be resized along with the form in the GUI environment. It causes all controls of that type, which are the full width of the form, to automatically resize horizontally when the form is resized. It will also resize the height of the lowest control of that type. The form may also be resized in only one direction by entering "V" (Vertical) or "H" (Horizontal) after the Control Type. Alternatively, the 'anchors' property of the various controls on the form may be used and the form resizing capability enabled by entering a hash (#) instead of the control letter. V or H may also be appended to this character.

Help code The help code reference, if required, for form level help information. It must be an existing code. Look-up facilities are available.

Hide Help Checking this box will cause the help display to ignore help information for controls which are invisible to the user at run-time.

Event Entry-points

Pre-Form Optional label^routine specifier enabling a routine to be called (via a "do") immediately after the form has been loaded and displayed and before any control receives focus.

Pre-Exit An optional label^routine specifier enabling a routine to be called (via a "do") before this form is terminated. When triggered, %lzsex (exit status) has already been defined. It is possible to cancel a form termination by setting %lzfoc in the pre-exit event or by re-setting focus (see programmers guide).

Application An optional label^routine specifier enabling a routine to be called (via a "do") from any point within the form activity when the user presses the "application" function key. The program is presented with %lzitem containing the current control having focus and enables control level activities to be performed as required.

Use Right Mouse Click

Check this box if the application event can be fired from a right mouse button click. NB This only applies to Grids, Listboxes and Outline controls.

Menu Details

Disable Utils Enter a Y (y) here to inhibit access to the utility menu from this form otherwise set to N (n) or leave blank.

Inherit options If set to "Y" this form will, at run-time, inherit the menus of it's latest ancestor form which has it's own menus. It only applies if this form does not have any menus of it's own. It cannot inherit the menus from a form which also has inherited it's menus.

Show menubar If set to 'Y' then menubar, for character terminals, will always be displayed. If 'N' the menubar is displayed only when the menu function key is pressed.

Text Form Details

Variable If a text processing form is required this field must be set to a variable reference within which

the text (as array elements) will reside. A form cannot be changed to or from a text form. Once established as a text form it will remain so.

Chars For text forms only, this item enables a list of characters, entered as ASCII values separated by commas, to be allowed through the reject character filter. It can be used, for example, to allow the standard field delimiter through for a text form.

Word wrap Requested for text forms only. If auto word wrap is required, set this check box to "on".

Read only For text processing forms only, this check box, if set "on" will not permit any editing of the text displayed though cursor up/down and paging is provided.

Character input fields (applicable to character terminal operation only)

Start If entered, overrides the standard start character for input fields

End If entered, overrides the standard end character for input fields

Extent If entered, overrides the standard extent character for input fields

Look-up If entered, overrides the standard end character for input fields with look-up facilities.

Highlight Codes (applicable to character terminal operation only)

Form A highlight code in the range 0-19 to define the foreground and background colour or attributes of the form. It defines both the box line colours and the form 'fill-in' background colour. A default is provided and a look-up list is available.

Focus This defines the code to be used to highlight the current interactive control. It is utilised to enable a fast, visual recognition of the current field for input. Any highlight code may be utilised but it is recommended that a highlight is the reverse colour combination to the normal. Thus, if green on white is used for the default input field then a reverse code would give white on green. This format provides the most visually pleasing presentation.

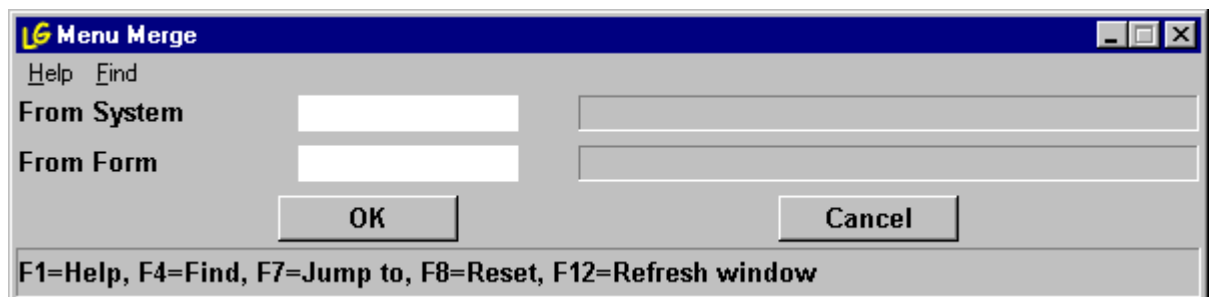
Skip This defines the highlighting required for controls which have been disabled using the skip facility. If not set then disabled controls appear unchanged.

2.4 MENUS

Menus consist of a menubar containing one or more menus and, for each menu, a number of entries as required. The menubar maintenance is invoked when the menu maintenance facility is invoked.



- Code** This is a free form reference to those menus defined for this form and a look-up is available on those items. It becomes the primary menu title for a list of facilities.
- Title** Upto 20 characters to appear as the menu header. For Character terminals, the sum of the menu titles plus, if enabled, the utilities menu header, cannot exceed the width of this form.
- Merge** If checked, the edit facility maintains a source system and form for the selected menu header. This mechanism forges a link to the source form though the title is given for this form. It enables menu sharing thus removing duplication. The menu header code must exist for the "from" system and form.



NOTE care should be taken with the merge facility in that the link is established at design time but is used dynamically. If you do not issue the source form then the link cannot be established at run-time. For this reason we recommend either creating a specific "dummy" form which will hold a number of menus for use by that application system and which is always distributed with that system, or make a dummy entry in your program to effectively force auto generation of the source form:

```
;d form^lzsrl(source_system, source_form,"000000")
```

Edit

For non-merged menus, the maintenance form as shown below will appear.

- Option code** This is a free-format code to define the menu option and also the sequence of appearance. Codes collate in the normal Mumps collating sequence thus numerics appear before alpha and uppercase appears before lowercase. You may thus structure the sequence of the menu entries utilising this code.
Recall of existing parameters can be made using the look-up facility.
- Description** The name to appear in the menu. For a "splitter" bar enter a single hyphen ("-").
- Call entry** The application program entry point consisting of an optional label plus the program name. The program must be preceded by the '^' sign.
- Switch variables** An optional list of variable names, delimited by commas and limited to a maximum of 1 subscript level, may be defined. If any variables are defined then this menu entry is available to a user only when all variables in this list exist within the run-time application data and have non-null values. If an entry is not available it will still appear in the menu but will not be selectable and will appear "greyed out".
- Not Avail Msg** An optional message code identifying the message to be used instead of the standard one if this option is not available at run-time due to missing switch values. Such a message must be a type 1 (hold and display).
- Check variable** If this menu entry requires "checking" specify a variable here (local/global with optional subscripts). At run-time, if the variable exists and has a non-null value, the entry will appear checked with, for character terminals, an asterisk appearing before the entry description.
- Hide variable** If this menu item is to be made invisible under certain circumstances then assign a variable (with an optional subscript) to this field. If, at run time, such a defined variable does not exist or has a null value then the menu item will be rendered invisible. If no hide variable is defined here then the menu item will always be visible but may be disabled via the switch variables above. If a menu item is made invisible via this mechanism then the switch variable details will not make it visible (even if enabled).
- Bitmap Index** if a bitmap (Icon) is to appear against this menu item enter the bitmap index number (1-999) to

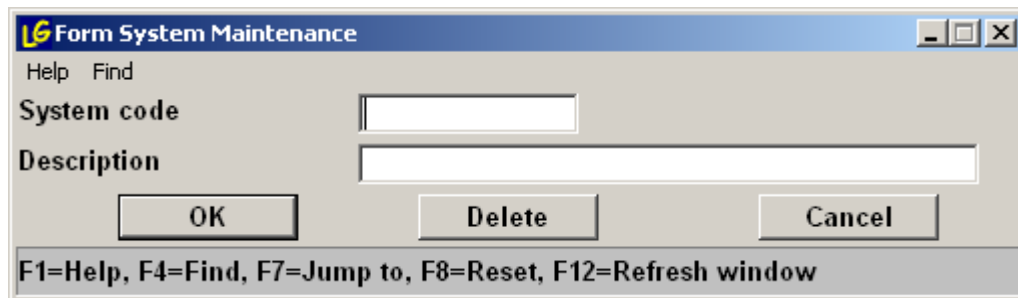
be used. The index number is that reference used when the bitmaps are loaded into Lastic GUI.

Parent Item This is a display only field and identifies, for child menu items, the parent menu item.

Child Items This will list those existing menu items within this main menu which may be (or are, if selected) child items of this menu entry. Select (or de-select) those items required. Items which are children of another item or are themselves parent items will not be listed. A menu item may not be a child of more than one parent nor may a parent also be a child of another.

2.5 FORM SYSTEM MAINTENANCE

This enables a new form system to be created or an existing one to be edited. When selected from the Form menu, the following form is displayed.



The image shows a Windows-style dialog box titled "Form System Maintenance". It has a blue title bar with a yellow "L" logo. Below the title bar, there are menu items "Help" and "Find". The main area contains two labels: "System code" and "Description", each followed by a text input field. Below these fields are three buttons: "OK", "Delete", and "Cancel". At the bottom of the dialog, there is a status bar with the text "F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window".

System code

Enter the code for the system (or use the look-up facility to select an existing system).

Description

This will be displayed for existing systems. For new systems a description must be entered. It is used purely for information.

3. MAINTENANCE

This chapter describes the maintenance facilities available within the Lastic User Interface system applicable to form construction.

Maintenance facilities for the "environment" and for terminal types etc. will be found in the System Manager's Guide.

To access the facilities described below, type d ^lzmnu at the Mumps prompt, enter your user_id as requested then select from the menu function provided.

3.1 MESSAGES

The Lastic User Interface provides a message table which is used by applications to inform a user of an error or process status. Such messages are displayed along the message bar on line 24 of the screen or in the messagebar of the form in the GUI environment. To display a message, the application will **set %lizer="message_code"** or to a text string containing the message and it's activity within an event.

The message maintenance facility enables frequently used messages to be defined rather than entering the full details each time within their applications. The messages details to be defined including highlighting requirements and severity level.

The screenshot shows a 'Message maintenance' window. It contains the following elements:

- Title Bar:** Message maintenance
- Buttons:** Help, Find
- Fields:**
 - Code:** A text input field.
 - Description:** A text input field.
 - Message:** A text input field.
 - Type:** A dropdown menu.
 - Highlight code:** A dropdown menu.
 - Display time:** A dropdown menu.
- Action Buttons:** OK, Cancel, Delete
- Status Bar:** F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window

Code Messages are coded so that the code described above only need be presented to the Lastic User Interface by the application. However, should the developer forget to define such a message coded in an application then the code itself is displayed instead of the message.

Description The description is used for look-up/confirmation purposes

Type The type of message is designated by one of three code values.

0	No message display but hold on current control
1	Display message and hold on current control
2	Display message but continue to next control

Display time An optional field, for use with type 2 above. A time (in seconds) may be set forcing this message to be retained for that period of time (as a minimum).

Highlight code The highlight code (0-19) corresponds to one of a range of colour/shading/other attribute combinations which are defined specifically by terminal type to allow for terminal capability differences. This feature enables, for example, error, warning and status messages to be differentiated by presentation. A default value is generated based upon the 'type' entered. For error messages, if a 'bell' is required then, within the selected highlight code, an ASCII 7 value should be included.

3.2 UTILITY MENU

The utility menu is a menu which may be invoked from virtually any task using the Lastic User Interface. It may not be invoked if the user is already operating a utility triggered from this menu - that is, utilities may not be layered. This facility enables entries to be added, removed or modified within the utility menu.

Utilities maintenance

Help Find

Utility code

Description

Call entry

Switch variables

Not Avail message

Check variable

OK Cancel Delete

F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window

- | | |
|-------------------------|---|
| Utility code | This is any code defining the utility menu entry. Functions are displayed in alphabetic order of this code. |
| Description | Used in the utility menu for selection purposes. It should therefore be specified in a consistent manner suitable for such a presentation. |
| Call entry | The label^program which will be invoked should this option be selected through the pop-up utility menu. |
| Switch variables | This may contain one or more variable names (max. of 1 subscript), delimited by commas, which are used to determine if this entry may be selected.
All variables listed must exist and have non-null values in order for that item to be available. If this is not the case then the entry will still be listed though disabled. |
| Not Avail msg | Instead of using the standard message a specific message can be linked to this utility option to identify the reason why the requested facility is not currently available. |
| Check variable | If this menu entry requires "checking" specify a variable here (local/global with optional subscripts). At run-time, if the variable exists and has a non-null value, the entry will appear checked. |

3.3 DATA TYPES

Data typing enables database fields to be defined in terms of size, validation, help, look-ups etc. such that, when creating a control on a form, the appropriate information can be loaded automatically into the editor property grid thus enabling standardisation of such form controls.

This facility enables data-type details to be created, amended or deleted.

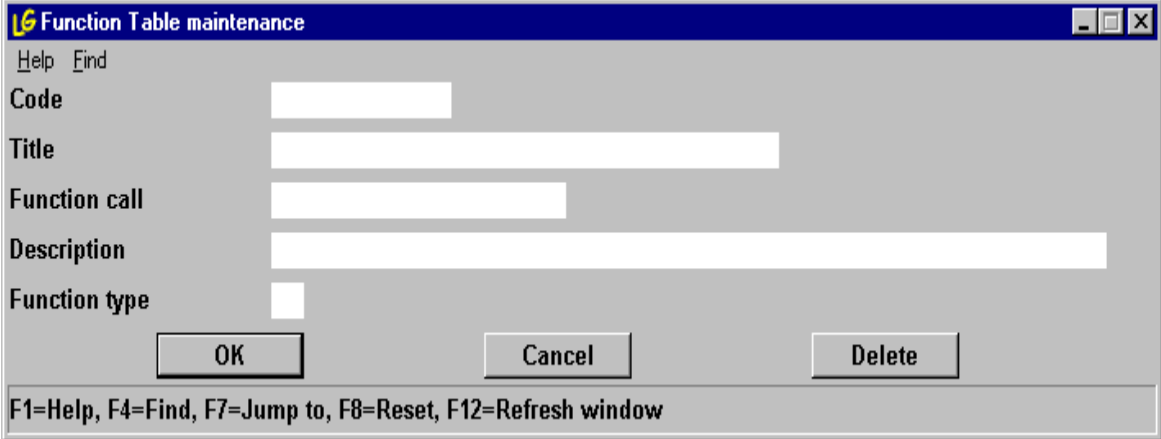
Data type	The data-type code to be maintained. A look-up facility is available on this field.
Description	For look-up and confirmation purposes, a description of the data type must be entered.
Prompt	The prompt field value is automatically provided as a default for the text property.
Width	The maximum number of characters for this item.
Translate	Optional, may be either LU (lower to upper) or UL (upper to lower). Again, this is used for defaulting for corresponding input items.
Help code	A help text code to provide a default for corresponding input items. A look-up facility is provided with this item.
Function code	An input function code (see Function tables below) to provide a default for corresponding input items. A look-up facility is provided with this item. Input functions have only 1 parameter (input value).
Look-up code	A look-up code to provide a default for corresponding input items. A look-up facility is provided with this item.
Validation	
Pattern	Optional pattern for input field validation. Only one pattern is available, entered in the normal Mumps fashion as the 'string' to follow "?" such that, if true, the validation succeeds. For example, the entry of: 1A4N1"/"5N would create, at run-time, a pattern match test of: i %Izsv?1A4N1"/"5N
Function	A validation function code (see Function tables below) to provide a validation mechanism within the data type parameters. Validation functions differ to input functions in that they have two parameters (1 - input value, 2- data-type code).

3.4 FUNCTION TABLE

Functions are available both within datatype validation and also as separate library procedures which can be invoked within control event operations.

This facility enables a list file to be maintained of such callable functions.

This is separate to the Validation codes functions which are described in a later section.



The image shows a Windows-style dialog box titled "Function Table maintenance". It has a menu bar with "Help" and "Find". Below the menu bar are five text input fields labeled "Code", "Title", "Function call", "Description", and "Function type". At the bottom of the dialog are three buttons: "OK", "Cancel", and "Delete". A status bar at the very bottom contains the text: "F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window".

Code Free text format code for uniquely identifying this particular function across all systems.

Title Free text for confirmation or look-up assistance.

Function call The label and program name of the called function. Within the called routine the entrypoint should be coded in the form: **label(x[,y])**
Where x is an application function variable containing the input field value and y a second parameter, for validation type functions only, containing the data type code. Upon return the function must quit with a value as the error code. For further details of function programming please see section 4.3.

Description A free form narrative for further identification of this particular function.

Function type If data types are not in use this will be forced to a value of "I" (for input) and not requested. If data types are in use then enter either "I" (for Input types - single parameter of item value) or "V" (for Validation types - dual parameters of item value and data type code).

3.5 HELP

Help information can be applicable at application, form, control and grid cell levels. Help texts are defined against reference codes which can then be linked to individual applications, forms, controls or grid cells.

This facility enables help text to be defined or amended.

Help text maintenance

Help Find

Help code [] Title []

Help text []

Linked help [] []

See Also

Help code	Description

OK Cancel Delete

F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window

Help code A code to identify this help text within this system.

Description Free text for confirmation and selection through look-up lists.

Help text Entered in a document control, the text is wrapped and as many lines as required may be entered.

Linked help It is possible to link a secondary help item to this help text. It is recommended that help texts defining the format of the response are created for database value "types" (i.e. dates, account codes etc.). When creating specific data items within a form, the help for such an item is context sensitive. This help can then be given the code of the help text of the format information. The help system will then automatically pick up the linked text and append it to the primary help text for the display thus providing a context sensitive text followed by the format help details.

See Also This grid enables help codes to be entered which will be appended to the help text as hypertext links. A look-up facility is available for help codes. Enter a code then press <RETURN>. The description will be displayed for confirmation. To remove a link, clear the helpcode cell and press <RETURN> the description will be removed and, if not the last row of the grid, the defunct row will be removed.

These links will only be available to the user when the "Link Help" checkbox in the system parameters has been checked (see scrm/general) to invoke the new help facility.

3.6 LOOK-UP TABLE

This function enables look-up list build routines to be encoded for ease of incorporation into various application forms. Look-up codes are available across all systems. The form below is used to maintain Look-up table details.

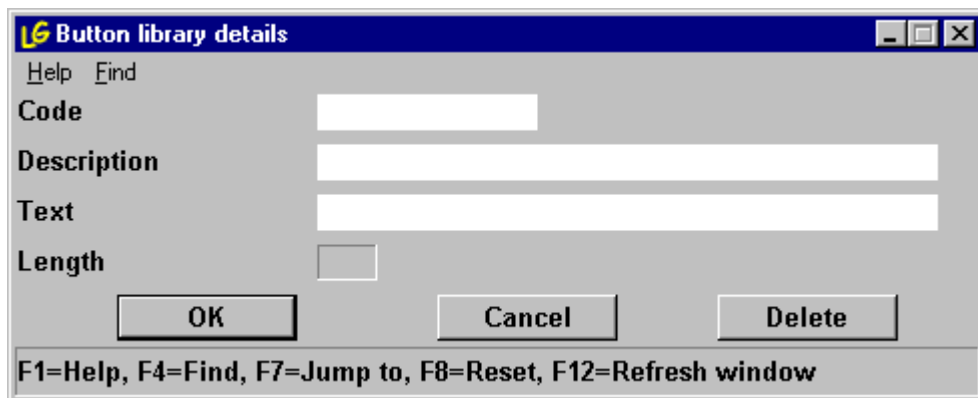
The screenshot shows a Windows-style dialog box titled "Look-up Table maintenance". It has a menu bar with "Help" and "Find". The main area contains the following fields and controls:

- Code**: A single-line text input field.
- Title**: A single-line text input field.
- List build call**: A single-line text input field.
- Select made call**: A single-line text input field.
- Max list items**: A small numeric input field.
- Form width**: A small numeric input field.
- Description**: A multi-line text input area.
- Auto list**: A checkbox.
- List Heading**: A single-line text input field.
- Search message**: A single-line text input field.
- Buttons**: "OK", "Cancel", and "Delete" buttons.
- Status bar**: Displays "F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window".

- Code** A free text code (excluding spaces and commas) to define the look-up mechanism.
- Title** Free text for confirmation and look-up selection purposes. It appears as the look-up list form caption.
- List build call** The program entry point in which the generation of the list will be performed.
- Select made call** An optional entryptoint triggered if a selection has been made. `%ltsel` will be defined at this point and is available to the called routine.
- Max list items** An optional parameter. If set (to a numeric value) then the list build routine must not exceed this number of selections in the list. If it would exceed this count then the list will be closed at the maximum number and a message code of `'%maxlist'` returned in the error variable.
- Form width** A number in the range 32 to 78 inclusive. This permits variable width look-up forms.
- Description** A free text field to further identify this look-up facility - especially where multiple look-ups are available on the same 'file' though for different purposes.
- Auto list** If "Y" ("y") entered here the look-up will be triggered automatically when positioned on the corresponding item and that item has a blank value. A selection must be made by the user.
- List Heading** Optional text heading to appear in the list form above the items.
- Search message** Optional message code. If entered the message will be displayed when the look-up is invoked to inform the user of the action being performed. The message type must be 2 (warning).

3.7 BUTTON LIBRARY

This feature provides a means of standardising button texts which can be loaded by the application prior to invoking the applicable forms.



Button library details

Help Find

Code

Description

Text

Length

OK Cancel Delete

F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window

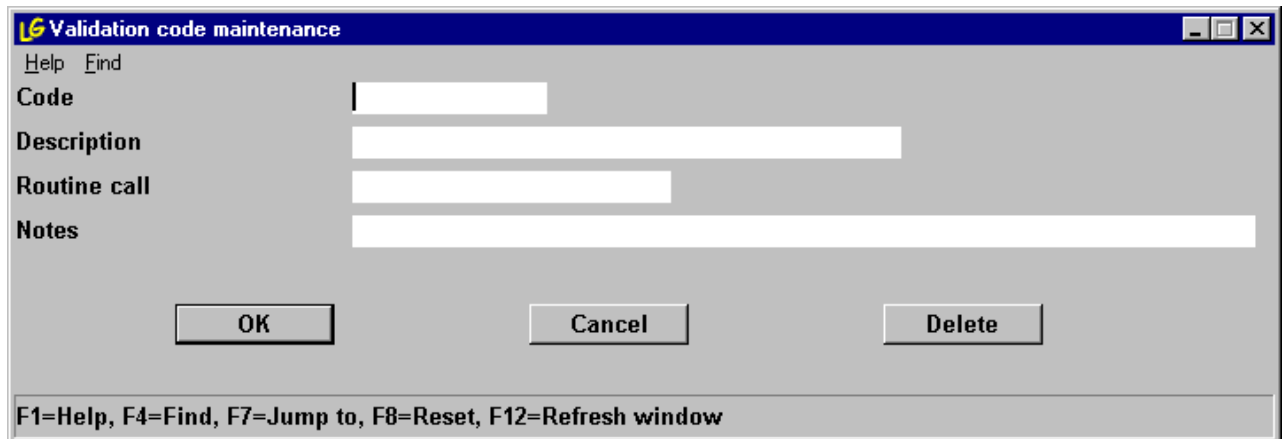
Code Enter a code by which this button is to be recognised and accessed. For amendment a look-up facility is available on this item.

Description A title for confirmation and look-up purposes.

Text The actual text to be displayed in the button. For centralising, spaces should be prefixed and suffixed where required. When entered, the length of this text is displayed for information only.

3.8 VALIDATION CODES

This maintenance facility enables the validation code feature of control event handling to be utilised. Validation procedures are written within library routines as functions. The function will receive a single parameter (%lzsv) passed by reference and will quit with a value being the error message code. A valid value will cause a null valued return. Since %lzsv is passed by reference it is possible to generate secondary values within lower level subscripts of this variable - such as the result of a date conversion - so that, in the application routine event code both user entered value and converted values may be picked up from this variable.



The image shows a dialog box titled "Validation code maintenance". It has a menu bar with "Help" and "Find". Below the menu bar are four input fields: "Code", "Description", "Routine call", and "Notes". At the bottom of the dialog are three buttons: "OK", "Cancel", and "Delete". A status bar at the very bottom contains the text: "F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window".

Code Any code require to identify the validation procedure within the form control parameters

Description For reference and look-up only, the description of this validation procedure.

Routine call The program entry point consisting of label^program. It will be called in the form:
s %lzser=\$\$label^program(.%lzsv)
and the entryptoint within the validation library must be:
label(x) ;procedure

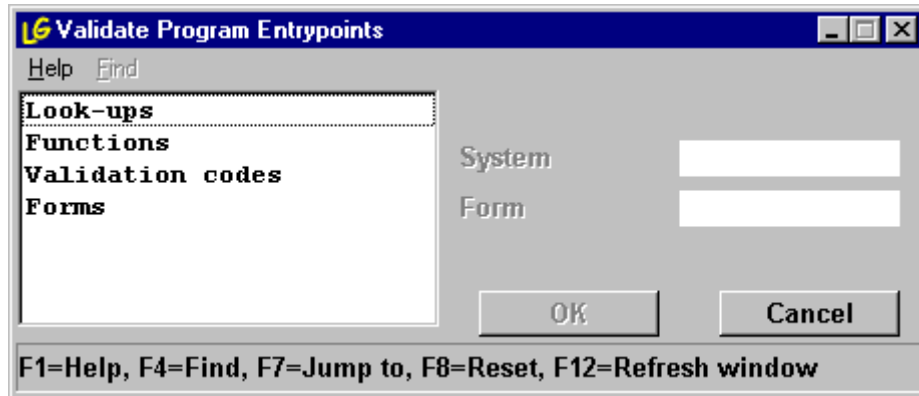
Note the "call by reference" form of the parameter.

Notes An additional free format narrative to expand upon the above description.

3.9 VALIDATE PROGRAM ENTRYPOINTS

This routine enables checking of all program entrypoints linked through the Form Builder parameters. It covers look-ups, functions, validation codes and forms.

When invoked, a form is displayed as show below

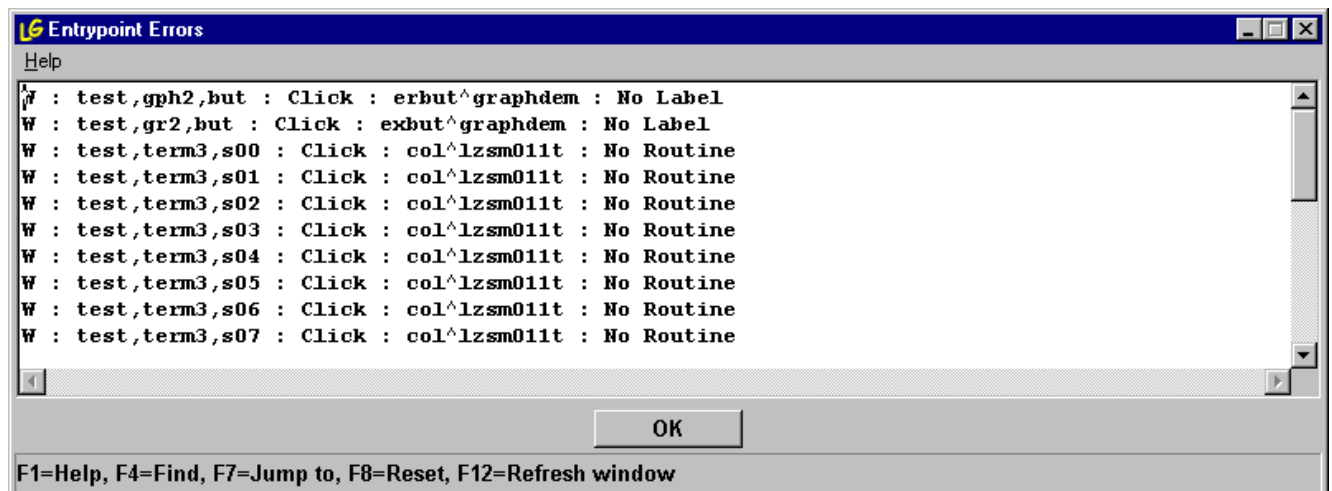


The list box contains a list of:

- Look-ups
- Functions
- Validation codes
- Forms

and is a multi-select type. If the "forms" option is selected the user can select a single system or all (using an asterisk). If a single system is selected the one or all forms can be selected (again using asterisk for all).

If the "ok" button is pressed the routine commences checking all entries as requested and build a list of any discrepancies. If any errors are encountered these are displayed, at the end of the checking, is a document control as exemplified below.



The format of the line is: Type: reference: event: label^routine: error

Where:

Type is on of	F	function
	L	look-up
	W	window - form
	M	menu

Reference is one of:

direct code	for functions and look-ups
system,form (,control)	for forms
system, form, menu, menu entry	for menus

4. CONTROL PROPERTIES AND EVENTS

4.1 PROPERTIES

Alignment	For Picture controls, this may be null, S (Stretch), P (Proportional) or F (Fill form).
Allow characters	Override of default characters filtered out by the Lastic User Interface. ASCII values of characters to be allowed in this control are defined, delimited by commas.
Anchors	Override of the default anchors (LT). A control may be anchored to a combination of L(ef), R(ight), T(op) or B(ottom). When the form is resized controls will remain "anchored" to those margins against which anchors have been defined. NOTE: When using anchors, for form resizing use "#" as the resize control type.
Autoselect	For Input controls within Lastic GUI to operate in "replace" mode. When set to 'Y' the existing text of that input control will be selected upon receiving focus and data entry will replace the existing value.
Bring To Front	Used currently only by the splitter control, where controls of the same type (Windowed or graphical) are overlaid use this setting to force the selected control to be on top of the other where their areas overlap.
BorderStyle	Applicable to GUI environment only this defines the style of the control border and may be L(owered), R(aised) or N(one).
Datatype	For input controls, defines the data type code reference used to assist the generation of this control.
Description	A brief identification of the control. It has no operational significance and is used to assist the developer in identifying the purpose of the control.
Editable	For Comboboxes. Set to Y to provide editable combobox, N to provide a fixed dropdown list only. The default is 'N'.
FrameSize	By default, the drawn frame border is half a cell height less than the control's top and bottom boundaries and half a cell width less than the control's left and right boundaries. To draw the frame border at the top and bottom boundaries enter "H", to draw the frame border at the left and right boundaries enter "W" and, for all boundaries enter "B" (Both).
FrameStyle	For Frames. Values available are FR (Frame – Raised), FL (Frame – Lowered), BR (Box Raised) and BL (Box Lowered).
Glyph Style	For BitButtons within Lastic GUI to define the location of the glyph relative to the text. A=Above, B=Below, R=to the Right and L= to the Left. The default is A.
Height	The number of lines deep for the control. For comboboxes, it is the number of lines utilised by the drop down element when invoked.
Help code	Help text table reference
Highlight code	Identifies the presentation highlight code (0-19) for displaying the control (denotes foreground/background colouring etc.)
Hint	Free text to appear when mouse hovers over the control.
Jump-to	If set to 'Y' indicates this control will be given focus if the "jump-to" function key is clicked or, if a button, then in the GUI environment it becomes the default and is given focus and fires a

"Click" event when the <CR> key is pressed.

Justify	Allows the control text to align L(eft), R(ight) or C(entered). For some controls where justification is available the Centre option is not provided.
Look-up code	References the look-up table record defining the entypoint to the associated database search routine.
Maxchars	Defines the maximum number of characters permitted in this control. If not defined, the maximum number is the same as the width.
Multiple select	For listboxes, setting to 'Y' indicates more than one item may be selected at a time.
Name	The unique code given to a control within that form.
Password protect	If set to 'Y' indicates the input data will be "hidden" from the user's view.
Position	The top left location in terms of line number and column number of the control within the form. The format is line;column and value are relative to the form with the top left of the form border being 0;0.
Radio set number	For radio buttons, this is the group reference. Radio buttons of the same group work together. (Range 0-9)
Read only	When set indicates that the control's value cannot be changed though it may still receive focus.
Selected highlight code	For controls where elements may be selected by the user, this indicates the highlight code to use when displaying those elements which have been selected.
Selected highlight code current	For controls where elements may be selected by the user, this indicates the highlighting to be given for an element which is both selected and is the one currently having focus.
Sequence	The taborder of the focusable controls starting at 1.
Style	Applicable to the Splitter and indicates whether the resizing is to operate V(ertically) or H(orizontally).
Terminators	Overrides, for character terminals, the default terminator. For example, <CR> would terminate an input field though it could be set to allow both <CR> and, say, '*' by setting this property to 13,43
Text	The fixed textual element of the control. For text controls, this is the "value" shown. For other controls it specifies the "prompt" element.
Translate	May be set to UL indicating convert upper to lower case or set to LU indicating convert lower to upper case. If not set, no conversion is performed.
Variable ref.	Identifies the variable from which the control display is taken and into which user entries are stored. Some controls may utilise complex structures within the defined reference. The reference can be to a local or global variable and multiple subscripts are permitted.
Width	The width of the control in terms of columns.
Width override variable	If defined, then, at run-time, if a numeric value is held within this variable reference this value is

used as the width of the control rather than the value defined above.

Wordwrap

If set to 'Y' indicates that text will wrap on word boundaries. If not set, the GUI environment provides a horizontal scrollbar while the character terminal environment wraps on the last column of the line.

4.2 CONTROL EVENTS

Gotfocus	Activated when a control is given focus. If an attempt to give focus to a control fails due to interim event processing the initial target control's gotfocus event does not fire.
Change	Activated when the value of the control changes.
Click	Activated when the control's status is changed by the user. Includes changing the "focus" of an element within the control for such as listboxes and grids.
DoubleClick	Activated by the selection of a multi-element control, e.g. pressing <SpaceBar> on an element within a listbox or double clicking the mouse within a grid.
Drag/Drop	Fires within Lastic GUI only. For those controls available as Drag/Drop sources the Drag/Drop event will fire when dropped onto an available target. The Event fires within the source control and indicates the details of the target control.
Lostfocus Next	Activated when a control loses focus to another control with a taborder (sequence) greater than itself. It is fired for all controls between the original control and the control receiving focus.
Lostfocus Previous	Activated when a control loses focus to another control with a taborder (sequence) less than itself. It is fired only for the control which is losing focus.

4.3 CONTROL EVENT PROPERTIES

Datatype Val. Indicates if the validation associated with the datatype for this control is to be applied at run-time ('Y/N')

Function code The code or name of the 'Input' function to be activated within this event processing.

Function Parameters

Override of the standard parameter of the input value as passed to this function. The entered parameters must correspond to the target function.

Mandatory Indicates this control must have some value.

Refresh controls Lists the control names, separated by commas, to be re-displayed upon completion of the event processing.

Routine entrypoint

Application program entrypoint, invoked by a "do" to be called following any other validation and function processing.

Validation code The reference to the validation code function processing to be performed.

Validation parameters

Override of standard parameter of the control value to be passed to the associated validation routine.