

Lastic System Software

System Management

Version 6.1
Jan 2013

CONTENTS

- 1. Introduction**
- 2. Software Installation**
 - 2.1 Loading the Software**
 - 2.2 Installing the Software**
 - 2.2.1 Entering your licence key**
 - 2.3 System Set-up**
 - 2.4 Quick set up for Terminal types**
 - 2.5 Using the Manager Facilities**
- 3. Presentation & Terminal Types**
 - 3.1 Screen Maintenance - General**
 - 3.2 Screen Maintenance - Terminal Types**
- 4. Application Program Release**
 - 4.1 File Device Maintenance**
 - 4.1.1 Named Files**
 - 4.2 Device Type Maintenance**
 - 4.3 Using the Distribution System**

1. INTRODUCTION

The system management facilities are provided with all Lastic Software to enable presentation and terminal type definition to be accomplished. Certain facilities are available only for the User Interface development environment - these are identified within this document.

Where GUI facilities are employed, the Lastic GUI user manual is also provided giving details of the set-up facilities provided in the PC environment.

1.1 New Features

1.1.1 Access to control routines

There are new control routines accessed through the new menus:

lgmenu for Lastic GUI

lcmenu for Lastic Character Terminal

start these using d ^lgmenu or d ^lcmenu as appropriate.

The menus and the system control routines have been re-designed to improve presentation.

1.1.2 Revised distribution routines

These now provide for distribution file encryption and for auto removal of the source code on the target system. The deletion of source code operated only for Cache based systems.

The new ^lzsinst (installation routine) will now permit multiple releases to be imported and then installed with a single run of ^lzsinst. Please be aware that the install routine will action the releases in the sequence of the release ID codes - they are used as subscripts within the saved install files. We would recommend that multiple release updates are not performed where pre and post install routines are to operate within any of those releases.

2. SOFTWARE INSTALLATION

2.1 LOADING THE SOFTWARE

This section deals with the loading and installation of the distributed M software. The installation and set-up of the GUI software for your PC is described in the Lastic GUI User Manual.

Lastic software is distributed in one or more files with names in the form LZxDIST.GLO where x depends upon the system area. For example:

LZSDIST.GLO is used to distribute Screen Handler development environment

LZRDIST.GLO is used for the Rule Based System

LZGDIST.GLO for the Lastic GUI development environment

LZWDIST.GLO for the Lastic Web Handler

The files contain both global data and routines held in a global save format which, when restored into your system, will create a global called ^lzsrel containing all the distributed elements.

For Cache environments use ^%GI to load the contents of this file.

It is possible that, for some M implementations, the normal global restore routine will not restore this file. If this is the case then a simple restore routine can be created with the following code:

```
A      R "File Name:", ?30, fn
        Q: fn=""
        S T=$I
        O .....:2 ;open file in the form applicable to your M system
        I '$T W "File cannot be opened",! G A
        U fn R x,y
        U T W x,! ,y,!
B      U fn R x,y
        G END:x=""
        S @x=y
        G B
END     C fn
        Q
```

The distribution also contains a routine save file (LZSINST.RTN) which contains a single M routine (^lzsinst). This must be restored into your M system using the normal routine restore facility.

2.2 INSTALLING THE SOFTWARE

When the global data has been restored to create the global ^lzsrel as described above, the install routine (^lzsinst) can then be run to distribute the contents of this release.

```
>d ^lzsinst
```

This routine will display the distribution header and will then list the globals and routines being installed. When installation is complete the program will delete the release global ^lzsrel.

NB More than one distribution file may be rolled in using the global restore facility and lzsinst will install each in the order of the installation release 'codes'. If there is a specific order required then roll in and run lzsinst for each release separately.

^lzsinst will also log the installation header for each release into ^lzsinsth(date, time, sequence).

If you are a new user or are installing a new product you will receive a licence key as a second distribution file which must also be loaded in the above manner.

2.2.1 ENTERING YOUR LICENCE KEY

Evaluation software licence keys are provided within the software. If you have purchased the software then you will be provided with a licence key either with the software distribution as a file for importing or as a paper key which requires keying in.

Once the licence key is entered it does not require re-entry unless your licence agreement has been changed or you are adding further products.

To enter your licence key from a paper key, locate the Software Licence Key document and then run the key entry routine.

```
>d ^lickey
```

The program will request the information as shown on your licence key document.

Customer namexxxxxxxxxx

Licence ref: xxxxxxxxxxxx

Code	Key
xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx

Enter the keys exactly as shown on your licence key document (including the customer name). The Licence ref, Codes and Keys are all uppercase alpha.

When all codes have been entered, press <Return> when another code is requested.

The program will validate the entered key. If valid it will display either "Licence Stored" or, if an update, "Licence Updated".

If an error is detected, a message such as "Licence Key Error" will be displayed and the full information will be requested again.

For existing users, the installation is now complete though please refer to the release notes for any further requirements.

For new users, please proceed to the system set-up (below).

2.3 SYSTEM SET-UP

For Cache users, this chapter can be ignored since the system comes pre-configured for Cache environments.

Having loaded the released software as described in 2.1 and 2.2 above, for the first installation only, it is also necessary to run the set up routine ^Izssetup which will have been loaded by the install routine.

This set-up routine will enable you to configure the overall environment and to set up a terminal type which can then be used to operate the Lastic character terminal environment.

To run the set-up routine, type, at the Mumps prompt:

```
>D ^Izssetup
```

The set up routine will commence by displaying:

```

Lastic Screen Handler set up

1 = Environment details
2 = Terminal types
Select option      _
```

enter "1" and press <return>. The set up routine will now display, item by item, the following:

```

Lastic Screen Handler set up

Use defaults (Y/N)      Y

1=DSM, 2=DTM, 3=ISM, 4=MSM  _

Esc seq identifier      27
_____

Echo off M code         u $i:("S")
_____

Echo on M code          u $i:("")
_____

Routine directory global  ROUTINE
_____

Update (Y/N)           _
```

2.3.1 Use defaults (Y/N)

The system set up provides a mechanism for setting the environment details automatically depending upon the host M implementation. It is recommended that this option is selected causing the default values to be generated. The values generated will be displayed below prior to updating. If you select "N" at this point, the existing values are retrieved and displayed (the program proceeding to section 2.3.3 below).

2.3.2 1=DSM, 2=DTM, 3=ISM, 4=MSM

This question is asked if the response to the "Use defaults" option above was "Y". Enter the number corresponding to the M implementation you are using. Note ISM includes all Intersystems M products except for DSM, MSM and DTM.

The values for the escape sequence identifier, echo on/off and the routine global are generated and displayed below to allow for amendment if required.

2.3.3 Esc seq identifier

When using function keys, cursor movement keys etc, the terminal sends escape sequences to the host computer. Mumps systems notify their programs that an escape sequence has been received by sending a specific number. This is generally either 27 or 0 (depending upon your Mumps implementation). To determine what this number is for your installation simply perform the following (in programmer mode).

>R *X

Now press a function key or cursor movement key

>W X

The value of the number will be displayed.

If the number displayed by the set up is the same as that determined by the above procedure then you may merely press <return> to accept the current value.

If the number is different then enter the correct number and then press <return>.

2.3.4 Echo off M code

The Lastic User Interface requires character echo to be turned off - this is particularly important when the Lastic GUI environment is employed since we do not want messages sent by the PC to the database server to be echoed back to the PC. The Lastic software will turn echo off when a form is invoked and will turn it back on whenever a call is made back to the application program or when the form is terminated. In order to enable character echo to be turned off it is necessary to inform the system how to do so for your Mumps implementation. Again this differs widely by implementation though the majority of implementations permit this feature through the "Use" command. You must enter the actual Mumps program code. If the existing program code shown is that required then you may merely press <return> to accept the existing value.

Because the Screen Handler makes extensive use of escape sequences for positioning and highlighting the value of \$X will be greater than the true screen position of the cursor causing line wrapping. You should therefore disable the use of screen margins for wrapping by setting the margin (or width) to zero and also enable escape sequence processing if not automatically enabled. This should be accomplished within this "echo off" code.

2.3.5 Echo on M code

This is the reverse of the above. The Screen Handler executes this code before terminating a form and before invoking an application program event routine. Enter the program code to turn echo back on or press <return> only if the existing code shown is correct. It is recommended that escape sequence processing is not turned back on within this code since it could affect "type ahead" capabilities for your users.

2.3.6 Routine directory global

This is the name of the global (or "pseudo" global) which contains the names of the routines to be found in the directory (or UCI). It is utilised by the Screen Handler's distribution system to select routines. It may be a name such as "ROUTINE" or <space>. Some Mumps implementations do not provide such a global. If this is the case then press <return> at this point and refer to the section on the distribution system before attempting to use that feature.

2.3.7 Update (Y/N)

Enter "Y" to confirm changes and to update the system. Enter "N" if no update is required.

The set up routine will then return to the initial option display.

2.4 QUICK SET UP FOR TERMINAL TYPES

This section is required only for Lastic Character terminal environments.

The Lastic user Interface, for character terminals makes use of function keys, terminal highlighting and character graphics drawing. It is necessary to define one or more terminal types suitable for your M implementation and physical terminal before using the software.

This facility provides the capability to generate both default settings and specifically selected function keys. Having set up such terminal types they may be amended using the system management function (see section 3.2 below).

Start the set-up routine by typing:

```
>d ^lzssetup
```

The set up routine will commence by displaying:

Lastic Screen Handler set up

1 = Environment details

2 = Terminal types

Select option _

enter "2" and press <return>

The set-up routine will then request, item by item, the following:

Lastic Screen Handler set up

Terminal type code _____
Description (max 30 chars) _____
Default Function keys (Y/N) _

if "Default Function keys" is "Y"

1=DSM, 2=DTM, 3=ISM, 4=MSM _

Keyboard: 1=Terminal, 2=PC _

The following is displayed (the keys and the default message vary depending upon the keyboard type selected) and the "Proceed (Y/N)" question asked. The example below shows the PC keyboard selection.

Function	Key	Function	Key
Abort/exit	F12	Backspace	Backspace
Carriage return	Return	Clear field	F8
Cursor up	Cursor up	Cursor down	Cursor down
Cursor left	Cursor left	Cursor right	Cursor right
Delete	Delete	Help	F1
Insert/over	Insert	Jump to	F9
Last character	End	Look-up	F4
Page up	Page up	Page down	Page down
Menu	F3	Refresh screen	Home
Reset input	F7	Application	F6
Control exit	Tab		

Default message (msg code =deftname)
F12=Exit End=Append F8=Clr F7=Reset F9=Jump to F3=Menu F6=Appkey F4=Find

Proceed (Y/N) _

If "Default Function keys" is "N" the following is displayed with the function keys being requested one at a time.

Note Function keys selected for inclusion in default message

Function	Key	Function	Key
Abort/exit	27, 91, 52, 126	Backspace	127
Carriage return	13	Clear field	27, 91, 50, 52, 126
Cursor up	27, 91, 65	Cursor down	27, 91, 66
Cursor left	27, 91, 68	Cursor right	27, 91, 67
Delete	27, 91, 51, 126	Help	27, 79, 80
Insert/over	27, 91, 50, 126	Jump to	27, 79, 83
Last character	27, 91, 49, 55, 126	Look-up	27, 79, 81
Page up	27, 91, 53, 126	Page down	27, 91, 54, 126
Menu	27, 79, 82	Refresh screen	27, 91, 49, 126
Reset input	27, 91, 49, 57, 126	Application	27, 91, 49, 56, 126
Control exit	9		
Default message (msg code =defname) - max 78 chars			

Proceed (Y/N) —

2.4.1 Terminal type code

Enter a "code" for the terminal type to be created (max 6 alpha-numeric characters) . The code is used for subsequent identification of this terminal type. You can set up default details for types other than the one you are currently using.

If no entry is made the set up will terminate. If this code already exists the entry is rejected and another code must be chosen.

2.4.2 Description (max 30 chars)

This is purely for reference. The description must not include the tilde "~" character.

2.4.3 Default Function keys (Y/N)

Enter "Y" if you require the system to generate a default set of keys for the various functions. Enter "N" if you want to select your own.

If the response is "N" the program will continue as at step 2.4.6 below.

2.4.4 1=DSM, 2=DTM, 3=ISM, 4=MSM

Select the M environment for which the terminal type is to be generated. You can generate default terminal types for M implementations other than the current host.

2.4.5 Keyboard: 1=Terminal, 2=PC

Select the appropriate user device. A Terminal keyboard is assumed to be a VT200 and upwards compatible type.

The program will then generate a default set of function keys and a default message based upon the keyboard type and the M implementation. These keys will be displayed. The default message is that which will appear at the foot of the screen as an aid to the operator for non-obvious function keys. The message code ("def" concatenated with the terminal type name) is displayed and this message can later be modified using the message maintenance facility.

The program then requests if the user wishes to proceed (see 2.4.8 below)

2.4.6 Function key usage

If default function keys was not selected then the set up routine will display a series of function key actions. For each action you must select a specific key on the keyboard and press it once. When you press the key to be used for a function the set up routine will display the key value or escape sequence which is generated. You should make a note of each key that is chosen for each function for your user documentation and for inclusion in the default message which you can define later. You do not have to note the values generated which are displayed purely for information. You cannot select the same key for more than one action. Individual key requirements are detailed below.

Abort/exit

Pressing this key when running a form will cause that form to terminate and return to the application program with an "aborted" status. This is the equivalent of alt-F4 under MS-Windows.

Backspace

This key causes the character to the left of the cursor position to be deleted. Note, Some terminals have options to set this as either ASCII 8 (ctrl-H) or as delete (ASCII 127). The default generation sets this to ASCII 127. You can subsequently modify this, or any other value, using the terminal details maintenance (see section 3.2 below)

Carriage return

This is the normal field input terminator. Press the normal carriage return or "enter" key.

Clear field

When running a form, pressing this key will cause the current field to be cleared (set to null) thus enabling a new value (if required) to be entered without deleting individual characters or editing the existing value.

Cursor up

The key selected here causes a move back to the previous input-type field in sequence.

Cursor down

This key will cause the "focus" to move to the next field in sequence within the window.

Cursor left

When in an input field, this key will cause the cursor to move one character position to the left unless it is already at the first character position in that field. For button type fields it will cause focus to be given to the previous control.

Cursor right

This key causes the cursor to be moved one character position to the right within an input field. The cursor will not move further than the next available character position for input. For button type fields this will cause focus to be given to the next control.

Delete

This key causes the character at the cursor position to be deleted.

Help

Pressing the key selected here will cause the help form to be invoked displaying the required help text.

Insert/over

The character terminal facilities provides both character insert and overwrite modes. The current mode is shown by the character at the extreme right of the bottom line on the terminal screen. An "I" represents insert mode while an "O" represents overwrite mode. The user can toggle between these two modes at will by pressing the key selected here. When the mode is changed the displayed mode character will automatically change.

Jump to

When a form is created a focusable field (input, button etc) can be designated as the jump-to item. Typically, this would be the field requesting the "save" action or similar. It's purpose is to avoid the user having to use the cursor down key to position on that item. Select the key you wish to designate as the "jump-to" key which causes this movement.

Last character

Pressing this selected function key will enable a user to move the cursor immediately to the end of the current input field enabling data to be appended to that item without having to move the cursor position character by character.

Look-up

Within the software input fields may have look-up facilities. The selected function key here will cause any such look-up facility to be invoked.

Page up

Forms and controls have the capability of paging details (such as the look-up window, listboxes, grids and multi-line inputs). This selection designates which key is used to trigger a page-up action.

Page down

The reverse action to the previous item. This selection designates which key is used to trigger a page-down action.

Menu

Some application Forms have a menu bar attached enabling actions, other tasks etc to be performed while within this application. The function key defined here will cause the menubar to become active.

Refresh screen

If a terminal screen should become corrupted through such as message broadcasts or if a terminal has to be replaced while in the middle of a user's task then this key causes the whole screen, including all levels of forms currently active, to be redrawn leaving the user still positioned on the same field.

Reset input

This key will enable a user to reset the current input field to its value at the point when editing commenced. It provides a means of removing editing errors.

Application

A developer specific function key. This key causes the "application" event for a form to be invoked presenting details of the current control enabling context sensitive processing to be performed.

Control Exit

For document, grid and hypertext controls, since they use all cursor keys for navigation, it is necessary to use another method for exiting such controls. If the user presses this defined key followed by cursor left or cursor up then the focus moves to the previous control in sequence. Pressing cursor right or cursor down causes focus to be given to the next control in sequence.

2.4.7 Default message (msg code =defname) max 78 chars

Where the user has defined their own function keys the default message can now be created. It is advised that the message specifies non-obvious function keys. Obvious keys include cursor movement, carriage return etc and should not be included at the expense of other keys. This message (the code is given in parentheses and consists of "def" concatenated with the terminal type name) may be modified later using the message maintenance facility.

2.4.8 Proceed (Y/N)

Enter "Y" to continue to the next screen else enter "N" to terminate this set up without any update. If the user selects the proceed-"Y" option then the set up routine displays the next screen, item by item.

```

                                Lastic Screen Handler set up

Terminal type code              xxx

Monitor: 1=B/W, 2=Colour       _
Graphics char set: (Y/N)       _
VT200 compatible (Y/N)         _

Set as default type (Y/N)      _

To modify these details use the function 'scrm' and select terminal details

Update (Y/N)                   _
```

2.4.9 Monitor: 1=B/w, 2=Colour

For generation of highlight escape sequences, select the type of monitor applicable.

2.4.10 Graphics char set: (Y/N)

For certain terminals and emulators, the default character set does not provide line drawing and requires, for such actions, the invocation of a graphics character set. If this is the case, respond with "Y". If there is no special graphics character set then respond with "N". For non-graphics character sets the line drawing characters generated are compatible with the "Terminal" font face. Lastic GUI emulator font "lasticGUI" has the same characteristics as the terminal font face and requires no graphics character set.

2.4.11 VT200 compatible (Y/N)

If this terminal type is VT200 or upwards compatible enter "Y" here. VT200 compatibility enables the set-up routine to provide fast form drawing and to enable 132 column operation. The Lastic GUI emulator has these capabilities.

If it is not VT200 compatible, enter "N". In this case form painting is a little slower and you will not have 132 column operation support (though these can be added/amended later).

2.4.12 Set as default type (Y/N)

If this is to be the default terminal type enter "Y" otherwise enter "N". The default terminal type is selected if no overriding value is presented to the screen handler at run-time on a specific terminal. It is recommended that the most common terminal type becomes the default type. The default terminal type can be changed subsequently.

2.4.13 Update (Y/N)

Enter "Y" if the terminal type details are to be updated else enter "N".

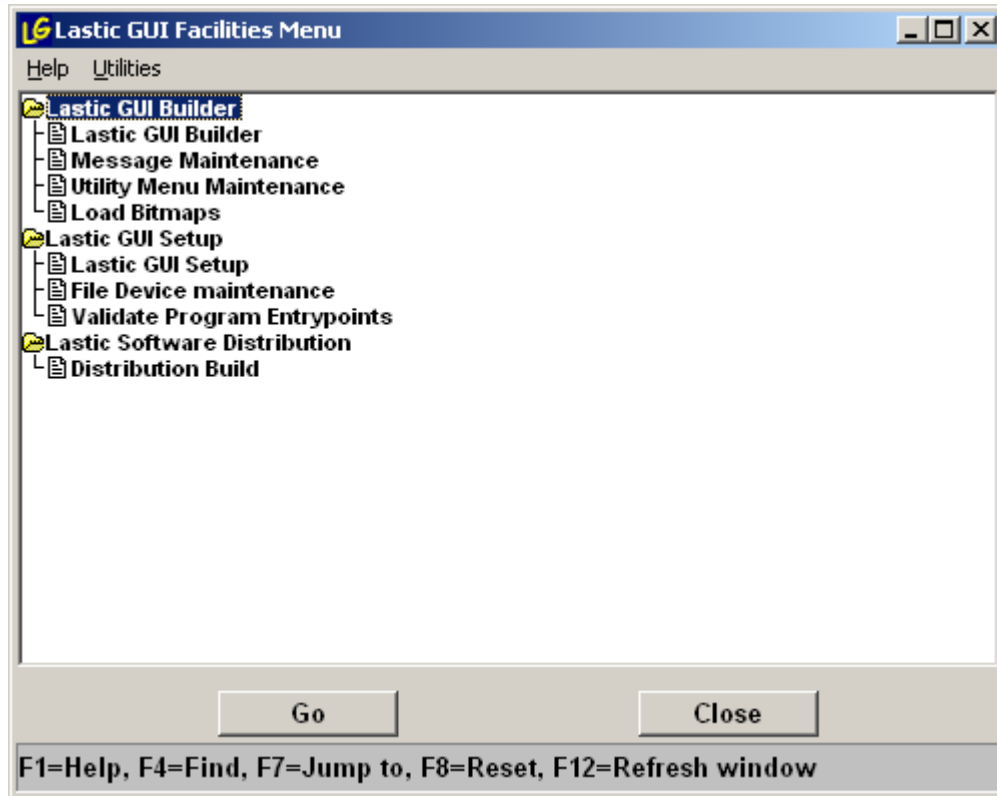
Though the terminal type details created are basic they are sufficient for you to now operate the software on your system and thus you may now fully configure your system as required.

2.5 USING THE MANAGER FACILITIES

To invoke the manager software:

- >D ^lgmenu for GUI operation
- >D ^lcmenu for Character terminal operation

This causes the following display in GUI mode. An equivalent is shown in Character Terminal mode.



Select the facility required by either double-clicking the item or highlight the item then click the 'Go' button.

To exit the system, press the 'close' button.

3. PRESENTATION & TERMINAL TYPES

This section deals with the overall presentation and operation of the software and also enables terminal type definitions to be maintained.

Once the basic environment and a default terminal type has been defined the detailed aspects of the system can be maintained using the facilities available within the routine Lastic GUI set up (or Lastic CHUI set up for character terminal operation).

Lastic Software utilises a system variable %lzt, to identify the terminal type in use, including GUI, in order to communicate with the terminal/PC.

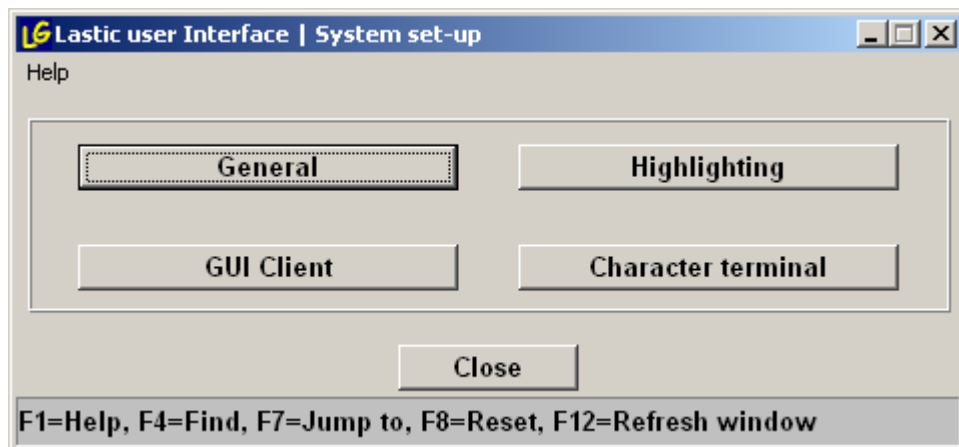
The software provides for a default terminal type to be defined such that, if %lzt is not defined the default will be used. For mixed environments these types of facilities are usually already implemented though through a different variable. In these cases it merely requires the setting up of those terminal types within the Lastic Software (using the same codes) and then copying the existing variable into the Lastic variable %lzt within such as a menu routine.

For Lastic GUI a terminal type of "lw" is provided

Lastic software for character terminals simulates many of the controls found in a GUI environment including text boxes, command and options buttons, check, list and combo boxes as well as fixed and variable output texts. The screen maintenance facilities enable, amongst other items, the presentation of such controls to be defined.

To invoke the facilities select the function "scrm" from the manager functions (see section 2.5 above).

This will cause a window as shown below to be displayed:



3.1 SCREEN MAINTENANCE - GENERAL

This option is selected by pressing the "general" button on the window as shown in the introduction to this chapter.

It provides for the definition, and amendment, of presentation formats, reject character filters and various environment dependent characteristics. The maintenance screen is shown below:

The screenshot shows a window titled "System set-up | General" with a menu bar containing "Help" and "Find". The main area is a table with two columns: "Item" and "Value". The table is organized into sections: General, Character Terminal, Input control, Checkbox control, Radio Button control, Command Button control, and Listbox Actions. At the bottom, there are "OK" and "Cancel" buttons, and a status bar with keyboard shortcuts: F1=Help, F4=Find, F7=Jump to, F8=Reset, F12=Refresh window.

Item	Value
General	
Reject chars (ASCII values)	1,2,3,4,5,6,7,9,10,11,12,14,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,126
Wild card char (ASCII val)	42
Echo off M code	u \$t{"";"S"}
Echo on M code	u \$t{"";""}
Def form timeout (seconds)	999999
Default term type	trt
Error trap routine	ermes^lmenu
Bespoke Help routine	
Help focus on text	False
Use Hypertext help	True
Save on lostfocus previous	False
Routine global	ROUTINE
Use Data Types	True
Character Terminal	
Esc. Seq Ident (ASCII val)	27
Trap serial line errors	False
Screen size	
Lines	24
Columns	80
Input control	
Field Start	[
Field End]
Field Extent	-
Field Look-up End]
Checkbox control	
Field Start	[
Field End]
On Character	X
Radio Button control	
Field Start	(
Field End)
On Character	*
Command Button control	
Field Start	(
Field End)
Listbox Actions	
Cursor up/down to pos.	True
Select on cursor pos.	True

OK Cancel

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

GENERAL

Reject chars.

You may enter all control characters here. When the screen handler is initialised it loads the reject character list and removes from that list any characters which must be permitted including any

single characters defined within the "function keys" for this terminal type and the escape sequence identifier.

To reduce application validation it is suggested that your standard global field delimiter is included within the list thus making it impossible for such to even reach the application. In this way no program need test for such a character.

Entry of reject characters are made as series of Ascii values, delimited by commas. There is no specific order of characters required. The five lines are provided to leave room for additional filtering out codes if required later. Remember that such codes may have values between 0-255 and that no comma must follow the last value on a line.

Wild card character

For use within look-up actions, this character, initially defined as an asterisk, may be embedded throughout a look-up item input field. The form of the look-up list build code enables full pattern matching to be utilised against such a selection. It is also used to denote 'any character' within the item input filter and terminator parameters - more about this later on. The input requires the ASCII value of such a character to be entered.

Echo off/on eXecute code

This is the executable code required, for this implementation of Mumps, to turn off character echo. If left null, hidden fields will repeat the last character entered before blanking it out. It also causes additional overhead since characters are echoed and then repeated by the screen handler. The 'on' request is the parallel code to turn character echo back on. It should be noted that upon any application call or function call, or when a window is terminated, echo is turned on. The echo off M code should also ensure escape sequence processing is turned on though it should not be turned off again within the echo-on code.

Default form time out

To permit windows to abort if no action has occurred within a specified time period, this parameter permits such a time period, in seconds, to be defined. Enter a high value to effectively disable time out. (min value = 60 seconds).

Default terminal type

Though it was stated earlier that the terminal type must be passed to an application utilising the Lastic Screen Handler, in truth, if it was not passed then the system will use the default type defined here.

Error Trap Routine

If you want to install error trapping, this provides a simple mechanism wherein the trap is set within every event handler immediately prior to calling your own event handling routines. It is then reset upon return from your routines. You will have to build your own procedures for the error trap.

Bespoke Help routine

If you wish to implement your own help system you may do so by specifying the entry point of the run-time help application here. If defined, this routine will be called instead of the standard mechanism. It should be noted that the call to this routine also involves a series of parameters. For details of this facility please refer to the Help section of the programmer guide.

Help focus on text

If true, within the Character terminal operation, focus will be given to the text control when a help form is invoked. If unchecked, focus is initially given to the first control on the help form..

Use Hypertext Help

If true, the hypertext help mechanism will be invoked when using help.

Save on lost focus previous

If true, Input field target variables are updated with the newly edited value even when moving to a

previous control. If false, an Input field value is reset to it's previous value when moving back to a previous (lower tab order) field.

Routine directory global

This is the name of the global holding the routine directory. It requires that the routine name is the first subscript of this global. It is used merely for identifying valid routines when building a software release. The routines themselves are loaded using the "zload" command. Typically, such global names would be:

ROUTINE	(InterSystems)
<space>	(VAX-DSM)

You may prefix the global name with extended syntax ([...]) if required.

If your mumps implementation does not support a routine global (or pseudo global) you can always generate such a global and give the name here.

Use Data types

An optional facility provides for the definition of data types for input items which contain default values for both ease of generation of window items and to provide consistency of item definition. Such data items can also carry validation parameters which may be invoked. This item defines whether data-types are to be employed by the developer. If set to false data-types are not requested.

CHARACTER TERMINAL

Esc seq identifier

Enter the ASCII value of the character presented (on a single character read) when an escape sequence is sent to the application from the keyboard.

For example Intersystems implementation sends ASCII 27
 VAX-DSM sends ASCII 0

Trap serial line errors

Certain environments can cause an error trap back to M when a network disconnect (or similar) occurs. Though the network may re-instate connections the generation of the error can cause the software to abort. If you want to trap such conditions such that the users merely continue processing when the network returns then set this item to true. Please note that this trap captures all M read errors thus ctrl-C will be trapped. We therefore recommend that this is not implemented in your development area.

Screen size (Screen Handler development only)

This enables the maximum number of lines and columns per screen to be defined. Ranges are 24-99 for lines and 80-132 columns. It must be noted that no screen should be used with a smaller definition than this size since the screen handler will automatically use the last line defined for the message bar.

Note GUI development allows forms up to the size of the screen being used for that form creation.

Input field definition characters

Input fields may be given a field extent character to define the size of such a field - initially provided as '_' though it may be amended but not removed. If no specific character is required it is suggested that it be changed to a space character. In addition, an input field may have left and right boundary characters to delineate the position and size of the item. Furthermore, it is possible to define an alternative right boundary character for use where that input field has an associated look-up facility. These are initially set as the characters '[' and ']'. They may, however, be amended. Though such characters may not be removed, in order to make them 'invisible' they may be set to a space character.

The system requires the character form to be entered here.

Check box definition characters

A tick box (or check box) is an on-off button type input field accepting only a carriage-return to reverse the status. The boundary characters for such a box can be defined here (defaulted to '[' and ']'). In addition, the character used to denote such a box turned 'on' may be defined. The de-

facto standard is the letter 'X' (upper case).

Radio button definition characters

Radio (or options) buttons provide a similar mechanism to tick boxes but, within a 'set' of buttons, only one may be turned on at any time. The act of turning one such button on will automatically turn all other buttons in the set to 'off'. Definition of the format for such buttons are achieved here in the same manner as the tick box definition.

Command button definition characters

Normal (command) buttons have textual content presented through a defined variable but, for 'dumb' terminals may have boundary characters defined. These are defaulted to parentheses characters though may be amended to whatever 'printable' characters are available.

List Box Cursor Up/down to position

List boxes, by default, use cursor up and down to move out of the control upwards or downwards and use cursor left and right to move between entries within the list box.

By setting this to true the above actions are reversed with cursor up/down moving between entries and cursor left/right moving out of the control.

List box Select on cursor position

If true, single select list boxes will automatically make the current entry the selected entry. If not set, only pressing carriage return will re-set the selection.

Highlight definition

This form is primarily for Character Terminal operation though the default usage for controls is used within the GUI form builder.

HL code	Usage
0	Applies window base, text & op
1	Help window base
2	Menubar
3	Look-up base button/select
4	Apps ip, tick boxes & R. Butns
5	List box
6	Error message
7	
8	Warning message
9	
10	Application window crnt item

Default usage	HL Code
Desktop	0
Text (label) controls	0
Output controls	0
Input (Edit) controls	4
Button & Select controls	3
Checkbox controls	4
Radio button controls	4
Listbox ctrls base	5
Listbox ctrls selected	15
Listbox ctrls selected & crnt	17
Hyperlink controls	9

OK Cancel

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

The top part of the window defines utilisation of the various codes. The descriptions are provided but may be changed by the user. Such descriptions are displayed in highlight code look-ups available where such codes are requested. As shown, some codes have no description and, therefore, will not be included in the look-up lists though they may actually be entered for highlighting purposes.

The lower half of the window defines default values for different aspects of the system.

The desktop default code denotes the base screen highlighting to enable screen refresh to reset areas of the base screen - it will not change the colour of the overall desktop.

The first block represents code values which will be provided as defaults when creating window items. This also applies to the first two items (normal window) in the second block.

Those entries referring to Menubar, Help and Look-up windows, if changed, will automatically cause the coding of those windows to be amended.

Finally, the error and warning message entries will also cause amendment to existing messages where the existing highlight coding of those messages corresponds to the default value before change.

3.2 SCREEN MAINTENANCE - TERMINAL TYPES

This option is selected by pressing the "terminal" button on the window as shown in the introduction to this chapter.

This aspect of the system set-up defines both the types of terminals available and what actions and responses are involved with such devices.

The maintenance function is selected from the screen maintenance facility options:

Terminal parameters consists of type details, line drawing, function key usage and highlight coding. The first set of parameters for a terminal type are presented upon entry to this facility.

GUI CLIENT

Item	Value
Connection Type	1
Default Message	%deflw
Sync Start Char	28
Sync End Char	29
Message Sep Char	30

Code

The type code for this type. It shouldn't be necessary to create another GUI client since the type "lw" should operate for all situations. A look-up is provided to list the existing types present.

Desc

The name/description of this terminal type

Connection type

- 1 = Normal start up (Via Telnet)
- 2 = LGSserver start up (via raw TCP) – requires using ^lgserver

Default message

A default message is displayed at the foot of each form to provide quick help (i.e. identifying the function keys such as help, look-up etc. A facility on the Maintenance menu is also provided for the creation of such a message.

Sync Start Char

Messages between the PC and the M server start with a pre-defined character which will never appear within the messages themselves. ASCII 28 is recommended.

Sync End Char

Messages between the PC and the M server end with a pre-defined character which will never appear within the messages themselves. ASCII 29 is recommended.

Message Sep Char

For certain messages it is necessary to replace characters such as CR or LF which would otherwise cause problems when being received into the M system. This is the replacement character. ASCII 30 is recommended.

CHARACTER TERMINAL

System set-up | Character Terminal

Help Find Maintenance

Type Name

Item	Value
Terminal mode	1
Box Drawing	
Start Escape	27,40,48
End Escape	27,40,66
ASCII Character Values	
Top Left	218
Top Right	191
Bottom Left	192
Bottom Right	217
Horizontal line	196
Vertical Line	179
Left 'T'	195
Right 'T'	194
Function Escape Sequences	
Abort	27,91,52,126
Backspace	8
Delete	27,91,51,126
Page Up	27,91,53,126
Page Down	27,91,54,126
Terminator	13
Refresh Form	27,91,49,126
Last Character	27,91,49,55,126
Clear Field	27,91,50,52,126
Jump To	27,79,83
Application	27,91,49,56,126
Cursor Up	27,91,65
Cursor Down	27,91,66
Cursor Left	27,91,68
Cursor Right	27,91,67
Help	27,79,80
Look Up	27,79,81
Menu	27,79,82
Reset Input	27,91,49,57,126
Insert/Overtyp	27,91,50,126
Control Exit	9
Default Message	defnt
Line set-up	
Normal (80 columns) setup	27,91,63,51,108
Wide (132 columns) setup	27,91,63,51,104
Highlight Code escape sequences	
0	27,91,50,50,59,51,50,59,52,48,109
1	27,91,50,50,59,51,48,59,52,51,109
2	27,91,50,50,59,51,55,59,52,49,109
3	27,91,50,50,59,51,55,59,52,52,109

OK Cancel

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

The example shown defines a telnet VT-200 compatible type for Cache on NT.

Type

The type code for this type. It shouldn't be necessary to create another GUI client since the type "lw" should operate for all situations. A look-up is provided to list the existing types present.

Name

The name of this terminal type

Terminal mode

Modes are 0 (Std ANSI mode) and 1 (VT200 compatible).

Box Drawing (line graphics)

The 'start' and 'end' items are for defining escape sequences (as a series of ASCII value equivalents separated by commas) to turn character graphics on and off for the terminal type. In the case above, no escape sequences are required. Assuming a terminal is in such character graphics mode the individual components of a box can then be defined. Such consists of corners, "T"s and straights and are defined as ASCII value equivalents.

Function keys

This set of parameters defines the 'function key' code sequence generation for the recognised activities available within the screen handler.

The definitions of these actions are described on section 2.4.2 above.

Single character ASCII values may be entered which will trigger actions by 'normal' single keyboard keystrokes instead of the detection of an escape sequence.

Default message

The default message is defined per terminal since it can be used for initial instruction to the user. The default provided gives a list of the primary function keys. Since such keys can differ between terminal types it is necessary to have a default message for each type. The name of the message is displayed for confirmation.

Line Set Up

For Lastic GUI terminal types only this must be entered as executable M command(s) to configure the line for speed, parity etc. If the line does not require any change then this can be left empty.

Normal (80 Column) set up

To enable switching from 132 to 80 columns, enter the escape sequence here.

Wide (132 column) set up

To enable switching to 132 columns wide, enter the escape sequence here.

Highlight code Escape Sequences

The system provides 20 codes (0-19) for use to render items on the screen with foreground/background colours etc. The use of these colours was shown in the Highlighting form described above.

Each code, if defined, will consist of a series of ASCII values, separated by commas, which produce an escape sequence to be actioned by the terminal. The example above shows settings for foreground and background colour combinations on a PC.

There are 20 normal highlighting options plus 'attributes off' and screen clear sequences.

For colour monitors, the 'attributes off' sequence should be the same as the base colours (in this case the same as code 0).

The Screen Handler window parameters utilise the following highlight codes as shown in the utilisation definition within the general details.

Maintenance Menu

This provides facilities to maintain a message and also to save this terminal type details as another type (i.e. to make a copy).

4. APPLICATION PROGRAM RELEASE

The Lastic software includes a release system to enable application programs and associated globals to be transferred from development to live areas as a set of released software. The release facilities provided are those utilised for the release of the Lastic system software. This chapter describes this release mechanism.

4.1 FILE DEVICE MAINTENANCE

The software distribution facility utilises sequential files as applicable to the host operating system environment. This enables software to be transferred not only between "UCIs" on your system but also can be transferred between heterogeneous systems. To achieve this, Lastic provide a facility for defining such files. This maintenance facility is accessed through the "fdm" function.

File device details

Help Find Device types

User code %tfr-out

Description Transfer out

Device code e:\Releases\temp\tfr.glo

Device type out_file

Open parameters "e:\releases\temp\tfr.glo":("NW")

Close parameters "e:\releases\temp\tfr.glo"

File type 0

OK Cancel Delete

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

User code

This is the 'code' for this file which will be known to the user such that they do not have to be aware of the internal file name or device code. For named files (where the file name is supplied at run-time) this must be the name of the variable (no subscripts) which will contain the file name at run-time.

Description

Used merely for selection list and confirmation purposes. It must be defined but is not utilised by the system.

Device id

The internal device code or file name. It may include directory specifiers if required.

For named files (file name supplied at run-time) where the M implementation uses the actual file name as the device code (rather than numbers) this must be the same as the user code (i.e. the variable to contain the name of the file). Where the M implementation uses device numbers this will be the file number in all cases.

Device type

This corresponds to device type attributes maintained by the facility described subsequently. A look-up facility is provided on this item. If the required type is not defined it may be created by invoking the maintenance through the pop-up menu attached to this window (see 5.2).

Open parameters

This is the 'operand' of a open command to open this device. This operand will be actioned as entered thus it is important that any required quotes are included.

Close parameters

As for the open parameters above.

Input/Output

This defines whether the file is for input or output. This enables the application program to determine whether the selected file is of the correct type and also enables look-ups to be selective in the files that are displayed. Entry must be either I (input) or O (output). Upper and lower case are accepted.

4.1.1 NAMED FILES

You can also define a file device with which you can open any file. This mechanism is used within the distribution facility when using the Named file option.

To create file type "%lzofile" suitable for using the named file option:

- | | | |
|----|------------------|---|
| 1. | Type | %lzofile |
| 2. | Device code | If your M system used file codes (eg 10, 51) enter that number
If your M system references file names, enter the type (in this example %lzofile) |
| 3. | Device type | Enter "out_file" - use look-up to find this. |
| 4. | Open parameters | Enter for the variable %lzofile to contain the file name |
| 5. | Close parameters | Enter for the variable %lzofile to contain the file name |
| 6. | Input/Output | Enter "O" |

For example

	Cache:	MSM
Type	%lzofile	%lzofile
Device code	%lzofile	51
Device type	out_file	out_file
Open parameters	%lzofile: ("NW")	51: (%lzofile, "w")
Close parameters	%lzofile	51
Input/Output	O	O

4.2 DEVICE TYPE MAINTENANCE

This is invoked through the menu attached to the file device window and the terminal type window described above. When selected the program will display a window as exemplified below:

Format	width	lines	Escape sequence
pn	80	24	27,91,63,51,108
pr	80	24	
ps	132	24	27,91,63,51,104
ln	132	24	
lr	132	24	
ls	132	24	

Device type

A code for reference purposes. It is referenced by the file device details (see previous section).

Description

For selection and confirmation purposes only.

Format

This heading lists the formats of: p = portrait, l=landscape, n=normal compression, r=reduced, s=small

Width

The maximum number of characters printable on a line when in this format.

Lines

The maximum number of lines per page when in this format.

Escape sequence

A series of ASCII values to be used to send the required command to the device to action the switch to this format. Two lines are provided in order to hold sufficient escape sequence values. Each ASCII value is delimited by a comma though there is no comma at the end of a line.

OK

Press this button to signify the acceptance of any changes made. Use the window abort key if no action is required.

NOTE: For non-printer files the escape sequences could be used for other purposes such as writing tape marks etc.

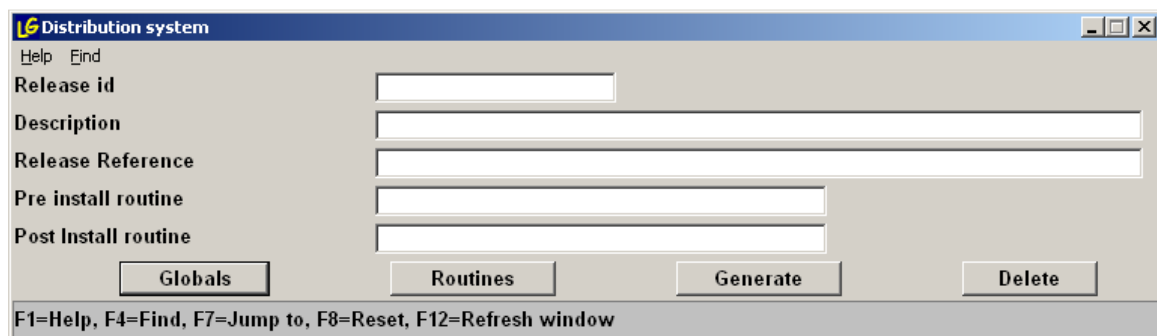
4.3 USING THE DISTRIBUTION SYSTEM

The software distribution system provides:

- i) Definition of release contents
(routines, globals and, optionally, pre and post installation processes)
- ii) Generation of release file
(including auto generation of all directly associated globals)
- iii) Distribution of release file within live area (installation)

This chapter will describe each of the above areas in detail.

To commence the distribution process select the distribution build (db) function. This will provide a window as exemplified below.



Enter a release id (or use the look-up facility to select an existing release reference).

The Description and Release Reference fields are for your information only and have no processing implications.

You can, optionally specify pre and post installation routines to be run automatically during the installation procedure ([label]^routine).

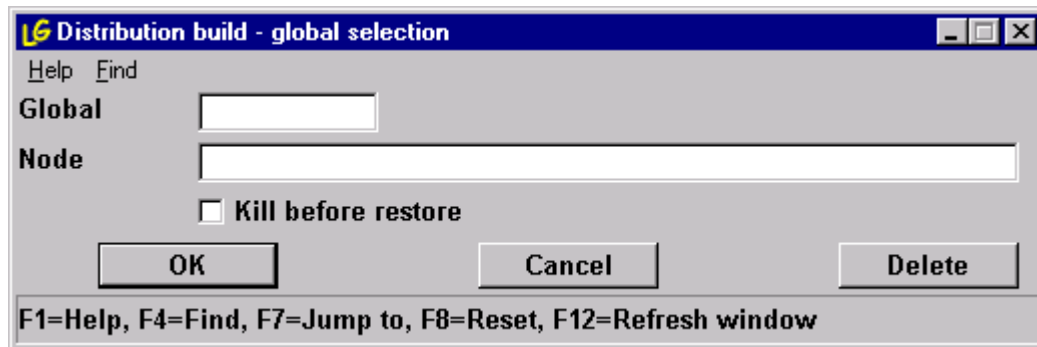
The pre-installation routine can be included within the distribution itself. If it is included the routine will first be installed and then run. If it is not found then it is checked for existence on the system. If not found the installation will be aborted.

You can force an abort of the installation from within your pre-installation program by setting an error message (text not a code) in %lzser and then quitting.

If the routine ran satisfactorily then quit with %lzser either undefined or set to null.

4.3.1 GLOBALS

To select globals for transfer (which cannot be automatically detected) press the 'globals' button on the initial window. This will invoke the global selection window as shown below:



Global

Enter a global name. Globals already included may be selected through a look-up list though globals not known to this release id must be specified. The global must exist in order to be selectable.

Node

This is optional and permits the selection of a global node (to the depth required) defining each subscript value applicable. do not use quotes around any subscript. All subscripts are separated by commas. Data at this node and all lower level nodes will be copied for release.

A look-up is available on this item to determine any previously defined nodes for this global.

If not selected the whole global will be released. The node must exist to be selectable.

Kill before restore

Where such globals already exist within the target system it is possible to either merge with the existing global structure or to replace the specified node (and lower level subscripts).

If replacement is required enter "Y" (y) here. If a merge is required enter "N" (n).

Buttons

Press the "ok" buttons to save this global reference, press "cancel" to abort this selection and return for another or press "delete" to remove the global reference from the list of globals to be released.

When complete, abort this window to return to the master release control window.

4.3.2 ROUTINES

Selection of the routines button in the control window will cause the routine selection window to be invoked as shown below.



Routines may be selected by entering either the full name or a partial name plus the system defined wild card character. A look-up facility is also available to list previously selected routines.

This mechanism requires the definition of the routine directory global item shown in the general details window (see section 3.1 above).

Selected routines may be added ("ok" button) or deleted from the list ("delete" button). incorrect selection may be aborted using the "cancel" button.

When selection is complete the window is aborted and return is made to the control window above.

4.3.3 GENERATION

Having selected the globals and routines to be released the user may now select the generate button. Selecting this will invoke the generation window as shown below.

Generate distribution

Help Find File

Distribution file

☒ **Fixed file** ☐ **Named file** **Max size (Mbytes)**

File name

☐ **Encrypt** ☐ **Delete source code**

☐ **Auto gen global refs** ☐ **Include button name library**

OK **Cancel**

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

Fixed File

Choose the Fixed file option if you want to select an existing file definition naming a specific file. This is compatible with previous versions and will cause the generation of a single distribution file. Selecting this option disables the definition of the maximum file size.

Named File

Choose the named file option if you want to define your own file name and path (you must have created the file definition for "%lzofile" to use this option - see section 4.1.1 above). You must use this option if you want to generate multiple distribution files each with a maximum size.

Max size (MBytes)

Using the Named file option, you can cause the distribution mechanism to split the distribution into a number of files each of a maximum size to enable easy inclusion on diskettes for example. Enter the size in the format n..mmm (e.g. 1.4 to define 1.4 MBytes). You may enter up to 3 decimal places. If you enter a Max size then, when entering the named file, do not specify a file extension. The Lastic distribution system will create file extensions starting with 001.

File Name

For Fixed files you may select the required output "flat file" from those defined in the file device facility (output files only) either by direct entry or through using the look-up facility. Please note that file devices must be defined for your Mumps implementation.

For Named files, enter the full path and file name (without file extension if using the Max Size option).

Encrypt

Check this box if the routines within the distribution file are to be encrypted. The routine ^lzsinst will de-encrypt the routines before saving. If using this mechanism we recommend that the routine ^lzsinst is held on the target site in compiled form only (delete the source code when installed).

Delete Source Code

Check this box if the source code is to be deleted (Cache only) after compiling.

Auto gen global refs

This is a tick box and, if set, will cause the system to perform the following before final distribution is generated.

- i) Each routine selected is scanned for all occurrences of:
"form^lzsr" "get^lzsr(" "%lzser=" "loadf^lzpr(" "= "....

The final item being to identify literal parameters containing grid cell look-up codes and help codes.

For each occurrence of "form^lzsr" or "get^lzsr" found the system will obtain the system and window code and log these to a temporary work stack.

For each occurrence of "%lzser=" found the system will obtain the message code and likewise log this entry.

For each occurrence of "loadf^lzpr" found the program will obtain the print handler form and associated block details and log those entries

- ii) When all scanning has been completed the program will then scan the windows parameters and also log all references to data-types, help message codes, look-up table codes, function codes, validation codes

NOTES Button parameters cannot be detected thus the button library global can be logged as a single entity for transfer of the whole button global (^lbut).

- iii) The log entry is then accessed and all entries are copied to the global selection list with each entry flagged as "kill before restore".

4.3.4 INSTALLATION

Software installation must not be run when anyone is using any software for which new versions are being installed.

Using global restore (e.g. Cache ^%GI), restore the distribution from the generated file into the target UCI/Namespace. If multiple files have been generated using the Max size option above, restore all extensions of that file (it doesn't matter what order they are restored in).

Multiple releases (with different release IDs) may be restored and then updated with a single install run.

Run ^lzsinst

ROUTINES

Run-time routines

lzsr	Screen handler run-time control
lzsrinit	Initialisation routine
lzslib	Library routine - used by the screen handling system
lzsliba	Library - msgbox, GUI DDE, Print , XFer etc
lzslibb	Library - control property amend routine
lzslibc	Library - char terminal drag/drop action
lzslibh	Library - Help operations
lzslibv	Lastic validation code library
Character terminals	lzsr0c Run time primary driver lzsr02 Help and look-up control lzsr03 Menu control lzsr04 Form pasting and refresh lzsr06 Internal function library lzsr09 Form load lzsr011 MessageBox function lzsr0c0 Form as control driver lzsr0cB Command button driver lzsr0cC Combobox driver lzsr0cCE Editable Combobox Driver lzsr0cD* Document control drivers lzsr0cG* Grid control drivers lzsr0cH* Hypertext control drivers lzsr0cl Input control driver lzsr0cL List box control driver lzsr0cN* Outline control drivers lzsr0cR Radio button driver lzsr0cS Select control driver lzsr0cX Checkbox driver lzsr0cZ Terminal editor driver lzsr0ev Control event handler lzsr0kev Function key event handler lzsr0men Menu handler
Lastic GUI	lzsr3c Run-time handler lzsr3cG Grid build/action lzsr3cN Outline control build/action lzsr3cW Workbench build/action lzsr32 Help/look-up controller lzsr33 Menu controller lzsr34 Pasting and refresh lzsr36 Internal function library lzsr38 Load form parameters lzsr39 Send form parameters lzsr310 Graph operations lzsr3ev Control event handler lzsr3kev Function key event handler lzsr3men Menu handler

General system environment maintenance routines

lzsus	Set up facilities
lzsm007	File device maintenance
lzsm008	Device type maintenance
lmenu	System software menu routine
lzsinst	System installation routine
lzssetup	Character terminal based set-up routine

GLOBAL LAYOUTS

Global name: ^lzs

Global description: system parameters

Field no	Description	format	length
^lzs("d")			
1	delimiter for system globals	X	1
^lzs("def","ttype")			
1	default terminal type	X	10
^lzs("devtype",term_type)			
1	term_type code	X	10
2	description	X	30
^lzs("devtype",term_type,1,code)			
	where code =	ln (landscape normal size - eg 10 cpi) lr (landscape reduced character size - eg 12 cpi) ls (landscape small character size - eg 16 cpi) pn (portrait normal size - eg 10 cpi) pr (portrait reduced character size - eg 12 cpi) ps (portrait small character size - eg 16 cpi)	
1	code	X	2
2	no. of chars per line	999	3
3	no. of lines per page	999	3
4	set on esc sequence (1st part)	999,999...	40
5	set on esc sequence (2nd part)	999,999...	40
^lzs("echo")			
1	echo on executable command	X	30
2	echo off executable command	X	30
^lzs("errortrap")			
1	error trap entrypoint	[label]^routine	17
^lzs("file",device_name)			
1	device name	X	10
2	description	X	30
3	device system 'code'/file name	X	20
4	terminal type	X	10
5	device open paramters	X	30
6	device close parameters	X	30
7	input/output	"I/O"	1
^lzs("helpfocus")			
1	set focus to text control	0/1	1
^lzs("hl","def",n)			
1	Default highlight codes by object type n highlight code	99	2

GLOBAL LAYOUTS

Global name: ^lzs

Global description: system parameters

Field no	Description	format	length
^lzs("hl","utl",n) Highlight code usage descriptions			
1	n - highlight code	99	2
2	Description of use	X	30
^lzs("hlprtn") bespoke help system entryptpoint program call for help system			
1		X^X	17
^lzs("if") input field formatting			
1	input field extent character	X	1
2	input field start character	X	1
3	input field end character	X	1
4	Input field look-up end char	X	1
^lzs("ipupd") input field update action			
1	action mode when loosing focus to previous control 0 = Reset displayed input from target variable 1 = update target variable with displayed input		
^lzs("lbcursor") list box cursor action			
1	action mode 0 = left/right moves vertically, up/down exits list box 1 = up/down moves vertically, left/right exits list box	N	1
^lzs("lbselect") list box select current entry (single select list boxes)			
1	action mode 0 = do not make selection = current entry 1 = make selection = current entry	N	1
^lzs("lg","mon",\$j) Lastic GUI flag for message monitoring			
1	=1	N	1
^lzs('m') M implementation control			
1	escape seq identifier	N	3
^lzs("nbf") normal button format			
1	reserved		0
2	field start char	X	1
3	field end char	X	1
^lzs("prod",x) Lastic software products in use			
	x= SH (Screen Handler), LG (GUI), EL (Rule Based System) CO (Contacts), PH (Print Handler)		
1	=1 if product used	N	1

GLOBAL LAYOUTS

Global name: ^lzs

Global description: system parameters

Field no	Description	format	length
^lzs("r",n)			
1	reject characters where n = 1-5 ASCII values of reject characters	999,999...	40
^lzs("rbf")			
1	radio button field formatting 'on' character	X	1
2	field start character	X	1
3	field end character	X	1
^lzs("rder")			
1	read error trap in use	null/1	1
^lzs("rtngbl")			
1	routine global directory name global name	X	8
^lzs("scrsz") Screen max size			
1	max lines	N	2
2	max columns	N	3
^lzs("tbf")			
1	tick box formatting 'on' character	X	1
2	field start character	X	1
3	field end character	X	1
^lzs("termtype",term_type) Terminal types			
1	Type code	X	10
2	Type description	X	30
3	Mode	N	2
4	GUI sync start char	N	3
5	GUI sync end char	N	3
6	GUI text separator char	N	3
^lzs("termtype",term_type,"b",code) box drawing characters			
	where code =		
	bl	bottom left corner	
	br	bottom right corner	
	h	horizontal line	
	tl	top left corner	
	tr	top right corner	
	v	vertical	
1	ascii value of character	999	3
^lzs("termtype",term_type,"b","gi")			
1	box - graphics on esc seq	999,999...	15

GLOBAL LAYOUTS

Global name: ^lzs

Global description: system parameters

Field no	Description	format	length
^lzs("termtype",term_type,"b","go")			
1	box - graphics off esc seq	999,999...	15
^lzs("termtype",term_type,"defmsg")			
1	default bar message id	X	10
^lzs("termtype",term_type,"f",code) function key escape sequences where code =			
	a	abort	
	ap	application function key	
	c	terminator (eg carriage return)	
	d	cursor down	
	el	erase to left (eg backspace)	
	er	erase from right (eg delete)	
	ex	control exit (for document & grid controls)	
	h	help	
	l	cursor left	
	lc	last char jump	
	lu	look-up	
	opt	pop-up menu	
	pd	page down	
	pu	page up	
	r	cursor right	
	ref	screen refresh	
	ri	reset input field	
	u	cursor up	
	cf	clear field	
	io	insert/overtyp	
1	\$ZB ascii value esc seq	999,999...	20
^lzs("termtype",term_type,"h",code) highlight codes per terminal type where code =			
	0-19		
	cs	clear screen	
	"off"	all attributes off	
1	ascii value escape sequence	999,999...	20
^lzs("timeout")			
1	window time out in seconds	N	6
^lzs("tmode",n) Terminal type mode			
1	Terminal type mode	N	2
2	Description	X	30
3	Spare		
4	Spare		
5	Spare		

GLOBAL LAYOUTS

Global name: ^lzs

Global description: system parameters

Field no	Description	format	length
^lzs("tp") Text proc controls			
1	Paragraph terminator	127	3
2	Tab character	9	1
3	Tab spacing	8	1
4	Reserved		
^lzs("tp",1,N) Text processing edit functions			
1	Function number	N	2
2	Function description	X	30
^lzs("use") data-types & tagged text input usage			
1	data types in use	0/1	1
2	tagged text level	0/1/2	1
^lzs("w")			
1	wild card character (ASCII value)	999	3