STITLE, A PACKAGE TO CREATE SCROLLED MOVIE TITLES

David J. Kennison
NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000
email: kennison@ncar.ucar.edu (128.117.64.4)

1.0 INTRODUCTION 1
1.1 Usage 1
1.2 Miscellaneous Information 2
1.2.1 Portability 2
1.2.2 Required Library Packages 2
1.2.3 History 2
2.0 ROUTINES 3
2.1 FTITLE (MTST) 3
2.1.1 Usage 3
2.1.2 Arguments 3
2.1.3 Input Data Read By FTITLE 3
2.1.4 Internal Parameters That Affect FTITLE 4
2.2 STITLE (CRDS,NCDS,IYST,IYND, . . .) 4
2.2.1 Usage 4
2.2.2 Arguments 4
2.2.3 No Input Data Read by STITLE 6
2.2.4 Internal Parameters That Affect STITLE 6
2.2.5 Algorithm 6
2.3 SLOGAP (TIME,MTST) 7
2.3.1 Usage 7
2.3.2 Arguments 7
2.4 SLGETx (PNAM,xVAL) 7
2.4.1 Usage 7
2.4.2 Arguments 7
2.5 SLSETx (PNAM,xVAL) 8
2.5.1 Usage 8
2.5.2 Arguments 8
2.6 SLRSET 8
2.6.1 Usage 8
2.6.2 Arguments 8
2.7 SLUBKG (IPOC) 8
2.7.1 Usage 8
2.7.2 Arguments 9
3.0 INTERNAL PARAMETERS 9
4.0 ERROR MESSAGES 12
5.0 EXAMPLES 15

STITLE, A PACKAGE TO CREATE SCROLLED MOVIE TITLES

David J. Kennison
NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000
email: kennison@ncar.ucar.edu (128.117.64.4)

1.0 INTRODUCTION

The package STITLE is used to create movie titles. The titles can be scrolled, faded in, and/or faded out. A background color and one or more foreground colors can be specified.

1.1 Usage

The package STITLE has eight user entry points, as follows: FTITLE, STITLE, SLGETI, SLGETR, SLOGAP, SLSETI, SLSETR, and SLRSET. There is also a routine called SLUBKG that is called by STITLE and exists solely to be replaced by the user.

FTITLE stands for "fixed titles" and is used when the following conditions are met:

- Each group of text lines is to fit on one frame (no scrolling).
- All text lines in a group are to be written in the same color.
- Vertical spacing of text lines is to be done automatically.
- Text lines are to be centered horizontally.
- There are to be no more than 80 characters per text line (including PLOTCHAR "function codes").
- In production mode, blank frames are to be generated before and after each group of title frames.
- No more than 120 text lines are to be displayed on a single frame.

STITLE stands for "scrolled titles" and is used when the above conditions are not met.

SLOGAP is used to output blank-frame gaps in a manner consistent with FTITLE.

SLSETI and **SLSETR** are used to set the values of "internal parameters" whose values affect the behavior of FTITLE and/or STITLE.

SLGETI and **SLGETR** are used to retrieve the current values of internal parameters.

SLRSET is used to reset the values of all internal parameters of STITLE to the original defaults.

SLUBKG is not to be called by the user. It is called by STITLE just before and just after the drawing of titles on each frame. The default version of the routine does nothing. A user version may be supplied to add desired graphics to the frame.

Each of the routines of STITLE is described in detail in the section “ROUTINES”.

Internal parameters are discussed in detail in the section “INTERNAL PARAMETERS”.

Error messages are described in the section “ERROR MESSAGES”.

1.2 Miscellaneous Information

1.2.1 Portability

STITLE conforms to the FORTRAN-77 and GKS standards.

1.2.2 Required Library Packages

STITLE uses the NCAR Graphics packages COLCONV, GFLASH, PLOTCHAR, and SETER.

1.2.3 History

STITLE replaced the previous routines AUTOTITLE and SCROLL at NCAR.

In April, 1980, the package was rewritten, changed to use the new PWRITX instead of the old PWRX, and made portable.

In June, 1989, the package was converted to be compatible with the FORTRAN-77 and GKS standards, and was made to use PLOTCHAR and GFLASH. User entries SLGETI, SLGETR, SLSETI, and SLSETR were added. Fade-in and fade-out were added.

In January, 1993, the package was changed to use clipping built into PLOTCHAR. An error in the handling of continuation lines was corrected. The names of all internal entities were changed to start with ‘SL’. Parameter interface routines were made consistent with those in other packages. STITLE was changed to accept “cards” of arbitrary length and FTITLE was changed to accept 80-character text lines.

In July, 1995, provision was made for the use of more than one foreground color, the number of frames per second was made an internal parameter, fade-in/fade-out capabilities were enhanced, the routine SLRSET was added, user access to the routine SLOGAP was enabled, and error recovery was implemented.

2.0 ROUTINES

2.1 FTITLE (MTST)

The routine FTITLE is used for unscrolled movie titles. It reads, from standard input, the information necessary to define the desired title frames, and then calls STITLE to create those frames.

2.1.1 Usage

To call the subroutine FTITLE, use the following FORTRAN statement:

```
CALL FTITLE (MTST)
```

2.1.2 Arguments

MTST (an input expression of type INTEGER) is a switch indicating whether this is a “real” run or a “test” run:

0 means “real run”.

1 means “test run”.

During real runs, each title frame is repeated as many times as necessary to display it for a user-specified time period (at 'NFS' frames per second). The internal parameters 'TM1', 'TM2', and 'TM3' specify where blank frames are to be generated: 'TM1' seconds worth of blank frames are placed before the first title frame, 'TM2' seconds worth are placed between each pair of consecutive title frames, and 'TM2'+'TM3' seconds worth are placed after the last title frame. (Blank frames were to allow for splicing; they are probably no longer necessary.) If the user has turned on fade-in (by setting the internal parameter 'FIN' non-zero), and/or fade-out (by setting the internal parameter 'FOU' non-zero), the required fade-in and fade-out frames are generated, as well.

During test runs, each title frame appears from one to three times (once for fade-in, once for the stationary holding time, and once for fade-out); each of these frames has a legend indicating how many seconds the frame represents in a real run. Each sequence of blank frames will be replaced by a single frame with a message indicating how many seconds worth of blank frames it represents.

2.1.3 Input Data Read By FTITLE

FTITLE reads data from the unit specified by the internal parameter 'ICU', which has the default value 5, to specify standard input; this parameter can be given a different value to specify that another unit should be read instead.

The input data are read in groups. Each group represents one title frame. There can be any number of groups. FTITLE keeps processing groups until a group with NCDS = 0 is read or an end-of-file is encountered on the input unit. A group consists of the following:

- A header line from which variables NCDS, TIME, and SIZE are read, using the FORTRAN format “(I5,2F5.1)”.

  NCDS is the number of text lines that follow. If NCDS = 0, FTITLE returns to the calling routine.

  TIME is the time, in seconds, that the title frame should be displayed (not including fade-in and fade-out, if any).

  SIZE is the desired character size, given as a multiplier of a default height specified by the value of the internal parameter 'PSZ', the default value of which is 21 (out of 1024). Values of SIZE from .75 to 2.5 are recommended.

- Text lines, each containing one line of the title frame. PLOTCHAR function codes may be used as specified in the documentation for that package. Characters should not appear beyond column 80.

2.1.4 Internal Parameters That Affect FTITLE

The internal parameters 'ALN', 'BGB', 'BGC', 'BGF', 'BGG', 'BGR', 'FGB', 'FGC', 'FGF', 'FGG', 'FGR', 'FIN', 'FOU', 'GSZ', 'ICO', 'ICU', 'LOG', 'LX1', 'LX2', 'LY1', 'LY2', 'MAP', 'NFS', 'NXE', 'NXS', 'ORV', 'PSZ', 'SBK', 'SFG', 'TM1', 'TM2', 'TM3', 'VPB', 'VPL', 'VPR', 'VPT', and 'WID' all affect the behavior of FTITLE in one way or another. Some of these have been mentioned above; all are described in the section “INTERNAL PARAMETERS”.

2.2 STITLE (CRDS,NCDS,IYST,IYND, . . .)

(The remaining arguments are **TMST**, **TMMV**, **TMND**, and **MTST**.)

This routine creates scrolled movie titles. STITLE reads no card images; all information is passed to it through its argument list.

2.2.1 Usage

To call the subroutine STITLE, use the FORTRAN statement

```
CALL STITLE (CRDS,NCDS,IYST,IYND,TMST,TMMV,TMND,MTST)
```

2.2.2 Arguments

CRDS (an input array, dimensioned NCDS, of type CHARACTER*n, where “n” is greater than or equal to 21) is the “card input buffer”. This array must be filled, prior to calling STITLE, either by internal manipulations or by reading n-character “cards”. Each element of the array CRDS represents one line on the scroll (or, sometimes, a continuation of a previous line) and contains the following:

- **Columns 1-5**: MX, the X coordinate of the line of text on the scroll. This is normally a value between 1 and 1024, inclusive. Exactly how the line of text is positioned relative to the specified X coordinate depends on the value of ICNT (in columns 14-15).

If the value -9999 is used for MX, it indicates a continuation line: characters from columns 21 through “n” are just appended to the characters from the previous card to form the line of text. Any number of continuation cards may be used, but the total number of characters in a line of text must not be greater than 500.

Trailing blanks are omitted from each card, including those that are followed by a continuation card; thus, if there are to be blanks between the characters from one card and the characters from a continuation card, those blanks must be placed in columns 21 and following of the continuation card.

On a continuation card, columns 6-20 are ignored.

- **Columns 6-10**: MY, the Y coordinate of the line of text on the scroll. MY may range from -9999 to 99999.
- **Columns 11-13**: ICLR, the index of the color to be used for the line of text. If this field is blank, the default foreground color specified by the value of the internal parameter 'FGC' will be used.
- **Columns 14-15**: ICNT, the centering option:

  0 means “start the text at MX”.

  1 means “center the text about MX”.

  2 means “end the text at MX”.
- **Columns 16-20**: SIZE, the desired size of the characters to be used in writing the line. SIZE is given as a multiplier of a default height specified by the value of the internal parameter 'PSZ', the default value of which is 21 (out of 1024). Values of SIZE from .75 to 2.5 are recommended.
- **Columns 21-n**: Text for this line (or for continuation of a line when MX = -9999).

NCDS (an input expression of type INTEGER) is the dimension of the array CRDS (i.e., the number of card images in it).

IYST (an input expression of type INTEGER) is the Y coordinate that will be at the center of the screen when the text is first displayed.

IYND (an input expression of type INTEGER) is the Y coordinate that will be at the center of the screen when the text is last displayed.

TMST (an input expression of type REAL) is the time in seconds that the scroll will be stationary at IYST. One second is recommended.

TMMV (an input expression of type REAL) is the time to move the scroll from IYST to IYND. This should be the time required to read the text aloud at slow to normal speed.

TMND (an input expression of type REAL) is the time that the scroll will be stationary at IYND. One second is recommended.

MTST (an input expression of type INTEGER) is a switch to indicate whether this is a “real” run or a “test” run.

0 means “real run”.

1 means “test run”.

During real runs, frames are created for the fade-in sequence (if the user has turned on fade-in by setting the internal parameter 'FIN' non-zero), the stationary sequence at the start (if TMST is non-zero), the scrolling time (if TMMV is non-zero), the stationary sequence at the end (if TMND is non-zero), and the fade-out sequence (if the user has turned on fade-out by setting the internal parameter 'FOU' non-zero).

During test runs, only selected frames are created: a frame for the fade-in sequence (if fade-in is turned on), a frame for the stationary time at the start, a set of frames representing the scrolling sequence, a frame for the stationary time at the end, and a frame for the fade-out sequence (if fade-out is turned on). Each has a legend indicating either for how many seconds the frame will be shown or, if it is part of a scroll sequence, how many seconds into the scroll time it occurs; during real runs, these legends are omitted, of course.

Fade-in and fade-out are also affected by the values of the internal parameters 'BGF', which selects the type of fade-in/fade-out to be used for the background color, and 'FGF', which serves the same function for the foreground color. (The older parameters 'SBK' and 'SFG' may still be referenced, but their use is no longer recommended: setting 'SBK' has the effect of giving 'BGF' an appropriate value and setting 'SFG' has the effect of giving 'FGF' an appropriate value.)

2.2.3 No Input Data Read by STITLE

STITLE receives all input through its argument list; unlike FTITLE, it doesn’t read anything from an input unit.

2.2.4 Internal Parameters That Affect STITLE

The internal parameters 'ALN', 'BGB', 'BGC', 'BGF', 'BGG', 'BGR', 'FGB', 'FGC', 'FGF', 'FGG', 'FGR', 'FIN', 'FOU', 'INC', 'LOG', 'LX1', 'LX2', 'LY1', 'LY2', 'MAP', 'NFS', 'NXE', 'NXS', 'ORV', 'PSZ', 'SBK', 'SFG', 'VPB', 'VPL', 'VPR', 'VPT', and 'WID' affect the behavior of STITLE in one way or another. Some of these have been mentioned above; the others are described in the section “INTERNAL PARAMETERS”.

2.2.5 Algorithm

STITLE in effect moves the specified text up through the screen window, outputting the frames required to generate a movie sequence of the duration specified by the user.

At each frame STITLE skips plotting lines of text that are completely outside of the screen window. Lines of text that are partially in the window are clipped by PLOTCHAR.

2.3 SLOGAP (TIME,MTST)

This routine is used to output TIME seconds worth of blank frames in a manner consistent with FTITLE. During a real run, it actually produces blank frames, but during a test run, it produces a single frame saying how many seconds worth of blank frames will be produced during a real run.

2.3.1 Usage

Use the following FORTRAN statement:

```
CALL SLOGAP (TIME,MTST)
```

2.3.2 Arguments

TIME (an input expression of type REAL) specifies the desired number of seconds worth of blank frames.

MTST (an input expression of type INTEGER) is a switch to indicate whether this is a "real" run (MTST = 0) or a "test" run (MTST = 1).

2.4 SLGETx (PNAM,xVAL)

These routines are used to get the current values of internal parameters that affect the behavior of STITLE and FTITLE. The "x" stands for either an "I" or an "R". See the section "INTERNAL PARAMETERS" for a complete list of all the internal parameters.

2.4.1 Usage

Use one of the following FORTRAN statements:

```
CALL SLGETI (PNAM,IVAL)
CALL SLGETR (PNAM,RVAL)
```

2.4.2 Arguments

PNAM (an input constant or variable of type CHARACTER) specifies the name of the parameter whose value is to be retrieved. Only the first three characters of the string are examined.

IVAL (an output variable of type INTEGER) is the current value of the internal parameter. If the internal parameter is of type REAL, INT(real value) will be returned.

RVAL (an output variable of type REAL) is the value of the internal parameter. If the internal parameter is of type INTEGER, REAL(integer value) will be returned.

2.5 SLSETx (PNAM,xVAL)

These routines are used to set the values of internal parameters that affect the behavior of STITLE and FTITLE. The “x” stands for either an “I” or an “R”. See the section “INTERNAL PARAMETERS” for a complete list of all the internal parameters.

2.5.1 Usage

Use one of the following FORTRAN statements:

```
CALL SLSETI (PNAM,IVAL)
CALL SLSETR (PNAM,RVAL)
```

2.5.2 Arguments

PNAM (an input constant or variable of type CHARACTER) specifies the name of the parameter to be set. Only the first three characters of the string are examined.

IVAL (an input expression of type INTEGER) is the desired value of the internal parameter. If the internal parameter is of type REAL, it will be given the value REAL(IVAL).

RVAL (an input expression of type REAL) is the desired value of the internal parameter. If the internal parameter is of type INTEGER, it will be given the value INT(RVAL).

2.6 SLRSET

This routine is called to return all the internal parameters of STITLE to the default state.

2.6.1 Usage

Call SLRSET by executing the statement

```
CALL SLRSET
```

2.6.2 Arguments

SLRSET has no arguments.

2.7 SLUBKG (IPOC)

2.7.1 Usage

This routine is not to be called by the user. It is called by STITLE as title frames are being produced. The default version of the routine does nothing but RETURN; a user version can be supplied to draw a background on the title frame. In such a user-supplied version, care should be taken not to alter the state of GKS or SPPS in ways that will interfere in an

undesirable way with the drawing of the titles. The details of using this routine have not all been worked out, but the example named “slex02” illustrates one possible use of it..

2.7.2 Arguments

IPOC (an input expression of type INTEGER) indicates the position of the call to SLUBKG in the routine STITLE, as detailed in the following table:

Value of IPOC	Position of call to SLUBKG
-1	Just before drawing titles on a “fade-in” frame.
+1	Just after drawing titles on a “fade-in” frame.
-2	Just before drawing titles on a “start” frame.
+2	Just after drawing titles on a “start” frame.
-3	Just before drawing titles on a “move” frame.
+3	Just after drawing titles on a “move” frame.
-4	Just before drawing titles on an “end” frame.
+4	Just after drawing titles on an “end” frame.
-5	Just before drawing titles on a “fade-out” frame.
+5	Just after drawing titles on a “fade-out” frame.

3.0 INTERNAL PARAMETERS

The table below describes all of the internal parameters of STITLE; for each, it gives the name of the parameter, its default value, and its function in the package.

Name	Default value	Function
'ALN'	0	An integer flag to control whether alignment frames with dots in the corners are produced. 'ALN'=0 means “suppress them”, 'ALN'=1 means “produce them”, and 'ALN' = 2 means “produce them and label the frames as “alignment frames”.
'BGC'	0	An integer specifying the color index to be used for the background color. If 'BGC' = 0, GKS is depended upon to fill the background; otherwise, STITLE does a GFA call to fill it.
'BGF'	-2	An integer specifying the type of background fade-in/fade-out. The value -1 implies that background fades are suppressed. The value -2 implies a fade from/to black by mapping RGB values to HSV values and varying V. A non-negative value of the form “n_i x 1000 + n_o” implies a fade-in from the color with color index “n_i” and a fade-out to the color with color index “n_o”; for example, the value 11012 implies fading in from color 11 and fading out to color 12. Use “999” to suppress one of the fades; for example, the value 26999 implies fading in from color 26, but no fade-out, while the value 999026 implies no fade-in, but a fade-out to color 26. These fades are done using linear interpolation in the RGB color system.

Name	Default value	Function
'BGR', 'BGG', 'BGB'	Device default.	Real values, in the range 0. to 1., to be used for the red, green, and blue components of the background color. When one of these values is set, it defines a component of the color associated with the current value of the color index 'BGC'.
'FGC'	1	An integer specifying the color index to be used for the default foreground color. All informative information on “test run” frames and all title lines for which no other color index is specified will be in the color specified by this color index.
'FGF'	-2	An integer specifying the type of foreground fade-in/fade-out. The value -1 implies that foreground fades are suppressed. The value -2 implies a fade from/to black by mapping RGB values to HSV values and varying V. A non-negative value of the form “n_i x 1000 + n_o” implies a fade-in from the color with color index “n_i” and a fade-out to the color with color index “n_o”; for example, the value 11012 implies fading in from color 11 and fading out to color 12. Use “999” to suppress one of the fades; for example, the value 26999 implies fading in from color 26, but no fade-out, while the value 999026 implies no fade-in, but a fade-out to color 26. These fades are done using linear interpolation in the RGB color system. Fades are done on all the foreground colors known to STITLE because SLSETR has been called to define their RGB components or because their color indices have appeared on card images defined by the argument CRDS.
'FGR', 'FGG', 'FGB'	Device default.	Real values, in the range 0. to 1., to be used for the red, green, and blue components of a specified foreground color. The index of the color being defined may be specified in parentheses following the 'FGR', 'FGG', or 'FGB'; for example, the statement “CALL SLSETR('FGR(2)',.5)” defines the red component of the color associated with color index 2 to be .5. If no subscript appears, the color index defined by the current value of 'FGC' is the one whose definition is affected. A color can also be defined by calling the GKS subroutine GSCR and specifying the color index on one of the card images in the CRDS array. Note, however, that any color which is to be used solely in a user-supplied version of the routine SLUBKG and which is desired to be subject to fade-in/fade-out should be defined by means of SLSETR calls. Characters may be drawn in any of the foreground colors.
'FIN'	0.	Number of seconds during which to fade in titles. The background color and foreground color are faded in independently. Each color is faded in in the manner specified by the value of either 'BGF' or 'FGF', whichever is appropriate.
'FOU'	0.	Number of seconds during which to fade out titles. The background color and foreground color are faded out independently. Each color is faded out in the manner specified by the value of either 'BGF' or 'FGF', whichever is appropriate..

Name	Default value	Function
'GSZ'	40.	Value for interline spacing, defined in terms of a 1024x1024 grid. Used only by FTITLE.
'ICO'	1	Centering option. Set to 0 to get left edges lined up at X-coordinate 64, and to 2 to get right edges lined up at X-coordinate 960. Set to 1 for centered text. Used only by FTITLE.
'ICU'	5	Unit number for reading input. Used only by FTITLE.
'INC'	300	Vertical spacing between test frames.
'LOG'	4	FORTRAN logical unit number for opening WISS (Workstation-Independent Segment Storage).
'LX1'	0	Integer, in the range 0 to 32767, specifying the X coordinate of the lower left corner of the viewport. This parameter is superseded by 'VPL', which should be used instead.
'LX2'	32767	Integer, in the range 0 to 32767, specifying the X coordinate of the upper right corner of the viewport. This parameter is superseded by 'VPR', which should be used instead.
'LY1'	0	Integer, in the range 0 to 32767, specifying the Y coordinate of the lower left corner of the viewport. This parameter is superseded by 'VPB', which should be used instead.
'LY2'	32767	Integer, in the range 0 to 32767, specifying the Y coordinate of the upper right corner of the viewport. This parameter is superseded by 'VPT', which should be used instead.
'MAP'	100	If greater than zero, the value to which PLOTCHAR's internal parameter 'MA' is to be set; otherwise, the user will be assumed to have set it. The default values of 'MAP' and 'ORV' are such as to create scrolled titles in the normal way; other values may be used to create titles that scroll upwards on the surface of the globe or that scroll off into space (a la Star Wars); the appropriate transformations need to be embedded in a user-written version of the PLOTCHAR routine PCMPXY. (This is still rather new; eventually, there will be an example showing how to do this.)
'NFS'	24.	The number of frames per second. The default, 24, is appropriate for film. For videos, one should use 'NFS' = 30.
'NXE'	512	Analogous to the STITLE argument IYND, to allow for limited scrolling in the X direction. 'NXE' must be within the current STITLE window.
'NXS'	512	Analogous to the STITLE argument IYST, to allow for limited scrolling in the X direction. 'NXS' must be within the current STITLE window.
'ORV'	1.E12	If non-zero, the value to which PLOTCHAR's internal parameter 'OR' is to be set; otherwise, the user will be assumed to have set it. See the discussion of the parameter 'MAP', above.
'PSZ'	21.	Default character height on a 1024 by 1024 grid.
'SBK'	0	This parameter has been superseded by 'BGF', which should be used instead. "SBK' = 0 is equivalent to 'BGF' = -2, while 'SBK' = 1 is equivalent to 'BGF' = -1.

Name	Default value	Function
'SFG'	0	This parameter has been superseded by 'FGF', which should be used instead. "SFG' = 0 is equivalent to 'FGF' = -2, while 'SFG' = 1 is equivalent to 'FGF' = -1.
'TM1'	1.	Number of seconds worth of blank frames (at 'NFS' frames/second) generated before any title frames produced by a call to FTITLE. Used only by FTITLE.
'TM2'	.5	Number of seconds worth of blank frames (at 'NFS' frames/second) between sets of title frames, and after the last set of title frames, produced by a call to FTITLE. Used only by FTITLE.
'TM3'	0.	Number of additional seconds worth of blank frames (at 'NFS' frames/second) generated after the last title frame produced by a call to FTITLE. 'TM3' may be negative to suppress some of the frames specified by 'TM2'; the number of seconds of blank frames after the last title frame will be the sum of 'TM2' and 'TM3'. Used only by FTITLE.
'VPB'	0.	A real, in the range from 0. to 1., specifying the Y coordinate of the bottom edge of the STITLE viewport. The STITLE viewport is used in two ways: 1) characters are clipped at its edges; 2) it defines the rectangle which is filled by STITLE when the value of 'BGC' is non-zero. Changing the values of 'VPB', 'VPL', 'VPR', and 'VPT' changes the values of 'LX1', 'LX2', 'LY1', and 'LY2' (and vice-versa); use of the names 'VPB', 'VPL', 'VPR', and 'VPT' is now preferred.
'VPL'	0.	A real, in the range from 0. to 1., specifying the X coordinate of the left edge of the STITLE viewport. See also 'VPB', above.
'VPR'	1.	A real, in the range from 0. to 1., specifying the X coordinate of the right edge of the STITLE viewport. See also 'VPB', above.
'VPT'	1.	A real, in the range from 0. to 1., specifying the Y coordinate of the top edge of the STITLE viewport. See also 'VPB', above.
'WID'	9	Workstation identifier for WISS (Workstation-Independent Segment Storage); this is used internally in the calls to GFLASx.

4.0 ERROR MESSAGES

When an STITLE routine detects an error condition, it calls the routine SETER, which is the principal routine in the error-handling package for NCAR Graphics. (There is a programmer document describing SETER and associated routines; see that document for complete information about error-handling in NCAR Graphics.)

By default, SETER prints a line and STOPs. The line printed will look something like this:

```
ERROR    3 IN SLSETR - INVALID KEYWORD: xxx
```

The error number (“3”, in the example) may be of use to a consultant (to determine exactly where the error occurred), but is not otherwise meaningful. The actual error mes-

sage consists of the name of the routine in which the error occurred (“SLSETR”, in the example), a blank, a minus sign, another blank, and, lastly, a short description of the error.

All errors are “recoverable” in the sense that, if the user program puts SETER into “recovery mode”, control will be returned to the caller of the STITLE routine in which the error occurred. In some cases, it is then possible to take remedial action to get around whatever problem has occurred; in any case, the error flag can be cleared and execution of the user’s program can continue.

When SETER is in recovery mode (and, occasionally, even when it is not), error messages may have a somewhat more complicated form, like this:

```
SLSETI/SLSETR - INVALID KEYWORD: xxx
```

What this particular error message says is that SLSETI called SLSETR, which detected an error condition and called SETER. Upon getting control back from SLSETR, SLSETI detected the fact that SLSETR had logged an error. It augmented the error message by prepending its own name, followed by a slash, and then passed control back to the user. Of course, there can be more than two such levels of routine calls indicated in the error message: in a few cases, seven or eight routine names may be listed, each separated from the next by a slash.

The various error conditions in STITLE are described in the list below. Each bulleted item includes an error message and a thumb-nail description of the error. The items in the list are arranged in alphabetical order. If you get an error message with one or more prefixed subroutine names, as described above, omit them and look up the result in this list. Note that, since STITLE routines sometimes call other routines, elsewhere in NCAR Graphics, that can detect error conditions and call SETER, the error message you get by calling an STITLE routine may not be listed here, but in the programmer document for some other package.

- **FTITLE - PREMATURE EOF ON CARD INPUT UNIT**

  FTITLE has encountered an end-of-file on the card input unit in a place where one should not have been.

- **FTITLE - READ ERROR ON CARD INPUT UNIT**

  FTITLE has executed a FORTRAN READ statement and the “ERR=. . .” exit has been taken.

- **FTITLE - TOO MANY INPUT CARDS IN GROUP**

  The first card of a title group contains a value of NCDS greater than 120. FTITLE can’t handle that many lines.

- **FTITLE - UNCLEARED PRIOR ERROR**

  When FTITLE was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **SLBKGD - ERROR RETURN FROM GQFACI: i**

A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **SLBKGD - ERROR RETURN FROM GQFAIS: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **SLGCLR - TOO MANY COLORS DEFINED**

  More than 256 colors have been made known to STITLE. It has no room in which to maintain information about them all.

- **SLGETI - UNCLEARED PRIOR ERROR**

  When SLGETI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **SLGETR - INVALID KEYWORD: xxx**

  The first argument in the call to SLGETR is not the name of one of the known internal parameters of STITLE.

- **SLGETR - UNCLEARED PRIOR ERROR**

  When SLGETR was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **SLGWID - ERROR RETURN FROM GQACWK: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **SLGWID - NO ACTIVE WORKSTATIONS**

  STITLE uses the routine SLGWID to determine the ID of the first active workstation (to use in calls dealing with color) and it has found that there are currently no active workstations at all.

- **SLOGAP - ERROR RETURN FROM GQTXCI: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **SLOGAP - UNCLEARED PRIOR ERROR**

  When SLOGAP was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **SLRSET - UNCLEARED PRIOR ERROR**

  When SLRSET was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **SLSCLR - TOO MANY COLORS DEFINED**

More than 256 colors have been made known to STITLE. It has no room in which to maintain information about them all.

- **SLSETI - UNCLEARED PRIOR ERROR**

  When SLSETI was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **SLSETR - INVALID KEYWORD: xxx**

  The first argument in the call to SLSETR is not the name of one of the known internal parameters of STITLE.

- **SLSETR - UNCLEARED PRIOR ERROR**

  When SLSETR was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **STITLE - ERROR RETURN FROM GQCLIP: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **STITLE - ERROR RETURN FROM GQCNTN: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **STITLE - ERROR RETURN FROM GQPLCI: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **STITLE - ERROR RETURN FROM GQPMCI: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **STITLE - ERROR RETURN FROM GQTXCI: i**

  A GKS routine has returned a non-zero error flag with value “i”. This most likely means that GKS is in the wrong state.

- **STITLE - UNCLEARED PRIOR ERROR**

  When STITLE was called, there was an unrecovered outstanding error. The routine does not continue: it forces the error message for the outstanding error to be printed and then substitutes this one for it.

5.0 EXAMPLES

The code for the NCAR Graphics example named “tstitl” can be acquired and run with the command

```
ncargex tstitl
```

The resulting metafile will be called “tstitl.ncgm”. This example demonstrates minimal functioning of STITLE. The command

```
ncargex -n tstitl
```

can be used to get the code for the example (in a file called “tstitl.f”), but not to run it. After modifying the code, one can use the commands

```
ncargf77 tstitl.f
a.out
ctrans gmeta
```

to compile and execute the code and view the resulting metafile.

The example “slex01” is more complicated. The example “slex02” was added in 1995; it is still more complicated yet and illustrates the use of some features that are not mentioned in the examples. Both of these may be acquired and run in the same manner as “tstitl”. It will be useful for the user to read the code of “slex02” and try some of the modifications that are suggested there.