$INSPECT For GX

1.   Introduction

This document describes the 32-bit GX-only version of $INSPECT released with GSM SP-19. The program name of the "GX version of $INSPECT" is $32INSP although it is not necessary to remember this new program name as $INSPECT has been modified to automatically chain to $32INSP when running on GX.

2.   Overview

The following prompts displayed by 16-bit $INSPECT:

$48 INSPECT: *filename*   UNIT:*uuu*

are replaced in $32INSP by the following window:

As with $INSPECT, $32INSP will allow the inspection of both text files (type TF) (e.g. S.ABCDEF) and report files (type RS) (e.g. L.ABCDEF).

If a file prefix is not specified (e.g. the "File" is entered as ABCDEF) and that file does not exist on the unit specified then, if the Unit is **not** "$PR", $32INSP (and $INSPECT) will attempt to open a file with an S. prefix appended (e.g. S.ABCDEF). If a file prefix is not specified (e.g. the "File" is entered as ABCDEF) and that file does not exist on the unit specified then, if the Unit **is** "$PR", $32INSP (and $INSPECT) will attempt to open a file with an L. prefix appended (e.g. L.ABCDEF).

If the file specified is a Data Library (created and maintained using $DLIB) a further "Member" prompt will appear. For example:
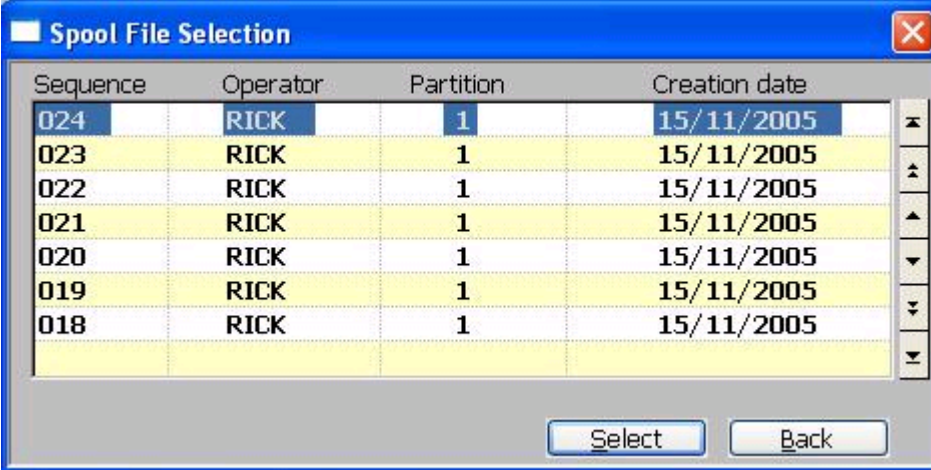
The Member must be a Source or Report file within the specified Data Library.

2.1 Inspecting Files on a Spool Unit

If the unit you specify is a spool unit, $32INSP will scan the volume to determine if more than one copy of the report file is present on the spool unit. If more than one copy is present on the spool unit, $32INSP displays a window that allows you to select the version of the file. For example:

Note that the alternative versions are displayed in reverse order of sequence number so that, in general, the most recent version of the print report will be displayed first.

As with $INSPECT, $32INSP automatically handles the situation where a large report (i.e. one that exceeds the spool unit allocation size) spans multiple "part files" on a spool unit.

2.2 The Main Inspect Window

When a File and Unit have been specified, $32INSP commences by displaying the start of the file in the main Inspect Window. For example:

```
■ File Inspection of L.BO$CON on unit C22 - Line 1 of 6220                              ⊠

SPEEDBASE II V3.16   CONSOLE HANDLING RTNS           BO$CON        17/01/06  09.18.27  PAGE    1


  1 0000      PROGRAM BO$CON   "CONSOLE HANDLING RTNS"
  2           $OPT PR
  3           LOAD-MODULE
  4           *
  5           *#BUILD        :$SDL32
  6           *#COPY         :S.XX    :$CL
  7           *#COPY         :S.X0    :$CL
  8           *#LIST         :LST     :0
  9           *#OPTION       :BL
 10           *#OPTION       :ST
 11           *#OPTION       :CB
 12           *#ENDBUILD
 13           *
 14           * COPYRIGHT 1998 TIS SOFTWARE LTD
 15           * Source creation date 27/04/1998
 16           *
 17           * This DLM contains the following:-
 18           *
 19           * MSG$
 20           *
 21           * TYP$/TYPF$/TYPU$
 22           * TYPU$ is GX only put a function number in type ahead
 23           *
 24           * RDSCR$/GTSCR$/PTSCR$

  Next    Back    Find   Find Next   Go To   Keywords                                  Exit
```

All navigation through the file is achieved using the scroll bar or the various buttons on the bottom of the window.

3.   Comparison of $INSPECT Command and $32INSP Functions

The following table describes the traditional $INSPECT commands and their equivalent functions in $32INSP:

Command	$INSPECT Description	$32INSP equivalent
B	Back to first line	Ctrl-Home key or use the up-down scroll bar on the right of the Inspect Window.
C	Close extract file	Replaced by the "Close Extract File" entry on the menu activated by a right-mouse click.
E	Exit to process next file	Replaced by the "Exit" button. This button closes the main Inspect Window and returns to the file selection window (see section 2). The current file name and unit are preserved as the default for the "next" file.
F	Find a line containing a specified string and display a page with the line at the top of the page. Stop searching at the end of the file.	Replaced by the "Find" button. Further find options are available (see below).

FA	Find All. Find a line containing a specified string and display a page with the line at the top of the page. If the string is not found before the end of file is reached, continue searching from the start of the file to the current position.	Replaced by the "Find" button. Further find options are available (see below).
FS	Find and Stay. Find a line containing a specified string and display a page with the line at the top of the page. Stop searching at the end of the file but retain the current position rather than positioning at the end of file (as the F command).	Replaced by the "Find" button. Further find options are available (see below).
G	Get a line starting with a specified string **at the start of the line** and display a page with the line in the middle of the page. Stop searching at the end of the file.	Replaced by the "Find" button. Further find options are available (see below).
GA	Get All. Get a line starting with a specified string **at the start of the line** and display a page with the line in the middle of the page. If the string is not found before the end of file is reached, continue searching from the start of the file to the current position.	Replaced by the "Find" button. Further find options are available (see below).
GS	Get and Stay. Get a line starting with a specified string **at the start of line** and display a page with the line in the middle of the page. Stop searching at the end of the file. Stop searching at the end of the file but retain the current position rather than positioning at the end of file (as the G command).	Replaced by the "Find" button. Further find options are available (see below).
L	Go to line number	Replaced by the "Go To" button, supplying an absolute number.
LE	Go to last line	Ctrl-End key or use the up-down scroll bar on the right of the Inspect Window.

M	Memorise current position	Replaced by the "Toggle Bookmark" entry on the menu activated by a right-mouse click. The line that is book-marked is the one currently under the mouse cursor when the right-mouse button is clicked. Unlike $INSPECT which only permitted up to 10 bookmarks, an unlimited number of bookmarks are available in $32INSP. A book-marked line is indicated by a blue-background.
N	Re-enter narrow mode	Not required for GX.
P	Page (i.e. move forward so that the bottom line of the current page becomes the top line of the new page)	Replaced by the "Next" button.
-P	Page backwards (i.e. move backwards so that the top line of the current page becomes the bottom line of the new page)	Replaced by the "Back" button.
R	Recall memorised position	Replaced by the "Next Bookmark" and "Previous Bookmark" entries on the menu activated by a right-mouse click.
TO	Turn (obsolete) translation mode on.	Not supported by $32INSP.
TF	Turn (obsolete) translation mode off.	Not supported by $32INSP.
W	Enter wide mode (if available)	Not required for GX.
WS	Used to print or write out selected parts of the file under inspection.	Replaced by the "Write Start" entry on the menu activated by a right-mouse click. Unlike $INSPECT, the $32INSP "start line" is the one currently under the mouse cursor when the right-mouse button is clicked.
WE	Used to print or write out selected parts of the file under inspection.	Replaced by the "Write End" entry on the menu activated by a right-mouse click. Unlike $INSPECT, the $32INSP "end line" is the one currently under the mouse cursor when the right-mouse button is clicked.
$LS	Prompts you for the starting column of the part of the line to be displayed.	Not directly supported but effectively replaced by the left-right scroll bar on the bottom of the Inspect Window.
n	(a number between 0 and 9999). Moves forward *n* lines and displays the page thus selected	Replaced by the "Go To" button by prefixing the "Line" number with a "+" character. Line navigation is also possible using the up-down scroll bar on the right of the Inspect Window.

+*n*	(a number between 0 and 9999). Moves forward *n* lines and displays the page thus selected	Replaced by the "Go To" button by prefixing the "Line" number with a "+" character. Line navigation is also possible using the up-down scroll bar on the right of the Inspect Window.
-*n*	(a number between -9999 and -1). Moves backward *n* lines and displays the page thus selected	Replaced by the "Go To" button by prefixing the "Line" number with a "-" character. Line navigation is also possible using the up-down scroll bar on the right of the Inspect Window.
?	Display current filename, unit, line number, total line number and percentage through the file.	Not required. This information, **with the exception of the percentage through the file**, is permanently displayed on the window caption.
<CR>	Following an F, G, P, -P, *n* or -*n* instruction it causes the instruction to be repeated, otherwise it is ignored	Selects the default button. Note that the <TAB> key cannot be used to move between the buttons.

## 4.	New Features in $32INSP

In addition to the vastly improved look-and-feel, $32INSP includes some new features that are not available in $INSPECT.

### 4.1	Keystrokes Supported by $32INSP Only

The following keystrokes supported by $32INSP are not available in $INSPECT:

| Key	Meaning in $32INSP
<F2>	Go to Next Bookmark (see R command)
SHIFT <F2>	Go to Previous Bookmark (see R command)
<PAGE DOWN>	Equivalent to the "Next" button
<PAGE UP>	Equivalent to the "Back" button
<DOWN ARROW>	Next line
<UP ARROW>	Previous line
<LEFT ARROW>	Scroll 1 character to the left
<RIGHT ARROW>	Scroll 1 character to the right

### 4.2	The Go To Button

The functionality of the $INSPECT L, *nnnn*, +*nnnn* and -*nnnn* commands are available by selecting the Go To Button. If the "Line" supplied is purely number (e.g. 1234) it represents an absolute line number, equivalent to the $INSPECT L command.

If the "Line" supplied is prefixed by a "+" character (e.g. +1234) it represents a relative line-number forwards in the file, equivalent to the $INSPECT *nnnn* and +*nnnn* commands.

If the "Line" supplied is prefixed by a "-" character (e.g. -1234) it represents a relative line-number backwards in the file, equivalent to the $INSPECT -*nnnn* command.

4.3　The Find and Find Next Buttons

The functionality of the $INSPECT F, FA, FS, G, GA, GS and, in some cases, <CR> commands are available by selecting the Find and Find Next Buttons. Selecting the Find button produces the following dialogue box:
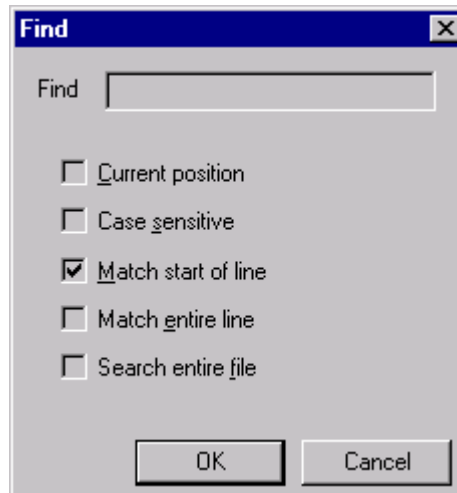
The "Current position" tick-box allows the find to start from either the current position (ticked) or the start of the file (not ticked).

The "Case sensitive" tick-box is new for $32INSP and allows case insensitive searches (all searches in $INSPECT are case sensitive). Note that wildcards or regular expressions are not recognised in either $32INSP or $INSPECT search strings.

The "Match start of line" tick-box allows a $INSPECT Find (not ticked) or $INSPECT Get (ticked) to be executed. **Important Note: Unlike the $INSPECT Get, which positions the target line halfway on the screen, the $32INSP Find/Get always positions the target line at the top of the screen.**

The "Match entire line" tick-box, which is mutually exclusive with "Match start of line" option is new for $32INSP. If this option is ticked the search string must match an entire line (i.e. not just a sub-string of a line).

The "Search entire file" tick-box allows the search to wrap around at the end of the file. The following combinations of the "Current position" and "Search entire file" tick-boxes are possible:

| Current position	Search entire file	Search algorithm
Unticked	Unticked	Search from start of file to end
Unticked	Ticked	Search from start of file to end
Ticked	Unticked	Search from current position to end of file
Ticked	Ticked	Search from current position to end of file;

		continue from start of file to current position

In all cases, if a string is not found a warning box appears but the current line number remains unchanged. That is, **all** $32INSP text searches are equivalent to the $INSPECT FS and GS commands. The less than useful option of positioning at the End of File when the search fails ($INSPECT F and G commands) is not supported by $32INSP.

The "Find Next" button, which is only active after a Find has been executed, simply repeats the previous Find using the same string and tick-box options (i.e. the "Find Next" button is equivalent of the $INSPECT <CR> command immediately after the F command.

4.4   The Keywords Button
The Search for Keyword function is new for $32INSP. Selecting the "Keyword" button produces the following window:
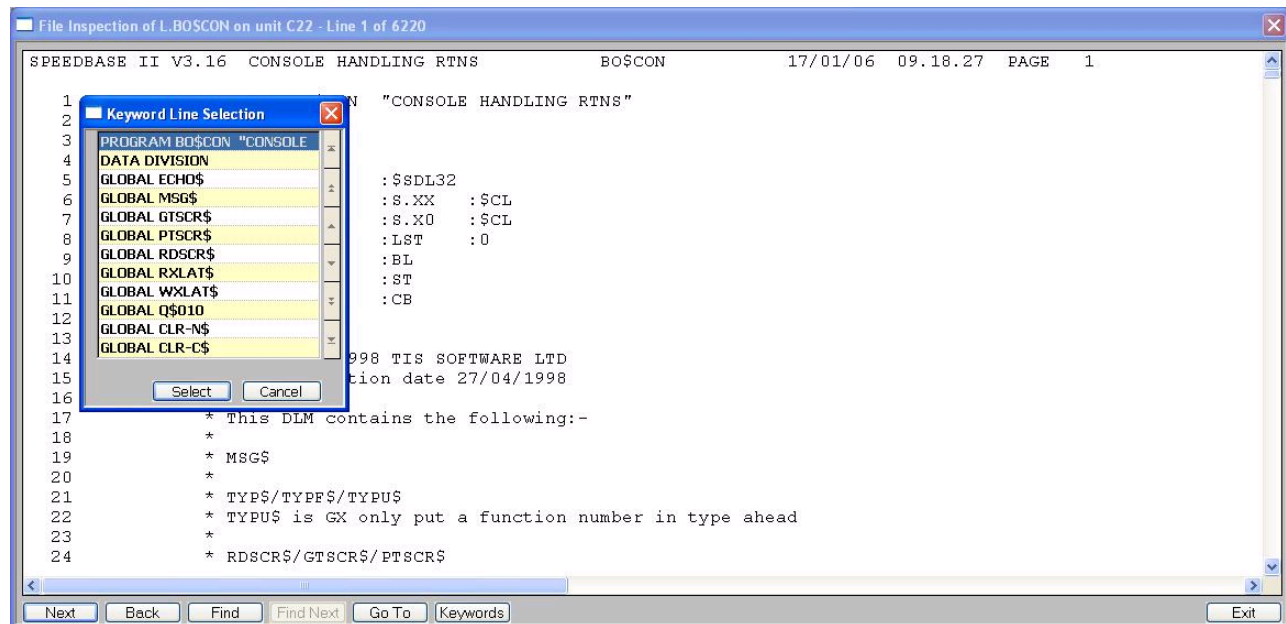
The list contains all the "keywords" encountered in the file. Simply selecting a given keyword takes you to the line position in the file that contains that keyword (positioned at the top of the screen).

At the time of writing (GSM SP-20) the following keywords are recognised:

    PROGRAM
    FRAME
    DATA
    LOCAL
    GLOBAL
    COMMON
    EXTERNAL

```
LINKAGE
WINDOW
PF
SECTION
ENTRY
PROCEDURE
UNLOAD DIVISION                         (not supported in GSM SP-19, or earlier)
LOAD DIVISION                           (not supported in GSM SP-19, or earlier)
.BOOK                                   (not supported in GSM SP-19, or earlier)
```

For text files the keyword search commences from the 1st character position. Leading white-space characters (i.e. SPACE and TAB) are ignored.

For listing files the keyword search commences from the 17th character position which is the character position of the start of the source line text in listing files generated by $SDL32. When searching from this character position, leading white-space characters (i.e. SPACE and TAB) are ignored.

4.5   Customisation
Clicking on the icon on the left of caption bar produces a small menu that, amongst other options, allows the Text Font and Button Font used in the Inspect Window to be customised. Note that only fixed-pitch the fonts are available in the Text Font selection dialogue. These fonts are remembered for the next time an Inspect Window is displayed.

5.   WS - Set Write Start and End Positions
The "Write Start" and "Write End" menu entries are used to print or write to an output file from the selected parts of the files under inspection. Each time that a "Write Start" instruction is keyed you are prompted for the **file-id** and **unit-id** of the output file, unless an output file is already open. The default extract file name is D.$INSPE on unit $PR.

The new file will be produced as a text file if the file-id you supply begins with an F. or S. prefix and is on a direct access unit which is not a spool unit. In all other cases a relative sequential print file will be produced. You can use this convention to convert from text to print file format, or vice versa.

The "Write Start" instruction sets the **write start pointer** to the line that is addressed by the mouse cursor when the right mouse button is clicked.

The "Write End" instruction causes lines to be written to the output file from the line addressed by the write start pointer up to and including the line that is addressed by the cursor when the right-mouse button is clicked. If the line pointed to by the write start pointer is later in the file than the current line selected by the mouse cursor, an error message box is displayed.

IMPORTANT NOTE: The "Write Start" line is selected when the right mouse button is clicked in order to select the "Write start" menu option. Unlike $INSPECT, $32INSP does

not assume that the top line of the screen is the "Write Start" line. The "Write End" line is selected when the right mouse button is clicked in order to select the "Write end" menu option. Unlike $INSPECT, $32INSP does not assume that the top line of the screen is the "Write Start" line.

Once the lines have been written to the output file, the following message box is displayed:
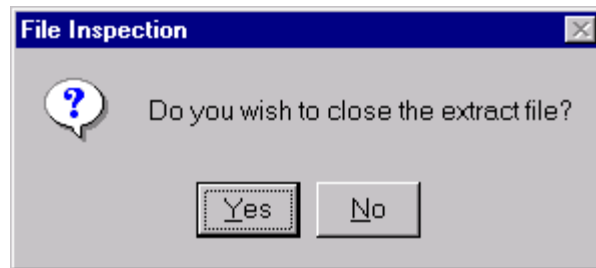
This provides the option of correctly closing the output file, or leaving the file open so you can continue extending it.

The "Close extract file" option can be used to close the extract file immediately without any further writes taking place.

If you attempt to end the file inspection, while an output file is open (i.e. if the "Write Start" option has been used without a corresponding "Write End" option) $32INSP will display the following warning message box:
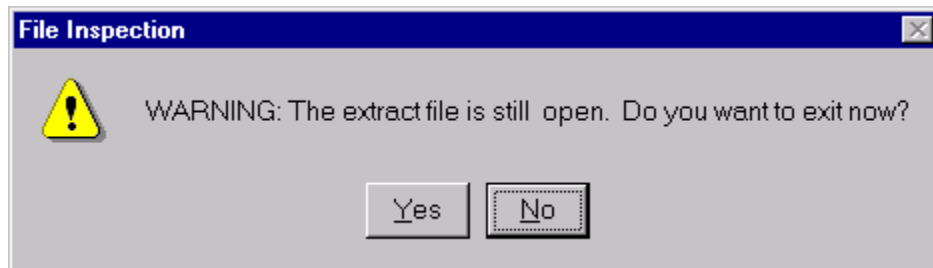
Select "Yes" to confirm the inspection of the current file is to end. Select "No" if you have made a mistake and wish to close the output file, using the "Write End" option, before ending the inspection.

6.   Known Restrictions
There is currently no facility to mark and copy text from the Inspect Window (e.g. the combination of $INSPECT and GX Mark/Copy or <SYSREQ> X are not available).

The ButtonStyleScrollBar option is not recognised for the Inspect Window.

The text of the "last find" is not preserved when a new file is selected.

7.   Executing $32INSP from a 32-bit Program

For GSM SP-23, and later, $32INSP can be EXEC'ed from a 32-bit application using the following calling convention:

        EXEC "$32INSP " USING *in*

where *in* is a control block of the following format:

```
        01    IN
          02   INVERS     PIC 9(4) COMP     * Block Version = 1 or 2 or 3
          02   INFID      PIC X(8)          * File name to be inspected
          02   INUID      PIC X(3)          * Unit of file to be inspected
          02   INDLIB     PIC X(8)          * Data library (SPACES if none)
          02   INERR      PIC 9(9) COMP     * Returned error
                                           * 1 = Not GX
                                           * 2 = Invalid block version
                                           * 3 = Cannot open file
*
*     The following fields are only recognized if the Block Version is 2,
*     or higher.
*
          02   INTITP     PIC PTR           * Pointer to #00 terminated
                                           * Inspect window title;
                                           * or #FFFFFFFF or #00000000
                                           * if no window title
*
*     The following fields are only recognized if the Block Version is 3,
*     or higher.
*
          02   INPBTN     PIC 9 COMP        * 1 = Print button required
                                           * 0 = No print button
          02   INPBTR     PIC 9 COMP        * 1 = Print button pressed
                                           * 0 = Normal exit
```

If a parameter block has been passed to $32INSP then the initial file window will not appear.   If the file cannot be opened an error will be returned in INERROR.