Appendix A - DAZZLER Test Stream

```
COM **************************************************************
COM *       THIS DAZZLER STREAM IS DESIGNED TO PROVIDE EXAMPLES  *
COM *            OF THE USE OF THE VARIOUS STATEMENTS (BOTH      *
COM *            OPTIONAL AND MANDATORY) THAT CAN BE USED.       *
COM **************************************************************
COM *
COM * All these comment statements (indicated as so by the ’COM’ in
COM * the first three positions) are ignored by the program,
COM * as are any blank statements like the ones following:


COMment statements do not have to have a blank/asterisk (’ *’) in
COM positions 4 and 5 and are not printed. In addition, as no
COM statement contains information after position 66, it is possible
COM to code printable comments in positions 67 - 80 inclusive.

COM * In these examples, comments PRECEDE the actual coding, and
COM * blank statements have been inserted for ease of reading.
COM *
COM **************************************************************


COM **************************************************************
COM *
COM *    1)   The first decision to be made concerns the
COM *         printer. Some consideration should be given
COM *         to the volume and required content. Many options
COM *         are available. The default values assumed if
COM *         this statement is not present are that dates take the
COM *         format dd/mm/yy and that all lines should be
COM *         printed with an automatic page-throw before each
COM *         set of statements. The following example suppresses the
COM *         automatic page-throw, prints dates as mm/dd/yy
COM *         and only prints the 10675th and 10676th calls in
COM *         detail.
COM *
COM *         The program always prints every statement read, with
COM *         the exception of comment or blank statements, plus
COM *         a last page showing the total number of calls made.
COM *
COM **************************************************************

PARUS>PF1067501

COM **************************************************************
COM * As a reminder, all the preceding statements in this example
COM * are totally optional and, if the default settings for the
COM * PAR statement above were required, then nothing need
COM * have been encoded.
COM **************************************************************

COM **************************************************************
COM *
COM *    2)   The following example shows the use of the
COM *         iterative function on the header statement and the use
COM *         of both types of overwrite statement.
COM *         Reading from the left, the header statement specifies
COM *         the following:
COM *
COM *         Do an ’ISRT’ call 9 times in succession using PCB1
COM *         with 1 SSA, iteratively increasing the number from
COM *         its start value of 100 by 100 and overwriting the
COM *         I/O Area in position 14 for a length of 5 and in
COM *         position 95 for a length of 4.
COM *
COM *         The next statement causes the first SSA to have the
COM *         segment name in positions 1 to 8 inclusive (all
COM *         segment names are 8 characters long and left-justified)
COM *         followed by 42 spaces. The default format (character
COM *         format) is used as no entry was made in positions
COM *         5 to 8 inclusive. For the same reason, this default
COM *         also applies to the remaining IOA statements.
COM *
COM *         These last four statements cause the I/O Area to be set
COM *         up for a total of 170 characters. The NNNNN entries
COM *         will be iteratively overwritten by the values from
COM *         the *HD statement.
COM *
COM *         Note that the order of the five statements following
COM *         the *HD statement is immaterial: the same results are
COM *         achieved no matter what order they are specified in.
COM *
COM **************************************************************

*HD ISRT 009    01 01 00100 00100 0145    0954
S01      001 50 ITEMID
IOA      001 50 ITEMID/MITEM/NNNNN*...............................
IOA      051 50 ..............................ITEMID/MMACD/*NNNN**
IOA      101 50 ...........................................END OF
IOA      151 20 SEGMENT ITEMID --->:

COM **************************************************************
COM *
COM *    3)   The following example shows a single call
COM *         (’ISRT’) using two SSAs.
COM *         The *HD statement specifies an insert using PCB1 and 2
COM *         SSAs.
COM *         The two SSAs and the IOA are created as explained
COM *         above in the second step.
COM *
COM **************************************************************

*HD ISRT        01 02
S01      001 50 ITEMID  (MITEM   = ITEMID/MITEM/00100*)
S02      001 50 LAGER
IOA      001 50 THIS IS A LAGER SEGMENT THAT CONTAINS NO SEQUENCE
IOA      051 50 FIELD WHATSOEVER - HOWEVER FOR INTERNAL CONTROL PU
IOA      101 50 RPOSES -0101- END OF SEGMENT LAGER --->:



COM **************************************************************
COM *
COM *    4)   The following example shows a single call
COM *         (’GHU’) using two SSAs.
COM *         The *HD statement specifies a ’GHU’ using PCB1 and 2
COM *         SSAs.
COM *         The two SSAs are created as explained
COM *         above in the second step.
COM *
COM **************************************************************

*HD GHU         01 02
S01      001 50 ITEMID  (MITEM   = ITEMID/MITEM/00700*)
S02      001 50 PRODST  (SPKEY   = ITEMID-00700/SPKEY-701*****)

COM **************************************************************
COM *    5)   POSITION AT BEGINNING AND PRINT THE LOT
COM *
COM *         This is achieved by issuing an unqualified get
COM *         unique which automatically retrieves the first
COM *         segment in the data base and follows this with
COM *         successive get nexts until the end of the data
COM *         base is reached. This condition is indicated
COM *         by a status code of ’GB’.
COM *         The two *HD statements are translated as follows :
COM *         Issue a ’GU’ using PCB1 and then issue a ’GN’,
COM *         also using PCB1. Keep issuing a ’GN’ call
COM *         until a status code of ’GB’ is returned in the
COM *         PCB.
COM *
COM **************************************************************

*HD GU          01
*HD GN       GB 01

COM **************************************************************
COM *   LAST) PRINT THE NUMBER OF EACH TYPE OF SEGMENT
COM *         ACCESSIBLE BY PCB3
COM *
COM *         Internally, this statement will cause an unqualified
COM *         GU on PCB3 to be issued, thus positioning at
COM *         the beginning. Unqualified GNs are then issued
COM *         until a status code of ’GB’ is returned in the
COM *         PCB. This is apparently the same as the previous
COM *         two *HD statements in combination (but on a different
COM *         PCB), but the SUMMARY statement does not print out
COM *         each segment as it reads it. Instead, it only
COM *         prints the summary table at the end.
COM **************************************************************

SUMMARY 03

COM **************************************************************
COM *
COM *    As all blank statements and comment lines are ignored by
COM *    DAZZLER, the identical results from the stream above
COM *    would be achieved by the stream which now follows.
COM *
COM **************************************************************


PARUS>PF1067501
*HD ISRT 009    01 01 00100 00100 0145    0954
S01      001 50 ITEMID
IOA      001 50 ITEMID/MITEM/NNNNN*...............................
IOA      051 50 ..............................ITEMID/MMACD/*NNNN**
IOA      101 50 ...........................................END OF
IOA      151 20 SEGMENT ITEMID --->:
*HD ISRT        01 02
S01      001 50 ITEMID  (MITEM   = ITEMID/MITEM/00100*)
S02      001 50 LAGER
IOA      001 50 THIS IS A LAGER SEGMENT THAT CONTAINS NO SEQUENCE
IOA      051 50 FIELD WHATSOEVER - HOWEVER FOR INTERNAL CONTROL PU
IOA      101 50 RPOSES -0101- END OF SEGMENT LAGER --->:
*HD GHU         01 02
S01      001 50 ITEMID  (MITEM   = ITEMID/MITEM/00700*)
S02      001 50 PRODST  (SPKEY   = ITEMID-00700/SPKEY-701*****)
*HD GU          01
*HD GN       GB 01
SUMMARY 03
```