ALTERNATIVE TECHNOLOGIES
HIGH PERFORMANCE COBOL RUNTIME
IDM HOST INTERFACE
ROUTINES

USERS GUIDE

I. INTRODUCTION

The routines supplied in this Beta release provide a high performance supplement to the standard Britton-Lee Release 3.x host interface. This package is intended to run with VAX/VMS version 4.x using COBOL. While this is a Beta Release of this version of the package, the intent is to provide full compatibility with the standard BLI software.

Performance of user written host software is expected to be improved in several ways. First, a routine (AT-OPEN) is provided which allows the programmer to define a command (including commands which can not be included in stored commands such as "retrieve into" or "create") at program initialization. This eliminates the need to parse commands during normal processing when performance is critical. Second, the same routine allows the programmer to define a data buffer format for reading data from or writing data to the IDM. This eliminates the need to perform runtime binds on a per field basis. Third, two I/O routines are provided which allow the user to read (write) tuples (records) from (to) the IDM. These "cache" I/O routines should be more efficient than the usual runtime routines because (a) they allow the programmer to read (write) directly to (from) working storage, (b) they use low-level BLI I/O routines where possible, and (c) they read (write) multiple tuples (records) as the result of a single call. Fourth, the "cache" I/O routines may be freely interspersed without need for the programmer to be concerned with "clean-up" of the interface buffers.

The rules for using these routines are outlined below. It should be understood that deviations from the proper calling sequence can result in grave errors. As with all optimized software, great care should be taken to use the routines properly.

II. OVERVIEW OF ROUTINES

The following routines are provided:

AT-OPEN - define an IDM command and a data buffer format for subsequent use by AT-READ, AT-WRITE, or AT-NOPARSE.

AT-READ - retrieve data from the IDM and place in a data buffer. The retrieve command is defined by AT-OPEN and data is placed in the data buffer under control of a data buffer format defined by AT-OPEN.

AT-WRITE - write data to the IDM and place in a data buffer. The relation written to is defined by AT-OPEN and data is read from the data buffer under control of a data buffer format defined by AT-OPEN. This is essentially used for writing data to a single relation in a manner similar to FCOPY IN. It is NOT an append command.

AT-NOPARSE - establish a command pre-defined by AT-OPEN as the current command for use by the standard BLI runtime calls. This is used to eliminate a parse by pre-parsing. Command structures which have either parameters (stored commands) or substitution variables can be defined by AT-OPEN and then set as the current command using AT-NOPARSE. The command can be executed using IDM-IREXEC after the appropriate calls to IDM-IRSUBST or IDM-IRXSETP.

AT-CLOSE - close a command and data buffer definition that has been established by AT-OPEN. This routine MUST be called once for each call to AT-OPEN. It is useful to think of AT-OPEN and AT-CLOSE as the opening and closing of formatted files. It is a grave error to exit a program without closing all previously openeed IDM "files".

III. PROGRAMMERS REFERENCE

NAME: AT-OPEN

DESCRIPTION: Used to define an IDM command and a data buffer format for subsequent use by AT-READ, AT-WRITE, or AT-NOPARSE. It is a grave error to call AT-OPEN prior to calls to IDM-INITLIB and IDM-IROPEN. It is assumed that the appropriate database has been opened and that the appropriate range variables have been declared.

CALL FORMAT:

```
CALL "AT-OPEN" USING IDMRUN
                     BY REFERENCE QUERY
                     BY REFERENCE QTYPE
                     BY REFERENCE DBNAME
                     BY REFERENCE RELNAME
```

```
                BY REFERENCE RECDEF
                BY REFERENCE RECLEN
                BY REFERENCE IDMLIB-RETURN-CODE
        GIVING IDM-FILE.
```

PARAMETERS:

IDMRUN - PIC 9(9) COMP.

The IDMRUN context (like a channel)

BY REFERENCE QUERY - PIC X(n)

The text of a command of length n-1, it must be null terminated.

BY REFERENCE QTYPE - PIC 9 COMP.

The Alternative Technologies query type. See the description of HIPERF.LIB below for symbolic values.

BY REFERENCE DBNAME - PIC X(n)

A text database name of length n-1, it must be null terminated.

BY REFERENCE RELNAME - PIC X(n)

A text relation name of length n-1, it must be null terminated. This is the name of the relation being written to in the case of an AT query type of atCOPYIN.

BY REFERENCE RECDEF

An array of PIC 9(9) COMP. values defining the data buffer format and arranged in order of the target number specifying the target number, the data buffer data type, the number of bytes allocated in the buffer, and the offset position into the data buffer for the start of the field. It is assumed that every record will be identical in format. Space may be skipped in the buffer between fields. Fields need not be allocated in the buffer in target number order, but they MUST be defined in target number order. For example,

```
target number = 1
data type = iSTRING
number of bytes (field length)
      = 20
offset = 30
target number = 3
data type = iSTRING
```

number of bytes (field length)
= 20
offset = 1

is a legitimate data buffer format but

target number = 2
data type = iSTRING
number of bytes (field length)
= 20
offset = 1

target number = 1
data type = iSTRING
number of bytes (field length)
= 20
offset = 21

is NOT legitimate, since the target numbers are not in order.

BY REFERENCE RECLEN - PIC 9(9) COMP.
The length of each record in the data buffer. This allows AT-READ and AT-WRITE to skip FILLER in records.

BY REFERENCE IDMLIB-RETURN-CODE
The standard IDMLIB-RETURN-CODE. In most cases either RS-NORM or RE-FAILURE is returned. Only standard return codes are used in any case.

IDM-FILE - PIC 9(9) COMP.
A reference to the command and data buffer format defined by AT-OPEN. This value is returned. The user application MUST NOT alter this field.

NAME: AT-READ

DESCRIPTION: Used to retrieve data from the IDM and place it in a data buffer. The retrieve command is defined by AT-OPEN and data is placed in the data buffer under control of a data buffer format defined by AT-OPEN. AT-READ should not be called prior to an appropriate call to AT-OPEN. It is assumed that the appropriate range variables have been declared prior to a call to AT-READ (this can be done by parsing and executing the range declaration).

CALL FORMAT:

```
CALL "AT-READ" USING IDMRUN
                BY REFERENCE IDM-FILE
                BY REFERENCE BUFFER
                BY REFERENCE NUMRECS
        GIVING IDMLIB-RETURN-CODE.
```

PARAMETERS:

IDMRUN - PIC 9(9) COMP.
The IDMRUN context.

IDM-FILE - PIC 9(9) COMP.
The command and data buffer format returned by a call to AT-OPEN.

BUFFER - PIC X(n).
A data buffer. The number of bytes allocated must be large enough to hold as many records as will be read and in any case not less than the product of the record length (RECLEN defined in AT-OPEN) and the number of records to be read during this call to AT-READ (NUMRECS).
The buffer may be defined with subfields to aid in moving the data to/from the buffer, etc.
It must be remembered that the IDM null terminates character strings and space must be provided for the null (one byte per string field). Care should be taken to define sub-field data types and lengths to reflect the data buffer format used in the corresponding AT-OPEN.

NUMRECS - PIC 9(9) COMP.
The maximum number of records to be read by AT-READ on this call. AT-READ returns the actual number of records read into BUFFER.

NAME: AT-WRITE

DESCRIPTION: Used to write data to the IDM and place in a data buffer. The relation written to is defined by AT-OPEN and data is read from the data buffer under control of a data buffer format defined by AT-OPEN. This is essentially used for writing data to a single relation in a manner similar to FCOPY IN. It is NOT an append command. AT-WRITE sould not be called prior to a an appropriate call to AT-OPEN using a query type of atCOPYIN.

CALL FORMAT:

```
        CALL "AT-WRITE" USING IDMRUN
                        BY REFERENCE IDM-FILE
                        BY REFERENCE BUFFER
                        BY REFERENCE NUMRECS
                GIVING IDMLIB-RETURN-CODE.
```

PARAMETERS:

IDMRUN - PIC 9(9) COMP.
The IDMRUN context.

IDM-FILE - PIC 9(9) COMP.
The command and data buffer format returned by a call to AT-OPEN.

BUFFER - PIC X(n).
A data buffer. The number of bytes allocated must be large enough to hold as many records as will be written and in any case not less than the product of the record length (RECLEN defined in AT-OPEN) and the number of records to be written during this call to AT-WRITE (NUMRECS).
The buffer may be defined with subfields to aid in moving the data to/from the buffer, etc.
It must be remembered that the IDM expects null terminated character strings and space must be provided for the null (one byte per string field) and the null must be inserted.
Care should be taken to define sub-field data types and lengths to reflect the data buffer format used in the corresponding AT-OPEN.

NUMRECS - PIC 9(9) COMP.
The maximum number of records to be written by AT-WRITE on this call.
AT-WRITE returns the actual number of records written.

IDMLIB-RETURN-CODE - PIC 9(9) COMP.
The standard IDMLIB return status.

NAME: AT-NOPARSE

DESCRIPTION: Used to establish a command pre-defined by

AT-OPEN as the current command for use by the standard BLI runtime calls. This is used to eliminate a parse by pre-parsing. Command structures which have either parameters (stored commands) or substitution variables can be defined by AT-OPEN and then set as the current command using AT-NOPARSE. The command can be executed using IDM-IREXEC after the appropriate calls to IDM-IRSUBST or IDM-IRXSETP. AT-NOPARSE must not be called prior to AT-OPEN. It is assumed that the appropriate range variables have been declared on the IDMRUN context. It is an error to call AT-NOPARSE with an IDM-FILE of command type atCOPYIN.

CALL FORMAT:

```
CALL "AT-NOPARSE" USING IDMRUN
                  BY-REFERENCE IDM-FILE
                  GIVING IDMLIB-RETURN-CODE.
```

PARAMETERS:

IDMRUN - PIC 9(9) COMP.
The IDMRUN context.

IDM-FILE - PIC 9(9) COMP.
The command previously defined in a call to AT-OPEN.

IDMLIB-RETURN-CODE - PIC 9(9) COMP.
The return status. Normally RS-NORM or RE-FAILURE.

NAME: AT-CLOSE

DESCRIPTION: Used to close a command and data buffer definition that has been established by AT-OPEN. This routine MUST be called once for each call to AT-OPEN. It is useful to think of AT-OPEN and AT-CLOSE as the opening and closing of formatted files. It is a grave error to exit a program without closing all previously openeed IDM "files". AT-CLOSE must not be called prior to AT-OPEN.

CALL FORMAT:

```
CALL "AT-CLOSE" USING IDM-FILE
```

PARAMETERS:

IDM-FILE - PIC 9(9) COMP.
A reference to the command and data buffer definition established by a previous call to AT-OPEN. This reference MUST NOT be used in subsequent calls.

IV. HIPERF.LIB

This is the copy file which contains definitions for each symbol used by the AT HIGH PERFORMANCE ROUTINES. The symbols are as follows:

```
*
*    HIPERF -- Include file for AT COBOL HIPERFORMANCE ROUTINES
*         This file defines constants for these purposes:
*           - substitution types for AT-SET-VALS (not released).
*           - buffer types for ATREAD and ATWRITE.
*           - query types for ATOPEN, ATREAD, ATNOPARSE, and ATWRITE.
*
*       Copyright 1985, All Rights Reserved, Alternative Technologies
*       150 Felker Street, Suite E, Santa Cruz, California 95060
*

01 AT-SUBSTITUTION-TYPES.
        03 iSUB                 PIC 9(9) COMP   VALUE IS 0.
        03 iSET                 PIC 9(9) COMP   VALUE IS 1.

01 AT-BUFFER-TYPES.
*
*       used as a value for NUMRECS in AT-READ or AT-WRITE
*
        03 ONEREC               PIC 9(9) COMP   VALUE IS 1.
*
*       used as a value for NUMRECS in AT-READ or AT-WRITE
*
        03 MULTREC              PIC 9(9) COMP   VALUE IS 9999.

01 AT-QUERY-TYPES.
*
*       symbols for command types to be used in AT-OPEN calls.
*
*       retrieve command
*
        03 atRETRIEVE           PIC 9(9) COMP   VALUE IS 1.
*
*       retrieve into command
*
        03 atRETINTO            PIC 9(9) COMP   VALUE IS 2.
*
*       create command: database, relation, or index
*
        03 atCREATE             PIC 9(9) COMP   VALUE IS 3.
*
*       audit command
*
        03 atAUDIT              PIC 9(9) COMP   VALUE IS 4.
*
*       begin or end transaction
```

```
*
        03 atBEGINEND           PIC 9(9) COMP   VALUE IS 5.
*
*       miscellaneous commands for which no explicit
*       command type symbol exists such as permit, destroy,
*       or delete
*
        03 atOTHER              PIC 9(9) COMP   VALUE IS 6.
*
*       an append command
*
        03 atAPPEND             PIC 9(9) COMP   VALUE IS 7.
*
*       for a subsequent call to AT-WRITE
*
        03 atCOPYIN             PIC 9(9) COMP   VALUE IS 8.
*
*       not implemented:  use a retrieve command
*
        03 atCOPYOUT            PIC 9(9) COMP   VALUE IS 9.
*
*       stored commands
*
        03 atCMD                PIC 9(9) COMP   VALUE IS 10.
```

V.   COMMENTED COBOL EXAMPLE

```
IDENTIFICATION DIVISION.
PROGRAM-ID.             ATIDL.
AUTHOR.                 DAVID MCGOVERAN.
INSTALLATION.           BRITTON-LEE.
                        Copyright (c) 1985, Alternative Technologies
                        150 Felker Street, Suite E, Santa Cruz, Ca. 95060
DATE-WRITTEN.           $Header: atidl.cob,v 0.9 85/12/15 18:10:29 david E


ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. VAX-11.
OBJECT-COMPUTER. VAX-11.
*
*       set up a special symbolic character for null termination
*       of strings
*
SPECIAL-NAMES.
        SYMBOLIC CHARACTERS CHAR-NULL ARE 1.


DATA DIVISION.
```

```
WORKING-STORAGE SECTION.

COPY "IDMLIB".
*
*       include the definitions of special AT symbols
*
COPY "HIPERF".

*
*       define two "files"
*
01   IDM-FILE-1          PIC 9(9) COMP.
01   IDM-FILE-2          PIC 9(9) COMP.
01   IDMRUN              PIC 9(9) COMP.
*
*       set up the database name to be used
*
01   DBNAME.
*
*       initialize
*
        02 DATABASE      PIC X(5) VALUE IS "vino".
*
*       room to null terminate the string
*
        02 DBEND         PIC X.
*
*       set up the relation name to be used
*
01   RELNAME.
*
*       initialize
*
        02 RELATION      PIC X(5) VALUE IS "kinds".
*
*       room to null terminate the string
*
        02 REND          PIC X.
*
*       set up a command to be defined
*
01   QUERY1.
*
*       initialize
*
        02 QUERYTXT1             PIC X(20) VALUE IS "retrieve (k.all)".
*
*       room to null terminate the string
*
        02 QEND1                 PIC X.
*
*       set up a second command to be defined
*
01   QUERY2.
*
```

```
*       initialize
*
        02 QUERYTXT2    PIC X(60)
                VALUE IS "retrieve (k.all) where k.color= ""red"" ".
*
*       room to null terminate the string
*
        02 QEND2        PIC X.
*
*       define the data buffer format
*
01  RECDEF.
*
*       occurs five times - once per field plus one
*               termination field
*
        03 RECS OCCURS 5 TIMES INDEXED BY ITARGET.
*
*               target number of this field in sequential order
*
                05 TARG         PIC IS 9(9) COMP.
*
*               the IDM data type
*
                05 ITYPE        PIC IS 9(9) COMP.
*
*               maximum length of the data allowed in the buffer
*                       including room for null termination of
*                       strings
*
                05 ILENGTH      PIC IS 9(9) COMP.
*
*               offset into the data buffer for start of this
*                       field
*
                05 IOFFSET      PIC IS 9(9) COMP.
*
*       Alternative Technologies query type for the command
*               being defined
*
01  QTYPE       PIC 9 COMP.
*
*       Data buffer
*
01  BUFFER.
*
*       Allow room for twenty tuples to be read in this example
*
        03 BUFDEF OCCURS 20 TIMES INDEXED BY RECNO.
*
*               target 1 (kind) is 25 characters max
*
                05 KIND         PIC X(25).
*
*               kind is null terminated
```

```
*
                05 FILLER       PIC X.
*
*               target 2 (color) is 5 characters max
*
                05 COLOR        PIC X(5).
*
*               color is null terminated
*
                05 FILLER       PIC X.
*
*               target 3 (flavor) is 10 characters max
*
                05 FLAVOR       PIC X(10).
*
*               flavor is null terminated
*
                05 FILLER       PIC X.
*
*               target 4 (body) is 10 characters max
*
                05 BODY         PIC X(10).
*
*               body is null terminated
*
                05 FILLER       PIC X.
*
*       initialize the length of the record in the data buffer
*
01 RECLEN       PIC 9(9) COMP VALUE IS 54.
*
*       initialize the number of records to be read into the
*               data buffer
*
01 NUMRECS      PIC 9(9) COMP VALUE IS 20.

*       set up a single record image for display purposes
*
01  OUTPUT-RECORD.
                05 KIND         PIC X(25).
                05 FILLER       PIC X VALUE IS "|".
                05 COLOR        PIC X(5).
                05 FILLER       PIC X VALUE IS "|".
                05 FLAVOR       PIC X(10).
                05 FILLER       PIC X VALUE IS "|".
                05 BODY         PIC X(10).
                05 FILLER       PIC X VALUE IS "|".
                05 FILLER       PIC X(26).

*
*       a dummy input record
*
01  INPUT-RECORD        PIC X(80).
    88  INPUT-EMPTY                     VALUE SPACES.
    88  NO-MORE-INPUT                   VALUE "exit".
```

```
01  DONE-FLAG               PIC X.
    88  DONE                                VALUE "T".


PROCEDURE DIVISION.

MAIN.

    PERFORM 1000-INIT.
    PERFORM 2000-READ-INPUT-RECORD.
    PERFORM 3000-GET-OUTPUT.
    PERFORM 4000-GET-OUTPUT.
    PERFORM 5000-CLOSE.
    STOP RUN.

1000-INIT.

*
*       initialize runtime interface
*
    CALL "IDM-INITIDMLIB".
*
*       open the channel to the IDM
*
    CALL "IDM-IROPEN" GIVING IDMRUN.
*
*       parse the open database command
*
        CALL "IDM-IRIDL" USING IDMRUN DESCRIPTOR "open vino" GIVING
                                        IDMLIB-RETURN-CODE.
*
*       execute the open database
*
        CALL "IDM-IREXEC" USING IDMRUN.
*
*       parse the range statement
*
        CALL "IDM-IRIDL" USING IDMRUN DESCRIPTOR "range of k is kinds" GIVI
                                                IDMLIB-RETURN-CODE.
*
*       execute the range statement
*
        CALL "IDM-IREXEC" USING IDMRUN.
*
*       set the query type
*
        COMPUTE QTYPE = atRETRIEVE.
*
*       initialize the record definition
*
        PERFORM 1001-INIT-RECDEF.
*
*       call AT-OPEN on the first command definition (queryl)
```

```
*
        CALL "AT-OPEN" USING IDMRUN
                                BY REFERENCE QUERY1
                                BY REFERENCE QTYPE
                                BY REFERENCE DBNAME
                                BY REFERENCE RELNAME
                                BY REFERENCE RECDEF
                                BY REFERENCE RECLEN
                                BY REFERENCE IDMLIB-RETURN-CODE
                        GIVING IDM-FILE-1.
        IF RS-NORM DISPLAY "SUCCESSFUL ATOPEN-1.".
*
*       call AT-OPEN on the second command definition (query2)
*       Note that if the QTYPE were atCOPYIN, that we could
*       write the data back out to another relation (see
*       the second call to AT-READ below.
*
        CALL "AT-OPEN" USING IDMRUN
                                BY REFERENCE QUERY2
                                BY REFERENCE QTYPE
                                BY REFERENCE DBNAME
                                BY REFERENCE RELNAME
                                BY REFERENCE RECDEF
                                BY REFERENCE RECLEN
                                BY REFERENCE IDMLIB-RETURN-CODE
                        GIVING IDM-FILE-2.
        IF RS-NORM DISPLAY "SUCCESSFUL ATOPEN-2.".
*
*       this completes the initialization processing
*


1001-INIT-RECDEF.

*
*       set up the data buffer definition - in this example one
*               record definition serves for both queries but
*               this is only to simplify the example
*
        SET ITARGET TO 1.
        MOVE 1 TO TARG (ITARGET).
        MOVE iSTRING TO ITYPE (ITARGET).
        MOVE 26 TO ILENGTH (ITARGET).
        MOVE 0 TO IOFFSET (ITARGET).
        SET ITARGET TO 2.
        MOVE 2 TO TARG (ITARGET).
        MOVE iSTRING TO ITYPE (ITARGET).
        MOVE 6 TO ILENGTH (ITARGET).
        MOVE 26 TO IOFFSET (ITARGET).
        SET ITARGET TO 3.
        MOVE 3 TO TARG (ITARGET).
        MOVE iSTRING TO ITYPE (ITARGET).
        MOVE 11 TO ILENGTH (ITARGET).
        MOVE 32 TO IOFFSET (ITARGET).
        SET ITARGET TO 4.
        MOVE 4 TO TARG (ITARGET).
        MOVE iSTRING TO ITYPE (ITARGET).
        MOVE 11 TO ILENGTH (ITARGET).
        MOVE 43 TO IOFFSET (ITARGET).
        SET ITARGET TO 5.
*
*       AT-OPEN looks for a target number of 0 to know when all
*       fields have been defined
*
        MOVE 0 TO TARG (ITARGET).
        MOVE 0 TO ITYPE (ITARGET).
        MOVE 0 TO ILENGTH (ITARGET).
        MOVE 0 TO IOFFSET (ITARGET).
*
*       null terminate all the strings
*
        MOVE CHAR-NULL TO QEND1.
        MOVE CHAR-NULL TO QEND2.
        MOVE CHAR-NULL TO REND.
        MOVE CHAR-NULL TO DBEND.



2000-READ-INPUT-RECORD.

        DISPLAY "ALTERNATIVE TECHNOLOGIES HIGH PERFORMANCE IDM HOST INTERFA
        DISPLAY "FILES HAVE BEEN DEFINED. HIT RETURN TO READ AND DISPLAY."
*
*       dummy terminal input to provide a pause
*
        ACCEPT INPUT-RECORD.


3000-GET-OUTPUT.
*
*       blank the data buffer
*
        MOVE " " TO BUFFER.
*
*       set the number of records to read on this call
*
        MOVE 20 TO NUMRECS.
*
*       call AT-READ for the first command and data buffer format
*
        CALL "AT-READ" USING IDMRUN
                                BY REFERENCE IDM-FILE-1
*
*       tell it which data buffer to put the data into
*
                                BY REFERENCE BUFDEF(1)
*
*       and how many tuples to read
*
                                BY REFERENCE NUMRECS
```

```
                        GIVING IDMLIB-RETURN-CODE.
*
*       go print the tuples from the data buffer
*
        PERFORM 3100-PRINT-TUPLES VARYING RECNO FROM 1 BY 1
                                        UNTIL RECNO = NUMRECS.



4000-GET-OUTPUT.
*
*       this is the same as 3000-GET-OUTPUT except
*               it executes a different command
*
        MOVE " " TO BUFFER.
        MOVE 20 TO NUMRECS.
*
*       if AT-OPEN for IDM-FILE-2 had been called with
*       QTYPE = atCOPYIN, and with RELNAME="mykinds"
*       this would be a call to AT-WRITE which would write
*       the buffer to the mykinds relation.  Through appropriate
*       use of this technique, data can be transferred
*       dynamically between databases and relations via the host.
*
        CALL "AT-READ" USING IDMRUN
                                BY REFERENCE IDM-FILE-2
                                BY REFERENCE BUFDEF(1)
                                BY REFERENCE NUMRECS
                        GIVING IDMLIB-RETURN-CODE.
        PERFORM 3100-PRINT-TUPLES VARYING RECNO FROM 1 BY 1
                                        UNTIL RECNO = NUMRECS.



3100-PRINT-TUPLES.
*
*       routine to display a tuple
*
        MOVE CORRESPONDING BUFDEF(RECNO) TO OUTPUT-RECORD.
*
*       get rid of nulls used by IDM to terminate string values
*
        INSPECT OUTPUT-RECORD REPLACING ALL CHAR-NULL BY " ".
        DISPLAY OUTPUT-RECORD.
        MOVE " " TO OUTPUT-RECORD.



5000-CLOSE.
*
*       close up shop, calling AT-CLOSE for each "file" defined
*               using AT-OPEN
*       call IDM-IRCANCEL just to make certain there are
*               no dangling commands
```

```
*
       CALL "IDM-IRCANCEL" USING IDMRUN.
       CALL "AT-CLOSE" USING IDM-FILE-1.
       CALL "AT-CLOSE" USING IDM-FILE-2
*
*      close the idm channel
*
       CALL "IDM-IRCLOSE" USING IDMRUN.
```