

Examples for COBOL

This section contains examples of using direct Adabas calls in COBOL. The previously defined Adabas files are used in each example.

```

*
*** CONTROL BLOCK
01 CONTROL-BLOCK.
    02 FILLER PIC X(2) VALUE SPACES.
    02 COMMAND-CODE PIC X(2) VALUE SPACES.
    02 COMMAND-ID PIC X(4) VALUE SPACES.
    02 FILE-NUMBER PIC S9(4) COMP VALUE +0.
    02 RESPONSE-CODE PIC S9(4) COMP VALUE +0.
    02 ISN PIC S9(8) COMP VALUE +0.
    02 ISN-LOWER-LIMIT PIC S9(8) COMP VALUE +0.
    02 ISN-QUANTITY PIC S9(8) COMP VALUE +0.
    02 FORMAT-BUFFER-LENGTH PIC S9(4) COMP VALUE +100.
    02 RECORD-BUFFER-LENGTH PIC S9(4) COMP VALUE +250.
    02 SEARCH-BUFFER-LENGTH PIC S9(4) COMP VALUE +50.
    02 VALUE-BUFFER-LENGTH PIC S9(4) COMP VALUE +100.
    02 ISN-BUFFER-LENGTH PIC S9(4) COMP VALUE +20.
    02 COMMAND-OPTION-1 PIC X VALUE SPACE.
    02 COMMAND-OPTION-2 PIC X VALUE SPACE.
    02 ADDITIONS-1 PIC X(8) VALUE SPACES.
    02 ADDITIONS-2 PIC X(4) VALUE SPACES.
    02 ADDITIONS-3 PIC X(8) VALUE SPACES.
    02 ADDITIONS-4 PIC X(8) VALUE SPACES.
    02 ADDITIONS-5 PIC X(8) VALUE SPACES.
    02 COMMAND-TIME PIC S9(8) COMP VALUE +0.
    02 USER-AREA PIC X(4) VALUE SPACES.
*
*** USER BUFFER AREAS
01 FORMAT-BUFFER PIC X(100) VALUE SPACES.
01 RECORD-BUFFER PIC X(250) VALUE SPACES.
01 SEARCH-BUFFER PIC X(50) VALUE SPACES.
01 VALUE-BUFFER PIC X(100) VALUE SPACES.
01 ISN-BUFFER PIC X(20) VALUE SPACES.
*
*** ADDITIONAL FIELDS USED IN THE EXAMPLES
01 PROGRAM-WORK-AREA.
    05 COMM-ID PIC X(4).
    05 COMM-ID-X REDEFINES COMM-ID PIC S9(8) COMP.
    05 INPUT-KEY PIC X(8).
    05 RECORD-BUFFER-EX2.
        10 RECORD-BUFFER-A PIC X(8).
        10 RECORD-BUFFER-B PIC S9(3) COMP-3.
    05 RECORD-BUFFER-EX3.
        10 OPEN-RECORD-BUFFER.
            15 OPEN-RECORD-BUFFER-X PIC X(8).
            15 FILLER PIC S9(8) COMP.
        10 FILLER PIC X(18).
        10 UPDATED-XC PIC X(6).
        10 LAST-XD PIC X(8).
        10 FILLER PIC X(5).
    05 USER-DATA.
        10 RESTART-XD PIC X(8).
        10 RESTART-ISN PIC S9(8) COMP.
    05 SYNC-CHECK-SWITCH PIC 9 VALUE 0.
    05 AB-VALUE PIC S9(4) COMP-3 VALUE +500.
*

```

Example 1

- Find the set of records in file 2 with XB = 99.
- Read each record selected using the GET NEXT option.

Issue Open Command

```

EXMP1.
  MOVE      'OP' TO COMMAND-CODE.
  MOVE      'ACC.' TO RECORD-BUFFER.
  CALL      'ADABAS'
           USING CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
  IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX1ERR.

```

Issue Find Command

```

MOVE      'S1' TO COMMAND-CODE.
MOVE      'S101' TO COMMAND-ID.
MOVE      2 TO FILE-NUMBER.
MOVE      0 TO ISN-LOWER-LIMIT.
MOVE      0 TO ISN-BUFFER-LENGTH.
MOVE      '.' TO FORMAT-BUFFER.
MOVE      'XB,3,U.' TO SEARCH-BUFFER.
MOVE      '099' TO VALUE-BUFFER.
CALL      'ADABAS' USING CONTROL-BLOCK, FORMAT-BUFFER,
           RECORD-BUFFER, SEARCH-BUFFER, VALUE-BUFFER.
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX1ERR.
EX1A.
  IF ISN-QUANTITY = 0 GO TO EX1EXIT.

```

Read Each Qualifying Record

```

EX1B.
  MOVE      'L1' TO COMMAND-CODE.
  MOVE      0 TO ISN.
  MOVE      'N' TO COMMAND-OPTION-2.
  MOVE      'RG.' TO FORMAT-BUFFER.
EX1C.
  CALL      'ADABAS'
           USING CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
  IF RESPONSE-CODE = 0 GO TO EX1D.
  IF RESPONSE-CODE = 3 GO TO EX1EXIT.
EX1D.
  . . . PROCESS RECORD . . .
  GO TO EX1C.

```

Error Routine

```

EX1ERR.
*      .DISPLAY ERROR MESSAGE
*      .TERMINATE USER PROGRAM

```

Issue Close Command

```
EX1EXIT.
    MOVE      'CL' TO COMMAND-CODE.
    CALL      'ADABAS' USING CONTROL-BLOCK.
    IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX1ERR.
```

Example 2

- All records in file 1 are to be read in physical sequential order.
- Each record read is to be updated with the following values:
 - Field AA = ABCDEFGH
 - Field AB = 500
- User is to have exclusive control of file 1.

Issue Open Command

```
EXMP2.
    MOVE      'OP' TO COMMAND-CODE.
    MOVE      'EXU=1.' TO RECORD-BUFFER.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
    IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX2ERR.
```

Issue Read Physical Sequential Command

```
EX2A.
    MOVE      'L201' TO COMMAND-ID.
    MOVE      1 TO FILE-NUMBER.
    MOVE      0 TO ISN.
    MOVE      'GA.' TO FORMAT-BUFFER.
EX2B.
    MOVE      'L2' TO COMMAND-CODE.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
    IF RESPONSE-CODE = 0 GO TO EX2C.
    IF RESPONSE-CODE = 3 GO TO EX2EXIT.
    GO TO EX2ERR.
```

Update Record

- The same fields are to be updated as were read.
- The same CID and format buffer can be used for the update command.
- The ISN of the record to be updated is already in the ISN field as a result of the L2 command.

```

EX2C.
    MOVE      'A1' TO COMMAND-CODE.
    MOVE      'ABCDEFGH' TO RECORD-BUFFER-A.
    MOVE      AB-VALUE TO RECORD-BUFFER-B.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER-EX2.
    IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX2ERR.
    GO TO EX2B.

```

Error Routine

```

EX2ERR.
    . DISPLAY ERROR MESSAGE
    . TERMINATE USER PROGRAM

```

Close User Session

```

EX2EXIT.
    MOVE      'CL' TO COMMAND-CODE.
    CALL      'ADABAS' USING CONTROL-BLOCK.
    IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX2ERR.

```

Example 3 : User Session with ET Logic

During user session initialization, the user program is to display information indicating the last successfully processed transaction of the previous user session.

For each user transaction, the user program is to

- accept from a terminal 8 characters of input to be used as the key for updating files 1 and 2; and
- issue the Find command for file 1 to determine if a record exists with field AA = input key.

If no record is found, the user program is to issue a message. If a record is found, the user program is to

- delete the record from file 1; and
- add a new record to file 2: Field RA = input key entered.

Other fields are to contain a null value.

If the record cannot be successfully added, the user program is to issue a BT command and display an error message.

If both updates are successful, the user program is to issue an ET command.

Session Initialization

This section of the program is only executed during user session initialization.

- The OP command is issued with ET data of the previous session being read.
- A message is displayed on the terminal screen identifying the last successfully processed transaction of the user's previous session.

```

EX3.
  MOVE      'OP' TO COMMAND-CODE.
  MOVE      'E' TO COMMAND-OPTION-2.
  MOVE      'USER0002' TO ADDITIONS-1.
  MOVE      'PASSWORD' TO ADDITIONS-3.
  MOVE      'UPD=1,2.' TO RECORD-BUFFER.
  CALL      'ADABAS' USING
            CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
  IF RESPONSE-CODE = 9 GO TO EX3.
  IF RESPONSE-CODE NOT EQUAL TO 0
    GO TO EX3ERR.

EX3A.
  MOVE      COMMAND-ID TO COMM-ID.
  IF COMM-ID-X = +0
    GO TO EX3B.
* Display ET data (contained in RECORD BUFFER) on screen to inform user of
* last successfully processed transaction of previous user session.
  . . .DISPLAY ET DATA. . .
  GO TO EX3C.

EX3B.
*** No ET data received.
* Display message that no transactions were successfully processed during
* the previous user session
  . . .DISPLAY MESSAGE . . .
*** Transaction processing.
* This section is executed for each user transaction.
EX3C.
* . . .ACCEPT INPUT FROM TERMINAL. . .
* Issue Find command for file 1 to determine if record exists with field AA
* equal to input key entered.
EX3D.
  MOVE      'S4' TO COMMAND-CODE.
  MOVE      SPACES TO COMMAND-ID.
  MOVE      1 TO FILE-NUMBER.
  MOVE      0 TO ISN-LOWER-LIMIT.
  MOVE      '.' TO FORMAT-BUFFER.
  MOVE      'AA.' TO SEARCH-BUFFER.
  MOVE      INPUT-KEY TO VALUE-BUFFER.
  CALL      'ADABAS' USING
            CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER,
            SEARCH-BUFFER, VALUE-BUFFER, ISN-BUFFER.
  IF RESPONSE-CODE = 0
    GO TO EX3E.
  GO TO EX3ERR.

EX3E.
  IF ISN-QUANTITY NOT EQUAL TO ZEROS
    GO TO EX3F.
***No records found, issue message requesting correction.
  . . .ISSUE MESSAGE . . .
  GO TO EX3C.
*** Delete record from file 1.
* ISN of record to be deleted is already in ISN field and in hold
status
* as a result of the S4 command.
EX3F.
  MOVE      E3' TO COMMAND-CODE.
  CALL      'ADABAS' USING CONTROL-BLOCK.
  IF RESPONSE-CODE = 0
    GO TO EX3G.
  IF RESPONSE-CODE = 9
    GO TO EX3D.
  GO TO EX3ERR.

```

```
*** Add new record to file 2.
EX3G.
    MOVE      'N1' TO COMMAND-CODE.
    MOVE      2 TO FILE-NUMBER.
    MOVE      'RA.' TO FORMAT-BUFFER.
    MOVE      INPUT-KEY TO RECORD-BUFFER.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
    IF RESPONSE-CODE = 0
        GO TO EX3I.
    IF RESPONSE-CODE = 9
        GO TO EX3D.
*** Attempt to add new record not successful.
* Backout transaction.
* Notify user that error condition exists.
    MOVE      'BT' TO COMMAND-CODE.
    CALL      'ADABAS' USING control-block.
    IF RESPONSE-CODE = 0
        GO TO EX3H.
*** Backout not successful.
* Issue message indicating that the backout was not successful
    GO TO EX3ERR.
EX3H.
*** Backout successful.
* Issue message indicating the error condition detected while while
adding a
* new record
    GO TO EX3ERR.
*** Updates successfully executed.
* Issue ET command with ET data.
EX3I.
    MOVE      'ET' TO COMMAND-CODE.
    MOVE      'E' TO COMMAND-OPTION-2.
    MOVE      INPUT-KEY TO RECORD-BUFFER.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
    IF RESPONSE-CODE = 0
        GO TO EX3C.
    IF RESPONSE-CODE = 9
        GO TO EX3D.
*** Error Routine
EX3ERR.
*      . DISPLAY ERROR MESSAGE
*      . TERMINATE USER PROGRAM
      . . .
```