i Magazine 2016 夏号

IBM iアプリケーション開発 最新事情

Part2 ILE RPG：システムAPI・組み込みSQL活用術

サンプルコード

･･･････････････････････････

下記のコードは、i Magazine 2016 夏号の記事「IBM iアプリケーション開発 最新事情」Part Ⅱで紹介したサンプルコードです。

･･･････････････････････････

サンプル1

API QUSRJOBIによる現行ユーザー・プロファイル取得

\*\*free

 ctl-opt openopt(\*nocvtdata);

 ctl-opt dftactgrp(\*no);

 ctl-opt main(main);

 dcl-proc main;

 // API QUSRJOBIプロトタイプ定義

 dcl-pr QUSRJOBI extpgm;

 receiverVariable char(2000) options(\*varsize);

 lengthOfReceiverVariable int(10) const;

 formatName char(8) const;

 qualifiedJobName char(26) const;

 internalJobIdentifier char(16) const;

 errorCode char(32767) options(\*varsize:\*nopass);

 resetPerformanceStatistics char(1) options(\*nopass) const;

 end-pr;

 // API QUSRJOBIからのJOBI0600出力用データ構造

 dcl-ds receiverVariableDs qualified;

 numberOfBytesReturned int(10);

 numberOfBytesAvailable int(10);

 jobName char(10);

 userName char(10);

 jobNumber char(6);

 internalJobIdentifier char(16);

 jobStatus char(10);

 jobType char(1);

 jobSubType char(1);

 jobSwitches char(8);

 endStatus char(1);

 subsystemDescriptionName char(10);

 subsystemDescriptionLibraryName char(10);

 currentUserProfile char(10);

 end-ds;

 // APIエラー・データ用データ構造

 dcl-ds errorCodeDs qualified;

 bytesProvided int(10) inz(0);

 bytesAvailable int(10) inz(0);

 exceptionId char(7);

 exceptionData char(256);

 end-ds;

 // API QUSRJOBIへ渡す変数群

 dcl-s lengthOfReceiverVariable int(10) inz(0);

 dcl-s formatName char(8) inz(\*blank);

 dcl-s qualifiedJobName char(26) inz(\*blank);

 dcl-s internalJobIdentifier char(16) inz(\*blank);

 dcl-s resetPerformanceStatistics char(1) inz(\*blank);

 //ローカル変数

 dcl-s currentJobUserProfile char(10) inz(\*blank);

 dcl-s msg char(50) inz(\*blank);

 dcl-s res char(1) inz(\*blank);

 monitor;

 //引き数変数への値のセット

 lengthOfReceiverVariable = %size(receiverVariableDs);

 formatName = 'JOBI0600';

 qualifiedJobName = '\*';

 internalJobIdentifier = '';

 resetPerformanceStatistics = '1';

 // API QUSRJOBIでジョブ関連情報の取得

 QUSRJOBI(receiverVariableDs

 :lengthOfReceiverVariable

 :formatName

 :qualifiedJobName

 :internalJobIdentifier

 :errorCodeDs

 :resetPerformanceStatistics);

 currentJobUserProfile = receiverVariableDs.currentUserProfile;

 msg = 'Current Job User is: ' + %trim(currentJobUserProfile) + '.';

 on-error;

 msg = 'An error occurred.';

 endmon;

 dsply msg '' res;

 return;

 end-proc;

サンプル2 API open()でのエラー・ハンドリング例

C \_\_errno関数のILE RPGでのプロトタイプ定義 (引き数なし、戻り値はポインター)

ポインターerrnoPtrの位置から

4バイト分でint型変数とする

エラー番号のポインターを取得

int型のerrnoを文字型にキャスト

dcl-pr getErrnoPtr pointer extproc('\_\_errno');

end-pr;

 dcl-s errnoPtr pointer;

dcl-s errno int(10) based(errnoPtr);

 fd = open('/home/nakamura/test.txt'

 :O\_WRONLY+O\_CREAT+O\_TRUNC+O\_CCSID

 :S\_IRWXG+S\_IRWXO+S\_IRWX

 :1208);

 if fd < 0;

 errnoPtr = getErrnoPtr();

 msg = 'open operation failed. errno: ' + %char(errno);

 dsply msg '' wait;

 return;

 endif;

サンプル3　API open()のプロトタイプ定義

dcl-pr open int(10) extproc(\*cwiden:'open');

 path pointer value options(\*string);

 oflag int(10) value;

 mode uns(10) value options(\*nopass);

 convId uns(10) value options(\*nopass);

 textFileCreatConvId uns(10) value options(\*nopass);

 end-pr;

サンプル4　API read()プロトタイプ定義

dcl-pr read int(10) extproc(\*cwiden:'read');

 fd int(10) value;

 buf pointer value;

 nbyte uns(10) value;

 end-pr;

サンプル5. API write()プロトタイプ定義

dcl-pr write int(10) extproc(\*cwiden:'write');

fd int(10) value;

 buf pointer value;

 nbyte uns(10) value;

end-pr;

サンプル6　API close()プロトタイプ定義

dcl-pr close int(10) extproc(\*cwiden:'close');

fd int(10) value;

end-pr;

サンプル7　IFS関連APIを使用したILE RPGプログラム例

\*\*free

ctl-opt openopt(\*nocvtdata);

 ctl-opt dftactgrp(\*no);

 ctl-opt main(main);

 ctl-opt bnddir('QC2LE');

 dcl-proc main;

 /COPY QSYSINC/QRPGLESRC,IFS

 /COPY QSYSINC/QRPGLESRC,ERRNO

//ローカル変数

 dcl-s fromFilePath char(50) inz(\*blank);

 dcl-s toFilePath char(50) inz(\*blank);

 dcl-s fromFd int(10) inz(0);

 dcl-s toFd int(10) inz(0);

 dcl-s fileData char(30720) ccsid(1208) inz(\*blank);

 dcl-s rbyte int(10) inz(0);

 dcl-s wbyte int(10) inz(0);

 dcl-s errnoPtr pointer;

 dcl-s errno int(10) based(errnoPtr);

 dcl-s errMsg char(50) inz(\*blank);

 dcl-s msg char(50) inz(\*blank);

 dcl-s res char(1) inz(\*blank);

 //変数への値のセット

 fromFilePath = '/home/nakamura/test.txt';

 toFilePath = '/home/nakamura/test2.txt';

 // 既存ファイルのオープン

 fromFd = open(%trim(fromFilePath):O\_RDONLY);

 if fromFd < 0;

 errnoPtr = getErrnoPtr();

 errMsg = 'exist file open operation failed. errno: ' + %char(errno);

 dsply errMsg '' res;

 return;

 endif;

 //既存ファイルの読み取り

 rbyte = read(fromFd:%addr(fileData):%size(fileData));

 if rbyte < 0;

 errnoPtr = getErrnoPtr();

 errMsg = 'read operation failed. errno: ' + %char(errno);

 //既存ファイルのクローズ

 callp close(fromFd);

 dsply errMsg '' res;

 return;

 endif;

 //既存ファイルのクローズ

 callp close(fromFd);

 msg = 'read operation finished.';

 dsply msg '' res;

 //新規ファイルのオープン

 toFd = open(%trim(toFilePath)

 :O\_WRONLY+O\_CREAT+O\_TRUNC+O\_CCSID

 :S\_IRWXU+S\_IRWXG+S\_IRWXO

 :1208);

 if toFd < 0;

 errnoPtr = getErrnoPtr();

 errMsg = 'new file open operation failed. errno: ' + %char(errno);

 dsply errMsg '' res;

 return;

 endif;

 //新規ファイルへの書き出し

 wbyte = write(toFd:%addr(fileData):rbyte);

 if wbyte < 0;

 errnoPtr = getErrnoPtr();

 errMsg = 'write operation failed. errno: ' + %char(errno);

 //新規ファイルのクローズ

 callp close(toFd);

 dsply errMsg '' res;

 return;

 endif;

 //新規ファイルのクローズ

 callp close(toFd);

 msg = 'write operation finished.';

 dsply msg '' res;

 return;

 end-proc;

サンプル8

文字列で組み上げたSQLステートメントとしての実行例

...

 // SQLステートメントの成形

 stmt01 = 'SELECT dept\_name, dept\_mgr\_id ' +

 'FROM dept\_Info ' +

 'WHERE dept\_code IN (''';

 //入力された部門コードのステートメントへの組み込み

 for i = 1 to numOfDeptIds;

 stmt01 = %trim(stmt01) + %trim(deptIds(i));

 if i < numOfDeptIds;

 stmt01 = %trim(stmt01) + ''',';

 endif;

 endfor;

 stmt01 = %trim(stmt01) + ''')';

 // SQLステートメントの準備

 exec sql prepare Stmt01 from :stmt01;

 //カーソルの宣言

 exec sql declare CsrForStmt01 cursor for Stmt01;

 //カーソルのオープン

 exec sql open CsrForStmt01;

...

サンプル9　動的SQLステートメントとしての実行例

...

 // SQLステートメントの成形

 stmt01 = 'SELECT dept\_name, dept\_mgr\_id ' +

 'FROM dept\_Info ' +

 'WHERE dept\_code IN (';

 //入力された部門コード数に応じたパラメーター・マーカーの成形

 for i = 1 to numOfDeptIds;

 stmt01 = %trim(stmt01) + '?';

 if i < numOfDeptIds;

 stmt01 = %trim(stmt01) + ',';

 endif;

 endfor;

 stmt01 = %trim(stmt01) + ')';

 // SQLステートメントの準備

 exec sql prepare Stmt01 from :stmt01;

 //カーソルの宣言

 exec sql declare CsrForStmt01 cursor for Stmt01;

 //パラメーター・マーカーの個数に応じたカーソルのオープン

 select;

 when numOfDeptIds = 1;

 exec sql open CsrForStmt01 using :deptId1;

 when numOfDeptIds = 2;

 exec sql open CsrForStmt01 using :deptId1,:deptId2;

 when numOfDeptIds = 3;

 exec sql open CsrForStmt01 using :deptId1,:deptId2,:deptId3;

 when numOfDeptIds = 4;

 exec sql open CsrForStmt01 using :deptId1,:deptId2,:deptId3,:deptId4;

 ...

 endsl;

...

サンプル10

SQL記述子を用いた動的SQLステートメントとしての実行例

...

 // SQL記述子への記憶域の割り振り

 exec sql allocate descriptor 'paramMarkers' with max :numOfDeptIds;

 // SQL記述子内の要素数として入力された部門コード数をセット

 exec sql set descriptor 'paramMarkers' count=:numOfDeptIds;

 // SQL記述子に入力された部門コードをセット

 clear inExp;

 for i = 1 to numOfDeptIds;

 j = lengthOfDeptId \* (i - 1) + 1;

 valueLength = lengthOfDeptId;

 valueData = %trim(deptIds(i));

 exec sql set descriptor 'paramMarkers' value :i

 type=12, length=:valueLength, data=:valueData;

 //入力された部門コード数に応じたIN述部内の

 //パラメーター・マーカーの成形

 inExp = inExp + '?';

 if i < numOfDeptIds;

 inExp = inExp + ',';

 endif;

 endfor;

 // SQLステートメントの成形

 stmt01 = 'SELECT dept\_name, dept\_mgr\_id ' +

 'FROM dept\_Info ' +

 'WHERE dept\_code IN ({$INEXP$})';

 // IN述部にパラメーター・マーカーをセット

 stmt01 = %scanrpl('{$INEXP$}':inExp:stmt01);

 // SQLステートメントの準備

 exec sql prepare Stmt01 from :stmt01;

 //カーソルの宣言

 exec sql declare CsrForStmt01 cursor for Stmt01;

 // SQL記述子を用いたカーソルのオープン

 exec sql open CsrForStmt01 using sql descriptor 'paramMarkers';

 // SQL記述子への記憶域割り当ての解除

 exec sql deallocate descriptor 'paramMarkers';

...

サンプル11

SQL結果セットの受け渡しサポート (SQLストアード・プロシージャーを呼び出し、結果セットを受け取るILE RPG)

...

// SQLタイプ結果セット・ロケーター変数の宣言

 dcl-s rsQryDeptInfo sqltype(result\_set\_locator);

...

 // SQL記述子(グローバル)への記憶域の割り振り

 exec sql allocate descriptor global 'paramMarkers' with max :numOfDeptIds;

 // SQL記述子内の要素数として入力された部門コード数をセット

 exec sql set descriptor global 'paramMarkers' count=:numOfDeptIds;

 // SQL記述子に入力された部門コードをセット

 clear inExp;

 for i = 1 to numOfDeptIds;

 j = lengthOfDeptId \* (i - 1) + 1;

 valueLength = lengthOfDeptId;

 valueData = %trim(deptIds(i));

 exec sql set descriptor global 'paramMarkers' value :i

 type=12, length=:valueLength, data=:valueData;

//入力された部門コード数に応じたIN述部内の

 //パラメーター・マーカーの成形

 inExp = inExp + '?';

 if i < numOfDeptIds;

 inExp = inExp + ',';

 endif;

 endfor;

 // WHERE句の成形

 whereClause = 'dept\_code IN ({$INEXP$})';

 // IN述部にパラメーター・マーカーをセット

 whereClause = %scanrpl('{$INEXP$}':inExp:whereClause);

 //部門情報を取得するSQLプロシージャーの実行

 exec sql call qryDeptInfo(:whereClause);

 // SQLプロシージャーから戻された結果セットと

 //結果セット・ロケーター変数との関連付け

 exec sql associate result set locator (:rsQryDeptInfo)

 with procedure qryDeptInfo;

 //結果セットにカーソルを割り当て

 exec sql allocate crsQryDeptInfo

 cursor for result set :rsQryDeptInfo;

...

サンプル12

SQL結果セットの受け渡しサポート (ILE RPGから呼び出す結果セットを返すSQLストアード・プロシージャー)

CREATE OR REPLACE PROCEDURE qryDeptInfo(

 IN whereClause VARCHAR(1024))

LANGUAGE SQL READS SQL DATA

DYNAMIC RESULT SETS 1

BEGIN

 DECLARE numOfParamMarkers INTEGER;

 DECLARE SQLStmt VARCHAR(1024);

 DECLARE CsrQryDeptInfo CURSOR WITH RETURN TO CALLER FOR StmtQryDeptInfo;

 DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET numOfParamMarkers = 0;

 GET SQL DESCRIPTOR GLOBAL 'paramMarkers' numOfParamMarkers = COUNT;

 SET SQLStmt = 'SELECT dept\_name, dept\_mgr\_id FROM dept\_Info WHERE '

CONCAT whereClause;

 PREPARE StmtQryDeptInfo FROM SQLStmt;

 IF numOfParamMarkers > 0 THEN

 OPEN CsrQryDeptInfo USING SQL DESCRIPTOR GLOBAL 'paramMarkers';

 DEALLOCATE SQL DESCRIPTOR GLOBAL 'paramMarkers';

 ELSE

 OPEN CsrQryDeptInfo;

 END IF;

 RETURN;

END