

IBM i World 2022

IBM i コンテンツ (2022年6月版)

フリーフォーマットRPGでプログラムを書いてみよう

日本アイ・ビー・エム株式会社
テクノロジー事業本部
IBM Powerテクニカルセールス
澤田英寿

目次

1. フリーフォーマットRPGのご紹介
2. フリーフォーマットRPGでプログラムを書いてみよう
3. 補足情報

1. フリーフォーマットRPG (FF RPG)のご紹介

- ・既存のRPGプログラマーは新しい文法を簡単に学べる
- ・RPG以外のプログラマーにとっても、固定形式より学びやすい

```
ctl-opt bnddir('ACCRCV');
```

H仕様書

```
dcl-f custfile usage(*update);
```

F仕様書

```
dcl-ds custDs likerec(custRec);
```

```
dcl-f report printer;
```

D仕様書

複数の仕様書記述が
ひとつのソース内で完結

```
read custfile custDs;
dow not %eof;
  if dueDate > %date(); // overdue?
    sendOverdueNotice();
    write reportFmt;
    exec sql insert :name, :duedate into
      mylib/myfile;
  endif;
  read custfile custDs;
enddo;
*inlr = '1';
```

C仕様書

```
dcl-proc sendOverdueNotice;
  sendInvoice (custDs : %date());
end-proc;
```

既存RPGプログラマーにとってのフリーフォーマットRPGのメリット

- 構文のインデントによりメンテナンス性向上
if, for, dow (Do While) などプログラムの構造が分かりやすい

- 多くのフラストレーションの解消
『/FREE』と『/END-FREE』の記述が不要
固定形式での2行にわたる宣言を1行に

```
D getNextCustomer...
```

```
D          pr
```

vs

```
dcl-pr getNextCustomer;
```

- 長い文字列キーワードに対するコーディング・スペースの不足の解消

```
D HSSFCellStyle c
```

```
D          'org.apache.poi.hssf.-
```

```
D          usermodel.HSSFCellStyle'
```

vs

```
dcl-c HSSFCellStyle 'org.apache.poi.hssf.usermodel.HSSFCellStyle';
```

新しくRPGを始めるプログラマーにとっての フリーフォーマットRPG (FF RPG) のメリット

- 固定形式と比べると、FF RPGの学習はとても容易
 - Rational Developer for iエディターの色付け機能が改善されたことで、JavaやPHPなどと同じようなルック・アンド・フィールでRPGのコードを扱える
 - 桁位置などに戸惑う必要はなく、RPGの命令コードやキーワードなどを覚えるだけでよい

フリーフォーマットRPGになっても変わらないもの

- 全てのバージョンのRPGについて、以下は変わらない
 - 以前のバージョンの上位互換
 - RPGコンパイラによる固定形式のサポート
 - ビジネス・プログラミングにおいて重要な十進数演算に対するサポート
 - システムとの優れた統合
 - データベースの統合による容易なレコード・レベルI/Oのコーディング
 - データ域のサポート
 - 組込みSQLにおけるレコードレベルI/OとSQL I/Oの混在
 - ILE RPG
 - 全OPMと全ILE言語の間の呼び出し
 - RPGとJava間の呼び出し
 - XML文書の読み込み

追加のサポート

- 組込みSQL
 - SQLプリコンパイラーも新しいフリーフォーマット文法をサポート
- Rational Developer for i (RDi)
 - RDi9.0.1以降は、フリーフォーマットRPGをサポート
 - フリーフォーマットへの変換機能
 - H F D P 仕様書はフリーフォーマット変換しない

フリーフォーマットRPGの前提条件

- IBM i
 - IBM i 7.1以降
 - IBM i 7.1の場合、下記PTFを適用
 - RPGIV コンパイラー：PTF SI51640
 - SQL プリコンパイラー：DB2 PTF Group SF99701 レベル 26 以降
- クライアントPC (Editor)
 - Rational Developer for i 9.0.1以降 (当資料ではこちらを使います)
 - ※5250画面のSEUではフリーフォーマットはサポートされません。
構文エラーとなります。
 - Visual Studio Code
 - IBM iへの導入・使用法については、下記のリンクを参照
 - <https://www.imagazine.co.jp/feat-visual-studio-code-part1/>

(参考) RPGⅢ / ILE RPG (固定) / FF RPGの比較

*青字=IBM i ライセンスPGM,
ピンク字=OSS

	RPGⅢ	ILE RPG (固定フォーマット)	ILE RPG (フリーフォーマット)
開発想定アプリケーション	SoR	SoE / SoR	SoE / SoR
想定開発スタイル	ウォーターフォール	アジャイル ウォーターフォール	アジャイル ウォーターフォール
プログラム構造	モノシリック	モジュール化 (モノシリックも可)	モジュール化 (モノシリックも可)
推奨開発環境	5250 / GUI	GUI (5250も可)	GUI
ソースコード保管先 形式	IBM i ソース物理ファイル(SRCPF)	IBM i またはコードサーバ テキストファイル	IBM i またはコードサーバ テキストファイル
ソースエディター	SEU	SEU / RDi / VSCode	RDi / VSCode
ソースコード管理(SCM)	PDM	IBM i Merlin*1 または Git	IBM i Merlin または Git
ビルド管理	CLコマンド、ベンダーツール	IBM i Merlin または Jenkins	IBM i Merlin または Jenkins
ディプロイ管理	CLコマンド、ベンダーツール	IBM i Merlin または Jenkins	IBM i Merlin または Jenkins
プロジェクト管理 課題、リリース、テスト	ベンダーツール	IBM i Merlin または JIRA	IBM i Merlin または JIRA

*1 IBM i Merlin =IBM i Modernization Engine for Lifecycle Integration

<https://www.ibm.com/jp-ja/products/ibm-i-merlin>

2. フリーフォーマットRPGでプログラムを書いてみよう

1) 事前準備

- Rational Developer for i のインストール

- この資料では、開発ツールとして、Rational Developer for i を使用しています。
- 事前準備として、Rational Developer for i を開発用PCにインストールください。
- Rational Developer for i のインストールについては下記リンク先をご参照ください。

Rational Developer for i V9.6

<https://www.ibm.com/products/rational-developer-for-i>

120日間の無償使用できるダウンロードサイトです。

導入ガイドは下記のiCafeの資料を参考にしてください。

https://www.i-cafe.info/library/dl_rdiv9-5_installguide

使い方のガイドも下記の資料を参考にしてください。

https://www.i-cafe.info/library/rdi-v9-6_guide_dl

2) フリーフォーマットRPG 初心者向けチュートリアル

ここからは、下記の初心者向けのフリーフォーマットRPGの研修資料を基に、日本語化して構成してます。

Coding in Free-Form RPG IV - a beginner's tutorial (英語)

<https://www.ibm.com/support/pages/node/1108749>

- Chapter 1 Hello World
 - ソースの編集、コンパイル方法
 - コンパイル・エラー時のエラー・メッセージの見方
 - プログラムの実行
 - プログラムのデバッグ
- Chapter 2 General info about RPG
 - RPG概要
- Chapter 3 Variables and procedures
 - 定数、変数定義
 - プロシージャ
- Chapter 4 Introduction to files
 - データベース・ファイル、スプールファイルの使用
- Chapter 5 Display files
 - 表示ファイルの使用

Chapter1 Hello World

初期セットアップ

1. IBM Rational developer for i を起動
2. IBM i の接続を作成
3. ライブラリーを作成 (AAAARPG)
4. ソースファイル「QRPGLESRC」を作成 (次ページから詳細解説)
5. ソールメンバー「HELLO.RPGLE」を作成 (同上)
6. 作成したメンバーを編集

初期セットアップの詳細な手順については、下記リンク先を参照ください。
「さわってみよう Rational Developer for i V9.0」

<https://www.i-cafe.info/column/product/rational-developer-for-i>

Chapter1 Hello World

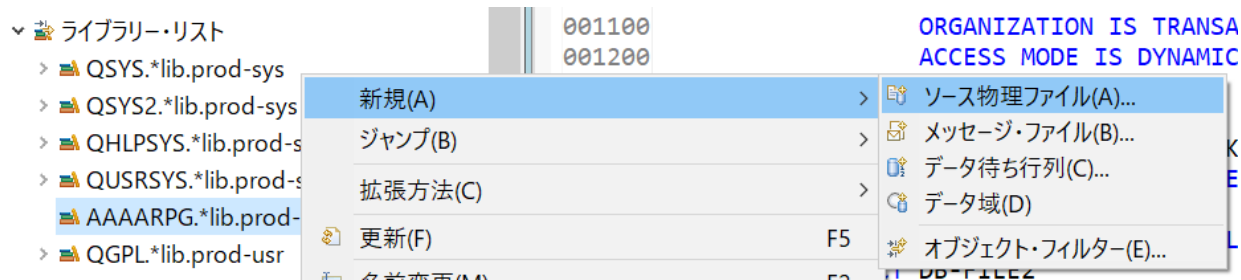
初期セットアップの解説

4. ソースファイル「QRPGLESRC」を作成

RDİ で ソース物理ファイルを作成する手順

1. 作成済みのライブラリー・フィルターを開き、ライブラリーを右クリック、メニューから "新規" → "ソース物理ファイル" をクリックします

ここでは、ライブラリー名は「AAAARPG」とします。



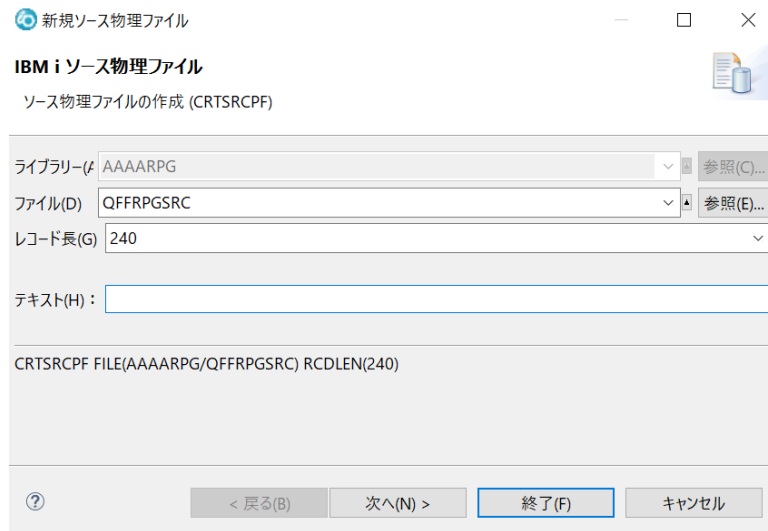
フリーフォーマットRPG は他の言語と同様に、テキストファイルでコーディングする事もできますが
本書では従来のソース物理ファイルとソース・メンバーを使用します

Chapter1 Hello World

初期セットアップの解説

4. ソースファイル「QRPGLESRC」を作成

2. 以下を入力し、「終了」をクリックします
ファイル：QFFRPGSRC
レコード長：240（ソース物理ファイルの最大長）



3. ライブラリーにソース物理ファイル QFFRPGSRC.*file.pf-src が作成されます

- AAAARPG.*lib.prod-usr
 - QFFRPGSRC.*file.pf-src
 - QGPL.*lib.prod-usr

Chapter1 Hello World

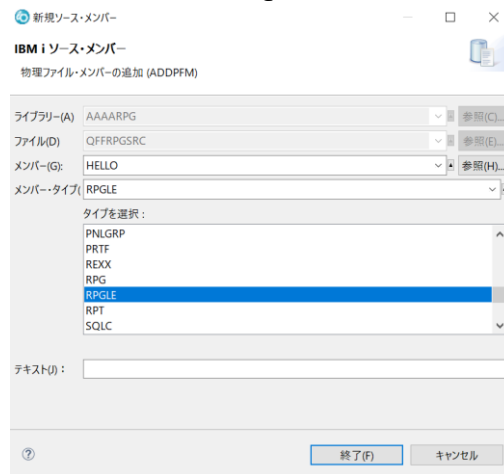
初期セットアップの解説

5. ソールメンバー「HELLO.RPGLE」を作成

RDİ で ソースメンバーを作成する手順

1. 作成したソース物理ファイルを右クリックし、メニューから "新規" → "メンバー" をクリックします
ソース物理ファイル名は QFFRPGSRC です、RSE では QFFRPGSRC.*file.pf-src と表示されます

2. 以下を入力し、「終了」をクリックします
メンバー : HELLO
メンバー・タイプ : RPGLE



3. ソース物理ファイルにメンバー HELLO.rpgle が作成されます
編集エリアに HELLO ソースメンバーがオープンされ、編集可能となります
閉じた場合は、RSE で HELLO.rpgle をダブルクリックし開く事もできます

Chapter1 Hello World

- RPGプログラムの編集

- 以下のコードを入力

※1-7列目までは使用できません。フリーフォーマットで使用できる列は、8-80列のみです。列の番号は、編集画面の上に記載されています。また、カーソルを合わせるとその列番号が表示されます

```
dsply 'Hello World';  
return;
```

- RPGプログラムのコンパイル

- ソースメンバー「HELLO.rpgle」を右クリックし、コンパイル（プロンプト）

→CRTBNDRPGを選択

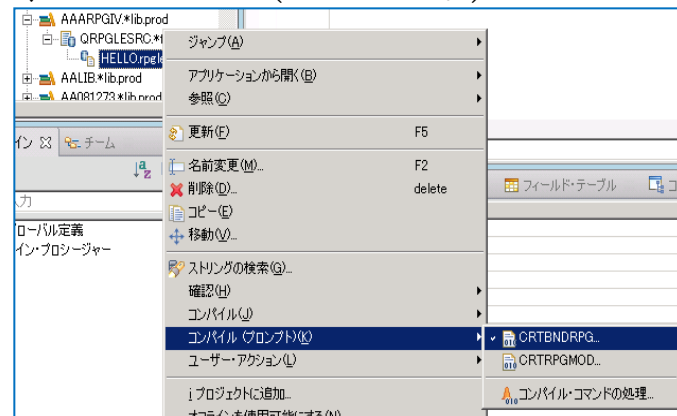
- 「リソースの保存」画面が表示されたら

「OK」をクリックし、保存

- パラメータを指定し、「OK」をクリックし、

コンパイルを実行

※パラメーターは特に指定がなければ
デフォルトでOK



Chapter1 Hello World

- RPGプログラムの実行

- 5250エミュレーターを起動し、ログイン
- 以下のコマンドを実行し、RPGプログラムを実行

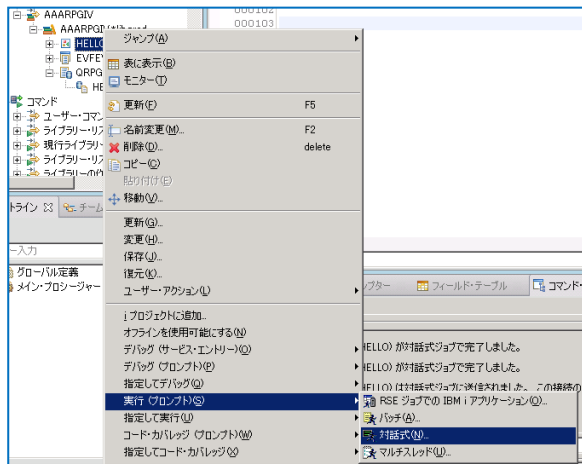
====> CALL AAAARPG/HELLO

実行後、5250エミュレーターに「DSPLY Hello World」のメッセージが表示されます

※表示されない場合は、「F10」キー(詳細なメッセージの組み込み)を押してください。

[補足]RDiからRPGプログラムを実行するコマンドを発行することも可能です。

- RPGプログラムを右クリック→「実行 (プロンプト)」→「対話式」を選択
実行後、5250エミュレーターに「DSPLY Hello World」のメッセージが表示されます



※RDiの接続を対話式ジョブに関連付けるために以下の手順を事前実施してください

- 5250 エミュレーターを起動し、ログイン
- 以下のコマンドを実行
- STRRSESVR NAME(yourhost) ※yourhost: IBM i の接続名

Chapter1 Hello World

- コンパイル・エラーの対処例 1/2

ここでは、コンパイラー・エラーの例を紹介します

- HELLO.RPGLEのソースメンバーを以下のように編集
-‘Hello World’をnameに変更

```
dsply name;
return;
```

- 再度、コンパイルを実行

-重要度30のエラーによりコンパイルが失敗します

-RDiのエラー・リストに「RNF7030: 名前または標識NAMEが定義されていない。」が表示されます

-RNF7030のエラーをダブルクリックすると「name」がハイライトされ、エラーの箇所を確認できます

-※RPGIVでは、変数名は大文字、小文字を区別されません。そのため、エラー・リストの「NAME」とソース内の「name」は同じであることに注意ください。

ID	メッセージ	重大度	行	位置	接続
RNS9308	コンパイルは停止しました。重大度30のエラーがプログラムで見つかりました。	50	0	AAAAARPG/QFFRPGSRC(HELLO)	DEMO00
RNF7030	名前または標識NAMEが定義されていない。	30	1	AAAAARPG/QFFRPGSRC(HELLO)	DEMO00
RNF7503	式に定義されていないオペランドが含まれています。	30	1	AAAAARPG/QFFRPGSRC(HELLO)	DEMO00

Chapter1 Hello World

- コンパイル・エラーの対処例 2/2

- 3. エラー箇所の修正

nameが定義されていないというエラーに対して、以下のように定義します。
プログラムの一番始めにdcl-sを使用し、nameを定義

```
dcl-s name char(10) inz('Jim');  
dsply name;  
return;
```

- 4. 再度、コンパイルを実行

- エラーなく、コンパイルが成功します

- 5. RPGプログラムを実行

- 実行後、5250エミュレーターに「DSPLY Jim」のメッセージが表示されます

```
5 > call hello  
      DSPLY Jim
```

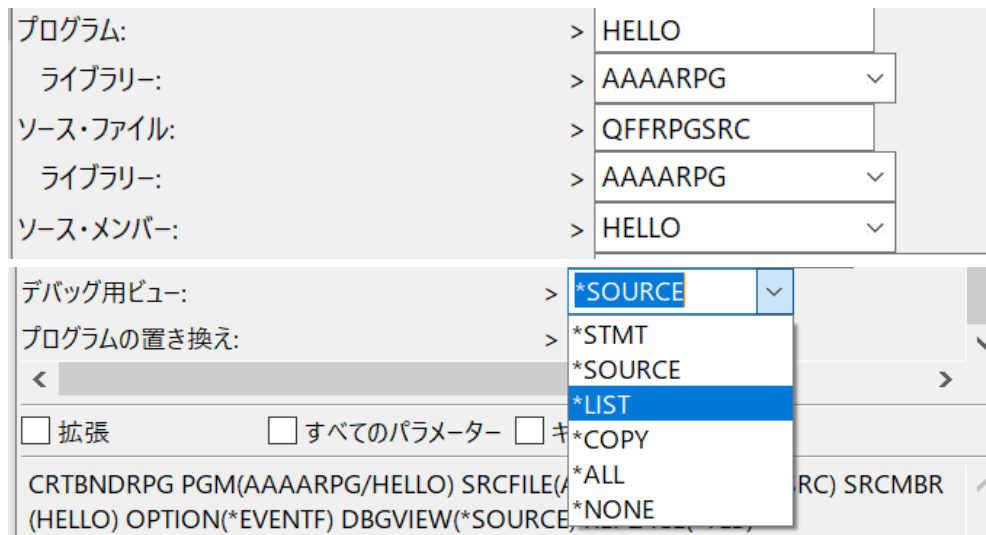
Chapter1 Hello World

- プログラムのデバッグ 1/2

デバッグの前にコンパイル時に「デバッグ用ビュー」のパラメーターを指定し、再度コンパイルします。

- ソースメンバー「HELLO.rpgle」を右クリックし、コンパイル（プロンプト）→CRTBNDRPGを選択
- 「デバッグ用ビュー（DBGVIEW）」パラメータに「*LIST」指定し、「OK」をクリックし、コンパイルを実行

※ 「デバッグ用ビュー（DBGVIEW）」パラメータは、*NONEと*STMT以外を指定します。
*ALL, *SOURCE, *COPYを選択しても、デバッグ可能です。



Chapter1 Hello World

- プログラムのデバッグ 2/2

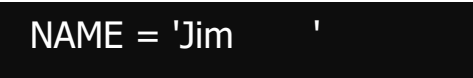
デバッガーを開始し、プログラムを実行し、デバッグを行ないます

※ここでは、5250エミュレーターからの手順を紹介しています。RDiにもデバッグの機能が備わっています。

以下のコマンドを実行し、デバッガーを開始

```
====> STRDBG PGM(AAAARPG/HELLO)
```

- `dsply name;`の行にカーソルを合わせ、F6キーを押し、停止点を追加、F3キーを押して、プログラムを終了
- 以下のコマンドを実行し、プログラムを実行

```
====> CALL AAAARPG/HELLO
```
- 停止点である`dsply name;`の`name`にカーソルを合わせ、F11キーを押し、`name`変数の値を表示
 `NAME = 'Jim'`
•Name返信に'Jim'の値が入っていることが確認できます。
- F3キーを押して、プログラムを終了
- 以下のコマンドを実行し、デバッガーを終了

```
====> ENDDBG
```

Chapter2 General info about RPG

- RPGプログラムでは、列ごとに以下の制限があります
 - 8-80列：フリーフォーマットのコード
 - 1-5列、81列以降：コメント
 - 7列、8列：固定形式のコード

Chapter3 Variables and procedures

- 定義ステートメント：DCL-x
 - 1) 定数 (constant) : dcl-c
 - 2) 独立フィールド (standalone field) : dcl-s
 - 3) データ構造 (data structure) : dcl-ds
 - 4) 配列 (array) : dcl-s xxx dim(n)
 - 5) プロトタイプ (prototype) : dcl-pr , end-pr
 - 6) プロシージャ (procedure) : dcl-proc , end-proc
 - 7) プロシージャ・インターフェース () : dcl-pi , end-pi

Chapter3 Variables and procedures

サンプル・プログラム

1) 定数 (constant) : dcl-c

- 以下のコードをソースメンバーに貼り付け、コンパイルし、実行します

定数を定義
MAX_ELEMSに100
を代入

SAMPLE1

```
dcl-c MAX_ELEMS 100;  
dcl-c default_city_name 'London';  
dsply max_elems;  
dsply default_city_name;  
return;
```

定数を定義
Default_city_name
に'London'を代入

実行結果

```
5 > call sample1  
DSPLY 100  
DSPLY London
```


Chapter3 Variables and procedures

サンプル・プログラム

2) 独立フィールド (standalone field) : dcl-s

SAMPLE2

```
dcl-s num int(10);
```

```
for num = 1 to 3;  
  dsply ('i = ' + %char(num));  
endfor;  
return;
```

for loop

numという名前で独立フィールドを定義
int(10) : 10digitsの整数

%char : char型に変換

+ : 文字列の連結

実行結果

```
call sample2  
DSPLY i = 1  
DSPLY i = 2  
DSPLY i = 3
```

Chapter3 Variables and procedures

サンプル・プログラム

3) データ構造 (data structure) : dcl-ds

```

dcl-ds info qualified;
  name char(10) inz('Sam');
  salary packed(9 : 2) inz(50000.25);
end-ds;
dcl-ds otherInfo likeds(info) inz(*likeds);
dsply (info.name
  + ' has a salary of'
  + %char(info.salary));
otherInfo.name = 'Joe';
otherInfo.salary += 10000;
dsply (otherInfo.name
  + ' has a salary of'
  + %char(otherInfo.salary));
return;

```

infoという名前のデータ構造を定義

qualifiedキーワード：
データ構造名の後にピリオドとサブフィールド名を続けて指定

likedsキーワード：別のデータ構造と同様に定義。サブフィールドは、パラメータに指定された親データ構造のサブフィールドと同じ

inz(*likeds)：親データ構造と同じ方法でサブフィールドを初期化

SAMPLE3

実行結果

```

call sample3
DSPLY  Sam           has a salary of50000.25
DSPLY  Joe           has a salary of60000.25

```

Chapter3 Variables and procedures

サンプル・プログラム

4) 配列 (array) : dcl-s xxx dim(n)

```

dcl-s dates date(*iso) dim(3);

dates(1) = %date(); // the current date
dates(2) = dates(1) + %days(1); // tomorrow
dates(3) = dates(1) + %years(1); // next year
dsply (%char(dates(1)) + ' '
+ %char(dates(2)) + ' '
+ %char(dates(3)));
return;

```

datesという名前の配列を定義
date(*iso) : ISO 形式
(yyyy-mm-dd)の日付タイプ
dimキーワード : 要素の数を指定

%date() : パラメータが指定されない場合、
現在の日付を返す組み込み関数
%days(1) : パラメータの日数を加算
%years(1) : パラメータの年数を加算

SAMPLE4

実行結果

```

call sample4
DSPLY 2022-06-06 2022-06-07 2023-06-06

```

Chapter3 Variables and procedures

サンプル・プログラム

5) プロトタイプ (prototype) : dcl-pr , end-pr

```

dcl-pr qcmdexc extpgm('QCMDEXEC');
theCmd char(3000) const;
cmdLine packed(15 : 5) const;
dbcs char(3) const options(*nopass);

end-pr;
dcl-s cmd varchar(100);

cmd = 'DSPJOB OUTPUT(*PRINT)';
qcmdexc (cmd : %len(cmd));
qcmdexc ('WRKSPLF' : 7);

return;

```

パラメータ
の定義

QCMDEXECという名前の
プロトタイプを定義

const : 読み取り専用参照パラメータを指定
options(*nopass) : パラメータの省略可

パラメータ・セパ
レーターは、コロン
(:)
※コンマでない

- SAMPLE5
- ※dcl-prでEXTPROC、EXTPGMキーワードを指定しない場合、デフォルトはEXTPROCキーワード
EXTPROC : プロシージャ、Javaメソッドの呼び出し
EXTPGM : プログラムの呼び出し
 - ※コマンド実行 (QCMDEXEC) API : 1 つのコマンドを実行する IBM 提供のプログラム
 - ※EXTPROC、EXTPGMキーワードのパラメータは大文字と小文字を区別します

実行結果

- DSPJOB OUTPUT(*PRINT)コマンドを実行
- WRKSPLFコマンドの実行結果画面を表示

Chapter3 Variables and procedures

サンプル・プログラム

6) プロシージャ (procedure) : dcl-proc , end-proc

7) プロシージャ・インターフェース () : dcl-pi , end-pi

DSPLY命令コードは最大値52byteという制限があります。ここでは、DSPLY命令コードを使用しないprintのサンプル・プログラムをご紹介します。

```

/if defined(*CRTBNDRPG)
ctl-opt dftactgrp(*no)
      actgrp(*new);
/endif
ctl-opt option(*srcstmt);
dcl-s num int(10) inz(25);

```

CRTBNDRPGコマンドによってコンパイルする場合

活動化グループを指定
ACTGRPパラメーターによって指定された活動化グループに関連付け、静的バインドを許可

```

print('his message is much longer than the 52 '
      + 'characters that DSPLY allows. '
      + 'The value of variable "num" is ' + %char(num));
return;

```

Print
プロ
シー
ジャー
の呼び
出し

コンパイル・リストにソース・ステートメントと同じ行番号を使用することを指定

SAMPLE6

次ページにつづく

Chapter3 Variables and procedures

サンプル・プログラム

6) プロシージャ (procedure) : dcl-proc , end-proc

7) プロシージャ・インターフェース () : dcl-pi , end-pi

前ページのつづき

プロシ
ジャー・イ
ンター
フェースを
指定 (名前
は不要)

```
dcl-proc print;
dcl-pi *n;
msg varchar(5000) const;
end-pi
dcl-pr printf extproc(*dclcase);
template pointer value options(*string);
dummy int(10) value options(*nopass);
end-pr;
dcl-c NEWLINE X'15';

printf(msg + NEWLINE);
end-proc print;
```

printという名前のプロシージャを定義

extproc(*dclcase) : 外部名が、「printf」と同じ名前であることを示す

改行 (X'15') をNEWLINEという名前で定義

実行結果

SAMPLE6

This message is much longer than the 52 characters that DSPLY allows. The value of variable "num" is 25

Chapter4 Introduction to files

- ファイル・ステートメント : DCL-F
 - DCL-Fで開始
 - ファイル名
 - キーワード

DEVICEキーワードを最初のキーワードにする
DISK, PRINTER, WORKSTN
DEVICEキーワードのデフォルトは DISK

- セミコロンで終了

Chapter4 Introduction to files

サンプル・プログラム

1) ファイルのレコードの読み込み

事前準備：最初に以下のコマンドを実行し、ファイルを作成

```
===> DSPOBJD OBJ(QGPL/QRPG*) OBJTYPE(*FILE) OUTPUT(*OUTFILE)
OUTFILE(AAAARPG/RPGTESTF)
```

open :
ファイルのオープン
read :
ファイルの読み込み
close :
ファイルのクローズ

```
dcl-f rpgtestf usropn;
open rpgtestf;
read rpgtestf;
dow not %eof;
  dsply ODOBNM;
  read rpgtestf;
enddo;
close rpgtestf;
return;
```

rpgtestfをファイルとして定義
DEVICEキーワードのデフォルトは DISK=入力用としてオープン
usropn : ファイルの最初のオープンはプログラマーが制御

ファイルの最初のレコードから最後のレコードまで一行ずつ読み込み

実行結果

```
call sample7
DSPLY  QRPGLSRC
DSPLY  QRPGRSRC
```

SAMPLE7

Chapter4 Introduction to files

サンプル・プログラム

2) スプールファイルへの書き出し

```

dcl-c QPRINT_LEN 132;
dcl-f qprint printer(QPRINT_LEN);
dcl-ds qprint_ds len(QPRINT_LEN) end-ds;

*inlr = '1';
qprint_ds = 'Hello';
write qprint qprint_ds;
qprint_ds = 'world';
write qprint qprint_ds;

```

132byteの
データ構造
を定義

`printer` : プリンターへのアクセスを提供し、
印刷出力の様式を記述
数値パラメータで長さを指定し、プログラム記
述ファイルを作成

`write` : プリンターファイル「qprint」に
「qprint_ds」を書き出し

SAMPLE8

実行結果

```

                スプール・ファイルの表示
ファイル . . . . : QPRINT                ページ/行  1/1
制御 . . . . . : _____             桁        1 - 75
検索 . . . . . : _____
* . . . + . . . 1 . . . + . . . 2 . . . + . . . 3 . . . + . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7 . . . +
hello
world

```

Chapter5 Display files

- 表示装置ファイルの使用
 - DCL-Fで開始
 - ファイル名
 - キーワード
- DEVICEキーワード：WORKSTN
- セミコロンで終了

Chapter5 Display files

サンプル・プログラム

1) 表示装置ファイルの使用

事前準備：以下のソースを名前：RPGDSPF、タイプ：DSPFのソースメンバーに貼り付け、コンパイルし、表示装置ファイルを作成します

```
A          R GETNAME
A
A          5 5'What is your name?'
A          NAME          25A I 5 25
A                                     CHECK(LC)
```

以下のコードをソースメンバーに貼り付け、コンパイルし、実行します

exfmt :
WRITE と
READ の組み
合わせ
2つの命令が
同じレコード
様式に対して、
続けて実行さ
れる

```
dcl-f rpgdspf workstn;
exfmt getname;
dsply ('Your name is ' + name)
*inlr = '1';
```

Rpgdspfをファイルとして定義
workstn : 表示装置ファイルであることを
指定

SAMPLE9

Chapter5 Display files

サンプル・プログラム

2) 表示装置ファイルの使用

実行結果

```
What is your name? _____
```

名前を入力

```
What is your name? IBM_____
```

Enterで実行

```
DSPLY Your name is IBM
```

3. 補足情報

(1) フリーフォーマットRPGを使う5つの理由

<https://www.i-cafe.info/column/product/5reasons-of-using-ffrpg>

(2) 【FFRPG】 第一回 とりあえず作ってみよう！

https://www.i-cafe.info/column/serials/ffrpg_intro_1

(3)特集 ILE & FF RPGのススメ ～RPG IIIを脱却し、IBM iのプログラム資産をより有効に活用

<https://www.imagazine.co.jp/ile-ffrpg-susume0000/>

(4)フリー・フォーマット型RPGの入門

https://i5php.jp/wp-content/uploads/2015/12/ibm_i_seminar_oss.pdf

(5) IBM i アプリ開発環境をモダナイズするRational Developer for i

<https://www.ibm.com/blogs/systems/jp-ja/rational-developer-for-i-pdm-perspective/>

(6) フリーフォーマットRPG勉強会資料

https://i5php.jp/wp-content/uploads/2016/08/2016seminar_freeformrpg.pdf

IBM Power Salon ご案内



IBM Powerユーザーのための自由な語り場。
オンラインサロンで、お客様同士、IBMスペシャリストと繋がりませんか
是非、お気軽にご参加ください。

内容：お客様によるDX事例、クラウド活用事例
 IBM i/AIX/Linuxの技術情報、サポート情報
 IBMスペシャリストによるQ&A 他

主催：日本アイ・ビー・エム（株）IBM Power 事業部

参加方法：オンライン開催
 どなたでも参加可能、無料
 事前申し込み不要

ご参加URL：ibm.biz/Powersalon-webex

お問い合わせ：NO1POWER@jp.ibm.com



第7回	5月11日(水) 9:00-10:00	IBM i エム・シーシー食品株式会社 石川様
第8回	6月 8日(水) 9:00-10:00	IBM i World 2022 Q&A
第9回	7月 13日(水) 9:00-10:00	IBM i 日本サニパック様 ご講演
-	8月 休会	
第10回	9月 14日(水) 9:00-10:00	お客様講演予定
第11回	10月 8日(水) 9:00-10:00	
第12回	11月 9日(水) 9:00-10:00	
第13回	12月 14日(水) 9:00-10:00	

IBM Community Japan

IBM i Club ご案内

■ IBM i Clubとは

- 自社システムとしてIBM iをご利用いただいている皆様同士で、各社の工夫や事例を紹介し合ったり、ディスカッションをしていただく場です。
- IBMの技術者も参加し、IBM iの最新情報のご提供や、ディスカッションに入らせていただくこともあります。
- 年5回(予定)、それぞれテーマを変えて開催します。

■ 2021年のテーマ例 (ご参加の皆様からいただいた課題をもとにテーマを選出しました)

- IBM i ユーザーハイブリッドクラウドへの道
- IBM i 人材育成の勘所
- 新技術の活用
- IBM i ユーザーのDX
- IBM i の優位性と今後

■ 開催期間

- 2022年2月24日から2022年12月末(予定)

■ 開催内容

- オンライン(Webex)セッションの開催: 2022年2月24日(木)を第1回とし年間5回(予定)
- 情報共有・ディスカッションの場: コミュニケーションツール(Slack)をご利用いただき、セッション以外の時も情報交換など可能です。

■ ご参加にあたってのお願い・ご注意点

- 守秘義務をお守りください。
- 営業活動を目的としたご参加はお断りいたします。
- IBM i Clubお申し込みには、事前にIBM Community Japanのメンバー登録が必要です。
- 開催期間の途中からのご参加も可能です。



■ 2022年開催予定

*日時・内容が変更になる可能性があります

	日時	実施内容	
1	2/24 (終了)	最新情報	「数字で見るIBM i小辞典」IBM 佐々木
		事例紹介	「利用部門からの要望」にどう対応しているか? トクラス(株)様
		ディスカッション	テーマ: これからのIT部門の役割
2	5月	最新情報	「IBM i 新リリース発表」
		事例紹介	(調整中)
		ディスカッション	テーマ: 人材確保・人材育成
3	7月	最新情報	
		事例紹介	
		ディスカッション	テーマ: 新技術の活用(仮)
4	9月	最新情報	
		事例紹介	
		ディスカッション	テーマ: DX(仮)
5	11月	最新情報	
		事例紹介	
		ディスカッション	

■ コース詳細・お申込み

<https://www.ibm.com/ibm/jp/ja/ibmcommunityjapan-product-community.html>

■ ご参考 昨年(2021年) 開催内容

https://higherlogicdownload.s3.amazonaws.com/IMWUC/2fde9da6-6e7d-43b4-bae3-7f25168bbbd0/UploadedImages/japan/2022/2021_IBM_i_Club.pdf

IBM i 関連情報

IBM i ポータル・サイト

<https://ibm.biz/ibmijapan>

月イチIBM Power情報セミナー「IBM Power Salon」

<https://ibm.biz/power-salon>

IBM i 関連セミナー・イベント

<https://ibm.biz/powerevents-j>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)

<https://ibm.biz/ibmiclubjapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

<https://www.imagazine.co.jp/IBMi/>

IBM i 研修サービス (i-ラーニング社提供)

<https://www.i-learning.jp/service/it/iseriess.html>

IBM Power Systems Virtual Server 情報

<https://ibm.biz/pvsjapan>

IBM i 7.5 技術資料

<https://www.ibm.com/docs/ja/i/7.5>

IBM Power ソフトウェアのダウンロードサイト (ESS)

<https://ibm.biz/powerdownload>

Fix Central (HW・SWのFix情報提供)

<https://www.ibm.com/support/fixcentral/>

IBM My Notifications (IBM IDの登録 [無償] が必要)
「IBM i」「9009-41G」などPTF情報の必要な製品を
選択して登録できます。

<https://www.ibm.com/support/mynotifications>

IBM i 各バージョンのライフサイクル

<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)

<https://www.ibm.com/support/pages/lifecycle/>



ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。