

# IBM i World 2022

IBM i コンテンツ (2022年12月版)

\*2024/9/5に一部修正

\*\*2024/10/8に再修正

## Db2 for i セキュリティ機能の秘密と使い方

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

修正履歴  
2024/9/5

P14のサンプルの構文に誤りがあったので  
下記のように修正しました。

## 誤り

```
CREATE MASK MY_LIB.M1 ON MY_LIB.PERMISSION_TABLE AS
PERMISSION_TABLE
  FOR COLUMN SALARY RETURN
  CASE WHEN ( CURRENT_USER = 'USER4  ')
  THEN
    PERMISSION_TABLE . NAME
  ELSE
    NULL END  ENABLE
```



## 正：修正後

```
CREATE MASK MY_LIB.M1 ON MY_LIB.PERMISSION_TABLE AS
PERMISSION_TABLE
  FOR COLUMN SALARY RETURN
  CASE WHEN ( CURRENT_USER = 'USER4  ')
  THEN
    NULL
  ELSE
    PERMISSION_TABLE . SALARY
  END
  ENABLE
```

修正履歴

2024/10/8

P15の文章に誤りがあったので  
下記のように修正しました。

## 誤り

- 次に、USER4が、SALARYの内容にマスクをかける設定を行います。



## 正：修正後

- 次に、USER4がSALARYの内容を参照できないように、マスクをかける設定を行います。

## 目次

1. Db2 for iのセキュリティ
2. 行・列レベルのアクセス制御（RCAC）の活用
3. フィールド・プロシージャーによるデータベース暗号化
4. 補足情報

# 1. Db2 for iのセキュリティ

## (1) IBM iセキュリティの強化

- ✓ 企業にとって、データは、最も価値ある資産であり、適切に保護される必要があります。
- ✓ Db2 for iのデータ・セキュリティの強化点としては、「カラム暗号化」と「RCAC」です。

パスワード (IBM i 7.5) ・より強力な暗号化方式 (SHA2-512) で暗号化  
 ・パスワードがパスワード規則に適合するかどうかをAPIで確認

権限変更 (IBM i 7.5)

- ・デフォルトの「\*PUBLIC」権限の値を「\*USE」に変更

権限収集 - トレース (IBM i 7.3-)  
 各IDのアクセス状況をトレース・報告

権限収集 - オブジェクト (IBM i 7.4) オブジェクトへのアクセスを精査し、必要な最低限の権限を報告

侵入検知 (IBM i 6.1-)

- ID・PWの連続間違いなど
- ・ GUIにてポリシー設定
- ・ 異常時リアルタイム発報

カラム暗号化 (IBM i 7.1-)

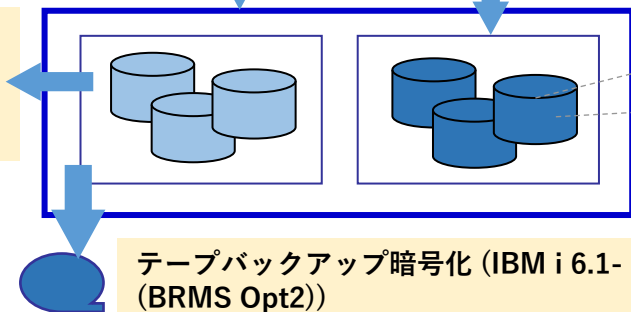
- ・ フィールド・プロシジャーによる実装

監査用追加カラム (IBM i 7.3-) OS  
 が自動でデータ付与

ASP 暗号化 (IBM i 7.1- 有償)

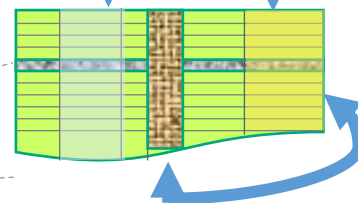
- ・ ディスク空間全体の暗号化

クラウドバック  
 アップ暗号化  
 (IBM i 7.2- 有償)



テープバックアップ暗号化 (IBM i 6.1- 有償 (BRMS Opt2))

- ・ ストレージ機種によりHWレベルの暗号化も可能



RCAC (IBM i 7.2-)

- ・ Row & Column Access Control レコード単位 または項目単位で制御
- ・ 行 (Row) アクセスに認証
- ・ 列 (Column) をマスク

POWER9、Power10による暗号化パフォーマンス向上  
 Power10によるメインメモリ暗号化

## (2) Db2 for i - データ・セキュリティ比較

下記の表は、Db2 for iの主要なセキュリティのテクノロジーを比較したものです。

- フィールド・プロシージャーは、カラム・データ暗号化のための手法として、IBM i 7.1から提供
- RCAC(行・列レベルのアクセス制御) は、IBM i 7.2から提供
- ビュー/論理ファイルは、初期のIBM i (OS/400) から提供

テクノロジー	フィールド・ プロシージャーに よるカラム暗号化	RCAC 列レベル アクセス制御	RCAC 行レベル アクセス制御	ビュー 論理ファイル
ユース・ケース				
サポートされるIBM i リリース	7.1, 7.2以上	7.2以上	7.2以上	all
列データへのアクセス制御	Yes	Yes	No	Yes
行データへのアクセス制御	No	No	Yes	Yes
実装方法	外部プログラム (複雑)	SQL ルール (シンプル)	SQL ルール (シンプル)	DDS /SQL
ソフトウェア・ベンダー対応	<ul style="list-style-type: none"> <li>• Crypt for Db2</li> <li>• Enforce</li> <li>• Assure Encryption など</li> </ul>	現時点ではなし	現時点ではなし	N/A
データ暗号化	Yes	No	No	No
ジャーナル内の暗号化	Yes	No	No	No
適用対象	SQL テーブル	DDS ファイル SQL テーブル	DDS ファイル SQL テーブル	DDS ファイル SQL テーブル

### (3) Db2 for i - データ・セキュリティ

ここからは、下記の主要な2つのデータセキュリティについて活用方法をご紹介します。

- ✓ 行および、列のアクセス制御（RCAC：Row and Column Access Control）
- ✓ データベース暗号化
  - ・ 業務アプリケーション内で、暗号化を実装できる
    - ただし、アプリケーション変更が必要
  - ・ フィールドプロシジャーを使用して、データベースに暗号化を実装できる
    - アプリケーションに対して透過的

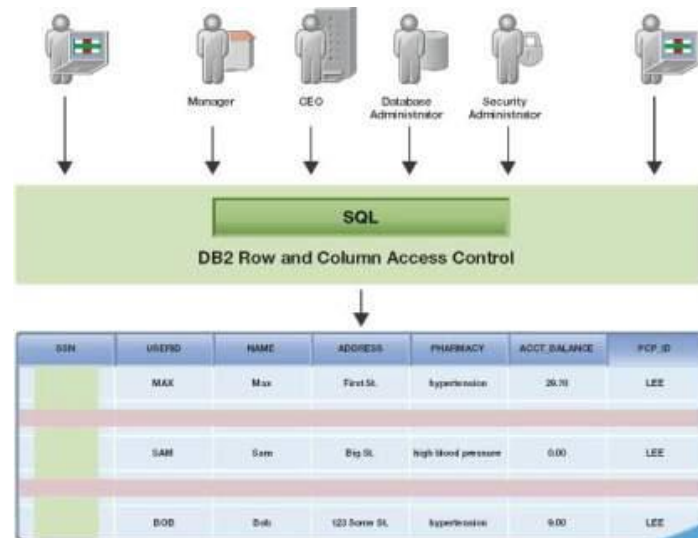
## 2. 行・列レベルのアクセス制御 (RCAC)の活用

### (1) 行・列レベルアクセス制御とは

- ✓ Db2 for iにおける追加のデータ・セキュリティ・レイヤー
- ✓ 表レベルのセキュリティをより完全なものにする
- ✓ 必要なデータのみアクセス可能にするための手法
- ✓ 表へのアクセスを行レベル/列レベル/その両方で制御する
- ✓ 2つのルール・セットを定義可能
  - 行への権限
  - 列のマスク
- ✓ \*ALLOBJの権限をもつ特権ユーザーもこの規則を適用される

### RCAC: Row and Column Access Control

IBM Advanced Data Security for i  
(IBM i Option 47)  
無償フィーチャー





## (2) IBM Advanced Data Security for i ( 5770-SS1 Option47 ) 概説

- ✓ 以下の機能を利用する場合に導入が必須になる
  - CREATE PERMISSION / CREATE MASK (RCAC)
  - RCACを活動化しているファイルをオープンする場合
- ✓ RCAC構成は、テーブル (\*FILE)内に存在し、SQEによって適用される
- ✓ RCACは、オブジェクト権限を置き換えるものではない
  - オブジェクト権限チェックをパスした後に、
    - 行レベル権限により、表示される行が制御される
    - 列マスクは、列データへのアクセスを全面的に、もしくは部分的に制限することができる
- ✓ RCACは、様々なインターフェイスで利用可能
  - Native DB、SQL、RPG、API
- ✓ 行レベル権限は、ビュー/論理ファイルを置き換え可能
- ✓ RCACを管理できるのは、QIBM\_DB\_SECADMユーザーのみ

```
5770SS1 45 暗号化 ASP の有効化
5770SS1 47 IBM ADVANCED DATA SECURITY FOR I
5770SS1 48 DB2 DATA MIRRORING
```

### (3) RCACを試してみよう (1/6)

ここからは実際にRCACを試してみます。

- ✓ 最初に、データベース管理者の権限 ( QIBM\_DB\_SECADM ) をRCAC操作するユーザーに与える必要があります。

- 例. ユーザー (SAWADA)にRCAC権限を与えるコマンド

**CHGFCNUSG FCNID(QIBM\_DB\_SECADM) USER(SAWADA) USAGE(\*ALLOWED)**

```
> CHGFCNUSG FCNID(QIBM_DB_SECADM) USER(SAWADA) USAGE(*ALLOWED)
関数 QIBM_DB_SECADM の使用法情報が変更されました。
```

- 権限が付与されたことは、下記のコマンドで確認できます。

**DSPFCNUSG FCNID(QIBM\_DB\_SECADM)**

```
機能使用状況の表示
機能 ID      : QIBM_DB_SECADM
機能名      : データベース機密保護管理者
記述        : データベース機密保護管理者機能

プロダクト  : QIBM_BASE_OPERATING_SYSTEM
グループ   : QIBM_DB

省略時の権限 : *DENIED
*ALLOBJ 特殊権限 : *NOTUSED

ユーザー   タイプ   使用法   ユーザー   タイプ   使用法
SAWADA     ユーザー *ALLOWED
```

## (参考)

IBM i 7.2から、「セキュリティ管理者とユーザーの分離」の機能が強化されている。OS全体のセキュリティ管理者と別に、データベースの管理を行うユーザーを置きます。

### ✓ IBM i 7.1以前

- ・オブジェクトに対して特権を付与 / 除去する場合、以下のいずれかが必要：
  - オブジェクトの所有者
  - オブジェクトの管理権限
  - \*ALLOBJ 特殊権限
- ・特権を付与する権限がある人は誰でも、これらの特権を要求する操作をまた実行することができる。  
例: SELECT 特権を付与することができるユーザーは、データの参照も許可されている。

### ✓ IBM i 7.2以降

- ・**セキュリティ管理者機能使用 (QIBM\_DB\_SECADM) により、ユーザーにオブジェクトに対する特権を付与 / 除去可能**
- ・登録されたユーザーは、セキュリティの管理は可能だが、データの参照は許可されない  
セキュリティ管理者は自分自身へ許可することは不可
- ・QSECOFR / 他のセキュリティ管理者のみがセキュリティ管理者機能使用の登録が可能

### (3) RCACを試してみよう (2/6)

行レベルのアクセス制御を実施してみます。

➤ 各ユーザー毎に、行への許可を与える例になります。

まず、テスト用のテーブルを準備します。

1. テスト用のスキーマの作成: `CREATE SCHEMA MY_LIB (MY_LIBという名前のスキーマ)`
2. テスト用のテーブルの作成: `NAME`フィールドはユーザー名、`SALARY`フィールドを給与額にします。  
`CREATE TABLE MY_LIB.PERMISSION_TABLE (NAME CHAR (5 ), SALARY DECIMAL(9,2))`
3. 下記のように、データを挿入します。

```
行      ....+....1....+....2
      NAME      SALARY
000001 USER1    1,000,000.00
000002 USER2    2,000,000.00
000003 USER3    3,333,333.33
***** ***** 報告書の終わり *****
```

### (3) RCACを試してみよう (3/6)

ここでは、下記のように、行レベルのアクセス制御を実施します。

- 操作するユーザーは、前頁のRCAC権限のあるユーザーで実施します。
- 各ユーザー毎に、行への許可を与える例になります。

```
CREATE PERMISSION MY_LIB.P1 ON MY_LIB.PERMISSION_TABLE
FOR ROWS WHERE
CURRENT_USER = 'USER1 '
ENFORCED FOR ALL ACCESS ENABLE
```

```
CREATE PERMISSION MY_LIB.P2 ON MY_LIB.PERMISSION_TABLE
FOR ROWS WHERE NAME = 'USER2 ' and CURRENT_USER = 'USER2 '
ENFORCED FOR ALL ACCESS ENABLE
```

```
CREATE PERMISSION MY_LIB.P3 ON MY_LIB.PERMISSION_TABLE
FOR ROWS WHERE NAME = 'USER3 ' and CURRENT_USER = 'USER3 '
ENFORCED FOR ALL ACCESS ENABLE
```

最後に、RCACを有効にするために、下記のコマンドを実行する

```
ALTER TABLE MY_LIB.PERMISSION_TABLE ACTIVATE ROW ACCESS CONTROL
```

もし無効にしたい場合は、

```
ALTER TABLE MY_LIB.PERMISSION_TABLE DEACTIVATE ROW ACCESS CONTROL
```

USER1は全てのレコードにアクセス可能にする ( USER1はマネジャーなので社員全員の給与が表示できる)

USER2は、NAMEが"USER2"のみアクセス可能  
( USER2は自分の給与のみ表示)

USER3はNAMEが"USER3"のみアクセス可能  
( USER3は自分の給与のみ表示)

### (3) RCACを試してみよう (4/6)

行レベルのアクセス制御が有効になっているか、各ユーザーで確認します。

1. 各ユーザーでサインオンして、QUERY for i で照会して、結果の確認を実施

```
SELECT * FROM MY_LIB.PERMISSION_TABLE
```

USER1は、全てのレコードにアクセス可能

USER2は、NAMEが"USER2"のみアクセス可能

USER3は、NAMEが"USER3"のみアクセス可能

```
NAME          SALARY
000001 USER1  1,000,000.00
000002 USER2  2,000,000.00
000003 USER3  3,333,333.33
***** ***** 報告書の終わり
```

```
NAME          SALARY
000001 USER2  2,000,000.00
***** ***** 報告書の終わり
```

```
NAME          SALARY
000001 USER3  3,333,333.33
***** ***** 報告書の終わり
```

### (3) RCACを試してみよう (5/6)

次に、列レベルのアクセス制御を実施してみましょう。

- 操作するユーザーは、前頁のRCAC権限のあるユーザー（QIBM\_DB\_SECADMをもつ）で実施します。
- SALARYの列にマスクをかけます。

1. まずは、USER4にすべての表の完全なアクセス権限を与えます。

```
CREATE PERMISSION MY_LIB.P5 ON MY_LIB.PERMISSION_TABLE
FOR ROWS WHERE
CURRENT_USER = 'USER4 '
ENFORCED FOR ALL ACCESS ENABLE
```

2. 次に、USER4がSALARYの内容を参照できないように、マスクをかける設定を行います。

```
CREATE MASK MY_LIB.M1 ON MY_LIB.PERMISSION_TABLE AS PERMISSION_TABLE
FOR COLUMN SALARY RETURN
CASE WHEN ( CURRENT_USER = 'USER4 ' )
THEN
NULL
ELSE
PERMISSION_TABLE . SALARY
END
ENABLE
```

USER4はSALARYの内容を参照  
することができなくなります。

3. 最後に、この設定を有効にします。

```
ALTER TABLE MY_LIB.PERMISSION_TABLE ACTIVATE COLUMN ACCESS CONTROL
```

### (3) RCACを試してみよう (6/6)

- ✓ USER4が表にアクセスすると、下記のようにSALARYがマスキングされて表示されます。

USER4はSALARYの内容を  
参照することができない

```
NAME          SALARY
000001  USER1  ++++++++
000002  USER2  ++++++++
000003  USER3  ++++++++
***** *****  報告書の終わり  **
```



## (参考1)

物理ファイルに、RCAC が活動化されているかどうかを、下記の2つの方法で確認できます。

➤ QIBM\_DB\_SECADM権限のあるユーザーでの実行が必要です。

1) DSPOBJAUTコマンドで確認できます。(行またはカラム・アクセス制御が活動中)

```

オブジェクト権限表示

オブジェクト . . . . . PERM100001 所有者 . . . . . SAWADA
ライブラリー . . . . . MY_LIB      1次グループ . . . . . *NONE
オブジェクトタイプ . . . . . *FILE   ASP 装置 . . . . . *SYSDAS
行またはカラム・アクセス制御 . . . . . :      :      :      :      :      :      :      :      :      :      :      :      :
権限リストによって保護されたオブジェクト . . . . . :      :      :      :      :      :      :      :      :      :      :      :      :
                                                    *NONE

ユーザー      グループ      オブジェクト
*PUBLIC
SAWADA          *EXCLUDE
                *ALL
  
```

2) ACSで「スキーマ」から「表」→「テーブル名」→「定義」を選択すると、確認できます。



## (参考2)

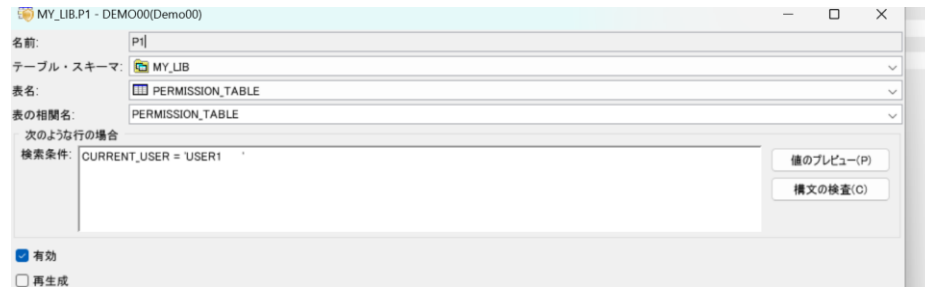
物理ファイルにどのようなRCACが設定されているかを確認する方法は下記になります。

➤ 行のアクセス制御については、

ACSで、「スキーマ」→「行の許可」を選択すると、設定を確認できます。

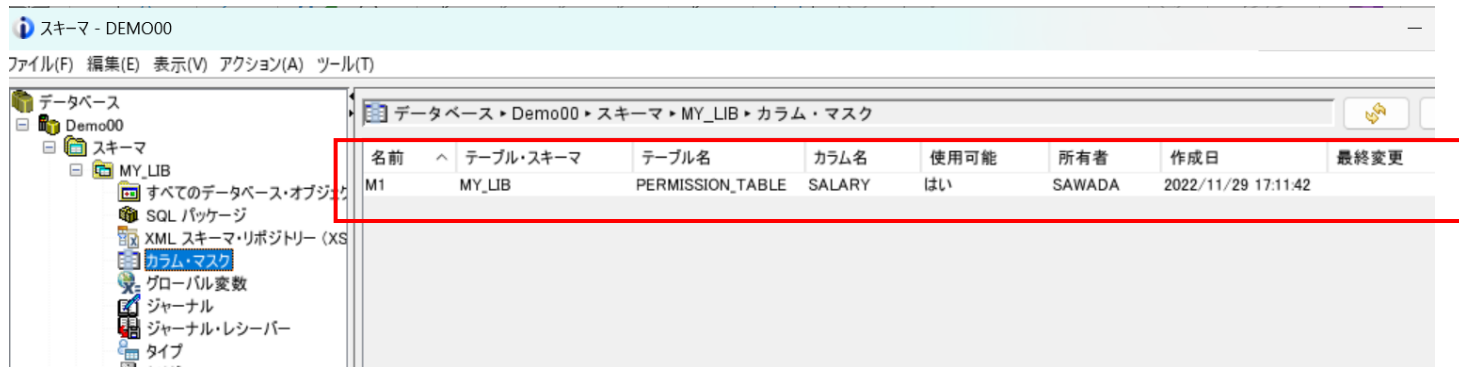


詳細は、例えばP1の定義については、下記のように確認できます。

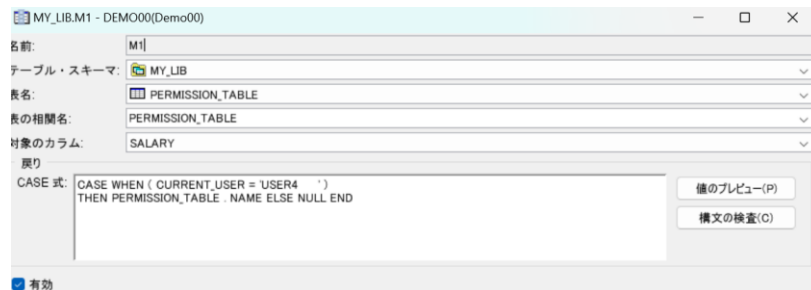


## (参考3)

- 列のアクセス制御については、「スキーマ」→「カラム・マスク」を選択すると、設定を確認できます。



詳細は、例えばM1の定義については、下記のように確認できます。



### 3. フィールドプロシージャによるデータベース暗号化

#### (1) データベース暗号化の概念

- ・ もう一つのDb2 for iのセキュリティ対策として、列の暗号化があります。
- ・ Db2 for iは、列の挿入/更新/読み取り時に呼び出されるフィールド・プロシージャを提供しています。（暗号化を有効にするために追加サポートされました）
- ・ 暗号化/復号化は、選択した暗号化APIを使用して実装する必要があります。（下図参照）

#### エンコード (暗号化)

データの取得

暗号化APIを使用した暗号化の実効

暗号化されたデータ保存

暗号化されたデータを含むように設定

鍵ストア  
・ ファイル

#### デコーディング (復号化)

暗号化されたデータの取得

オプションで許可検査を実行

復号化を実行するための暗号化API使用

復号化データの保管

復号化されたデータを含むように設定

## (参考) IBM iでの暗号化サポート

- ✓ IBM iでの主要な暗号化手法には、下記のようなものがあります。

当資料では、「暗号化サービスAPI」を利用した列の暗号化について解説します。

暗号化の手法	プログラム変更	使用目的	備考
Db2 API	必要	Db2 for i データ暗号化	SQL関数利用
暗号化サービスAPI	必要	Db2 for i データ暗号化	フィールド・プロシージャと組み合わせる
BRMS*	不要	バックアップ暗号化	システム(OS)以外のオブジェクト全体
テープ装置LTO	不要	バックアップ暗号化	LTOテープ装置の機能なので、暗号化SW組み込む必要あり
ASPの暗号化	不要	バックアップ暗号化	ディスク暗号化

## (2) フィールド・プロシージャー -暗号化拡張機能

- ✓ Db2 for iの拡張暗号化機能について
  - フィールド単位で、トリガーの設定が可能
    - ・暗号化ロジックと組み合わせることで、フィールド単位の暗号化が可能
    - ・フィールド・プロシージャーは、暗号化専用の機能ではない
- ✓ 利用方法
  - CREATE TABLE / ALTER TABLEで宣言する際に、下記の3つの処理を1つのプログラム（フィールド・プロシージャー）に書いておくことが必須
    - ・ CREATE TABLE / ALTER TABLE で宣言される際の処理
    - ・ エンコーディング【暗号化】の処理
    - ・ デコーディング【復号化】の処理
  - ILE C/RPG/COBOLのインクルードファイルが、QSYSINC/H/SQLFPで提供
  - トリガーが作動するタイミングの主なものは、下記の通り
    - ・ アプリケーション、SQLからのアクセス
    - ・ ネイティブレコードレベルのアクセス
    - ・ SELECT、RUNQRY、DSPPFM、CPYF、バイナリーFTPなど

## (2) フィールド・プロシージャー -暗号化拡張機能 ( 続き )

- ✓ フィールド・プロシージャーのコーディング手順について
  - パラメータの渡し方や、詳しいコーディングの作法は、下記のマニュアルに記載されています。ILE RPGとILE Cのサンプルプログラムも提供されています。

<https://www.ibm.com/docs/ja/i/7.5?topic=procedures-example-field-procedure-program>

- パラメータとして、「機能コード」に宣言時には、[8]、書き込み時には[0]、読み取り時には[4]が入ってプログラムが呼び出されるので、それに応じたコーディングを行う
- エンコードとデコードで、データが確実に元に戻るようコーディングするのはユーザーの責任です。

### (3) フィールドプロシジャーサンプルプログラムのご紹介 (1/5)

サンプルプログラムプログラムとして、ILE Cでのプログラム例をご紹介します。

```
***** Beginning of data *****
0001.00 #include "string.h" 100331
0002.00 #include <stdio.h> 100331
0003.00 #include <QSYSINC/H/SQLFP> 100331
0004.00 100331
0005.00 void encode(char *in, char *out, long length); 100331
0006.00 void reverse(char *in, char *out, long length); 100331
0007.00 100331
0008.00 main(int argc, void *argv[]) 100331
0009.00 { 100331
0010.00     short *funccode = argv[1]; 100331
0011.00     sqlfpFieldProcedureParameterList_T *optionalParms = argv[2]; 100331
0012.00     char *sqlstate = argv[7]; 100331
0013.00     struct sqlfpMessageText_t *msgtext = argv[8]; 100331
0014.00     int bytesToProcess; 100331
0015.00     sqlfpOptionalParameterValueDescriptor_T *optionalParmPtr; 100331
0016.00 100331
0017.00     if (optionalParms->sqlfpNumberOfOptionalParms != 1) 100331
0018.00     { 100331
0019.00         memcpy(sqlstate, "38001", 5); 100331
0020.00         return; 100331
0021.00     } 100331
0022.00 100331
0023.00     optionalParmPtr = (void *)&(optionalParms->sqlfpParmList); 100331
0024.00     bytesToProcess = *((int *)&optionalParmPtr->sqlfpParmData); 100331
0025.00 100331
0026.00     if (*funccode == 8) /* create time */ 100331
0027.00     { 100331
0028.00         struct sqlfpParameterDescription_t *inDataType = argv[3]; 100331
0029.00         struct sqlfpParameterDescription_t *outDataType = argv[5]; 100331
0030.00         if (inDataType->sqlfpSqlType != 452 && 100331
0031.00             inDataType->sqlfpSqlType != 453) /* only support fixed length char */ 100331
0032.00         { 100331
0033.00             memcpy(sqlstate, "38002", 5); 100331
0034.00             return; 100331
0035.00         } 100331
0036.00         memcpy(outDataType, inDataType, sizeof(sqlfpParameterDescription_T)); 100331
0037.00     } 100331
```



## (3) フィールドプロシジャーサンプルプログラムのご紹介 (2/5)

(続き)

```
0038.00                                     100331
0039.00     else if (*funccode == 0) /* encode */ 100331
0040.00     {                                     100331
0041.00         char *decodedData = argv[4];         100331
0042.00         char *encodedData = argv[6];         100331
0043.00         encode(decodedData, encodedData, bytesToProcess); 100331
0044.00     }                                         100331
0045.00     else if (*funccode == 4) /* decode */     100331
0046.00     {                                         100331
0047.00         char *decodedData = argv[4];         100331
0048.00         char *encodedData = argv[6];         100331
0049.00         reverse(encodedData, decodedData, bytesToProcess); 100331
0050.00     }                                         100331
0051.00     else /* unsupported option -- error */     100331
0052.00         memcpy(sqlstate, "38003", 5);        100331
0053.00 }                                             100331
0054.00                                             100331
0055.00 void encode(char *in, char *out, long length) 100331
0056.00 {                                             100331
0057.00     int i;                                     100331
0058.00     for (i=0;i<length; ++i) {                 100331
0059.00         out[length - (i+1)] = in[i];         100331
0061.00     }                                         100331
0064.00 }                                             100331
0065.00 void reverse(char *in, char *out, long length) 100331
0066.00 {                                             100331
0067.00     int i;                                     100331
0068.00     for (i=0;i<length; ++i) {                 100331
0069.00         out[length - (i+1)] = in[i];         100331
0071.00     }                                         100331
0074.00 }                                             100331
***** End of data *****
```

### (3) フィールドプロシジャーサンプルプログラムのご紹介 (3/5)

- ✓ フィールドプロシジャーのコンパイル手順
  - コンパイル時には、活動化グループ ACTGRP(\*CALLER)を指定
    - **CRTCMOD MODULE(SAWADALIB/FPROC) SRCFILE(SAWADALIB/QCSRC)**  
モジュールFPROCは22/12/05の15:52:55にライブラリーSAWADALIBに作成されました。
    - **CRTPGM PGM(SAWADALIB/FPROC) ACTGRP(\*CALLER)**  
プログラムFPROCがライブラリーSAWADALIBに作成された。
- ✓ フィールドプロシジャーを指定したテーブルの作成
  - SAWADALIB/FPROCが、フィールドプロシジャーのILEプログラム(10) はパラメータ。テーブルFPTESTを作成します。
    - **CREATE TABLE MY\_LIB.FPTEST(id char(2) , name char(10) fieldproc SAWADALIB.fproc(10));**
- 考慮点
  - テーブル作成時に、フィールドプロシジャーが実行されるため、事前にフィールドプロシジャーのコンパイルが完了していることが必須
  - ストアードプロシジャーとは異なり、CREATE PROCEDURE等などは必要なし

### (3) フィールドプロシジャーサンプルプログラムのご紹介 (4/5)

- ✓ 作成したテーブルへのデータの更新と表示を行います。
  - ファイルを**READ/WRITE/UPDATE**する度に、フィールドプロシジャーが実行される
  - エンコード/デコードが非対称なフィールド・プロシジャーを意図的に作成したところ、正しく暗号化されていたことは確認できました。
  - フィールド・プロシジャーを削除すると、データを読み書きが不可になります。

下記のように、2レコード追加しました。

- `INSERT INTO MY_LIB.FPTEST VALUES('01', 'ABCDEFGHIJ')`  
ステートメントは正常に実行されました (201 ミリ秒)
- `INSERT INTO MY_LIB.FPTEST VALUES('02', 'JIHGFEDCBA')`  
ステートメントは正常に実行されました (82 ミリ秒)

### (3) フィールドプロシジャーサンプルプログラムのご紹介 (5/5)

- ✓ 作成したテーブルへのデータの更新と表示 (続き)
  - テストした限りは暗号化された値を見る術はなし。  
ユーザープログラムが読み取る前にデコードされるため、  
暗号化されたデータの確認は不可です。

```
1 SELECT * FROM MY_LIB.FPTEST;|
```

ID	NAME
01	ABCDEFGHIJ
02	JHGFEDCBA

## 4. 補足情報

(1) Db2 for i 概要資料

<https://www.ibm.com/support/pages/db2-ibm-i>

(2) Db2 for i テクノロジーの更新

<https://www.ibm.com/support/pages/node/1116645/>

(3) Db2 for i セキュリティの機能拡張

<https://www.ibm.com/support/pages/node/1116597>

(4) Db2 Row & Column Access Control (RCAC) on IBM i Redbook

<https://www.redbooks.ibm.com/redpapers/pdfs/redp5110.pdf>

(5) Db2 Row & Column Access Control (RCAC) on IBM i Youtube 講義

<https://www.youtube.com/watch?v=rZrWuarqDPU>

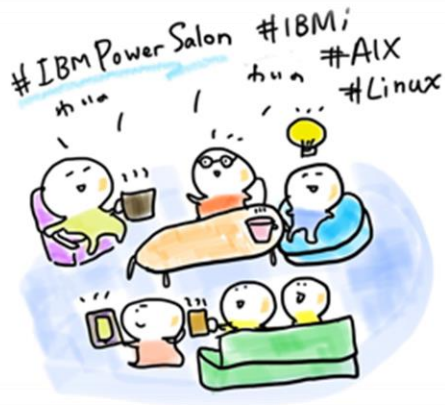
(6) フィールド・プロシージャを使用する列レベルの暗号化の提供

<https://www.ibm.com/docs/ja/i/7.5?topic=considerations-field-procedures>

(7) フィールド・プロシージャの暗号化プログラムの例

<https://www.ibm.com/docs/ja/i/7.5?topic=procedures-example-field-procedure-program>

# IBM Power Salon のご案内



IBM Powerユーザーのための自由な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

毎月第2水曜日 朝9時開店！

## ■ 12月: お客様講演 :開催済み■

日時： 2022年12月14日 (水) 9:00-10:00

ゲスト講師: 株式会社 松沢書店様

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](https://ibm.biz/Powersalon-webex)

お問い合わせ：[NO1POWER@jp.ibm.com](mailto:NO1POWER@jp.ibm.com)



IBM Power Salonの詳細はこちら  
→<https://ibm.biz/power-salon>

## IBM Power Salon バックナンバーのご案内

再生回数  
1,600回 突破!!

過去の講演はいつでもどこでも視聴可能、資料はダウンロードできます

	開催日	カテゴリー	テーマ	動画・資料リンク
第1回	2021/11/10	IBM i World	IBM Power Virtual Server	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第2回	2021/12/8	IBM i World	プログラム言語	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第3回	2022/1/12	AIX	35歳になったAIXからのご挨拶	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第4回	2022/2/9	お客様講演	IBM iで「DX推進」のその後 <「2025年の崖」問題を解決しながら「DX推進」> ゲスト講師: 株式会社ニイタカ 川端 様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第5回	2022/3/9	パートナー様講演	Legacy with DX - IBM iのデジタル変革実装の現場から ゲスト講師: 株式会社オムニサイエンス 下野 様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第6回	2022/4/3	セキュリティー	緊急特番! 脅威のサイバー攻撃にはこう備えよ	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第7回	2022/5/11	お客様講演	社運をかけた大プロジェクト: IBM iと 奇跡のスパイス ゲスト講師: MCC食品株式会社 石川 様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第8回	2022/6/8	IBM i World	IBM i World 2022: あなたの疑問全てお答えします	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第9回	2022/7/13	お客様講演	#サニパック流DXジャーニー、答えは「現場」が持っていた! ゲスト 講師: 日本サニパック株式会社 宇野 様, ベル・データ株式会社 田村 様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第10回	2022/9/14	パートナー様講演	Fujitsu Enterprise PostgresとIBM Powerで創る明日への希望と自由の世界 ゲスト講師: 富士通株式会社 石杜様、椎木様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第11回	2022/10/12	お客様講演	患者様の命のために - 保険科学西日本のDXを支えるIBM i ゲスト講師: 株式会社 保険科学西日本 太城 様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>
第12回	2022/11/9	お客様講演	立命館とIBM i ~これまでの歩みとこれからの展望~ ゲスト講師: 立命館大学 師井 様	<a href="#">・動画リプレイ</a> <a href="#">・資料</a>



# 大反響！IBM Power Salon お客様ご講演回

## 第4回 IBM Power Salon (2/9) IBM i

お題：  
**IBM iで「DX推進」のその後**  
＜「2025年の崖」問題を解決しながら「DX推進」＞



株式会社 ニイタカ 情報システム部 部長 川端氏ご登壇!

IBM i World 2021で大きな反響を呼んだ、株式会社 ニイタカの挑戦。

- ・ 中長期情報システム戦略は、実行段階へ、プロジェクトを通じて見えてきたことは？
- ・ RPGIV FFへのモダン化における、課題と副産物はこれだった！

[→動画リプレイ](#) | [→資料](#)

## 第7回 IBM Power Salon (2022/5/11) IBM i

お題：  
**50年振りの新工場建設、社運をかけた大プロジェクト!!** エム・シーシー食品株式会社は、どんな**奇跡のスパイス**を使ったのか？



エム・シーシー食品株式会社  
情報システムグループリーダー  
石川 真法氏 ご登壇!!

[→動画リプレイ](#) | [→資料](#)

## 第9回 IBM Power Salon (2022/7/13) IBM i

**サニパック流 DX ジャーニー、**  
答えは「現場」が持っていた!



日本サニパック株式会社  
SCMグループ  
デジタルトランスフォーメーション推進部 部長  
宇野 康典氏



ベル・データ株式会社  
東日本営業統括部 第2営業部  
サブマネージャー  
田村 浩和

[→動画リプレイ](#) | [→資料](#)

## 第11回 IBM Power Salon (2022/10/12) IBM i

**患者様の命のために -**  
**保健科学西日本のDXを支えるIBM i**



株式会社 保健科学西日本  
管理部 情報管理課 次長  
太城 義雄氏

[→動画リプレイ](#) | [→資料](#)

## 第12回 IBM Power Salon (2022/11/9) IBM i

**立命館とIBM i ~これまでの歩みとこれからの展望~**

立命館大学のこれまでの取り組み内容とこれからの展望と、  
IBM i/オープン系のハイブリッド技術者育成



立命館大学  
情報システム部 情報システム課  
師井 学氏

[→動画リプレイ](#) | [→資料](#)



過去の講演はいつでもどこでも視聴可能、  
資料はダウンロードできます



## 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](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>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

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

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>

IBM i 研修サービス (i-ラーニング社提供)

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

IBM Power Systems Virtual Server 情報

<https://ibm.biz/pvsjapan>

IBM i 情報サイト iWorld

<https://ibm.biz/iworldweb>

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](http://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。