

松尾 悦郎

株式会社ミガロ

システム事業部 システム2課

# JACi400を使って RPGでWeb画面を制御する方法

RPG 開発者のために  
RPG プログラムで、Web 画面を制御する方法を紹介する。  
ラジオボタンについてと  
画面遷移時の値の受け渡し方法を説明したい。



略歴  
1979年6月16日生  
2002年広島大学 理学部卒  
2006年株式会社ミガロ入社  
2006年6月システム事業部配属

現在の仕事内容  
SE。主に JACi400 を使った Web アプリケーションの開発を担当しており、システムの要件定義から納品・フォローまで行っている。

- はじめに
- ラジオボタン
- 第2画面で選択した値を、第1画面に戻す
- 最後に

## はじめに

昨今の世界的なインターネットの普及により、アプリケーションの Web 化はますます進んでいる。IBM i で稼働するアプリケーションも例外ではなく、Web 化のニーズが高まってきている。

これまで RPG プログラムをメインに、IBM i 上で稼働するアプリケーションの開発を行ってきた私にとっても、Web 化は避けては通れないものとなってきた。Java などの RPG 以外の言語を習得する必要性を感じているが、敷居が高くなかなか思うようにはいかず、少なからず歯がゆい思いもしていた。

### JACi400

そんなときに会ったのが、RPG プログラムで、Web アプリケーションを開発できる「JACi400」である。

Web アプリケーションを開発するには、Java などの言語を覚えなさいといっていたが、なんと JACi400 では、これまで私が身につけてきた知識

やノウハウをそのまま活かして、RPG で簡単に Web アプリケーションを開発できるのである。

しかも、HTML 画面との連携は、予め作成される RPG ソースに記述される。私たちは、ただデータの抽出やデータベースの更新を RPG ソースに記述すればよいし、DSPF を制御するように、HTML 画面も RPG プログラムで制御できる。従来の開発と同じように開発が可能なのである。

しかし、画面が DSPF から HTML になることで、画面で使用する部品が違うため、若干制御の方法を覚える必要がある。例えば、コンボボックスやラジオボタンがそうだ。これらは、いずれも Web 画面では必須の部品と言っても過言ではないだろう。

また、画面の遷移も異なってくる。こちらは、例えば、DSPF であれば 1 つのプログラムに複数の画面を持てるが、HTML だと 1 つしか持てない、といったことである。

今回は、RPG 開発者のために、RPG

プログラムで Web 画面を制御する方法のうち、ラジオボタンと、画面遷移時の値の受け渡し方法を紹介したいと思う。

## ラジオボタン

ラジオボタンは、Web 画面ではよく使用される部品である。いくつかの選択肢から 1 つを選ぶものであるが、これを RPG プログラムで制御できるのだ。

### ①画面で選択したものを、RPG プログラムで受け取る方法

画面で選択した値を RPG に受け取るにはどうしたらいいのだろうか。

まずは、HTML にラジオボタンを設定し、それぞれの項目に対して id 属性と、value 属性を定義する。id 属性は全て同じ値を指定するが（異なる値を指定すると正しく動作しないので注意）、value 属性はそれぞれに異なる値を指定する必要がある。

ここでは、東京・名古屋・大阪をラジオボタンで選択するものを作ろう。【図 1】

図1



図2

```

<td bgcolor="#ffffff" align="center">
  <div class="BLU11">
    <p><br>
      ++ ラジオボタングループ ++
    <br>
    <br>
    </p>
  </div>
  <p>
    <label>
      <input name="RBG01" type="radio" value="R1" checked id="RBG01">東京</label>
    <label>
      <input type="radio" name="RBG01" value="R2" id="RBG01">名古屋</label>
    <label>
      <input type="radio" name="RBG01" value="R3" id="RBG01">大阪</label>
  </p>

```

IRBG01には、画面で選択された値のHTMLで指定した"value値"がセットされます。  
 東京=R1  
 名古屋=R2  
 大阪=R3

図3

```

ラジオボタンで選択された値により、処理をコーディングするRPGIは以下ようになります。
0092.00 C050 * <YOURCODE>
0093.00 ---- * CHECK ACTION CODE JCACTN HERE AND PROCESS.
0094.00 C          JCACTN   IFEQ 'ED'
0095.00 C          GOTO ENDPGM
0096.00 C          ENDIF
0097.00 C**
0098.00 C          JCACTN   IFEQ 'ET'
0098.01 C**+ ラジオ ボタン
0098.02 C*** 東京が選択されたときの処理
0099.00 C          IRBG01   IFEQ 'R1'
0100.00 C          ENDIF
0100.01 C*** 名古屋が選択されたときの処理
0100.02 C          IRBG01   IFEQ 'R2'
0100.03 C          ENDIF
0100.04 C*** 大阪が選択されたときの処理
0100.05 C          IRBG01   IFEQ 'R3'
0100.06 C          ENDIF
0100.07 C**
0100.08 C          ENDIF
0101.00 C**
0102.00 C          GOTO T100
0103.00 ---- * YOUR CODE
0104.00 C050 * </YOURCODE>

```

id 属性は全て“RBG01”と指定する。value 属性は、東京“R1”、名古屋“R2”、大阪“R3”とそれぞれ指定する。【図2】

これで、画面からいずれかの都市を選択すると、id = RBG01 に HTML で指定したそれぞれの value 値がセットされるようになる。例えば、画面で東京を選択した場合は、“R1”が RPG プログラムに渡される。

HTML の設定が正しくでき、さらに画面で実行処理を行うと、JACi400 が RPG プログラムを呼び出し、RPG プログラムで id = RBG01 の値を受け取ることができるのだ。あとは、それぞれの値別に、処理を記述すればよい。【図3】

## ②ラジオボタンで選択した値を保持する方法

HTML を触ったことがある人ならば、ラジオボタンの初期値は、HTML ソースで“checked”を指定した項目になることを知っているだろう。そのため、画面で値を選択して実行したあと、次に画面表示するときには選択した値を保持せずに、常に初期値として指定した項目に戻ってしまう。

これでは都合が悪い。照会画面では自分が指定した内容は保持していたいし、前の画面に戻ったときも選択した値を保持していたいものだ。

ではどうするか。実はこれも RPG プログラムで対応できるのだ。選択時に受け取った値を、そのまま次の画面にも送ってあげればよい。

## 画面に送るための手順

それでは、画面に送るための手順を順に説明しよう。

### ・JACi400Designer での設定

最初に、JACi400Designer で設定を行う。

ラジオボタン id = RBG01 の設定で、Usage 項目を“Both”にし、Additional 項目にラジオボタンの項目数・value 属性の長さ・画面に表示する記述の長さを指定する。これで、ラジオボタンが入出力フィールドとして定義される。

ここでは、ラジオボタンの項目数を“3”・value 属性の長さを“2”・画面に表示する記述の長さを“10”に設定している。

### ・生成される DDS

JACi400Designer で各フィールドを設定したあと、JACi400 で DDS を作成する。

今回の例では、図4のような、INPUT と OUTPUT タイプの DDS が作成される。INPUT タイプは、画面から選択された値を受け取るためのものである。OUTPUT タイプのフィールド O01002 は、ラジオボタンの項目数を指定する。フィールド ORBBG01 には、選択フラグ・value 属性の値・画面に表示する記述内容を、ラジオボタンの項目数だけ指定する。【図4】

### ・RPG プログラムのコーディング

DDS が作成されると、いよいよ RPG プログラムのコーディングをしなければならない。

まずは各定義を行う。ラジオボタン情報用の配列を宣言し、同じくラジオボタン情報用の DATA STRUCTURE を定義する。配列のレコード長は 39 バイトである。これは、選択フラグ 1 バイト・value 属性の値 2 バイト・画面に表示する記述内容の 10 バイトをあわせた 13 バイトに対して、ラジオボタンの項目数 3 つ分の合計値である。

また、DATA STRUCTURE は、各項目別に選択フラグ・value 値・表示する記述内容を定義しておく。そして、配列の要素も“1R1 東京 0R2 名古屋 0R3 大阪”と内部で定義しておく。

## データセットの処理

以上の定義ができれば、次にデータセットの処理を行う。

まず、配列の要素“1R1 東京 0R2 名古屋 0R3 大阪”を DATA STRUCTURE にセットし、ラジオボタンの項目数“3”をフィールド O01002 にセットする。

このまま処理を終了すると、画面にはラジオボタンが表示され、“東京”にチェックが入った状態になる。しかしこれだと、常に画面のラジオボタンは東京が選択された状態になってしまう。

では、名古屋や大阪を選択状態にするにはどうしたらよいか、それは、画面で選択状態にする項目の選択フラグに“1”を、その他には“0”をセットすればよい。つまり、名古屋であれば、DATA STRUCTURE で定義した RD21 に“1”

をセットし、その他の RD11 と RD31 に“0”をセットするのである。同様に、大阪を選択状態にするには、RD31 に“1”をセットすればよい。

ここまでできたら、最後に DATA STRUCTURE の値をフィールド ORBG01 に転送して RPG から画面にデータを送信して完了となる。【図5】

これで、ラジオボタンで選択した値を、次回画面表示時に保持することができる。

実は、これと同じ方法でコンボボックスも制御でき、とても簡単である。ぜひ覚えていただきたい。

## 第2画面で選択した値を、第1画面に戻す

ここから画面遷移の方法を紹介しよう。

RPG 開発者にとっては、Web アプリケーションの画面遷移も、どのようにすればよいのかピンと来ないのではないだろうか。なぜなら、Web アプリケーションでは多くの場合、画面処理ごとに通信は切断されるからである。操作するユーザーからの入力を待ち続けているわけではない。

一方、従来の 5250 では、RPG プログラムが次の処理を待機している。それゆえ、制御方法が違い、どのようにしたらよいかわからないのである。

## JACi400の場合

JACi400 の場合は、メニューから画面が呼び出されると、JACi400 サブシステム配下にジョブが作成される。HTML を出力しても、CGI は終了するが、RPG プログラムは待機した状態になるのである。つまり、従来どおりの手法で開発ができるのだ。

また、アプリケーション構成は、HTML 画面 1 つに対して、RPG プログラム 1 つである。したがって、ある画面から別の画面に遷移するときは、もとの RPG プログラムから別の画面用の RPG プログラムを実行することになる。画面間でデータを受け渡しする場合には、何らかの手段で受け渡ししなければならない。

とはいえ、これまでの 5250 での開発と比べても、特別な手法ではないので安心していただきたい。

図4

```
JCTST3011
***** データの始め *****
0001.00 * HTML FILE: JCTST3.HTML
0002.00 A R JCTST3
0003.00 A IRBG01 00002
***** データの終わり *****

JCTST3010
***** データの始め *****
0001.00 * HTML FILE: JCTST3.HTML
0002.00 A R JCTST3
0003.00 A 001002 4S 0 TEXT('NUMBER OF VALUES ORBG
0004.00 A ORBG01 00039
***** データの終わり *****
```

図5

```
Radioボタン情報用の配列を宣言
0018.00 E010 * <YOURCODE>
0019.00 ---> * YOUR ARRAYS
0019.01 E @RD 1 1 39
0020.00 E010 * </YOURCODE>

Radioボタン情報用のDATA STRUCTURES
0025.00 * DATA STRUCTURES
0026.00 I010 * <YOURCODE>
0027.00 ---> * YOUR INPUT SPECIFICATIONS
0027.01 I DS
0027.02 I 1 39 RD
0027.03 I 1 1 RD11
0027.04 I 2 3 RD12
0027.05 I 4 13 RD13
0027.06 I 14 14 RD21
0027.07 I 15 16 RD22
0027.08 I 17 26 RD23
0027.09 I 27 27 RD31
0027.10 I 28 29 RD32
0027.11 I 30 39 RD33

0093.00 C050 * <YOURCODE>
0094.00 ---> * CHECK ACTION CODE JCACTN HERE AND PROCESS.
0095.00 C JCACTN IFEQ 'ED'
0096.00 C GOTO ENDPGM
0097.00 C ENDIF
0098.00 C**
0099.00 C JCACTN IFEQ 'ET'
0099.01 C** ラジオボタンワークリセット
0099.02 C CLEARRD
0099.03 C** ラジオボタン項目とアイテム数セット
0099.04 C MOVEL@RD,1 RD
0099.05 C Z-ADD3 001002
0099.07 C** ラジオボタン
0099.08 C**+ 東京が選択されたときの処理
0099.09 C IRBG01 IFEQ 'R1'
0099.10 C**+ ラジオボタン選択フラグに1えおセット
0099.11 C MOVEL'1' RD11
0099.12 C ENDIF
0099.13 C**+ 名古屋が選択されたときの処理
0099.14 C IRBG01 IFEQ 'R2'
0099.15 C**+ ラジオボタン選択フラグに1えおセット
0099.16 C MOVEL'1' RD21
0099.18 C ENDIF
0099.19 C**+ 大阪が選択されたときの処理
0099.20 C IRBG01 IFEQ 'R3'
0099.21 C**+ ラジオボタン選択フラグに1えおセット
0099.22 C MOVEL'1' RD31
0099.24 C ENDIF
0099.25 C**+ ラジオボタン情報を画面上フィールドに返す
0099.26 C MOVELRD ORBG01
0099.27 C**
0101.00 C ENDIF
0102.00 C**
0103.00 C GOTO T100
0104.00 ---> * YOUR CODE
0105.00 C050 * </YOURCODE>

0245.00 C510 * <YOURCODE>
0246.00 *-----
0247.00 * YOUR INITIALIZATION
0248.00 *-----
0249.00 C YRINIT BEGSR
0249.01 C Z-ADD3 001002
0249.02 C MOVEL@RD,1 ORBG01
0250.00 C ENDSR

@RD配列要素
0285.00 **
0286.00 R1 東京 R2 名古屋 R3 大阪
```

手順③の構成に合わせ  
DATA STRUCTURESを定義

- ① 選択フラグ
- ② Radio 値... HTML上の Value 値をセット
- ③ Radio 記述

## 画面遷移時の値の受け渡し方法

では、ここで第1画面から第2画面を呼び出し、第2画面の値を第1画面に返し、その値を第1画面で表示する例を紹介しよう。

### ・処理の流れ

まず処理の流れだが、第1画面と第2画面のRPGプログラムが別々であるため、第1画面から第2画面を呼び出す。呼び出された第2画面は、名称の一覧を表示する画面である。その一覧の中でいずれか1つにチェックを入れ実行すると、第2画面を終了し、第1画面の名称欄に第2画面で選択した名称をセットするものである。

これら全てを、RPGプログラムで処理を行う。

### ・RPGプログラムのコーディング

冒頭にも述べ、何度も繰り返すが、JACi400は画面1つに対してRPGプログラム1つである。

ここでは、第1画面のRPGプログラムを“JCTST1A”とし、第2画面のRPGプログラムを“JCTST1B”とする。

まず、第1画面のJCTST1Aでは、第2画面のJCTST1Bを、CALL命令で呼び出すようにする。このときJCTST1Bからの戻り値として、名称をパラメータ設定しておく。【図6】

次に、第2画面のJCTST1Bだが、選択した名称を戻りパラメータにセットして、プログラムを終了するだけでよい。

あとは、第1画面のプログラムJCTST1Aに制御が戻り、名称をパラメータで受け取って、その名称をフィールドにセットする。これで、第2画面で選択した名称が、第1画面で表示されるのである。

このように、第1画面から第2画面に遷移するときは、第1画面からCALL命令を実行し、値を引き継ぐときは、パラメータを設定して受け渡しを行えばよいのである。

いかがだろうか。この内容も、RPGプログラムを開発したことのある人にとっては、それほど難しいことではないはずだ。

## 最後に

今回いくつかRPGのコーディング例を紹介した。どうであっただろうか。どれも難しい内容ではないので、RPGプログラムの開発経験がある人ならば、すぐに理解できる内容であったと思う。

私がそうであったように、これまでWebアプリケーション開発に壁を感じていたRPGプログラムの開発者にとっては、少しはWebアプリケーションを身近に感じられるようになったのではないだろうか。

それでも、今回紹介した内容は、JACi400を使ってRPGプログラムでできる項目の一部に過ぎない。まずは、実際にJACi400を使って、その性能を実感していただきたいと思う。そして、これまでに培った知識とノウハウを十分に活用できるJACi400で、押し寄せてくるWeb化の波に対応してほしい。

■

### 現在の仕事内容（詳細）

SE。主にJACi400を使ったWebアプリケーションの開発を担当しており、システムの要件定義から納品・フォローまで行っている。以前はRPGプログラムの開発に携わっていたこともあり、JACi400の開発フェーズでは、RPGプログラム開発の管理を行っている。また、HAツールである\*noMAXの技術サポートも担当している。



図6

```

JCTST1A
0098.00      * GET YOUR DATA
0099.00      C                               EXSR GTDATA
0100.00 C060 * <YOURCODE>
0101.00 ---> * YOUR CODE
0102.00      C*++ ENTER PUSH
0103.00      C                               JCACTN   IFEQ '1 '
0104.00      C                               CALL 'JCTST1B'
0105.00      C                               PARM      PRMOUT 32...①
0106.00      C                               MOVELPRMOUT OEDT01 ...④
0107.00      C                               MOVELPRMOUT IEDT01
0108.00      C                               GOTO T200
0109.00      C                               ENDIF

0018.00 E010 * <YOURCODE>
0019.00 ---> * YOUR ARRAYS
0020.00      E                               @CK      5 1      ...選択
0021.00      E                               @BA      5 10     ...コード
0022.00      E                               @BB      5 32     ...名称
0023.00 E010 * </YOURCODE>

0030.00 I010 * <YOURCODE>
0031.00 ---> * YOUR INPUT SPECIFICATIONS
0032.00      I                               DS
0033.00      I                               1 5 @CK
0034.00      I                               1 1 CHK01
0035.00      I                               2 2 CHK02
0036.00      I                               3 3 CHK03
0037.00      I                               4 4 CHK04
0038.00      I                               5 5 CHK05
0039.00      I                               DS
0040.00      I                               1 50 @BA
0041.00      I                               1 10 LBA01
0042.00      I                               11 20 LBA02
0043.00      I                               21 30 LBA03
0044.00      I                               31 40 LBA04
0045.00      I                               41 50 LBA05
0046.00      I                               DS
0047.00      I                               1 160 @BB
0048.00      I                               1 32 LBB01
0049.00      I                               33 64 LBB02
0050.00      I                               65 96 LBB03
0051.00      I                               97 128 LBB04
0052.00      I                               129 160 LBB05
0053.00 I010 * </YOURCODE>

0089.00 C010 * <YOURCODE>
0090.00 ---> * YOUR CODE
0091.00      C                               *ENTRY  PLIST
0092.00      C                               PARM      PRMOUT 32...②
0093.00 C010 * </YOURCODE>

                                ↑
                                ↓
0308.00      C                               GTDATA  BEGSR
0326.00      C*+                               ...③
0327.00      C                               1      DO 5      J      20
0328.00      C                               @CK,J   IFEQ '1 '
0329.00      C                               MOVEL@BB,J PRMOUT
0330.00      C                               SETON      10
0331.00      C                               ENDIF
0332.00      C                               ENDDO
0333.00      C                               ENDSR

```

画面上で選択にチェックが入れられた場合、チェックされたところによりCHK01～05のいずれかに“1”が戻ってくる。

画面表示時、プログラム内でコードを画面のアウトプット用フィールドにセットする。

画面表示時、プログラム内でコードと対応する名称を画面のアウトプット用フィールドにセットする。