

株式会社ミガロ.

RAD事業部 技術支援課

[Valence]

Valence App Builder RPG連携テクニック



略歴

1973年8月16日生まれ
1996年3月 三重大学 工学部卒業
1999年10月 株式会社ミガロ 入社
1999年10月 システム事業部配属
2013年4月 RAD事業部配属

現在の仕事内容

Delphi/400 を中心としたテクニカルサポート対応や製品セミナーの講師などを担当している。

1. はじめに
2. Valence App Builder アプリ開発方法
3. Valence App Builder RPG 連携の基本
4. Valence RPG ToolKit 活用テクニック
5. さいごに

1. はじめに

Valence は、IBM i を Web 環境で活用するモダナイゼーション開発・運用ツールである。Valence には、次のような特長がある。

- ・ローコード開発ツールによりアプリの高速開発が可能
- ・ビジュアルな運用管理ツール群を搭載
- ・PC の Web ブラウザ、モバイル両方に対応

Valence の特長の1つであるローコード開発ツールが、「Valence App Builder」である。Valence App Builder を使用すれば、基幹システム開発に必要な入力・更新・照会アプリケーションが、ほぼノンコーディングで作成できる。

また、その作成方法はウィザードを使用したシンプルな手法なので、操作に慣れると、簡単なものであれば、ものの数分でアプリケーションを作成することが可能である。

Valence App Builder はノンコーディングで簡単にアプリケーションを開発できるが、複雑な業務ロジックを伴うアプリケーションを開発する場合には、RPG を組み合わせることも可能だ。

本稿では、Valence App Builder と RPG とを組み合わせるためのテクニックについて紹介する。

2. Valence App Builder アプリ開発方法

はじめに、Valence App Builder のアプリケーション開発方法を確認しておこう。Valence App Builder では、次の3つのステップでアプリケーションを作成する。

- 1 データソースの作成
- 2 ウィジェットの作成
- 3 アプリケーションの作成

「1 データソースの作成」では、ア

プリケーションで使用するデータ元となる IBM i 上のファイルを選択し、「2 ウィジェットの作成」では、データソースをもとに、表 (グリッド) やグラフ等のデータを表現するための部品を定義する。そして、「3 アプリケーションの作成」では、1つあるいは複数のウィジェットを画面上に配置し、アクション (動作) を定義することで、1つのアプリケーションとして完成させるのである。【図1】

実際の開発手順は、次のとおりだ。まず、データソースを作成する。データソースの作成は、IBM i の QUERY ユーティリティで、QUERY の定義を作成するのと似ている。データソース作成ウィザードを使用して、アプリケーションが使用する IBM i 上のファイルとカラム (フィールド) を選択・保存すればよい。

ウィザードではほかにも複数ファイルを使用した時の結合条件や、データの絞り込み条件、並び順なども指定できる。【図2】

次に、保存したデータソースをもとにウィジェットを作成する。ウィジェット

図1 Valence App Builder 3つの開発ステップ

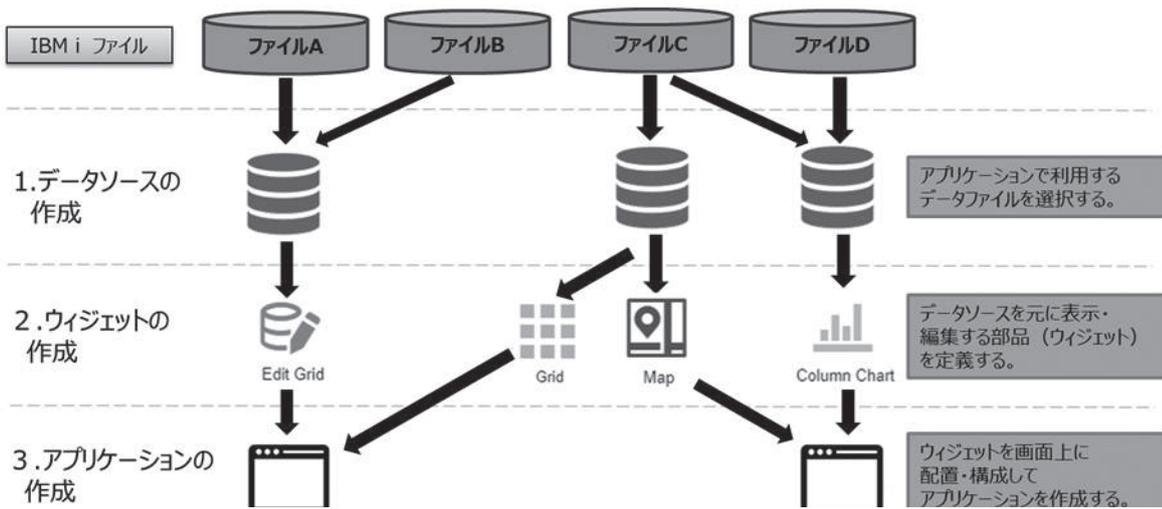
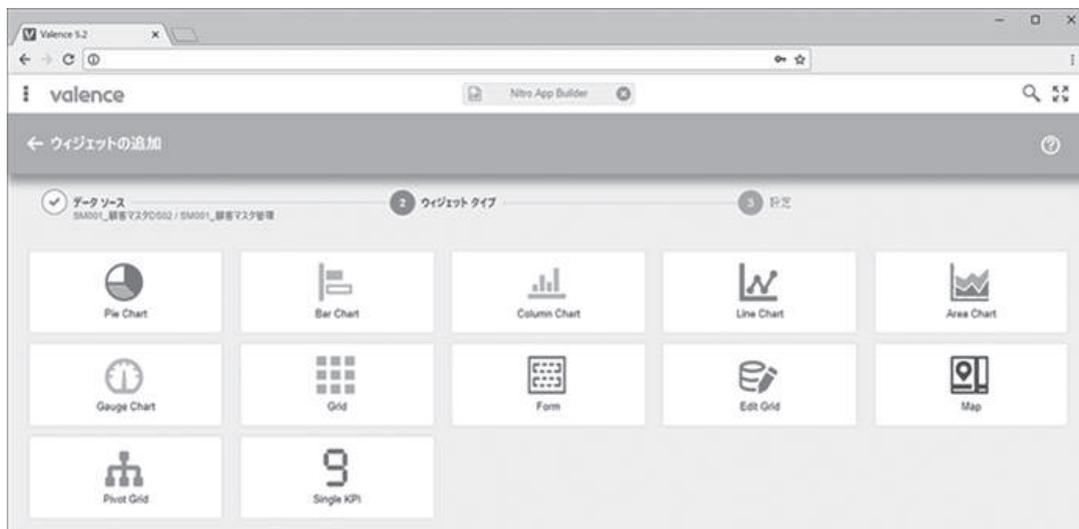


図2 データソース作成ウィザード



図3 ウィジェット一覧



- グラフ : 円グラフ、棒グラフ（横、縦）、折れ線グラフ、面グラフ、ゲージ（メーター）
- グリッド : グリッド（表）、編集用グリッド、クロス集計グリッド、
- その他 : フォーム、指標表示（シングルKPI）、地図

は、抽出したデータをさまざまな見せ方で表示するための部品である。2019年8月現在の最新版である Valence5.2には、表（グリッド）やグラフ、地図など全部で12種類のウィジェットが用意されている。【図3】

選択したウィジェットに応じた設定画面が用意されているので、データソース上のカラム（フィールド）を関連付ける。たとえばグリッドであれば、表示させる列の指定や書式の設定を行えばよい。また、グリッドやグラフのウィジェットでは、データソース上のレコードを絞り込むためのフィルタ設定が可能である。これにより、実行時にユーザーによる条件指定が行える。【図4】

必要なウィジェットが揃ったら、1つあるいは複数のウィジェットを組み合わせ、アプリケーションを作成する。複数のウィジェットを1つのワークスペースに配置した単一画面のアプリケーションや、複数のセクションを定義してワークスペースを増やすことで、複数画面をもつアプリケーションを作成できる。【図5】

ワークスペース上に複数のウィジェットを配置した場合や、複数のアプリケーションセクションを定義した場合には、ユーザーの操作に対するアクションを設定できる。たとえば、グリッドに表示された受注一覧の行をクリックした時に、選択した受注NOでデータの絞り込み（フィルタ）を行い、結果を詳細フォームに表示するといったアクションを設定できる。【図6】

以上が、Valence App Builderでアプリケーションを開発する手順である。基本的な部分はウィザードを使った簡単な定義で済み、一切のコーディングを行わずにアプリケーションが作成できる。

3. Valence App BuilderとRPG連携の基本

Valence App Builderでは、編集グリッド (Edit Grid) やフォーム (Form) の入力欄に関して、カラムの属性に合わせた入力可能文字の制御や項目の必須チェックなどを、ウィザードで設定できる。

しかし、たとえば複数項目間の相関エラーチェック等はウィザードでは設定できない。また、データソース上のカラム

(フィールド) 以外のファイルやフィールドを内部的に更新することはできない。こうした Valence App Builder の設定だけでは実現できない処理は、RPG ロジックを追加することで対応する。

Valence App Builderでは、下記のタイミングからRPGプログラムを呼び出して実行できる。

- 1 グリッドの行やグラフをクリック、あるいはウィジェットやアプリケーションセクションに追加したボタンをクリックした時
- 2 編集グリッド (Edit Grid) でレコードの追加/更新/削除を行う時
- 3 グリッド等でユーザーがフィルタ (絞り込み) を行う時

1については、アプリケーションの動作内容 (アクション) 設定画面でRPGプログラム呼び出しのアクションを追加できる (なお、ウィジェットやアプリケーションセクション上にボタンを追加するのも、この動作内容設定画面で行う)。【図7】

2と3は、それぞれウィジェットの設定画面で呼び出したいRPGプログラムのIDを追加すればよい。【図8】 【図9】

Valence App Builderでは、実行するRPGプログラムをテンプレートプログラムからコピーして作成する。テンプレートプログラムは、Valenceライブラリ上にあるQRPGLESRCの中に含まれている。【図10】

- 1 クリック : VALENCE52/
QRPGLESRC (EXNABBTN)
- 2 Edit Grid : VALENCE52/
QRPGLESRC (EXNABVAL)
- 3 フィルタ : VALENCE52/
QRPGLESRC (EXNABFLT)

1のクリック時に実行されるプログラム (EXNABBTN) のテンプレートプログラムは、【ソース1】である。テンプレートプログラムは、フリーフォームRPGで記述されている。

【1-①】が実行するプログラムの定義 (プロトタイプPRおよびプロシージャインターフェースPI) となっており、アプリケーションを作成する際には、ここにそれぞれプログラムIDを指

定する。

【1-②】がプログラムのメイン処理で、この中からProcessサブプロシージャを呼び出している。呼び出されたProcessサブプロシージャ【1-③】の中に、アプリケーションに必要なロジックを記述すればよい。

具体例を2つ紹介する。1つ目は、商品マスタメンテナンスの登録画面である。【図11】

フォームウィジェットを使用した簡単なアプリケーションで、商品CD、商品名、単価を入力して、登録ボタンをクリックすると、商品マスタに新規レコードを追加する。この登録ボタンをクリックした時に呼び出すRPGプログラム (TEC010) が、【ソース2】である。

【2-①】のプログラム定義では、プログラムIDの“TEC010”を指定している。そしてProcessサブプロシージャの中に必要な処理を記述している。【2-②】は、Processサブプロシージャの中で使用する変数の定義である。【2-③】は、フォームウィジェット上の入力項目の値を取得して、【2-②】で定義した変数に代入する処理である。

GetFormChar (文字) /GetFormNum (数値) は、ウィジェットに定義されたカラム (フィールド) の値を取得するために用意されたValenceのAPIである。

【2-④】がこのプログラムの主処理で、この中でキーフィールドの重複チェックおよび新規レコードの登録を行っている。その中にある【2-⑤】【2-⑥】が、処理結果をValenceアプリケーション側に返却する処理である。vvOutToJsonPairというValenceのAPIにより、処理結果が返却される。

この命令のパラメータにある“success:true (false)”は、処理結果の正常/エラーを表し、“msg”はエラー等のポップアップメッセージを、“info”は画面下部に表示されるお知らせメッセージを表す。

このプログラムを見るとわかるように、Valence App BuilderにおけるRPGにはフリーフォームRPGを使用するのだが、ValenceのAPIを使用する部分のみ、フリー形式 (/free ~ /end-free) で記述しており、その他のロジック部分は、従来どおりの固定長形式で記述している。

図4 ウィジェットのフィルタ設定

絞り込み対象とするフィールドを指定

ウィジェットの中に、絞り込み条件欄が追加される

フィールド	説明	選択子	初期値	大文字に自...	種
<input type="checkbox"/> F2_TANTO_X01	総当り能力	-	-	-	数値
<input type="checkbox"/> F1_R_TRHQ_CD	顧客先C/D	-	-	-	数値
<input checked="" type="checkbox"/> F1_R_TRHQ_NM	顧客先名	高C	-	-	数値
<input checked="" type="checkbox"/> F1_R_TANTO_CD	総当りC/D	-	-	-	数値
<input type="checkbox"/> F1_R_YUBN	郵便番号	-	-	-	数値
<input type="checkbox"/> F1_R_AISYO	...	-	-	-	...

図5 アプリケーションの作成

画面（ワークスペース）上に一つあるいは複数のウィジェットを配置できる

ジョブ名	CPU %	ステータス	サブシステム
004196/QTMHHTTP...	0.3	RUN	QHTTPSVR
004169/QTMHHTTP...	0.1	TMY	QHTTPSVR
004074/QLWSVRAD...	0.1	THDW	QHTTP...
004197/QTMHHTTP...	0	TMY	QHTTPSVR

つまりテンプレートプログラムの仕組み（構成）と Valence の API の使用方法さえマスターすれば、これまでどおりの RPG スキルをそのまま活用して開発できる。

なお、作成した RPG ソースは通常どおりにコンパイルしてオブジェクトを生成すればよいが、コンパイル時に RPG プリプロセッサ・オプションを “*LVL2” に変更する点に注意してほしい。【図 12】

2 つ目は、編集グリッド (Edit Grid) を使用したユーザーマスタメンテナンス画面である。【図 13】

このアプリケーションでは、入力画面上の連絡区分の値により、異なる必須項目エラーチェックを行う。レコード新規追加／更新／削除のタイミングで呼び出す RPG プログラム (TEC020) が、【ソース 3-1】 【ソース 3-2】 である。このプログラムは、テンプレートプログラム EXNABVAL をコピーして作成したものである。

【3-①】 のプログラム定義では、プログラム ID の “TEC020” を指定している。【3-②】 がプログラムのメイン処理となっている。パラメータ inMode は、Edit Grid の実行状態を表しており、レコードの新規追加時は “ADD”、編集時は “EDIT”、削除時は “DELETE” がセットされるようになっている。

そして実行状態に応じて、新規追加時であれば ProcessAdd サブプロシージャが、編集時であれば ProcessEdit サブプロシージャが、削除時であれば ProcessDelete サブプロシージャが呼び出されるので、それぞれに必要なアプリケーションロジックを各サブプロシージャに記述すればよい。

【3-③】 がこの ProcessAdd サブプロシージャの主処理で、連絡区分の値に応じたエラーチェック処理を実行している。その中にある 【3-④】 【3-⑤】 が Valence の API を使用した部分でフリー形式である（【3-④】 【3-⑤】 以外の箇所は、固定長形式で記述している）。

【3-④】 が、Edit Grid 上の編集画面で更新された項目の値を取得して変数に代入する処理である。値の取得には、GetValue を使用する。これは、パラメータに指定したファイル、フィールドの値を取得する Valence の API である。【3-

⑤】 に記述された SendError は、データの更新処理を中断し、エラーメッセージを出力する API である。

今回は 2 つの具体例をもとに、Valence App Builder から RPG プログラムを連携する方法を紹介した。

本章の最後に、Valence App Builder の各 RPG テンプレートプログラムから使用できる API 一覧を紹介する。【図 14】

なお各 API の詳細は、Valence Portal にある「Valence API ドキュメント」メニューに説明とサンプルが記載されているので、確認してほしい。

4. Valence RPG ToolKit活用テクニック

前章では、Valence App Builder から RPG プログラムを呼び出す方法を説明したが、他にも Valence には、RPG を活用して機能拡張を行う仕組みが用意されている。それが、RPG ToolKit である。ここでは RPG ToolKit の活用例として、ファイルのダウンロード機能、メール送信機能の実装を紹介する。

まず、ファイルのダウンロード実装例を紹介する。ファイルのダウンロードといっても大きく分けて、「1 あらかじめ保存しているファイルをダウンロードする」「2 動的にファイルを生成してダウンロードする」の 2 つが考えられる。今回は両方の手法を紹介する。

1 の保存ファイルダウンロードについて、具体例として商品カタログ PDF のダウンロード機能を紹介する。【図 15】

フォームウィジェット上にある [カタログダウンロード] ボタンをクリックすると、IFS 上に保存された選択商品 CD に関するカタログファイル ([商品 CD].pdf) をダウンロードする。このボタンをクリックした時に実行される RPG プログラム (TEC030) が、【ソース 4】 である。

【4-①】 が、IFS 上に保存された PDF ファイルの保管先パスを取得するロジックである。vvUtility_getValenceSetting は、Valence の設定情報を取得する API で、パラメータに “ROOT_PATH” を指定することで、IFS 上の Valence ルートフォルダを取得できる。

今回は取得したルートフォルダの配下にある “resources/pdfs” の中に、PDF ファイルがあらかじめ保存されている想定である。

【4-②】 では、対象となる商品 CD の PDF ファイルが存在するかどうかを判定している。vvIifs_pathExists は、パラメータに指定したファイルが存在するかどうかを確認するための API で、ファイルが存在する場合は “*ON”、存在しない場合は “*OFF” が返却される。今回はファイルが存在しない場合、エラーメッセージを Valence に渡している。

【4-③】 が、PDF ファイルを Valence 側に受け渡す処理である。vvout 構造体は出力するファイルを指定するもので、download フィールドに “1” をセットすると、クライアントへファイルを受け渡すモードとなる。そして file フィールドにダウンロードしたいファイルをセットし、vvOut_file という API を実行すると、ファイルが Valence 側へ渡される。

なおファイルのダウンロード機能を実装する場合は、Valence App Builder の RPG プログラム呼び出し設定画面にて、[ファイル返却時の処理] に “ダウンロード” を指定する。【図 16】

次に、2 の動的に作成したファイルダウンロードについて、具体例として担当者別受注一覧 Excel のダウンロード機能を紹介する。【図 17】

フォームウィジェット上にある [受注ダウンロード] ボタンをクリックすると、IBM i 上の受注ファイル (F_JUCHU_H) から、選択した担当者 CD の受注データを取得し、その結果をもとに動的に作成した Excel ファイルをダウンロードする。ボタンをクリックした時に実行される RPG プログラム (TEC040) が、【ソース 5】 である。

【5-①】 が、受注ファイルを取得するためのデータ抽出用 SQL 文である。フォーム上の担当者 CD を取得し、その値を使った SQL 文字列を作成している。【5-②】 が Excel ファイルのダウンロード処理である。vvOut_execSqlToSS は、SQL で取得した結果をもとに Excel ファイルを出力する API である。

なお今回の方法では、SQL をもとに Excel ファイルを作成しているが、データ構造体 (DS) をもとに Excel ファイ

図6 動作内容(アクション)の設定



図7 クリック時のRPG呼び出し



図8 Edit Grid 編集時のRPG呼び出し



ルを生成する vvOut_toSS という API も用意されている。READ 命令などファイル操作で取得したデータを出力する場合には、データ構造体に値をセットした上で、こちらの API を使用すればよい。

ダウンロードされた Excel ファイルを見ると、SQL で取得した結果が出力されているのがわかる。【図 18】

ただし、このままでは SQL により取得されたすべてのフィールドが出力されており、行タイトルには、フィールド ID がそのまま出力される。ロジックを追加することにより、出力する Excel の列設定をカスタマイズすることもできる。カスタマイズした TEC040 プログラムが、【ソース 6】である。

【6-①】で、列の設定を行うための vvSSCol 構造体の配列を定義している。そして【6-②】の部分で、Excel の列ごとに出力したいカラム（フィールド）および列のタイトルを設定している。【6-③】の vvOut_execSqlToSS にて、出力する SQL 文字列とともに、列定義の構造体配列をパラメータにセットすることで、列定義を含む Excel 出力が行える。【図 19】

最後に、メール送信機能の実装例を紹介する。ここでは、ファイルダウンロードで作成した Excel と同じものを添付してメール送信する機能を紹介する。【図 20】

フォームウィジェット上にある [メール送信] ボタンをクリックすると、メール送信先アドレスを入力するダイアログが表示される。RPG プログラム呼び出しのアクションを設定する際に、任意のパラメータを追加できるオプション設定があり、今回はメールアドレスとして、パラメータ“MAILTO”を追加している。【図 21】

メールアドレスを入力して、[OK] ボタンをクリックすると、動的に作成した Excel ファイルを添付したメールを、指定したアドレスに送信する。この [メール送信] ボタンをクリックした時に実行される RPG プログラム（TEC050）が、【ソース 7】である。

メール送信の API を使用する場合、【7-①】のように D 仕様書に“/define includeEMAIL”を追加する。【7-②】では、vvIn_char を使用している。これ

は、Valence 側で指定された文字列を取得する API で、追加パラメータに指定した“MAILTO”と“sid”を取得している。“sid”はセッション ID のことで、Valence を実行しているブラウザごとに一意になる文字列が取得できる。

【7-③】が、Excel ファイルを生成する部分である。今回はダウンロードではなく、IFS 上に Excel ファイルを一時保管している。vvOut 構造体の download フィールドに“F”を設定すると、IFS 上へファイルが作成できる。今回は、セッション ID を使用した一意な名称の Excel ファイルを生成している。

【7-④】が、メール送信内容を作成する部分である。vvMail 構造体に宛先、題名、本文、添付ファイル等の送信内容をセットすればよい。

【7-⑤】が、実際にメールを送信する処理である。vvMail_send を実行し、メールが正常に送信できた場合には“*ON”が、エラーとなった場合は“*OFF”が返却される。

メール送信処理が終わったら、IFS 上に一時保管した Excel ファイルを【7-⑥】で削除している。vvIfs_deleteFile は、IFS 上のファイルを削除する処理である。

今回のプログラムにより、指定されたメールアドレス宛てに、添付ファイル付きのメール送信が行える。【図 22】

なお、メール送信機能を使用する場合、あらかじめ Valence の設定画面にて、SMTP サーバーに関する設定をしておく必要がある。

5. さいごに

本稿では、Valence App Builder から RPG を連携する手法について紹介してきた。Valence App Builder は、3つのステップで簡単にアプリケーションが作成できる。

ローコード開発プラットフォームである Valence App Builder は、一切のコーディングを行わずに開発できる。もちろん業務ロジックなど複雑な処理にはプログラムの追加が必要となるが、テンプレートプログラムと、Valence の API 使用方法をマスタすれば、RPG スキルで簡単に処理が追加できる。

また今回紹介したとおり、RPG ToolKit を活用すれば、Valence App

Builder のアプリケーションに対し、さまざまな機能拡張も行える。ぜひ本稿を参考に、さらなる Valence App Builder の活用をご検討いただきたい。

M

図9 フィルタウィジェットのRPG呼び出し



図10 テンプレートRPGソース

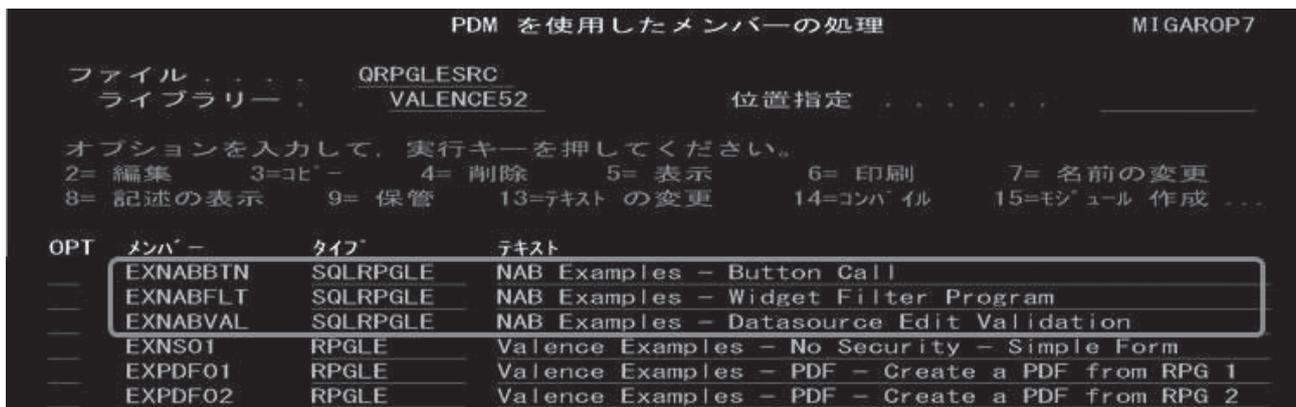
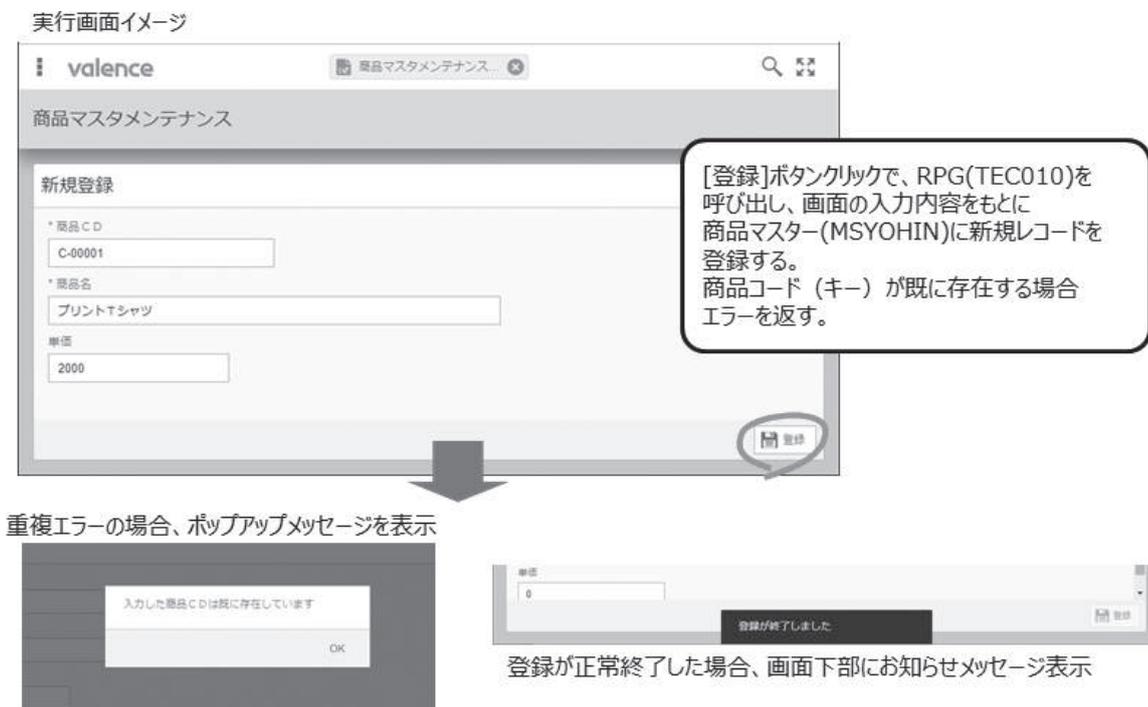
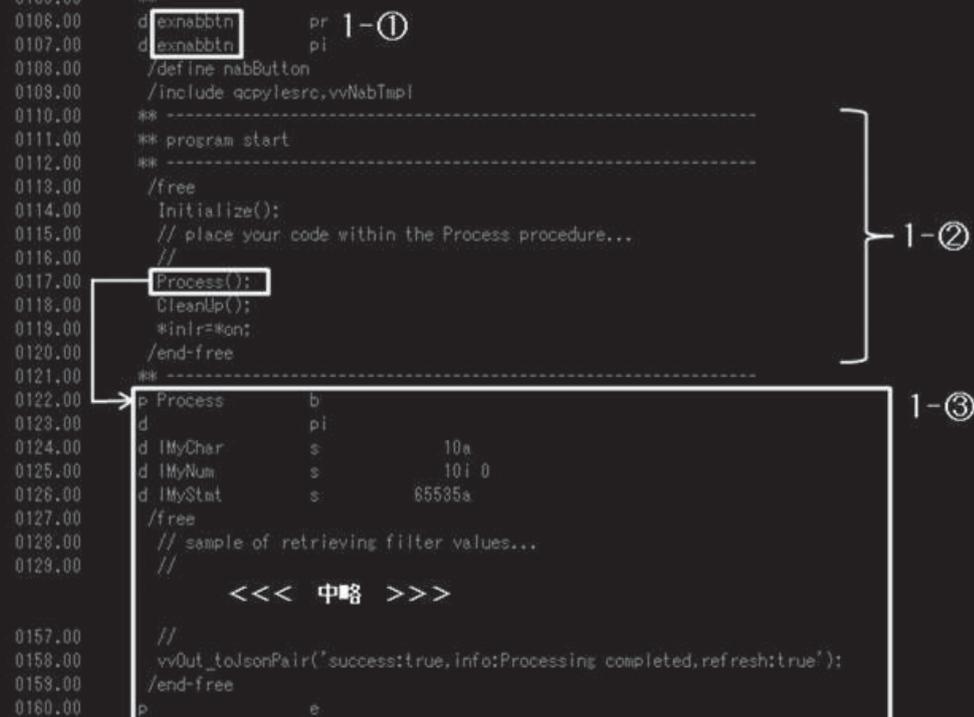


図11 ボタンクリック サンプルプログラム



ソース1 テンプレートプログラム(EXNABBTN)

```
0001.00 /copy qcpylesrc,vvHspec
0002.00 ** -----
0003.00 **      Copyright (C) 2008-2018 CNX Corporation
0004.00 ** -----
0005.00 **      Object ID: EXNABBTN
0006.00 **      Version: V5.2
0007.00
0008.00      <<<< 中略 >>>
0009.00
0104.00 **      NOTE: this must be compiled with RPGPPOPT = #LVL2
0105.00 ** -----
0106.00 d exnabbtn      pr 1-①
0107.00 d exnabbtn      pi
0108.00 /define nabButton
0109.00 /include qcpylesrc,vvNabImpl
0110.00 ** -----
0111.00 ** program start
0112.00 ** -----
0113.00 /free
0114.00     Initialize();
0115.00     // place your code within the Process procedure...
0116.00     //
0117.00     Process();
0118.00     Cleanup();
0119.00     *inlr=#on;
0120.00 /end-free
0121.00 ** -----
0122.00 p Process      b
0123.00 d             pi
0124.00 d lMyChar      s      10a
0125.00 d lMyNum       s      10i 0
0126.00 d lMyStat      s      65535a
0127.00 /free
0128.00     // sample of retrieving filter values...
0129.00     //
0130.00     <<<< 中略 >>>
0131.00
0157.00     //
0158.00     vvOut_toJsonPair('success:true,info:Processing completed,refresh:true');
0159.00 /end-free
0160.00 p
0161.00 /include qcpylesrc,vvNabImpl
```



ソース2 商品マスタメンテナンス新規登録 (TEC010)

```

0001.00 /copy qcpylesrc,vvHspec
0002.00 **
0003.00 ** TEC010:商品マスタメンテナンス新規登録
0004.00 **
0005.00 F*
0006.00 F* ファイル定義
0007.00 F*
0008.00 F* <商品マスタ>
0009.00 FMSYOHIN UF A E          K DISK
0010.00 F*
0011.00 d TEC010 2-① pr
0012.00 d TEC010 pi
0013.00 /define nabButton
0014.00 /include qcpylesrc,vvNabTapl
0015.00 **
0016.00 ** program start
0017.00 **
0018.00 /free
0019.00 Initialize();
0020.00 Process();
0021.00 CleanUp();
0022.00 *inlr=*on;
0023.00 /end-free
0024.00 **
0025.00 p Process          b
0026.00 d                  pi
0027.00 D VSYHNCD          S          10A
0028.00 D VSYHNNM          S          32A
0029.00 D VTANKA           S          9 0
0030.00 C*
0031.00 /free
0032.00 //フォーム上の値を取得
0033.00 VSYHNCD = GetFormChar('F1_SYHNCD'); //商品CD
0034.00 VSYHNNM = GetFormChar('F1_SYHNNM':'0'); //商品名
0035.00 VTANKA = GetFormNum('F1_TANKA'); //単価
0036.00 /end-free
0037.00 C*
0038.00 C*-----キー重複チェック
0039.00 C VSYHNCD CHAIN MSYOHIR 81
0040.00 C *IN81 IFEQ *OFF
0041.00 C*
0042.00 /free
0043.00 //エラーメッセージを送信
0044.00 vvOut_toJsonPair('success:false,'
0045.00 + 'msg:入力した商品CDは既に存在しています')
0046.00 /end-free
0047.00 C*
0048.00 C ELSE
0049.00 C*-----新規レコード登録
0050.00 C MOVEL VSYHNCD SYHNCD
0051.00 C MOVEL VSYHNNM SYHNNM
0052.00 C Z-ADD VTANKA TANKA
0053.00 C*
0054.00 C WRITE MSYOHIR
0055.00 C*
0056.00 /free
0057.00 //正常終了メッセージを送信
0058.00 vvOut_toJsonPair('success:true,refresh:true,'
0059.00 + 'info:登録が終了しました');
0060.00 /end-free
0061.00 C*
0062.00 C END
0063.00 p e
0064.00 /include qcpylesrc,vvNabTapl

```

2-②

2-③

2-⑤

2-④

2-⑥

図12 コンパイル時のオプション指定

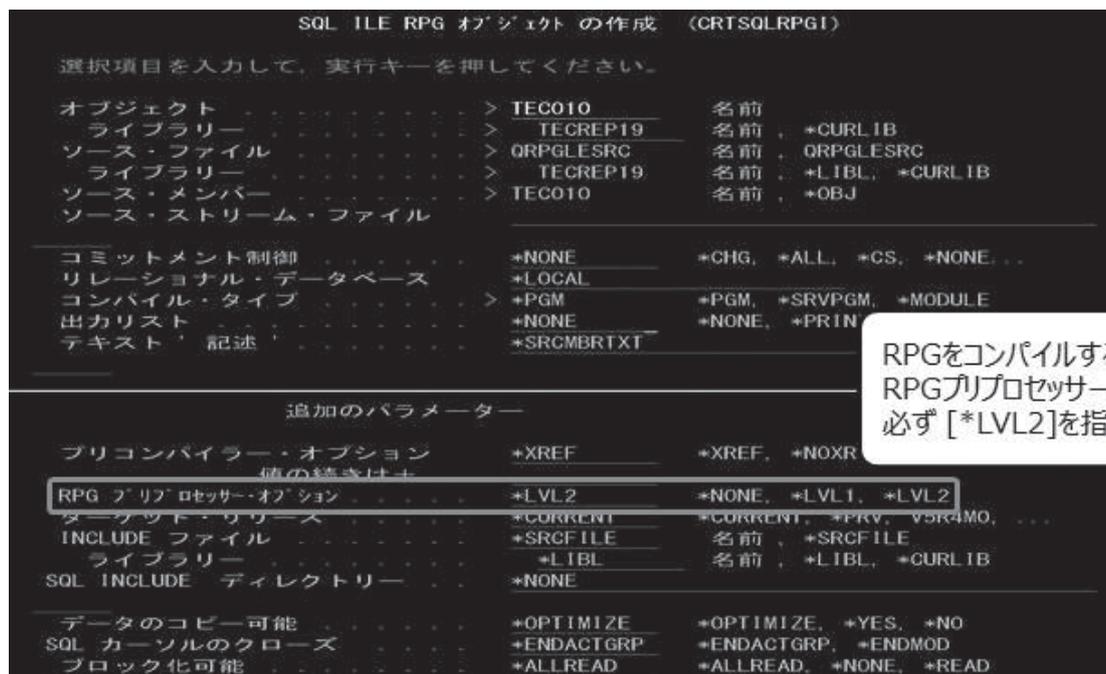
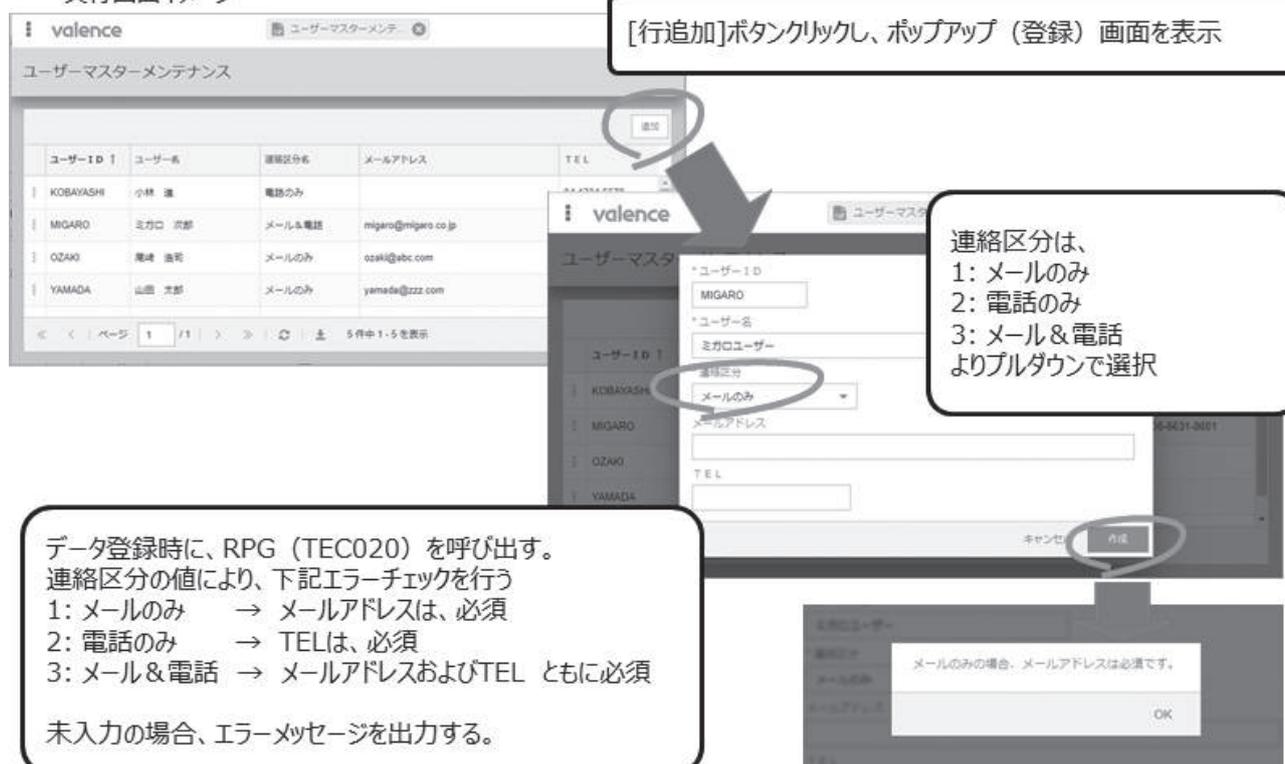


図13 Edit Grid サンプルプログラム

実行画面イメージ



ソース3-1 ユーザーマスタメンテナンス更新(TEC020)

```

0001.00 /copy qcpylesrc,vvHspec
0002.00 ** -----
0003.00 ** TEC020:ユーザーマスタメンテナンス
0004.00 ** -----
0005.00 F*-----
0006.00 F* ファイル定義
0007.00 F*-----
0008.00 F* <メッセージマスタ>
0009.00 FMMSGP IF E K DISK
0010.00 F*
0011.00 d TEC020 3-① pr
0012.00 d inMode 10a
0013.00 d inDataPtr *
0014.00 d outStopProcess...
0015.00 d n
0016.00 d TEC020 3-① pi
0017.00 d inMode 10a
0018.00 d inDataPtr *
0019.00 d outStopProcess...
0020.00 d n
0021.00 /define nabValidation
0022.00 /include qcpylesrc,vvNabTmpl
0023.00 ** -----
0024.00 ** program start
0025.00 ** -----
0026.00 /free
0027.00 Initialize();
0028.00
0029.00 // perform validations based on the mode...
0030.00 //
0031.00 if inMode = 'ADD';
0032.00 ProcessAdd();
0033.00 elseif inMode = 'EDIT';
0034.00 ProcessEdit();
0035.00 elseif inMode = 'DELETE';
0036.00 ProcessDelete();
0037.00 elseif inMode = 'POSTADD';
0038.00 ProcessPostAdd();
0039.00 elseif inMode = 'POSTDELETE';
0040.00 ProcessPostDelete();
0041.00 elseif inMode = 'POSTEDIT';
0042.00 ProcessPostEdit();
0043.00 endif;
0044.00
0045.00 Cleanup();
0046.00 *inlr=*on;
0047.00 /end-free
0048.00 ** -----
0049.00 p ProcessAdd b
0050.00 d pi
0051.00 **

```

ProcessAdd();

3-②



ソース3-2 ユーザマスタメンテナンス更新(TEC020)

```

0052.00 D LUSRKBN S 1A
0053.00 D LUSEMAL S 40A
0054.00 D LUSTEL S 14A
0055.00 D LERR S 5A
0056.00 **
0057.00 /free
0058.00 //----- 画面上の値を取得
0059.00 LUSRKBN = GetValue(*MUSER:'USRKBN'); //---連絡区分
0060.00 LUSEMAL = GetValue(*MUSER:'USEMAL'); //---メールアドレス
0061.00 LUSTEL = GetValue(*MUSER:'USTEL'); //---TEL
0062.00 /end-free
0063.00 C*-----連絡区分=1 (メールのみの場合)
0064.00 C LUSRKBN IFEO '1'
0065.00 C LUSEMAL IFEO *BLANK
0066.00 C MOVEL 'E0010' LERR
0067.00 C
0068.00 C ENDF
0069.00 C*-----連絡区分=2 (電話のみの場合)
0070.00 C LUSRKBN IFEO '2'
0071.00 C LUSTEL IFEO *BLANK
0072.00 C MOVEL 'E0020' LERR
0073.00 C
0074.00 C ENDF
0075.00 C*-----連絡区分=3 (メール&電話の場合)
0076.00 C LUSRKBN IFEO '3'
0077.00 C LUSEMAL IFEO *BLANK
0078.00 C LUSTEL OREQ *BLANK
0079.00 C MOVEL 'E0030' LERR
0080.00 C
0081.00 C ENDF
0082.00 C*-----エラーメッセージ取得
0083.00 C LERR IFNE *BLANK
0084.00 C LERR CHAIN MMSGR 31
0085.00 C *IN31 IFEO *OFF
0086.00 /free
0087.00 //エラーメッセージを送信
0088.00 SendError(MMSGTX);
0089.00 /end-free
0090.00 C ENDF
0091.00 C ENDF
0092.00 p e
0093.00 ** -----
0094.00 p ProcessDelete b
<<< 中略 >>>
0125.00 p e
0126.00 /include qcpylesrc,vvNabTmp1

```

3-④

3-③

3-⑤

図14 Valence App Builder API一覧

1. クリック (EXNABBTN)

I/O	API	概要
I	vvIn_char	処置パラメータの取得
I	GetFormChar	フィールド値取得(文字列)
I	GetFormNum	フィールド値取得(数値)
I	GetSelectionChar	選択したレコードのフィールド値取得(文字列)
I	GetSelectionNum	選択したレコードのフィールド値取得(数値)
I	GetFilterValue	フィルタ条件の取得
-	GetSqlStatement	関連するデータソースで実行されている SQL ステートメントの取得
O	vvOut_toJsonPair	ブラウザへ結果の返却
O	vvOut_data	ブラウザへ結果の返却(フィルタウィジェット実行)

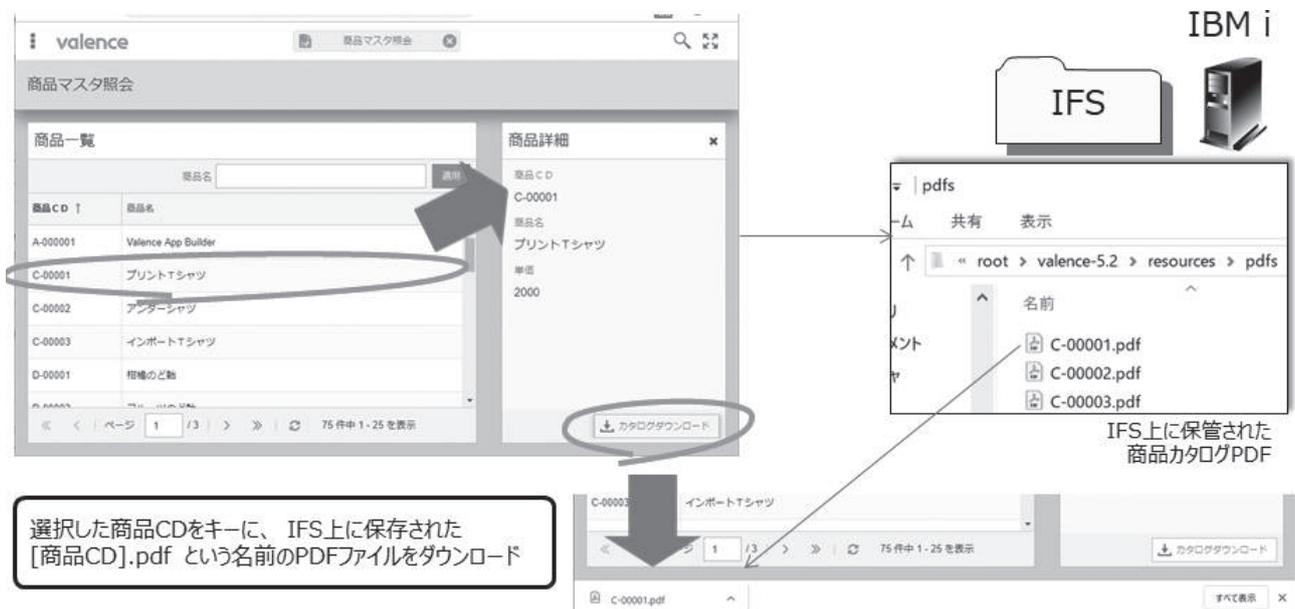
2. Edit Grid (EXNABVAL)

I/O	API	概要
I	GetValue	入力値の取得
I	GetFilterValue	フィルタ条件の取得
I	IsChanged	指定されたファイル/フィールドが変更されているかどうかを判断
-	SetValue	ファイル名、フィールド名の引数で設定、対象の値を更新
O	SetDoNotProcess	指定されたファイルへの追加、更新、削除処理をスキップ
O	SendError	処理の中断とメッセージの表示

3. フィルタ (EXNABFLT)

I/O	API	概要
I	GetValue	引数のフィルタ値の取得
-	SetValue	引数に設定したフィールドに対してフィルタ値を設定
O	AddFilterString	指定されたフィルタフィールドにフィルタを追加
O	WriteAllFilters	すべてのフィルタフィールドに対してフィルタを追加
O	SendError	処理の中断とメッセージの表示

図15 PDFファイル ダウンロード



ソース4 商品カタログPDFダウンロード(TEC030)

```

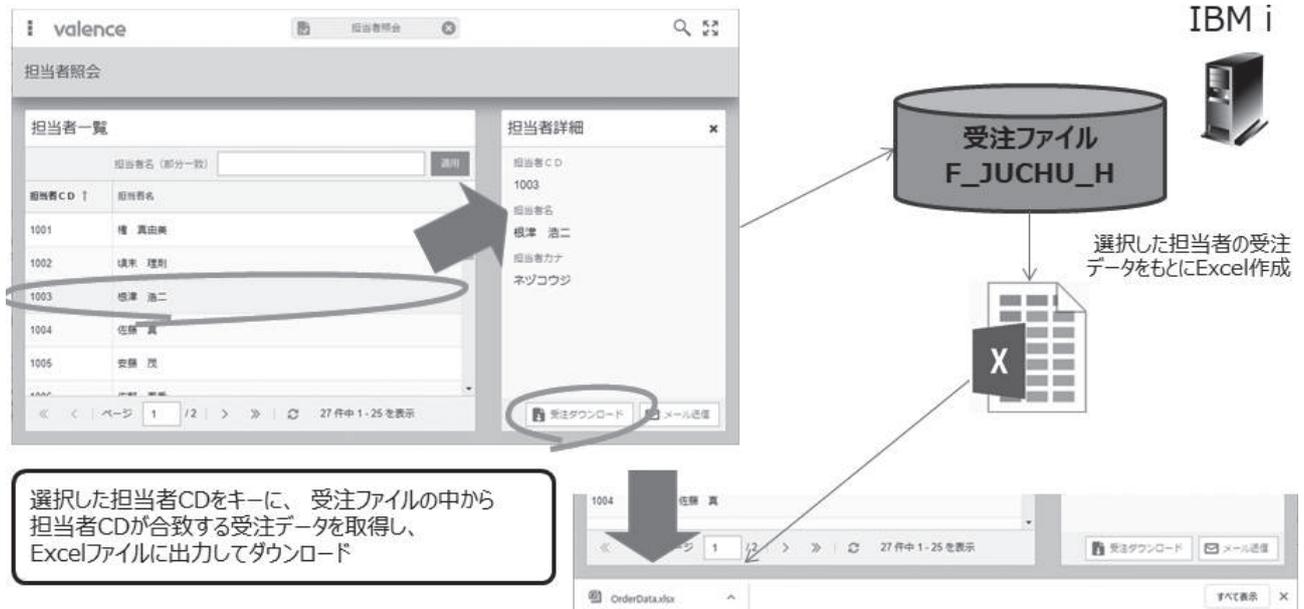
0001.00 /copy qcpylesrc,vvHspec
0002.00 **-----
0003.00 ** TEC030:PDFダウンロード
0004.00 **-----
0005.00 d TEC030 pr
0006.00 d TEC030 pi
0007.00 /define nabButton
0008.00 /include qcpylesrc,vvNabTpl
0009.00 **-----
0010.00 ** program start
0011.00 **-----
0012.00 /free
0013.00 Initialize();
0014.00 Process();
0015.00 Cleanup();
0016.00 *inlr=*on;
0017.00 /end-free
0018.00 **-----
0019.00 p Process b
0020.00 d pi
0021.00 D SYHNCD $ 10A
0022.00 D ROOTPATH $ 20A
0023.00 D PDFPATH $ 64A
0024.00 D FILENAME $ 14A
0025.00 D*
0026.00 /free
0027.00 //フォーム上の値を取得
0028.00 SYHNCD = GetFormChar('F1_SYHNCD'); //商品CD
0029.00
0030.00 //PDF保存先PATH取得 4-①
0031.00 ROOTPATH = vvUtility_getValenceSetting('ROOT_PATH');
0032.00 PDFPATH = %trim(ROOTPATH) + 'resources/pdfs/';
0033.00
0034.00 //PDFファイル名取得
0035.00 FILENAME = %trim(SYHNCD) + '.pdf';
0036.00
0037.00 //PDFファイルの存在チェック
0038.00 if not %vifs_pathExists(%trim(PDFPATH) + FILENAME); 4-②
0039.00 //ファイルが存在しない場合エラー
0040.00 vvOut_todsonPair('success:false,msg:PDFが存在しません。');
0041.00 else:
0042.00 //PDFファイルダウンロード 4-③
0043.00 vvOut.download = '1';
0044.00 vvOut.file = FILENAME;
0045.00 vvOut_file(%trim(PDFPATH) + FILENAME:vvOut);
0046.00 endif;
0047.00 /end-free
0048.00 p e
0049.00 /include qcpylesrc,vvNabTpl

```

図16 ファイル返却時の処理



図17 動的に作成したExcelファイル ダウンロード



ソース5 受注データExcelダウンロード(TEC040)

```

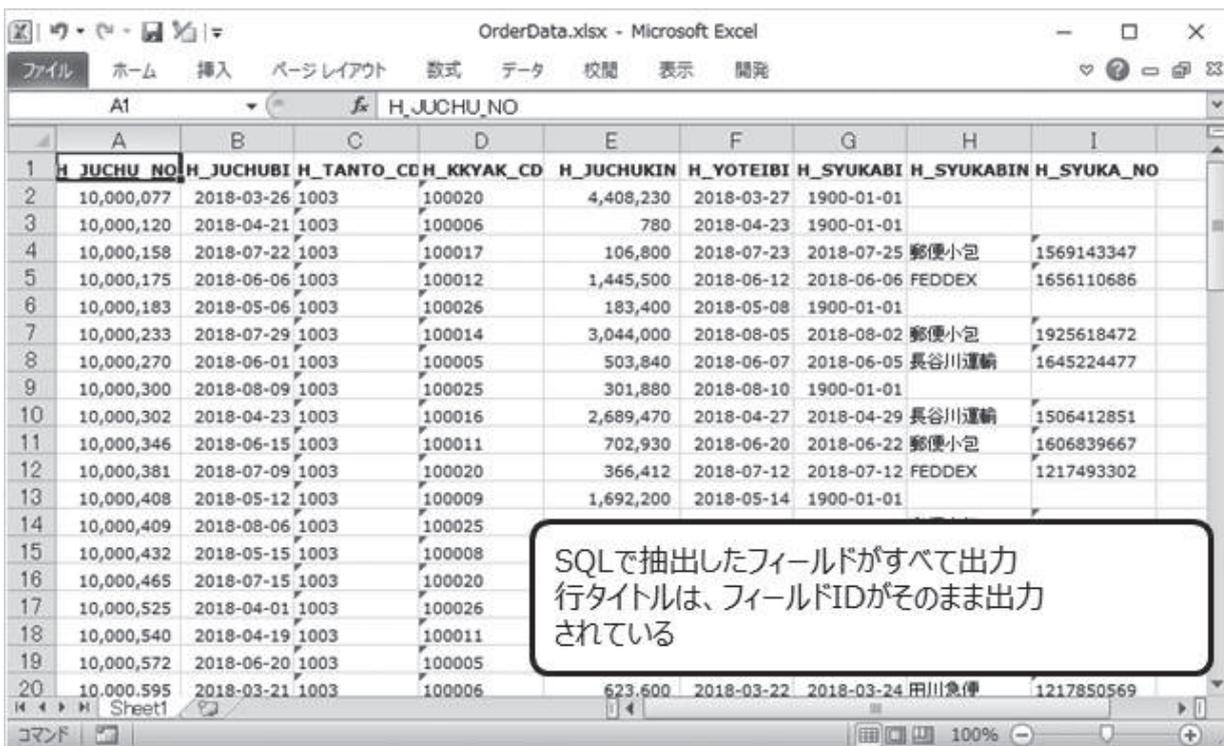
0001.00 /copy qcpylesrc.vvHspec
0002.00  **
0003.00  ** TEC040:受注データダウンロード
0004.00  **
0005.00  d TEC040      pr
0006.00  d TEC040      pi
0007.00  /define nabButton
0008.00  /include qcpylesrc.vvNabTpl
0009.00  **
0010.00  ** program start
0011.00  **
0012.00  /free
0013.00  Initialize();
0014.00  Process();
0015.00  Cleanup();
0016.00  *inlr=*on;
0017.00  /end-free
0018.00  **
0019.00  p Process      b
0020.00  d              pi
0021.00  D TANTOCD     S          4A
0022.00  D SOLSTR      S          32766A
0023.00  D*
0024.00  /free
0025.00  //フォーム上の値を取得
0026.00  TANTOCD = GetFormChar("F1_T_TANTO_CD"); //担当者CD
0027.00
0028.00  //SQL抽出条件
0029.00  SOLSTR = 'SELECT * FROM F_JUCHU_H'
0030.00  + ' WHERE H_TANTO_CD = ''' + TANTOCD + ''''
0031.00  + ' ORDER BY H_JUCHU_NO';
0032.00
0033.00  //SQL抽出結果をエクセルダウンロード
0034.00  vvOut.download = '1';
0035.00  vvOut.file     = 'OrderData.xlsx';
0036.00  vvOut_execSqlToSS(vvOut:SOLSTR);
0037.00  /end-free
0038.00  p              e
0039.00  /include qcpylesrc.vvNabTpl

```

5-①

5-②

図18 動的に作成したExcelファイル



ソース6 Excelファイルの加工を追加(TEC040)

```

0019.00  ** -----
0019.00  p Process      b
0020.00  d              pi
0021.00  D TANTOCD     S          4A
0022.00  D SQLSTR      S          32766A      6-①
0023.00  D COL         DS          LIKEDS(vvSSCol)
0024.00  D              DIM(5) INZ
0025.00  D*
0026.00  /free
0027.00  //フォーム上の値を取得
0028.00  TANTOCD = GetFormChar('F1_T_TANTO_CD');      //担当者CD
0029.00
0030.00  //SQL抽出条件
0031.00  SQLSTR = 'SELECT * FROM F_JUCHU_H '
0032.00  + ' WHERE H_TANTO_CD = ''' + TANTOCD + ''''
0033.00  + ' ORDER BY H_JUCHU_NO';      6-②
0034.00
0035.00  //表示列および列タイトル指定
0036.00  COL(1).SQLName = 'H_JUCHU_NO';
0037.00  COL(1).heading = '受注NO';
0038.00  COL(2).SQLName = 'H_JUCHUBI';
0039.00  COL(2).heading = '受注日';
0040.00  COL(3).SQLName = 'H_KKYAK_CD';
0041.00  COL(3).heading = '顧客CD';
0042.00  COL(4).SQLName = 'H_JUCHUKIN';
0043.00  COL(4).heading = '受注金額';
0044.00  COL(5).SQLName = 'H_TANTO_CD';
0045.00  COL(5).heading = '担当者CD';
0046.00
0047.00  //SQL抽出結果をエクセルダウンロード
0048.00  vvOut.download = '1';
0049.00  vvOut.file      = 'OrderData.xlsx';      6-③
0050.00  vvOut_execSqlToSS(vvOut:SQLSTR:%addr(COL):%elem(COL));
0051.00  /end-free
0052.00  p              e
0053.00  /include qcpylesrc.vvNabTmpl

```

図19 改良後に作成されたExcelファイル

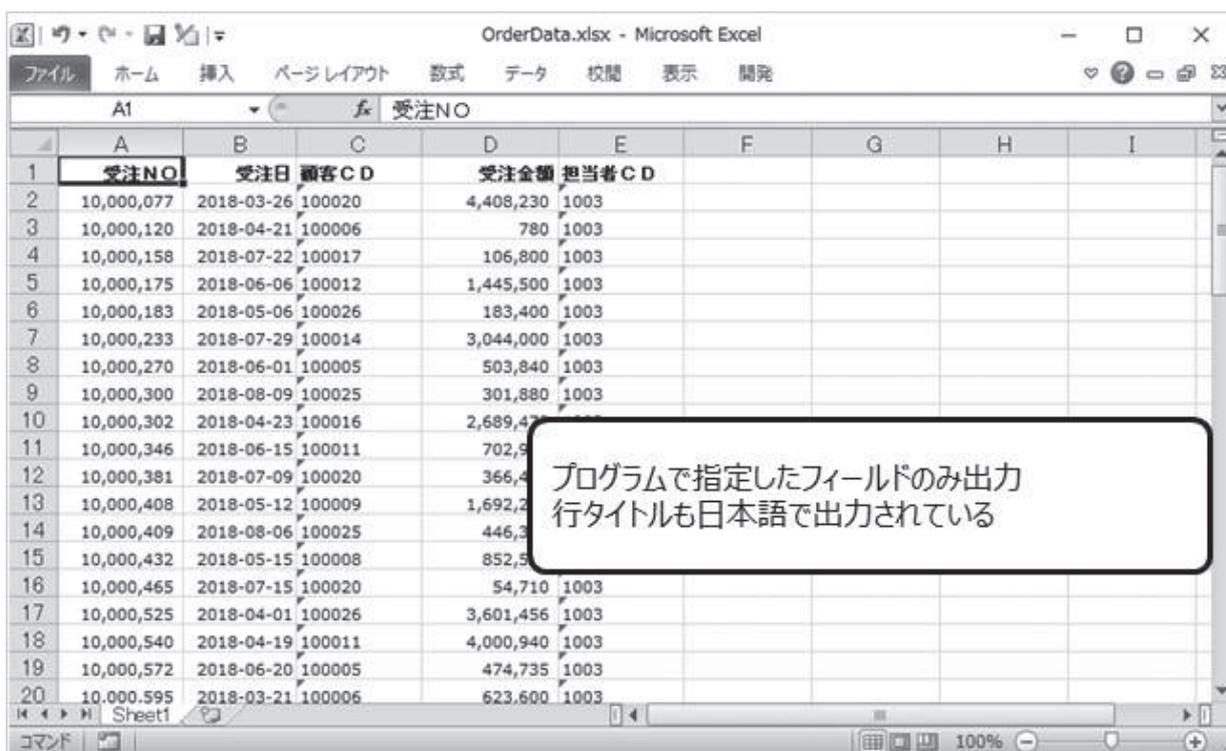


図20 メール送信



図21 RPGプログラム実行時に追加パラメータをセット



図22 メール受信結果



ソース7 メール送信(TEC050)

```

0001.00 /copy qcpylesrc,vvHspec
0002.00 ** -----
0003.00 ** TEC050:メール送信
0004.00 ** -----
0005.00 d TEC050      pr
0006.00 d TEC050      pi
0007.00 /define nabButton
0008.00 /define includeEMAIL 7-①
0009.00 /include qcpylesrc,vvNabTpl
0010.00 ** -----
0011.00 ** program start
0012.00 ** -----
0013.00 /free
0014.00 Initialize();
0015.00 Process();
0016.00 CleanUp();
0017.00 *inlr=*on;
0018.00 /end-free
0019.00 ** -----
0020.00 p Process      b
0021.00 d              pi
0022.00 D TANTOCD     S      4A
0023.00 D MAILTO     S      50A
0024.00 D TMPPATH    S      20A
0025.00 D SID        S      64A
0026.00 D SOLSTR     S      32766A
0027.00 D*
0028.00 /free
0029.00 //フォーム上の値を取得
0030.00 TANTOCD = GetFormChar('F1_T_TANTO_CD'); //担当者CD
0031.00 MAILTO = vvIn_char('MAILTO'); //送信アドレス 7-②
0032.00 SID = vvIn_char('sid'); //セッションID
0033.00
0034.00 //ファイル保存先取得
0035.00 TMPPATH = vvUtility_getValenceSetting('TEMP_PATH');
0036.00
0037.00 //SQL抽出条件
0038.00 SOLSTR = 'SELECT * FROM F_JUCHU_H'
0039.00         + ' WHERE H_TANTO_CD = ''' + TANTOCD + ''''
0040.00         + ' ORDER BY H_JUCHU_NO';
0041.00
0042.00 //SQL抽出結果をIFS上に出力 7-③
0043.00 vvOut.download = 'F';
0044.00 vvOut.file = Xtrim(TMPPATH) + SID + '.xlsx';
0045.00 vvOut_execSqlToSS(vvOut:SOLSTR);
0046.00
0047.00 //メール内容作成 7-④
0048.00 vvMail.from = 'info@migaro.co.jp';
0049.00 vvMail.to = MAILTO;
0050.00 vvMail.subject = '担当者別受注一覧報告';
0051.00 vvMail.body = '担当者CD=' + TANTOCD + '<br>'
0052.00             + 'の受注一覧を報告いたします。';
0053.00 vvMail.attachment = vvOut.file;
0054.00 vvMail.attachAlias = 'OrderData.xlsx';
0055.00
0056.00 //メール送信 7-⑤
0057.00 if not vvMail_send(vvMail);
0058.00     vvOut_toJsonPair('success:false,msg:送信エラー');
0059.00 else;
0060.00     vvOut_toJsonPair('success:true,info:送信しました');
0061.00 endif;
0062.00
0063.00 //出力ファイルを削除 7-⑥
0064.00 vvIfs_deleteFile(vvOut.file);
0065.00
0066.00 /end-free
0067.00 p      e
0068.00 /include qcpylesrc,vvNabTpl

```