

## [Valence]

Valence 最新バージョン  
進化のポイント

## 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 最新バージョンの進化点
3. Fusion5250 を使用したアプリの融合
4. App Builder の強化されたアプリ開発機能
5. さいごに

## 1. はじめに

Valence は、IBM i を Web 環境で活用できるモダナイゼーション開発・運用ツールである。Valence での Web アプリ開発は、以前は Sencha と呼ばれる JavaScript フレームワークを使用する方法のみであった。

しかし2018年8月に登場した「Valence 5.2」にて、ローコード開発ツール「Valence App Builder」が追加されたことにより、アプリの開発効率が飛躍的に向上した。

ウィザードを使用したシンプルな3ステップの作成手法でアプリ開発ができるので、簡単なものであれば、ものの数分でアプリを作成可能である。

App Builder は基本的にノンコーディングで簡単にアプリを作成できるが、複雑な業務ロジックを組み込みたい場合は、RPG を組み合わせることもできる。

App Builder の概要および RPG 連携テクニックについては、2019 年発行の

テクニカルレポートにある『Valence App Builder RPG 連携テクニック』で詳しく紹介しているので、参考にさせていただきたい。

この Valence だが、2020 年 6 月にバージョンアップし、Valence5.2+ (プラス) となった。さらに 2020 年 9 月の本稿執筆時点では、正式発表前ではあるが、まもなく Valence6.0 のリリースも予定している。

本稿では、Valence5.2+ 以降の最新バージョンにおける製品進化ポイントならびに開発ツール App Builder の強化されたアプリ開発テクニックを紹介する。

## 2. Valence 最新バージョンの進化点

Valence 最新バージョンの主な進化点は、以下のとおりである。

- (1) Valence Portal の機能強化
- (2) Web エミュレータ Fusion5250 の追加

## (3) App Builder アプリ開発機能の強化

(1) については、Valence Portal のユーザー認証機能や権限制御が強化された。具体的には、「2ファクタ認証」によるセキュリティ強化と、「NAB 権限」による App Builder の作成権限制御の強化である。

2ファクタ認証とは従来のパスワードによる認証に加え、利用者個人が所持するスマートフォン等の認証アプリを使用したデジタル認証コードを組み合わせることで、セキュリティを向上させる仕組みである。Google や Amazon 等でも使用されているセキュリティ強化の一般的なアプローチと同じである。【図1】

2ファクタ認証を有効にすれば、たとえ Valence にサインオンするユーザー ID とパスワードの情報が外部に漏洩したとしても、ユーザー本人所持のスマートフォン上で、都度発行される確認コードを正しく入力しない限り、Valence にサインオンできなくなる。

Valence で機密性の高いデータを扱う

## 図1 Valence 2ファクタ認証

ステップ1：ユーザー、パスワードでの認証



The image shows the Valence login interface. At the top is the 'valence' logo. Below it is a form with two input fields: 'ユーザー名' (Username) containing 'OZAKI' and 'パスワード' (Password) with masked characters. A 'ログイン' (Login) button is at the bottom. A link for 'パスワードを忘れたか?' (Forgot password?) is visible.

スマートフォンにインストールした認証アプリ  
(例：Google Authenticator)



ステップ2：確認コードによる追加認証



The image shows the Valence 2FA confirmation screen. It prompts the user to '確認コードを入力' (Enter confirmation code) and '認証アプリの確認コードを表示してください' (Display the confirmation code from the authentication app). There are six input boxes for the code. A link for 'デバイスを紛失しましたか?' (Device lost?) is at the bottom.

認証アプリに表示された確認コードが一致した場合のみ、ログイン可能となる

## 図2 Valence NAB権限

Valence Portal設定 - ユーザー 編集画面



The image shows the 'ユーザーを編集する "MIGARO"' (Edit user "MIGARO") screen in the Valence Portal. It contains several form fields for user details like 'ユーザーID', 'IBM i ユーザー', '名', 'Mid', '姓', 'Eメール', and 'パスワード'. Below these is a table for permissions. The 'NAB 権限' (NAB Permissions) column is circled in red.

グループ	環境	許可されたアプリケーション	許可されないアプリク	NAB 権限
データソース	<input type="radio"/> 権限無し <input type="radio"/> 編集	<input checked="" type="radio"/> すべて		
ウィジェット	<input type="radio"/> 権限無し <input type="radio"/> 編集	<input checked="" type="radio"/> すべて	<input checked="" type="checkbox"/> 編集機能付きウィジェットへのアクセス可	
アプリケーション	<input type="radio"/> 権限無し <input type="radio"/> 編集	<input checked="" type="radio"/> すべて		

ユーザーごとに Valence App Builderの作成権限が付与できる。データソース、ウィジェット、アプリケーション それぞれに対して権限レベルが設定可能。(ウィジェットについては、さらに編集機能の無い照会系ウィジェットのみ許可することも可能)

必要がある場合、この2ファクタ認証を有効にしておく効果的である。

次にNAB権限だが、これはValenceのユーザー登録に追加された機能である。

これまではValenceのグループ設定により、Valence PortalからApp Builderメニューを表示する・しないを制御することで、アプリ開発が可能なユーザーを制限することは可能であった。

しかし今回、NAB権限が追加されたことで、ユーザーごとにApp Builderの作成権限レベルを制御できるようになった。

たとえば、管理者があらかじめ用意したデータソースを一般ユーザーに提供し、一般ユーザーにはウィジェットやアプリケーションの作成だけを許可するといった使い方が可能である。あるいはウィジェットの作成権限はユーザーに付与するが、データを更新するためのウィジェット (Edit Grid) は不許可にすることで、データの参照目的のみでApp Builderの作成を一般ユーザーに開放するといった使い方もできる。【図2】

(2)のWebエミュレータFusion5250は、Valence5.2+で追加された新機能である。Valence PortalのメニューからFusion5250アイコンをクリックすると、ブラウザ上で5250エミュレータを実行できる。

これは従来のクライアントアクセスのような専用ソフトに依存せず、ブラウザだけで実行できるので、Windowsクライアントだけでなく、たとえばMacクライアントでもエミュレータが利用できる。【図3】

Fusion5250は単にブラウザで動作するエミュレータというだけでなく、Fusion5250という名前が示すとおり、他のValence機能との融合がポイントである(詳細は3.で紹介する)。

(3)は、ローコード開発ツールApp Builderの強化である。Valence最新バージョンでは、新たに2つのウィジェットが追加された。「タイトル」ウィジェットと「タイムライン」ウィジェットである。

タイトルウィジェットは、データソース上のデータをタイトル形式のパネルに表示す

る部品である。会社の受付システム等で見られるタッチパネルを使って、社員を選択するような画面が作成できる。【図4】

一方のタイムラインウィジェットは、データソースのデータを時系列に並べて表示する部品である。たとえば、売上日ごとにどんな商品がよく売れるのか、季節ごとにどんな商品が売れ筋なのか、など横軸の時系列に対してデータをプロットするようなインタラクティブなタイムラインを表示できる。【図5】

また最新バージョンでは、新たにアプリケーション内で扱えるアプリ変数機能が追加された。アプリケーション内の複数ウィジェット間で変数値を共有でき、さらにRPGで取得した値をクライアント側で保持することも可能になった。【図6】

ほかにもセキュリティ機能の強化や、スクリプト実行機能、RPGの新しいテンプレートなど多彩な機能強化がなされている(詳細は4.で紹介する)。

### 3. Fusion5250を使用したアプリの融合

PC5250エミュレータを日本語環境で使用する場合、ホストコードページに[930:日本語(カタカタ)]、あるいは[939:日本語(拡張ローマ字)]を使用するのが一般的である。ValenceのWebエミュレータであるFusion5250でも、同様にコードページが指定できる。【図7】

Valence Portalより[ポータル管理]を起動し、サブメニューより[言語]を開くと、言語ごとにFusion5250で使用するコードページ設定が表示されるので、[930-Japan Katakana Extended]、あるいは[939-Japan Latin Extended]を選択すればよい。

Fusion5250は、デフォルトでは一般的なエミュレータと同様に、黒背景・緑文字が基調のいわゆるグリーンスクリーンで表示される。IBM iユーザーにとっては馴染みだが、Valenceのほかの画面と比べると異質に感じるかもしれない。

Fusion5250には、テーマ設定が用意されている。3種類から選択できるが、このうちの[Valenceテーマ]は他のValenceアプリとの親和性がよいので、こちらを選択するとよい。【図8】

Fusion5250の特徴の1つがマクロ機

能である。これは、エミュレータ上でのユーザー操作をスクリプトとして定義しておき、コマンド操作を自動化する仕組みである。

マクロはJSON形式で記述し、Valenceインスタンス内のIFS上(/resources/fusionmacros/)に保管すればよい。ここではマクロの例として、IBM i上のCLプログラムをCALLし、特定のアプリを起動する例を紹介する。ここで使用するマクロファイルは、【ソース1】である。

マクロファイルは、「note」「showProgress」「steps」という3つの要素で構成される。noteは単なるメモで、動作に影響は与えない。showProgressは、マクロ実行中の途中画面を表示する、表示しないを切り替える。falseを指定すると、すべてのマクロが実行されたあとに画面が表示される。stepsは、実行する個々の処理ステップである。

【ソース1】のマクロは、エミュレータの7桁目、20行目、つまりエミュレータのコマンド入力ラインにカーソルをセットし、CLコマンドをタイプしたあと、[実行](ENTER)キーを押下するという動作を表している。

Fusion5250からマクロを実行する方法は、Valence Portalの[ポータル管理]→[アプリ]より、新規アプリを追加し、パス欄に以下の指定を行えばよい。【図9】

/build/production/Fusion/index.html?macro=[マクロファイル名](※拡張子.jsonは記述不要)

以上で完了である。これでValence Portal上に、新しいメニューが追加される。このメニューをクリックすると、Fusion5250が立ち上がると同時にマクロが実行され、既存のPC5250アプリがブラウザ上に立ち上がる。Valence PortalとPC5250アプリがシームレスに融合していることがわかる。【図10】

なお、今回はマクロをJSONファイルで直接作成する方法を紹介したが、マクロの記録機能を使用すると、Fusion5250上の操作を自動記録し、そのままValence Portalに新たなアプリとして追加できる。Fusion5250画面左下にある[REC]ボタンをクリックするだけで記録できるので、試してほしい。

図3 Fusion5250

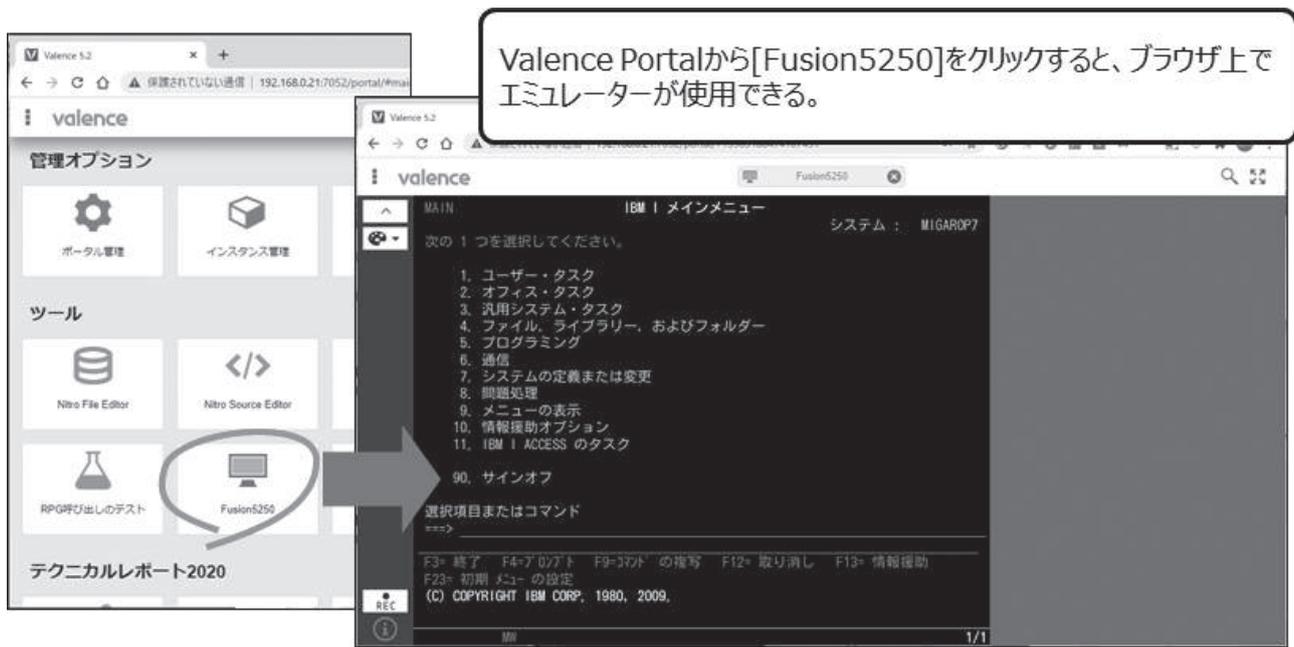


図4 タイルウィジェット



またマクロを使用した Fusion5250 は、Valence Portal からだけでなく、App Builder で作成するアプリからも呼び出せる。アプリケーション作成時に、「動作内容」画面で定義するアクションとして、Fusion5250 を指定する。もちろんマクロを含められるのだが、その際にパラメータとして、ウィジェット上のフィールド値やアプリ変数の値を渡すこともできる。【図 11】

つまりウィジェット上で選択したレコードのキーをパラメータにして、マクロを実行することにより、PC5250 アプリ上のキー項目に値を自動入力したり、RPG 側にパラメータとして値を渡したりできる。

この仕組みを使用すれば、App Builder と Fusion5250 とのシームレスなアプリ融合が可能になる。【図 12】

## 4. App Builderの強化されたアプリ開発機能

App Builder と RPG との連携例として、商品マスタメンテナンスの登録画面の作成方法を 2019 年発行のテクニカルレポートで紹介した。【図 13】

ここでは、ボタンをクリックしたときに実行される RPG プログラムを紹介した。ボタンクリックは、VALENCE52/QRPGLESRC (EXNABBTN) のテンプレートを使用する。【図 13】のサンプルで作成したプログラムが、【ソース 2】である。

この EXNABBTN プログラムは、Valence5.2+ 以降で記述方法が一部変更された。ポイントは、【ソース 2】の 1-①～③の部分である。

以前は、1-①のように d 仕様書としてプロシージャー宣言が必要だったが、これは不要になった。以前のバージョンで作成されたソースを再コンパイルする際には、この部分を削除する必要があるので注意してほしい。

1-②、1-③の部分は、レスポンスをクライアント側に返却する処理であるが、以前は、vvOut\_toJsonPair という Valence の汎用 API を直接記述する必要があった。最新バージョンでは、EXNABBTN 用の専用 API が追加されており、その中には、レスポンスを返却

する SetResponse という API が追加されている。【図 14】

これにより【ソース 3】のように、クライアントへのレスポンスの返却がよりシンプルに記述できるようになった（ただし、従来の記述方法でも正しく動作する）。

さらに専用 API になったことで、レスポンス結果として、クライアント上の特定機能を使用不可に変更するような制御が可能になった。

次に、ファイルのアップロードについてである。旧バージョンの Valence5.2 でも RPG Toolkit を活用することで、CSV ファイルや Excel ファイルのダウンロードを実装できたが、これまでファイルのアップロードはサポートされていなかった。

今回の最新バージョンでは、アップロードできるようになり、画像データの登録やデータの一括登録などが可能になった。今回紹介する例は、商品マスタメンテナンスにおける CSV データ取り込み処理の追加である。【図 15】

この例では、ファイルのアップロードは App Builder にて動作設定を定義し、RPG プログラムを組み合わせで作成する。App Builder では「RPG プログラム呼び出し」を追加し、設定画面を開く。

そしてプログラム実行前処理として、「Prompt For User Information」を選択し、パラメータを追加する。そしてパラメータ名に file、タイプに Upload を選択すればよい。もしアップロードするファイルの拡張子を制限する場合は、さらに Valid Extensions に許可する拡張子を指定する。【図 16】

そして RPG 側では、Valence に用意された RPG Toolkit の中にある vvIn を使用する。【ソース 4】

アップロード先は、vvIn.path に IFS ディレクトリを指定する。任意のディレクトリを指定可能だが、今回は、2-①のように Valence が保持しているテンポラリーディレクトリ (TEMP\_PATH) を指定している。2-②のように、vvIn\_file を使用するとファイルがアップロードされる。なお第 2 引数は、常に “\* NULL” を渡す。

これで IFS 上にファイルがアップロードされるわけだが、今回はこの CSV ファ

イルを使用して、物理ファイルである商品マスタ (MSYHIN) の内容を置き換える。IBM i には、CPYFRMIMPF というコマンドがあるので、今回はこのコマンドを RPG から実行することとした。

2-③のように “QCMDEXEC” を使用すると、RPG から CL コマンドが実行できる。

以上で CSV アップロード処理は完成である。CSV ファイルを使用したデータの一括登録は利用頻度が高いと思うので、今回の内容をもとに活用してほしい。

次に、Valence の新しいセキュリティ機能を紹介する。これを使用すれば、作成したアプリの中の一部機能を特定ユーザーのみ実行可能にできる。たとえば、先ほどの CSV データ取り込み処理を、特定のグループ (今回は「経理グループ」(ID:1010) を例とする) の場合のみ使用できるように権限設定する。【図 17】

セキュリティ機能も、App Builder の設定と RPG との組み合わせで実装する。

まず App Builder だが、アプリケーション作成画面に新たに追加された「セキュリティ」ボタンをクリックする。セキュリティ設定一覧が表示され、制御したい機能に対し任意の名前 (機能名) を付与し、あとは実行したい RPG のプログラム ID を指定すればよい。【図 18】

【図 18】で指定したプログラムは、VALENCE52/QRPGLESRC (EXNABSTART) というテンプレートを使用して RPG ロジックを作成する。App Builder で設定した機能名を使用不可にする場合は、disableFeature を呼び出せばよい。【ソース 5】

このサンプルでは、3-①のように RPG Toolkit に用意された vvSecurity API を使用して、実行中のユーザーが指定したグループに含まれるかを判断し、含まれない場合に disableFeature を実行している。

なお今回のサンプルでは使用していないが、現在実行中のユーザーはこのテンプレートプログラム内で、gUser という変数に格納されているので、これを使用することも可能である。

次に、入力画面等で使用する Form ウィジェットの機能を紹介します。ヘル

図5 タイムラインウィジェット

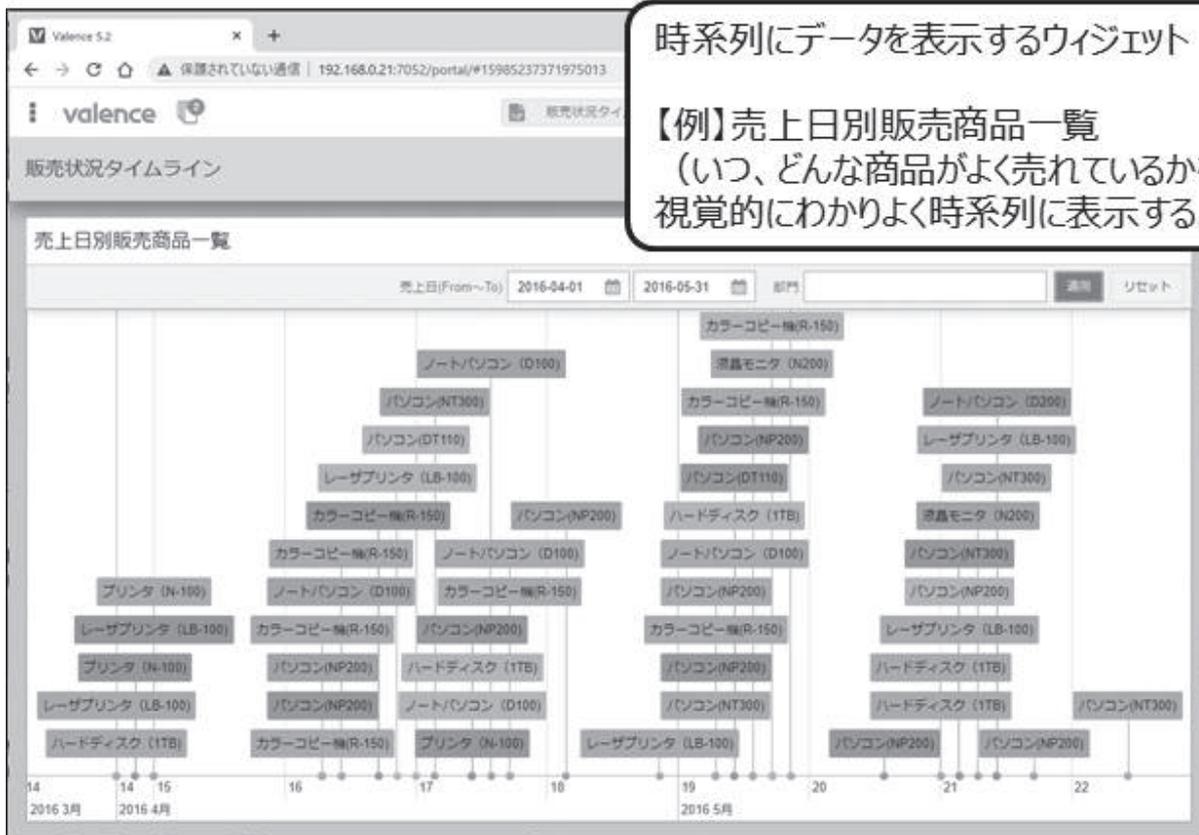


図6 アプリ変数

アプリケーション作成画面

アプリケーション内で共通に保持できる変数が定義できる。  
RPGプログラムから、値をセットしたり、  
RPGプログラムへ値を渡すことが可能。  
複数のウィジェット間で変数の共有も可能。

Name	Initial Value (Optional)	Pull from URL Parameter
ImageFilePath ①		⊗
nabInfo ①		
nabMsg ①		
nabMsgTitle ①		
nabPromptBeforeClose ①		

**Step 1**  
任意のアプリ変数を追加します。オプションで初期値を設定する。かつまたは、URL/パラメータによって値を設定します。

**Step 2**  
[動作内容]セクションのイベント（行のクリック、ボタンのクリックなど）でアプリ変数の値を設定します。

**Step 3**  
次のいずれかの領域でアプリ変数を使用します。

- 動作内容セクション：アプリ変数は、フィルターウィジェット、アプリバータイトルの設定、およびアプリの起動アクションで参照できます。
- RPGプログラム呼び出し：すべてのアプリ変数は、getAppVarを介してRPGプログラムで取得できます。setAppVarを使用して、アプリ変数を変更します。
- ウィジェット設定：ウィジェットにアタッチされている設定アイコン（ホバー時に表示可能）をクリックし、左側の[設定]セクションに表示される[アプリ変数にリンク]ボタンをクリックします。

パープログラムと呼ばれる機能である。Valence5.2では、ボタン等をFormウィジェットに配置し、そのボタンをクリックすることでRPGプログラムを実行する処理だけがテンプレートで用意されていた。しかし最新バージョンで追加されたヘルパープログラムを使用すると、リアルタイムな応答性を備えるインタラクティブな画面が作成できる。

今回紹介する例は、入庫入力を行う画面である。アプリ起動時に入庫日欄には、システム日付が初期セットされる。商品コードの値を変更したタイミングで、自動的に商品コードの存在チェックを行い、商品マスタに当該レコードが存在する場合は商品名を表示し、存在しない場合はエラーを表示する。【図 19】

ヘルパープログラムも、App BuilderとRPGの組み合わせで実装する。Formウィジェットの設定画面に追加されたヘルパープログラム欄に、実行したいRPGプログラムのIDを指定する。【図 20】

あわせてヘルパープログラムが呼び出させるタイミングを指定する。設定はこれだけである。実際の処理はすべてRPGで記述する。ヘルパープログラムは、VALENCE52/QRPGLESRC (EXNABFHLP) というテンプレートをもとに作成する。今回作成したTEC20PG30のソースは、【ソース 6】である。

このプログラムには、4-①、4-②の2つの処理がある。4-①は、初期表示時の処理であり、実行モードを示すgMode変数が“formRender”のとき(4-③)に、起動時を表している。今回は、4-④のようにSetValueを使用することで、フォーム上に初期値をセットしている。

gModeがブランクの場合はフィールド変更時(4-②)を表しており、4-⑤のようにgField変数に対象のフィールド名がセットされる。今回は、4-⑥のようにvvIn\_Charを使用して、フォーム上の値を取得し、商品マスタの存在チェックを行い、存在する場合は取得した商品名をフォームにセットする。存在しない場合は、4-⑦のようにSetErrorを使用して、エラーメッセージをフォームにセットしている。

またフォーカスを商品CD欄に再セットしている。このようにヘルパープログラムの追加により、Formウィジェット

上の画面制御が大幅に向上したことがわかる。

最後に、App Builderにおけるスクリプト機能を紹介する。最新バージョンでは、RPGによるロジック追加に加え、クライアント側のロジック追加としてスクリプトが記述できるようになった。

これにより、RPGに依存しないような画面上の制御も簡単に実装できる。スクリプトを追加する際は、動作内容の中からスクリプトアクションを追加すればよい。

【図 21】の枠内にスクリプトを記述すればよいのだが、スクリプトの実装例として、先ほどの入庫入力画面にQRコード読み込み機能を追加する。Valenceは、PCを使用したブラウザでの実行に加え、専用アプリを使用したモバイルアプリとしての実行も可能である。モバイルアプリの特徴は、デバイスの機能を活用できることである。今回は、デバイスのカメラ機能を活用したQRコード/バーコード読み込みの追加となる。実装例は【ソース 7】となる。

このスクリプトの追加により、モバイルアプリで実行したときだけ、カメラが起動するようになり、QRコードやバーコードが読み取れる。【図 22】

なお、PCブラウザからボタンをクリックして実行しようとした場合は、エラーメッセージが表示されるようになっている。

スクリプトソースのうち、取得したコードの値をForm上のどのフィールドにセットするか部分(下線部)だけを変更すれば、他のプログラムにもそのまま組み込めるのでぜひ活用してほしい。

## 5. さいごに

本稿では、バージョンアップした最新版Valenceの進化点を紹介してきた。

これまでは、既存のPC5250アプリを専用エミュレータ上で実行し、App Builderで作成したWebアプリをブラウザ上で実行するといったように、それぞれの仕組みを個別に使用するしかなかった。

今回のバージョンアップにより、これら異なる種類のアプリが融合され、1つ

のブラウザ上でシームレスに使用できるようになったので、ユーザーの利便性が大幅に向上した。

またApp Builderについても、従来からのシンプルな開発手法という特徴はそのままに、大幅に機能拡張されたので、より本格的なアプリケーションの作成にも対応できる。ぜひ本稿を参考に、さらなるValenceの活用をご検討いただきたい。

M

図7 Fusion5250 ホストコードページ

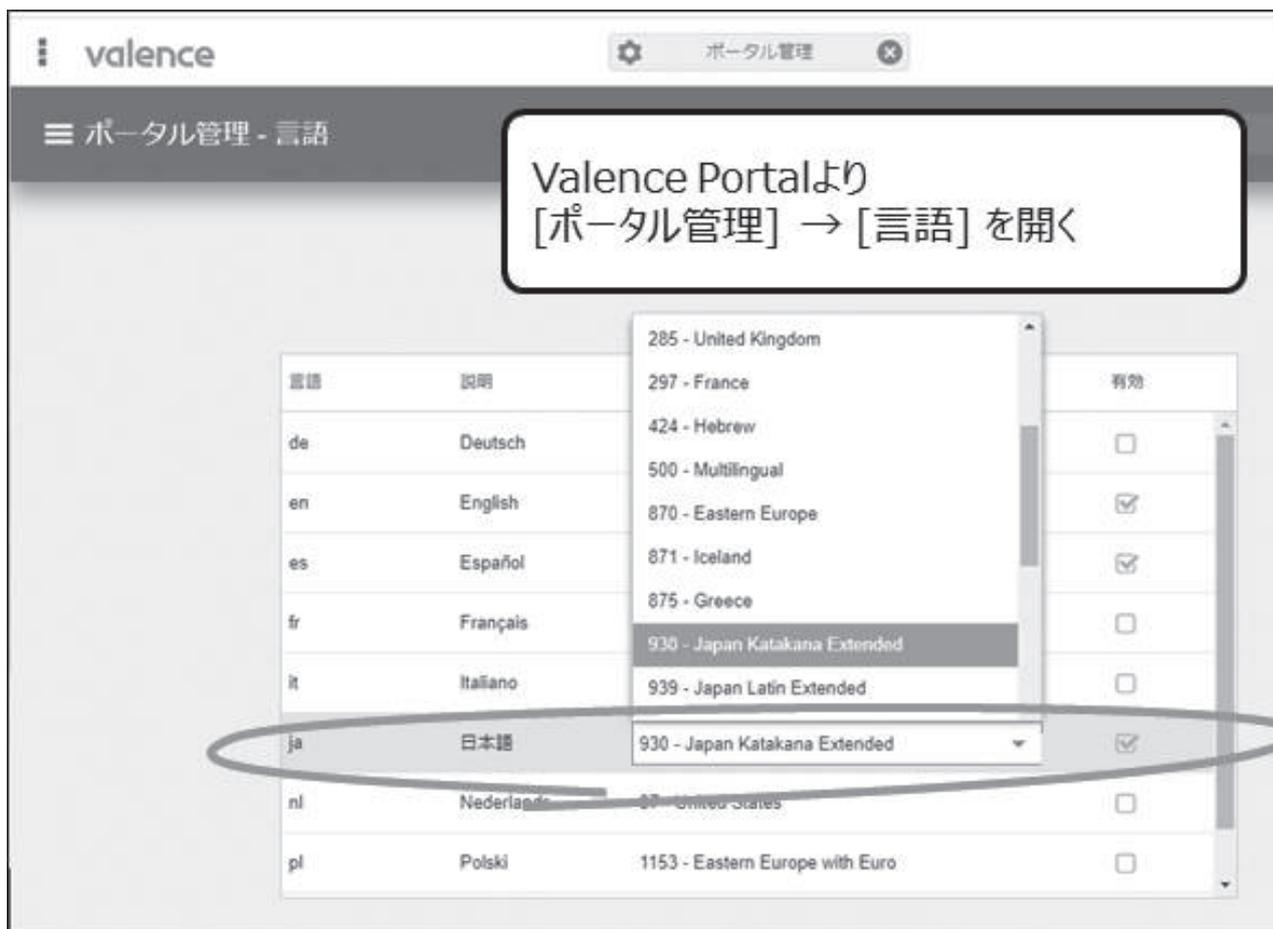
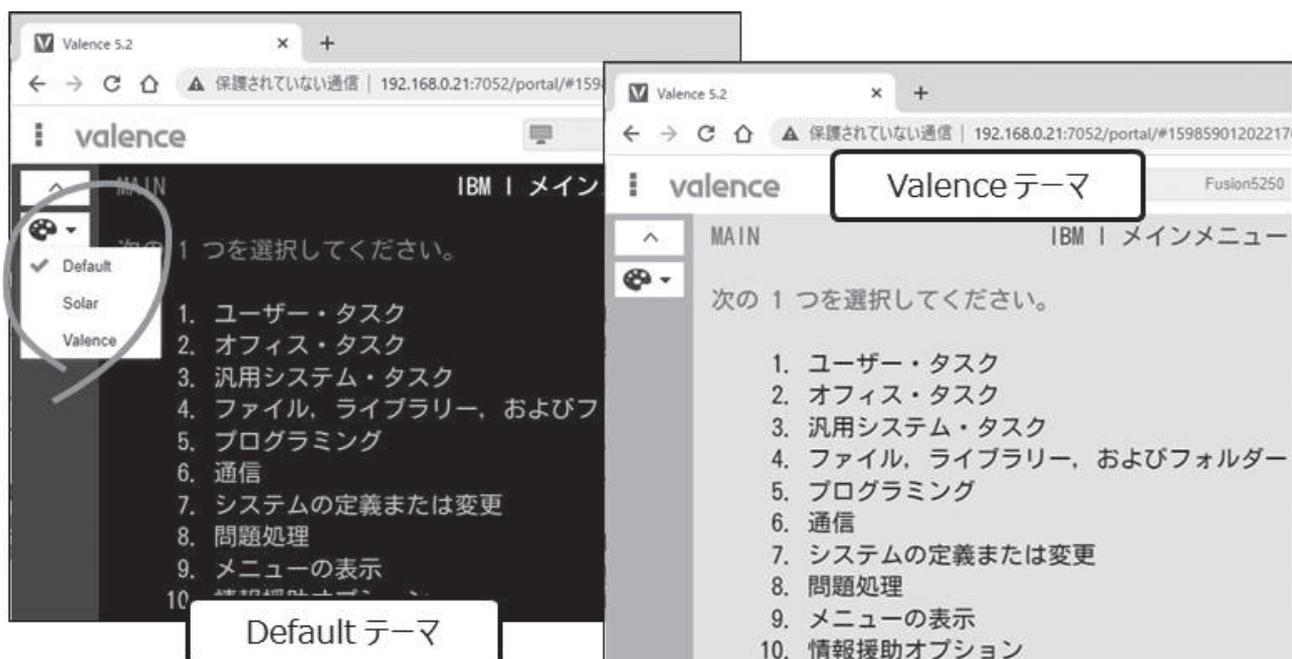


図8 Fusion5250 テーマの設定



```

{} 5250SampleApp.json ●
V: > resources > fusionmacros > {} 5250SampleApp.json > ...
1   {
2       "note": "iSite給与メニュー起動",
3       "showProgress":false,
4       "steps": [
5           {
6               "col":"7",
7               "row":"20",
8               "type":"CALL PXOLIBSO/START"
9           },
10          {
11              "action":"ENTER",
12              "control":false,
13              "shift":false
14          }
15      ]
16  }
    
```

図9 Fusion5250 マクロの呼び出し

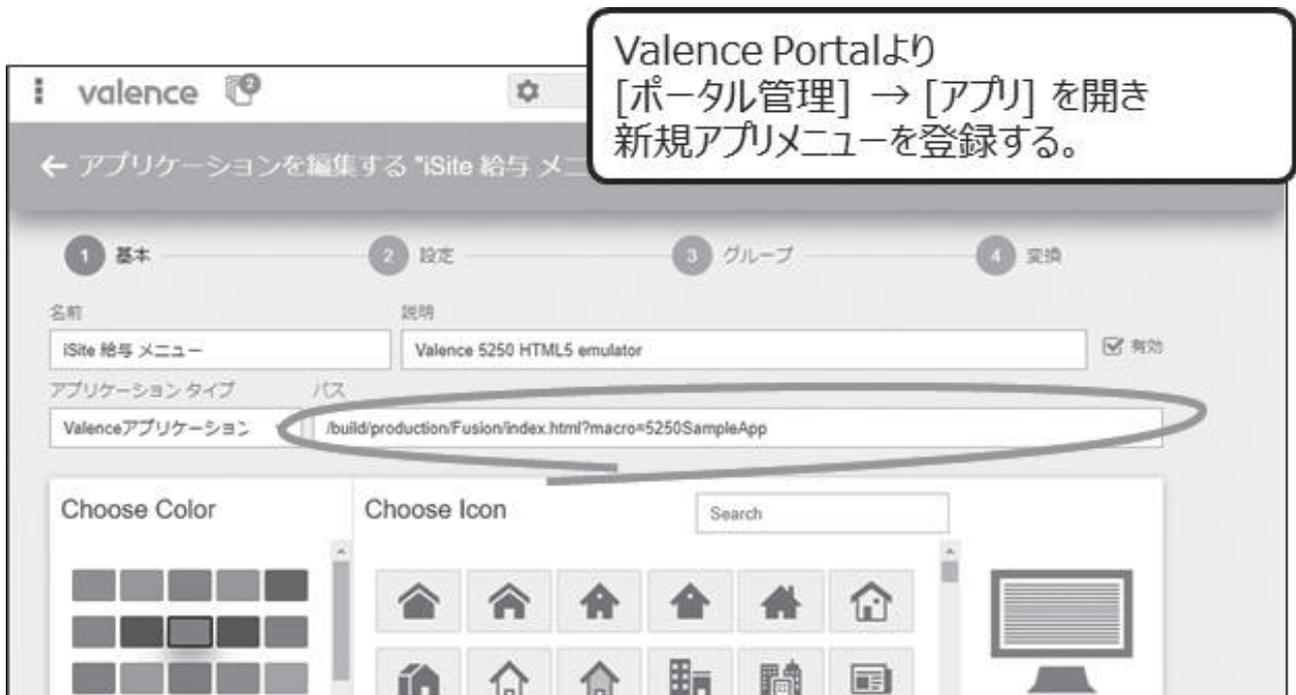


図10 Fusion5250 マクロ実行

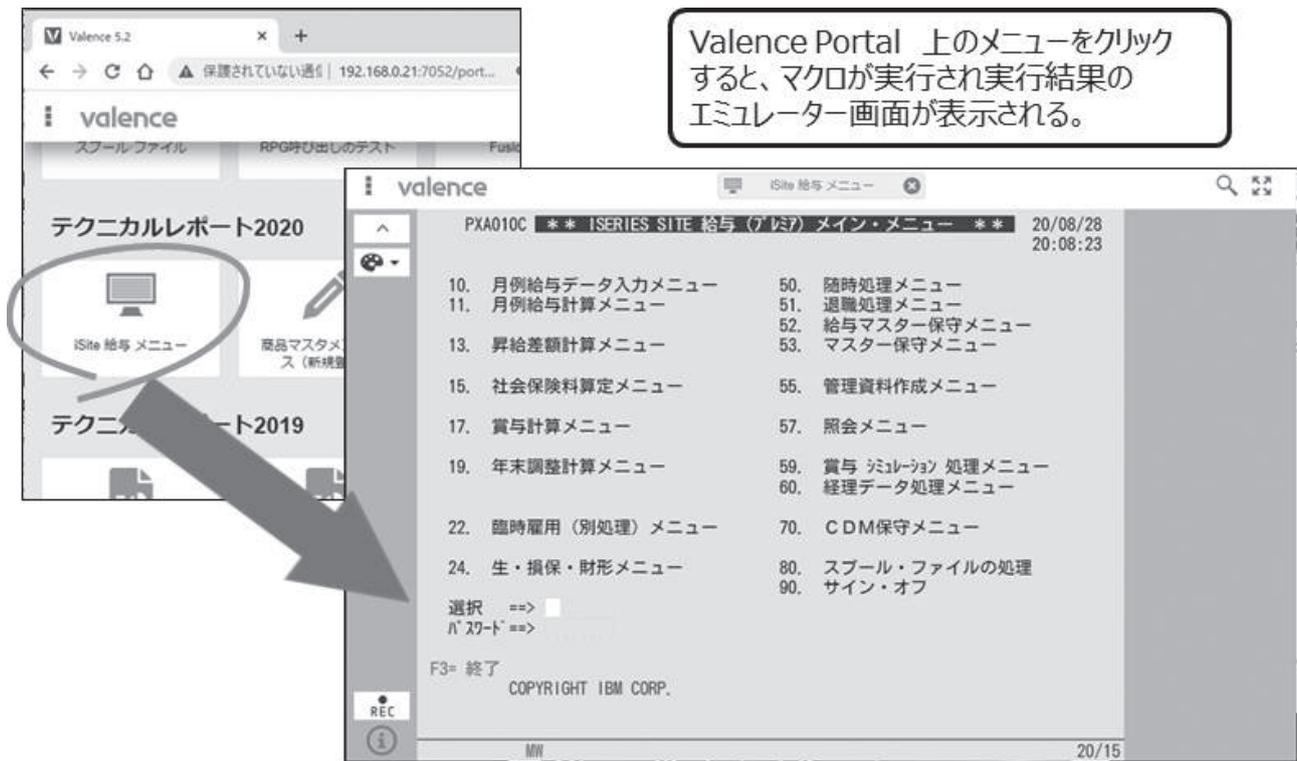


図11 Fusion5250 マクロパラメータ受け渡し

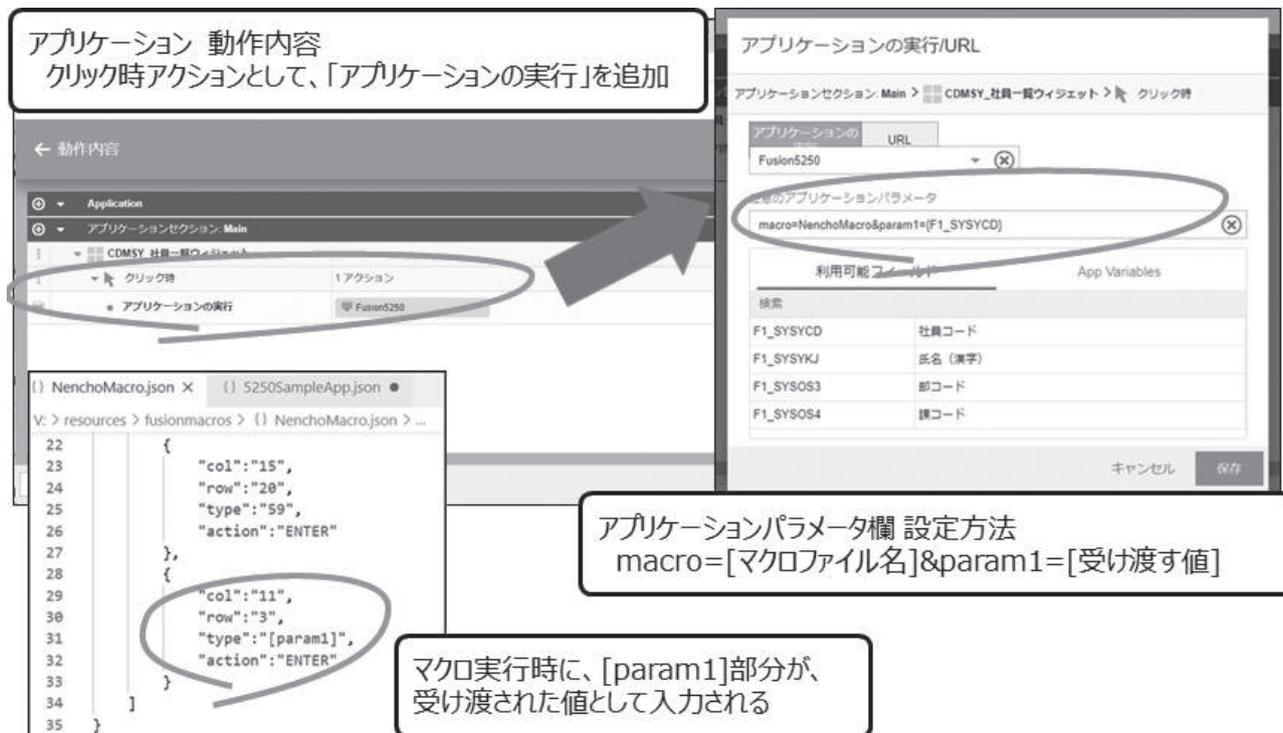
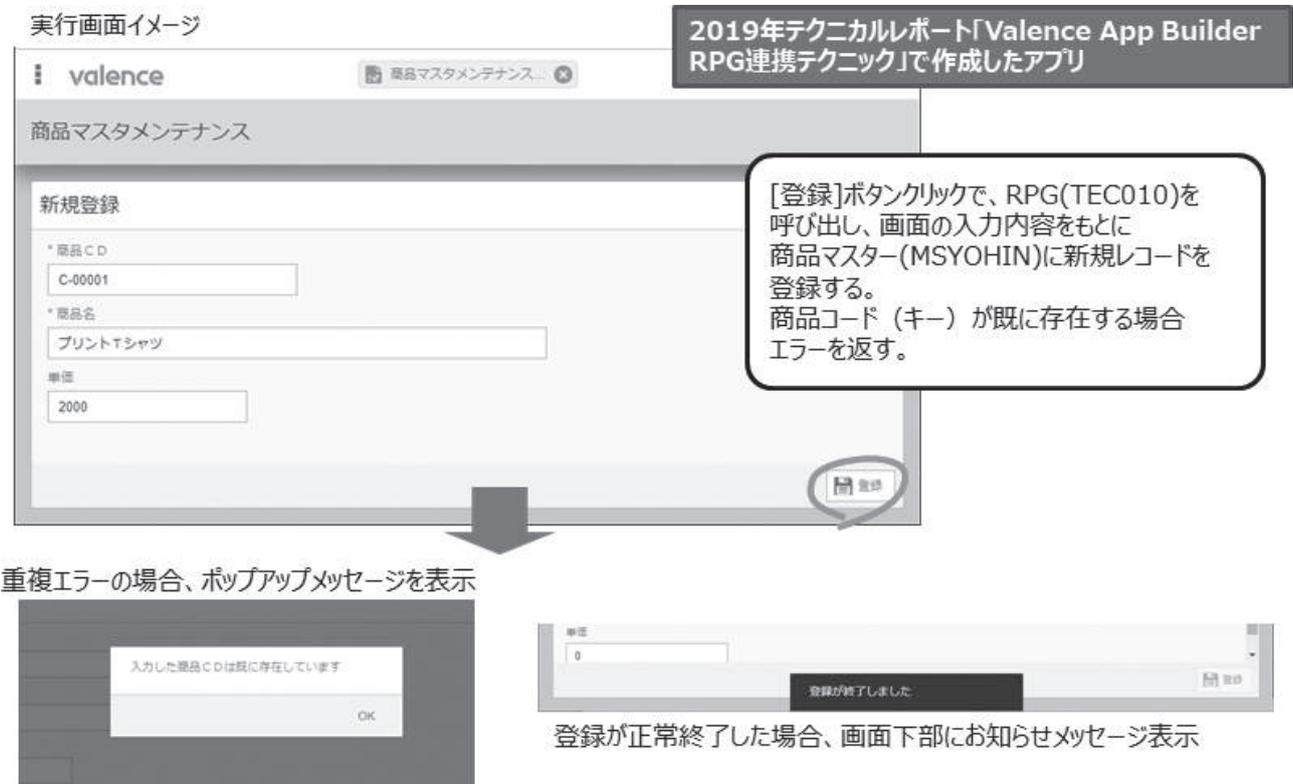


図12 App BuilderとFusion5250の融合



図13 ボタンクリック サンプルプログラム



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 pr
0012.00 d TEC010 pi 1-① 削除
0013.00 /define nabButton
0014.00 /include qcpylesrc,vvNabTmp1
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 D*
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
0040.00 C *IN81 IFEQ *OFF
0041.00 C* 1-②
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 1-③
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,vvNabTmp1

```

81

図14 ボタンクリック(EXNABBTN)新API

新API一覧

API	I/O	概要
GetAppVar	I	アプリ変数を取得
SetAppVar	O	アプリ変数をセット
SetResponse	O	レスポンスを返却

SetResponse パラメーター一覧

第1パラメータ	概要	記述例
success	処理の成否をセット	SetResponse('success':'true');
info	画面下部にトーストメッセージを表示	SetResponse('info':'終了しました。');
clearSelection	Grid行選択をクリア	SetResponse('clearSelection':'true');
refresh	データソース再読込	SetResponse('refresh':'true');
applyData	ウィジェットのフィールドに値をセット	SetResponse('applyData':'F1_NAME', '尾崎 浩司');
disableFeature	対象機能を使用不可にする	SetResponse('disableFeature':'FuncSave');
enableFeature	対象機能を利用可能にする	SetResponse('enableFeature':'FuncSave');

ソース3 SetResponseメソッド(TEC010を変更)

```

0038.00 C *IN81 IFEQ *OFF 1-②
0039.00 C*
0040.00 /free
0041.00 //エラーメッセージを送信
0042.00 SetResponse('success':'false');
0043.00 SetResponse('msg':'入力した商品CDは既に存在しています');
0044.00 /end-free
0045.00 C*
0046.00 C ELSE
0047.00 C*----新規レコード登録
0048.00 C MOVEL VSYHNCD SYHNCD
0049.00 C MOVEL VSYHNNM SYHNNM
0050.00 C Z-ADD VTANKA TANKA
0051.00 C*
0052.00 C WRITE MSYOHIR 1-③
0053.00 C*
0054.00 /free
0055.00 //正常終了メッセージを送信
0056.00 SetResponse('success':'true');
0057.00 SetResponse('refresh':'true');
0058.00 SetResponse('info':'登録が終了しました');
0059.00 /end-free
0060.00 C*
0061.00 C END

```

図15 ファイルアップロード サンプルプログラム

実行画面イメージ

CSVファイルアップロード

CSVファイル

IBM i

IFS

CPYFRMIMPF

商品マスタ MSYHIN

valence

商品マスターメンテナンス

商品マスタ

商品CD (From~To)

商品CD	商品名	単価	単位
4901306047569	野菜一日これ一本	98	本
4901777317604	南アルプスの天然水	98	本
4902424438529	アルコール除菌スプレー	350	本
4902738252026	苧麩畑 ブドウ味	230	個

アップロードが終了しました

[CSVデータ取込]をクリックするとアップロードダイアログが表示される。CSVファイルを選択し、[OK]をクリックすると、IFS上にファイルがアップロードされ、その後CPYFRMIMPFコマンドで商品マスタファイルを置き換える。

図16 ファイルアップロード RPGプログラム呼び出し設定

動作内容

Application

アプリケーションセクション

商品マスタウィジェット

行クリック

CSVデータ取込

クリック時

フィルターウィジェット

ウィジェットの表示/非表示

RPGプログラムの呼び出し/URL

アプリケーションの実行URL

ユーティリティ

アプリバーのタイトルの設定

RPGプログラムの呼び出し/URL

アプリケーションセクション: Main > 商品マスタウィジェット > CSVデータ取込 > クリック時

RPGプログラムの呼び出し URL アクション ファイル返却時の処理 プログラム実行前の処理

TEC20PG10 ? オプション

追加の詳細情報

パラメーター入力画面のタイトル

CSVファイルアップロード

\*ラベル

CSVファイルを選択してください。

必須 Valid Extensions

csv

\*パラメーター名

file

\*タイプ

Upload

キャンセル OK

キャンセル 保存

追加の詳細情報設定にて、下記を指定する  
パラメータ名 : file、タイプ : Upload  
拡張子を制限する場合、Valid Extensions  
を指定。

ソース4 CSVファイルアップロード(TEC20PG10)

```

0001.00 /copy qcpylesrc,vvHspec
0002.00 ** -----
0003.00 ** TEC20PG10 : CSVファイルアップロード
0004.00 ** -----
0005.00 ** -----
0006.00 /define nabButton
0007.00 /include qcpylesrc,vvNabImpl
0008.00 ** -----
0009.00 ** program start
0010.00 ** -----
0011.00 /free
0012.00 Initialize();
0013.00 Process();
0014.00 CleanUp();
0015.00 *inlr=*on;
0016.00 /end-free
0017.00 ** -----
0018.00 p Process      b
0019.00 d              pi
0020.00 D TMPPATH     S          80A
0021.00 D FILENM      S          80A
0022.00 D CMDSTR      S          255A
0023.00 D*
0024.00 /free
0025.00 //ファイル保管先IFSディレクトリの指定
0026.00 TMPPATH = vvUtility_getValenceSetting('TEMP_PATH');
0027.00
0028.00 //IFSへファイルアップロードを実施
0029.00 vvIn.path = %trim(TMPPATH);
0030.00 vvIn.ccsid = 943;
0031.00 vvIn_file(vvIN: '*NULL');
0032.00
0033.00 //アップロードしたファイル名を取得
0034.00 FILENM = %trim(TMPPATH) + %trim(vvIn.fileName);
0035.00
0036.00 /end-free
0037.00
0038.00 C*----- CPYFRMIMPFコマンドで物理ファイルMSYHINIに転送
0039.00 C          EVAL      CMDSTR =
0040.00 C          'CPYFRMIMPF '
0041.00 C          + 'FROMSTMF('' + %trim(FILENM) + '' ) '
0042.00 C          + 'TOFILE(MSYHIN) '
0043.00 C          + 'MBROPT(*REPLACE) FROMCCSID(943) '
0044.00 C          + 'RCDDL(*CRLF) ERRRCDOPT(*REPLACE) '
0045.00 C
0046.00 C*----- CPYFRMIMPFコマンドを実行
0047.00 C          CALL      'OCMDEXC'
0048.00 C          PARM          CMDSTR
0049.00 C          PARM          255          CMDLEN          15 5
0050.00 C*
0051.00 /free
0052.00 //-----アップロードしたCSVファイルを削除
0053.00 vvIfs_deleteFile(%trim(FILENM));
0054.00
0055.00 //-----レスポンスを返す
0056.00 SetResponse('success':'true');
0057.00 SetResponse('info':'アップロードが終了しました');
0058.00 /end-free
0059.00 p          e
0060.00 /include qcpylesrc,vvNabImpl

```

2-①

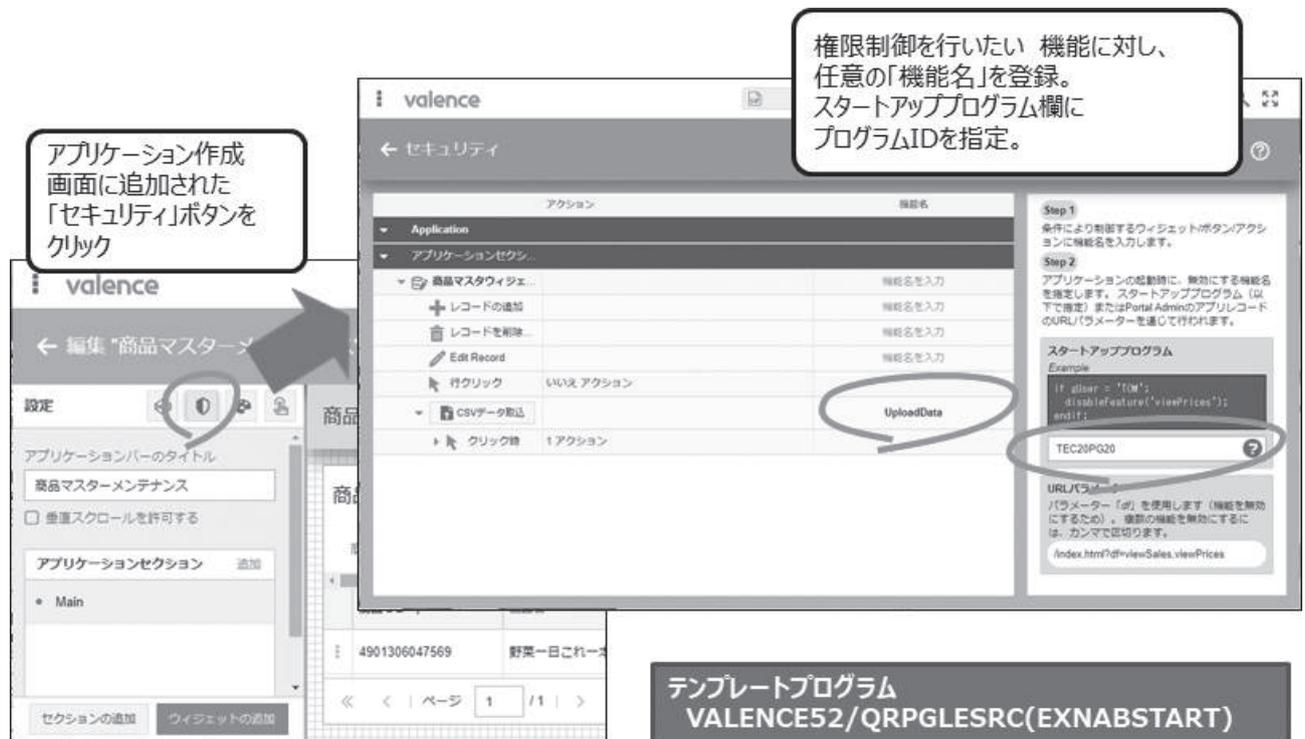
2-②

2-③

図17 セキュリティ機能 サンプルプログラム



図18 セキュリティ機能 設定画面



ソース5 スタートアッププログラム(TEC20PG20)

```

0001.00    /copy qcpylesrc,vvHspec
0002.00    ** -----
0003.00    ** TEC20PG20 : スタートアッププログラム
0004.00    ** -----
0005.00    ** -----
0006.00    /define nabStartup
0007.00    /include qcpylesrc,vvNabTmpl
0008.00    ** -----
0009.00    ** program start
0010.00    ** -----
0011.00    /free
0012.00    Initialize();
0013.00    Process();
0014.00    CleanUp();
0015.00    *inlr=*on;
0016.00    /end-free
0017.00    ** -----
0018.00    p Process      b
0019.00    d              pi
0020.00    /free
0021.00    //-----" 経理グループ" (グループID:1010) で無い場合
0022.00    if not vvSecure_isMember(1010);
0023.00    //----- CSVデータ取込 ボタン利用不可
0024.00    disableFeature('UploadData');
0025.00    endif;
0026.00    /end-free
0027.00    p              e
0028.00    /include qcpylesrc,vvNabTmpl
    
```

3-①

図19 Formウィジェット ヘルパープログラム サンプル

実行画面イメージ

アプリ起動時

Formウィジェット 起動時  
入庫日欄に システム日付を  
初期表示

商品CD入力終了時

商品CDが正しくありません  
商品マスターに該当の  
商品CDが存在しない場合  
エラーを表示し、フォーカスをセット

アルコール除菌スプレー  
商品マスターに該当の  
商品CDが存在する場合  
商品名をリアルタイムに表示

図20 Formウィジェット ヘルパープログラム 設定

Formウィジェット 設定画面 フィールド タブ

valence Nitro App Builder

← ウィジェットの編集 \*入庫入力フォーム - 入庫入力

ヘルパープログラムのプログラムID および実行タイミングを指定する。(下記のいずれか)

1. フォーム起動時
2. フォーム起動時+フィールド変更時
3. フォーム起動時+フィールド変更あるいはフォーカスロスト時

フィールド	フィールド	ラベル	フォーマッティ...	Editable	入...	色の設定
<input checked="" type="checkbox"/>	F1_SYHNC	商品CD				
<input checked="" type="checkbox"/>	F1_SYHNM	商品名				
<input checked="" type="checkbox"/>	F1_SURYO	入庫数量				
<input checked="" type="checkbox"/>	F1_NYDATE	入庫日				
<input checked="" type="checkbox"/>	F1_BIKO	備考				
<input type="checkbox"/>	F1_NYUKNO	入庫NO				

ヘルパープログラム

ヘルパープログラム

When should this be called?

TEC20PG30

At form creation and each time a field is changed or loses focus

At form creation

At form creation and each time a field is changed

At form creation and each time a field is changed or loses focus

Object ID: EXNABFHLP  
Version: V5.2

Description: NAB Examples - Sample Nitro App Builder Form Helper Template  
For use with Forms within Nitro App Builder.  
This program template enables the ability to hide/disable/show/enable form fields during user input (blur and/or change events). You may also filter the data in combo box fields in the form.

Copy from template EXNABFHLP

OK

テンプレートプログラム  
VALENCE52/QRPGLESRC(EXNABFHLP)

ソース6 ヘルパープログラム(TEC20PG30)

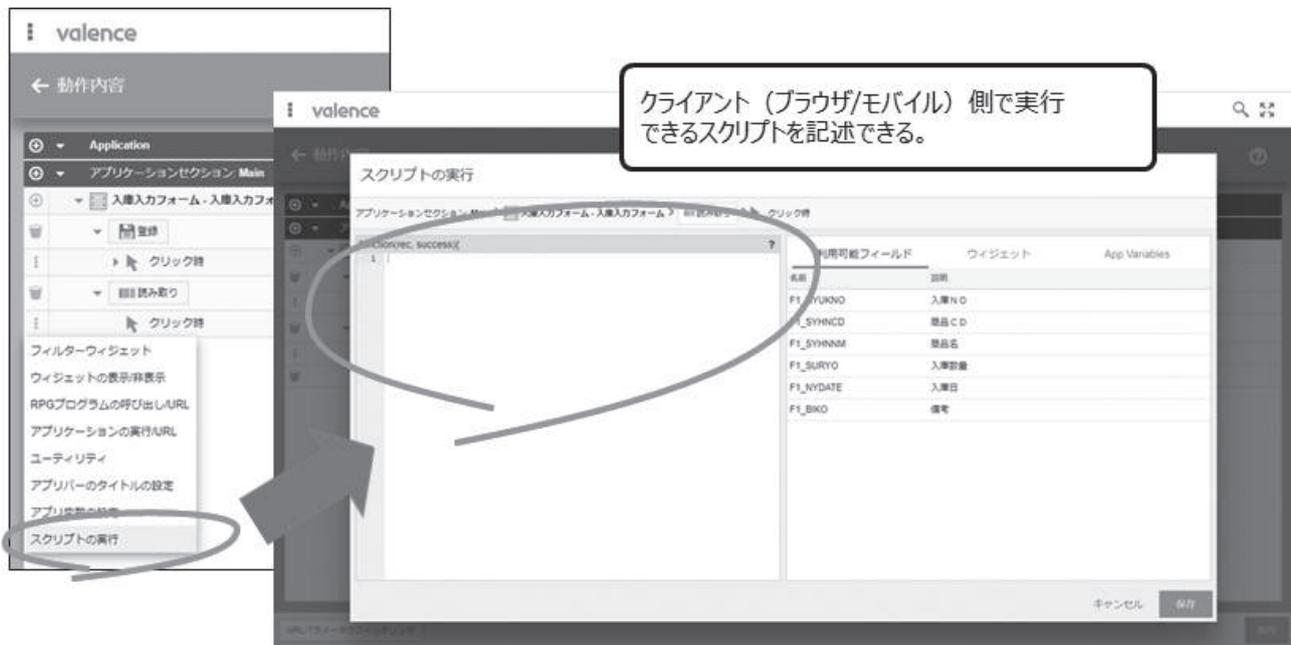
```

0001.00 /copy qcpylesrc,vvHspec
0002.00 ** -----
0003.00 ** TEC20PG30：フォームヘルパープログラム
0004.00 ** -----
0005.00 F*＜商品マスタ＞
0006.00 FMSYHIN IF E K DISK
0007.00 ** -----
0008.00 /define nabFormHelper
0009.00 /include qcpylesrc,vvNabTmpl
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 VSYCD S 20A
0023.00 D VSYNM S 60A
0024.00 D VERFG S 1A
0025.00 C*-----初期化
0026.00 C MOVEL *BLANK VSYNM
0027.00 C MOVEL *BLANK VERFG
0028.00 C*
0029.00 C*-----フォーム作成時処理 4-③
0030.00 C sMode IFEQ 'formRender'
0031.00 /free
0032.00 //-----入庫日欄にシステム日付を初期セット
0033.00 SetValue('F1_NYDATE':%CHAR(%DATE()));
0034.00 /end-free 4-④
0035.00 C ENDIF
0036.00 C*
0037.00 C*-----フィールド変更時処理
0038.00 C sField IFEQ 'F1_SYHNC'D' 4-⑤
0039.00 /free
0040.00 //----- 画面上的の商品C D取得
0041.00 VSYCD = vvIn_char('F1_SYHNC'D'); 4-⑥
0042.00 /end-free
0043.00 C*-----商品マスタに該当商品C Dが存在するか確認する
0044.00 C VSYCD IFNE *BLANK
0045.00 C VSYCD CHAIN MSYHIR
0046.00 C *IN31 IFEQ *OFF
0047.00 C MOVEL SYSYNM VSYNM
0048.00 C ELSE
0049.00 C MOVEL '1' VERFG
0050.00 C ENDIF
0051.00 C ENDIF
0052.00 /free
0053.00 //-----結果を画面上的の商品名欄にセットする
0054.00 SetValue('F1_SYHNNM':VSYNM);
0055.00
0056.00 //-----商品C Dが存在しない場合エラーを返す
0057.00 if VERFG < *BLANK:
0058.00 SetError('F1_SYHNC'D':'商品C Dが正しくありません');
0059.00 FocusField('F1_SYHNC'D');
0060.00 endif; 4-⑦
0061.00 /end-free
0062.00 C ENDIF
0063.00 C*
0064.00 p e
0065.00 /include qcpylesrc,vvNabTmpl

```

91

図21 スクリプトの実行設定



ソース7 スクリプト例(QR/バーコードリーダー連携)

```

function(rec, success){
  1- if (Valence.mobile.Access.isNativePortal()) {
  2-   Valence.mobile.Barcode.scan({
  3-     callback : function (response) {
  4-       if (Ext.isEmpty(response)) {
  5-         success(); return;
  6-       }
  7-       if (response.success) {
  8-         if (!response.data.cancelled) {
  9-           cmp.setValues({
 10-             F1_SYHNC : response.data.text
 11-           });
 12-           success();
 13-         } else {
 14-           Ext.Msg.alert('バーコードスキャン', 'キャンセル');
 15-           success();
 16-         }
 17-       }
 18-     }
 19-   });
 20- } else {
 21-   Ext.Msg.alert('バーコードスキャン', 'モバイルアプリの場合のみ使用可能です');
 22-   success();
 23- }
 24- }
 25- }
 26- }

```

図22 バーコード／QRコード読み取り

