

ORACLE

Oracle Forms 14cの新機能

2025年1月、バージョン[1.0]

Copyright © 2025, Oracle and/or its affiliates
Public

本書の目的

本文書では、リリース14.1.2.0の機能と強化された点の概要が説明されています。本文書は、14.1.2.0へのアップグレードに関するビジネス上の利点の評価と、説明した製品機能の実装およびアップグレードの計画を支援することのみを目的としています。

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料にならないでください。本文書に記載されている機能の開発、リリース、時期および価格については、弊社の裁量により決定されます。製品アーキテクチャの性質上、本書に記述されているすべての機能を安全に組み込むことができず、コードの不安定化という深刻なリスクを伴う場合があります。

目次

はじめに	5
実行時機能	6
ユーザー・インターフェース	6
テキスト項目	6
表示項目	8
プッシュ・ボタン	9
リスト項目	10
すべての境界線項目	11
グループ項目	11
値リスト（LOV）	12
グラフィック項目	13
アラート	13
エディタ	14
ログオン・ダイアログとパスワードの変更ダイアログ	14
キャンバス	15
タブ・バー	16
ウィンドウ	17
デフォルトのメニューとスマートバー	18
マルチレコード・ロック	19
データ・アクセス、監視、操作	24
REST	24
Continuous Query Notification	25
ロックのソート	25
その他の実行時機能	26
SYSTEM.CURSOR_ROW	26
イメージのコピー	26
組込みのMESSAGE	26
ユーザーのJavaバージョンの取得	26
クライアント構成の取得	27
開かれたモジュールの記録	27
WebUtil	27
クライアント構成	27
Forms Standalone Launcher	27
Web Start	29
Fusion Middleware Control	30
User Sessionsページ	30
Advanced Configurationページ	30

Builder	31
ログへのFSAL出力の公開	31
起動時のパフォーマンスの改善	31
複数のファイルを開けるダイアログ	31
メッセージ・バー内のデータベース名	32
フォーム・プロパティ – Application Name	32
Create XML After Save	32
RESTパッケージ・デザイナ (RPD)	32
まとめ	34

はじめに

本文書は、Oracle Forms 14c (14.1.2) の多くの新機能の一部に関する概要を説明するものです。本文書だけで、このリリースに導入されているすべての新機能と拡張機能の完全なコレクションを説明しているわけではありません。本文書では、読者が最新の製品バージョンを使用することで、説明されている任意の機能を活用できることを前提としています。

概要を説明する本文書は、新機能の一部に関する参考資料と紹介資料として使用することのみを目的としています。完全な詳細については、Oracle Form Builderのヘルプを含む製品ドキュメントを参照してください。

これらの機能とその他の新機能の詳細情報および使用の詳細については、[Oracle Formsドキュメントのライブラリ](#)とOracle Form Builderのヘルプを参照してください。

実行時機能

ユーザー・インターフェース

Formsのユーザー・インターフェースで使用可能なカスタマイズのレベルは、ほぼ20年間にわたって大半が同じままです。よりロードコードまたはコードなしのカスタマイズ・オプションが長い間待ち望まれていました。この新機能は、アプリケーションの最新化に向かう道の重要な一部です。

次の各セクションでは、さまざまな項目タイプとその他のFormsコンポーネントおよびこれらの改善点について説明します。機能、その使用方法、制限、その他の詳細については、Form Builderのヘルプを参照してください。

テキスト項目

テキスト項目（編集項目とも呼ばれます）は、エンドユーザーがデータを入力および編集するために使用するフィールドです。従来、これらの項目は、ユーザーがテキストを入力するときの挿入点を示すカーソルを備えた長方形のボックスとして表されていました。

フォントのスタイルとフィールドの色は変更できますが、その大部分は使用可能なカスタマイズの範囲内でした。この項目タイプには背景パターンを導入できますが、これは多くの場合、気を散らすものであると見なされ、テキストが若干読み取りにくくなりました。

このリリースには新しい属性セットが導入されたため、多くのさまざまなスタイルが使用可能になりました。次に、拡張機能のいくつかの例を示します。

隠されたデータ

隠されたデータ文字を選択できます。隠されたデータ・フィールド用としてどの文字を使用するかを選択できるようになりました。これには、Formsのデフォルトのログオン・ダイアログとパスワードの変更ダイアログが含まれます。この機能を有効にするために、コードなしの変更が必要です。単にFont、Image、Style Mapping（Registry.dat）で目的の値を設定するだけですみます。

```
default.concealedData.character=<Unicode value or single character>
```

たとえば、大きなドットを使用するには、値を\U25cfに設定します。これにより、次のように大きな円形のドットが表示されます。



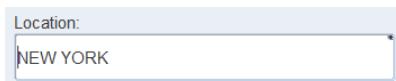
新しいプロパティConcealed Data Buttonを設定することで、隠されたデータの表示ボタンをフィールド内に組み込むことができます。これにより、ユーザーがボタンを押したときに、隠されたデータを人間が読める形式で一時的に表示できるようになります。ボタンを離すと、テキストは再度隠されます。



値リスト（LOV）ボタン

リスト項目に値リストが関連付けられている場合、アプリケーション開発者は、ユーザーがコードを一切含める必要なくLOVに簡単にアクセスできるようにすると便利な場合があります。この機能が有効化されている場合、ユーザーがフィールドに入力するときに、フィールドの右上に小さなボタンが表示されます。ユーザーがマウスをこのボタンに合わせると、マウス・ポインタが指のある手に変化します。このボタンをクリックすると、関連するLOVが表示されます。このデフォルトのLOVをカスタマイズすることはできません。カスタマイズされたLOVを表示するには、LOV属性を設定するために用意されているさまざまな組込み機能を使用してから、以前に表示した場合と同じようにLOVを表示します。

この機能は、この項目用のBuilderのプロパティ・パレットにある新しいプロパティ（LOV Button）を使用して有効化されます。



カスタム・フォント

これまでユーザーのマシンにあるフォントを使用することが可能でした。しかし、これを可能にするための構成の変更が面倒でした。このリリースでは、Registry.datで簡単な変更を行うことで、ユーザー側で（フォントの存在以外の）特別な変更を行う必要なく、ユーザーのマシンに存在するフォント名を使用するよう構成されたアプリケーション項目が実行時にレンダリングされます。たとえば、次のパラメータを“partial”に設定すると、設計時（または実行時）に参照されたフォント名の使用が試みられます。このフォントが見つからない場合、Javaフォント・マッピングが使用され、以前のバージョンで見られた動作になります。有効な値は、FULL、PARTIAL、NONEです。詳細については、[Working with Oracle Forms](#)ガイドまたはRegistry.dat内のコメントを参照してください。

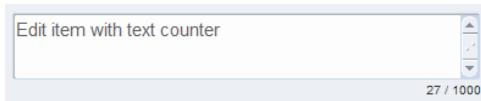
```
default.fontMap.defaultMapping=partial
```

次の図は、カスタム・フォントだけでなく、本文書でこの後説明するカスタムの境界線の色と側面の使用を示しています。



文字カウンタ

テキスト項目の文字カウンタを表示するには、Display Character Counterプロパティを有効化します。フィールドにフォーカスがある場合、項目の右下の近くにカウンタが表示されます。フォーカスがなくなると、これは消えます。この値は、フィールド内の文字数とその後ろに続くMaximum Lengthプロパティの値として表されます。この設定は、複数行または單一行のテキスト項目で使用できます。



プレースホルダと永続プレースホルダ

プレースホルダと永続プレースホルダは、項目のプロンプト/ラベルに対する最新のアプローチです。プレースホルダ・テキストを使用することで、項目の境界の外側で別途使用可能なスペースを必要とするプロンプトの使用を回避できます。この機能を使用するテキスト項目にユーザーが移動すると、ユーザーがフィールドにテキストを入力し始めるまで、プレースホルダ・テキストが表示されたまま残ります。同様に、すべてのテキストが削除されると、プレースホルダ・テキストが再表示されます。プロンプトを使用しないことが望ましい場合、Persistent Placeholderを有効化すると、プレースホルダ・テキストはフィールドのフォント・サイズの約70 %まで最小化され、左上隅に移動します。この結果、テキストがフィールドに入力された後でも、ユーザーにはプレースホルダ情報が表示され続けます。永続プレースホルダを最も効率的に使用するには、垂直に中央揃えされるユーザーの入力と永続プレースホルダ・テキストを十分に収容できるほどの項目の高さが必要であるください。プレースホルダの色は、Registry.datでグローバルに設定できます。詳細については、[Working with Oracle Forms](#)ガイドを参照してください。



表示項目

従来、Formsの表示項目は、無効化されたテキスト項目と似ていました。つまり、データが表示されましたが、ユーザーが項目に移動することはできませんでした。複数の新しい表示項目プロパティが導入されたため、この項目タイプをさまざまな方法で視覚的に表示できるようになりました。これらの新しい視覚表現を使用するには、項目のデータ型がNUMBER、INTEGER、RNUMBER、またはRINTEGERのいずれかである必要があります。

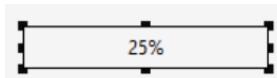
新しいプロパティ表示項目のUI Styleには、Standard、Progress Bar、Gauge、Half Gaugeの各オプションが含まれます。“Standard”は、従来の表示項目に使用されます。その他3つの新しいスタイルは次のとおりです。

プログレス・バー

プログレス・バーは、ワークフローまたはタスク・ステータスを視覚的に表示することを目的としています。プログレス・バーは、移動する塗りつぶしのバーが内部にある長方形として表されます。プログレス・バーは、水平または垂直に表示できます。項目の値が大きくなるにつれ、中身の詰まった長方形で満たされた領域の範囲も広くなります。このデータ値は、定義済の最小値に対する定義済の最大値のパーセンテージとして表示できます。最小値と最大値は両方とも、設計時または実行時に設定できます。また、表示されるデータ値を数値データ値として表示することもできます。ただし、表示できるのは整数のみであるため、項目内に端数値が入力されると、表示される値は次の整数まで丸められます。データベースに保存される実際の値は保存時に完全な値として残りますが、前述のように、この値はプレゼンテーション・レイヤーで視覚的に変更される場合があります。

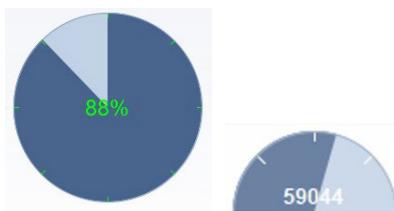


設計時には、レイアウト・エディタに、中央に“25 %”と表示された長方形としてプログレス・バーが表示されます。

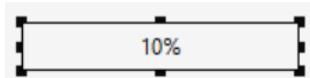


ゲージと半ゲージ

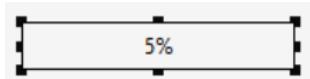
ゲージは、データ値の視覚的な表現を提供するという点でプログレス・バーと非常に似ています。プログレス・バーと同じ制約と制限がゲージに適用されます。



設計時には、レイアウト・エディタに、中央に“10 %”と表示された長方形としてゲージが表示されます。



設計時には、レイアウト・エディタに、中央に“5 %”と表示された長方形として半ゲージが表示されます。

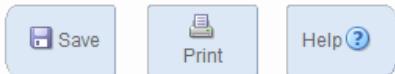


レイアウト・エディタの制限のため、ゲージをプログレス・バーとして表示することが必要でした。したがって、各項目タイプを特定しやすいようにさまざまなパーセンテージが使用されました。

プッシュ・ボタン

イメージとラベル

プッシュ・ボタンは以前、ラベルまたはイメージの表示のみをサポートしていましたが、両方を同時には表示していませんでした。しかし、両方を同時に表示できるようになりました。ラベルは、イメージの四方のいずれかに配置できます。ラベルの位置は、設計時にのみ設定できます。



ロールオーバー色スワップ

ボタンのRollover Color Swapプロパティを設定すると、ユーザーがマウスをボタンに合わせたときに、ボタンの前面色と背景色が切り替わります。



ロールオーバー・イメージ・スワップ

ボタン上のイメージは、ユーザーがマウスをボタンに合わせたときに、イメージ間で切り替えることができます。マウスがボタン領域を離れるとき、イメージは元のファイルに戻ります。この機能は、2つのイメージ・ファイル名の間に::（2つ並んだコロン）を使用して既存のプロパティIcon Filenameを拡張することで、有効化できます。

<input checked="" type="checkbox"/> Iconic	Yes
<input checked="" type="checkbox"/> Icon Filename	image1.png::image2.png

グラデーション色と角丸

ボタンの左側と右側を丸くしたり平らにしたりできるようになりました。曲がりの度合いは、ボタンの高さによって異なります。ボタンが短いほど、曲がりが大きくなります。

塗りつぶしの背景色に加えて、グラデーションを使用することもできます。



ボタンの境界線に色を付けると、ユーザーが特定のボタンを簡単に（視覚的に）特定しやすくなります。境界線の色はグローバルに設定され、実行時には変更できませんが、色が付けられた境界線は実行時に有効化および無効化できます。下の“[境界線の色と側面](#)”セクションを参照してください。



リスト項目

リスト項目を使用すると、ユーザーが特定の項目に関して使用可能な多数の値に簡単にアクセスできるようになります。リストには、一連の固定値を含めることができます。また、固定リストとともに、ユーザーがリスト内にある値とは別の値を入力できるテキスト・フィールドを用意することもできます。

コンボ・ボックス

コンボ・ボックス項目を使用すると、ユーザーが設定されている値リストから値を選択したり、用意されているテキスト・フィールドに別の値を入力したりできます。コンボ・ボックス機能は拡張され、自動補完が用意されました。自動補完を有効化すると、ユーザーがテキストを入力するときに、コンボ・ボックスの内部でリストのコンテンツが検索され、（これまでに入力されたコンテンツに基づいて）一致するコンテンツが見つかると、その文字列全体が表示されます。ユーザーがこの値を選択することを決定した場合、単に[Tab]キーまたは[Enter]キーを押して（またはマウスを使用して）このフィールドを離れるだけで、補完された値が選択されます。ただし、表示されている文字列が目的の文字列ではない場合、ユーザーは入力を続行でき、異なる一致コンテンツが見つかったら、これが同じ方法で表示され、もう一度、この候補を受け入れるか入力を続行するかを決定できます。検索は常にリストの一番上から始まり、そこから入力された下位の新しい文字列に向けて順に行われます。



スピニ・ボックス

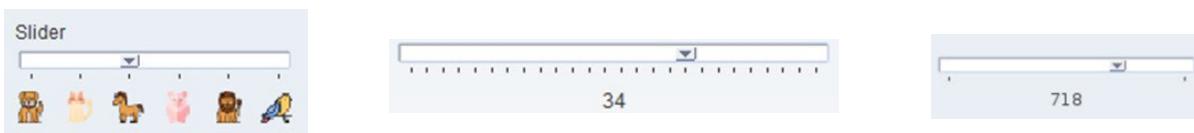
スピニ・ボックスは、Tリストと同じように動作します。ただし、スピニ・リストの場合、ユーザーがリストの最後までスクロールすると、リストの先頭に戻り、そこから続行されます。逆方向でも同じように動作します。先頭に達すると、次のリスト内の最後のエントリに移動します。

スライダ

スライダを使用すると、ユーザーがトラック上で位置バーを左右に移動することで、値を選択できます。温度計と同じように、トラック上にはリスト内の各値を表す目盛りがあります。目盛りが表示されるかどうかは、（設計時または実行時に設定された）最小および最大UI値が項目に設定されているかどうかによって異なります。項目で必要な数の目盛りを表示するのに十分な水平方向のスペースがあることが確認されると、目盛りは表示されますが、十分ではない場合、目盛りは表示されません。

スライダの現在のデータ値は、項目にそれを表示するのに十分な高さがある場合、（視覚的に次の整数に丸められて）項目の下に表示されます。データ値が表示されないようにプログラムによって非表示にする方法は存在しないため、データ値を非表示にするには単に項目を低くします。

スライダ・リスト・タイプでは、データ型が数値（NUMBERなど）である必要があります。



すべての境界線項目

境界線の色と側面

一部の項目タイプではサポートされていません。次の1つ以上の新しい項目プロパティを有効化することで、項目の境界線の色を設定できるようになりました。

- Show Border Color
- Rollover Border Color
- Specified Border Sides

各プロパティの目的の値はRegistry.datで宣言できます。これにより、これらのプロパティが有効化されている任意の項目にグローバル値が提供されます。設定されていない場合のデフォルト値は赤（255,0,0）です。

```
default.border.color=<desired color in R,G,B value> default.border.rolloverColor=<desired color in R,G,B value>
```

highlightColor属性は、プログラムによってのみ有効化できます。設定されていない場合のデフォルト値は赤（255,0,0）です。

```
default.border.highlightColor=<desired color in R,G,B value>
```

highlightColorを有効化するには、前述のとおり、目的の色を設定してから、次を実行します。

```
SET_ITEM_PROPERTY ('TEXT_ITEM1', BORDER_BEVEL, HIGHLIGHTED);
```

4つの境界側面のうちどれを表示するかを指定するには、Registry.datで、項目のSpecified Border Sidesプロパティを有効化してから、TRUEまたはFALSEまたは未設定（TRUEと同義）を使用して、どの側面を表示するかを指定します。次に例を示します。

```
default.border.right=false default.border.left=true default.border.top=false default.border.bottom=true
```



ブール項目

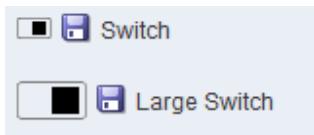
ブール項目は多くの場合、チェック・ボックスまたはラジオ・ボタンとして表されます。ただし、より新しい設計では、一部のブール項目はスイッチまたはトグル・ボタンとして表されます。

スイッチ

スイッチは、Check BoxまたはRadio Button項目のプロパティUI StyleのSwitchまたはLarge Switchを選択することで、有効化できます。Switch UI Styleには、チェック・ボックスで使用されるのとほぼ同じ垂直スペースが必要です。したがって、水平スペースには問題がないが、垂直スペースが制限されている場合、既存の項目の配置を一切変更することなく、このスイッチ・スタイルを使用してチェック・ボックスまたはラジオ・ボタンを簡単に置き換えることができます。Large Switchスタイルは、チェック・ボックスの約2倍の高さです。

スイッチを使用するには、単にスイッチ上の任意の場所をクリックして値を変更するだけですみます（キーボードを使用することもできます）。これは、チェック・ボックスまたはラジオ・ボタンで見られる動作と同じです。スイッチは、内部ボックスがその右側に配置されると、有効化されます。この内部ハンドルは、“オン”的位置（右側）にあるときに黒色で表示され、“オフ”的位置（左側）にあるときに白色で表示されます。

オブジェクトとラベルの間にイメージを組み込むこともサポートされるようになりました。



トグル・ボタン

ブール項目用のトグル・ボタンを選択するには、項目のUI StyleプロパティからToggle Buttonを選択します。このボタンは、押し下げられた位置にあるときに起動（オン）され、押し上げられた位置にあるときに解除（オフ）されます。このスタイルでは、ラベルとイメージの使用はサポートされていますが、プッシュ・ボタンとは異なり、イメージを基準にしてラベルを移動することはできません。イメージは常に、ラベルの上に表示されます。



Fill Patternプロパティ

ガラスは、Fill Patternの使用をサポートしている任意の項目で使用できる新しい“Fill Pattern”です。“ガラス”は、透明な背景を提供し、項目の背後にある色、イメージ、その他のオブジェクトを透かして見ることができます。この機能は、ページ/画面のタイトルのように表示される表示項目をプログラムによって設定できる簡単な方法を提供します。つまり、ボイラー・テキスト（グラフィック・テキスト）を使用する代わりに、Fill Patternを“glass”に設定した状態で表示項目を使用して、テキストをPL/SQLコードで制御できます。

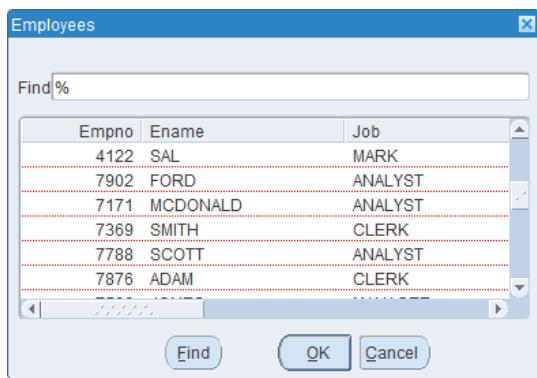


値リスト（LOV）

Formsの値リストは、多くの値を一度に表示するために使用する組込みの複合ウィジェットです。これはフィルタリング機能も備えているため、ユーザーが表示される値の数を必要なもののみに削減できます。FormsのLOVの問題の1つは、多くの列が表示される場合、各行が重なって表示されるため、行全体を表示するのが視覚的に難しくなる場合があることです。新しいRegistry.datの設定では、管理者が各行間に見やすい分割行を設定できるため、行を特定しやすくなります。この新しい設定はlovRowLine.colorであり、任意の有効なRGBまたはOLAF値に設定できます。たとえば、次のとおりです。

default.lovRowLine.color=255,0,0

これにより、次のイメージで見られるように、LOV内に見やすい赤い行が表示されます。



グラフィック項目

行の幅

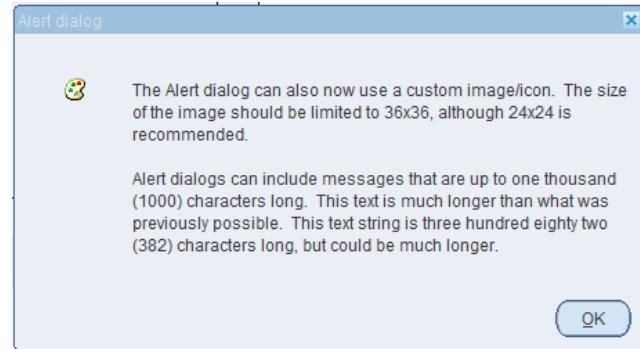
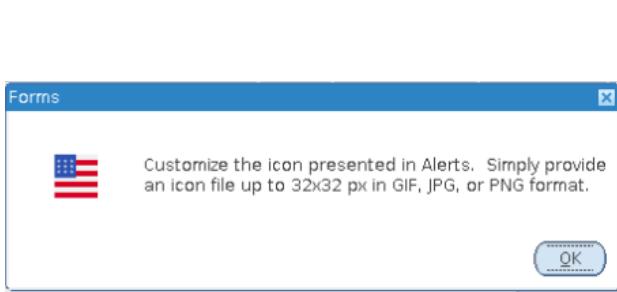
以前のバージョンでは、Oracle LookandFeelを使用する場合、グラフィック項目のline widthプロパティは無視されました。この影響を受けたグラフィック・タイプには、長方形、角丸長方形、楕円、多角形、フリーハンド、ポリライン、弧、線が含まれます。honorLineWidth Web構成設定をTRUEに設定することで、行の幅を設定できるようになりました。



アラート

カスタム・アラートは、以前の制限の200文字から増えて、最大1000文字をサポートできるようになりました。また、実行時にBuilderまたはICON_NAMEでAlertプロパティIcon Filenameを設定することで、カスタム・イメージも使用できるようになりました。たとえば、

```
SET_ALERT_PROPERTY ('ALERT1', ICON_NAME, 'my32x32image.png');
```



エディタ

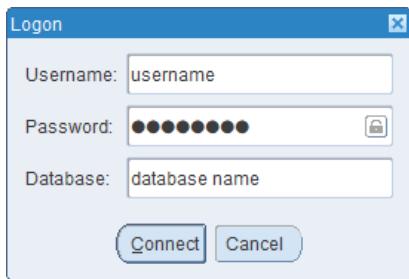
前述のテキスト項目に対して使用できるようになった文字カウンタと同様に、テキスト項目のデフォルトのエディタとカスタム・エディタにも文字カウンタが追加されました。カスタム・エディタの場合、文字カウンタは必要に応じて追加できますが、組込みエディタの場合、文字カウンタは常に有効化されています。



ログオン・ダイアログとパスワードの変更ダイアログ

隠されたデータ・ボタン

組込みのログオン・ダイアログとパスワードの変更ダイアログには、両方とも、“Show data”ボタンが含まれます。このボタンを使用すると、隠されたパスワードを人間が読めるテキストで表示できます。ボタンを離すと、テキストは再度隠されます。このボタンを組み込まないようにするには、環境変数FORMS_OBFUSCATE_CONCEALED_DATA=1に設定します。この変数を1に設定すると、すべての/任意の隠されたデータ・ボタンが表示されなくなります。これには、アプリケーション内のテキスト項目で明示的に有効化されているデータ・ボタンも含まれます。



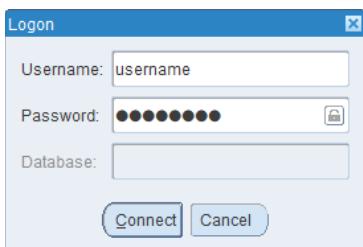
データベース名フィールド

ログオン・ダイアログでデータベース名フィールドを無効化するには、次を使用します。

```
SET_APPLICATION_PROPERTY(DATABASE_NAME_ENABLED, PROPERTY_FALSE);
```

このコードは、PRE-LOGONトリガーでのみ使用する必要があります。この設定を使用する場合、TWO_TASK（Linux/Unixサーバーの場合）またはLOCAL（MS Windowsサーバーの場合）を使用して、Formsの環境変数でデータベース名を設定すること前提としています。たとえば、次のとおりです。

```
TWO_TASK=orcl
```



キャンバス

スタックされたキャンバス・スプリッタ

スタックされたキャンバスは、長年にわたって多くのユースケースで使用されてきました。おそらくより一般的な使用方法は、スタックされたキャンバスを使用して大量の情報を収容する方法ですが、この場合、一度に表示される情報量は制限されます。ユーザーは、すべての情報を表示するためにスクロールバーを使用してビューを動かしていました。

2つのスタックされたキャンバスを使用した、新しいスタックされたキャンバス・スプリッタを使用すると、アプリケーション開発者は、1つのキャンバスをもう1つのキャンバスの上に配置できるだけでなく、ユーザーがキャンバスをもう1つのキャンバスの上にドラッグできる“スライダ”ハンドルを表示することもできます。このハンドルは、ユーザーがマウスを使用して動かすことも、プログラムによって動かすこともできます。また、このハンドルを固定し、ユーザーが動かせないようにすることもできます。これは、あまり重要ではない情報は隠すが、必要に応じて簡単に使用できるようにするための理想的な機能です。

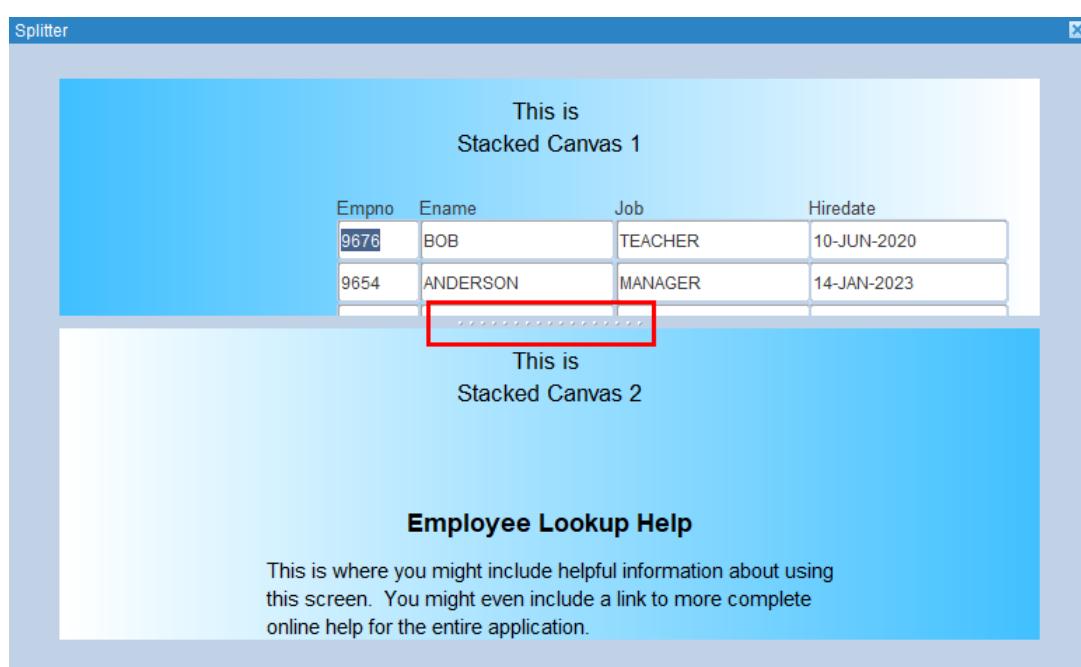
このスプリッタをプログラムによって配置するには、次のように組込みのSET_VIEW_PROPERTYを使用します。

```
SET_VIEW_PROPERTY (<CANVAS NAME>, SPLITTERBAR_POS, <POSITION>, <ANIMATION RATE>)
```

バーを無効化し、ユーザーが手動で動かせないようにするには、引数SPLITTERBAR_ENABLEDを使用します。たとえば、次のとおりです。

```
SET_VIEW_PROPERTY (<CANVAS NAME>, SPLITTERBAR_ENABLED, PROPERTY_FALSE);
```

スプリッタ・バーの位置は、スプリッタ・バーが無効化されていても、変更できます。



キャンバスの背景イメージ

キャンバスの背景イメージは、新しいキャンバス・プロパティBACKGROUND_IMAGEを使用して、プログラムによって設定できます。イメージ・ファイルは、サーバーのファイル・システム上に存在することも、URLから使用することもできます。URL（完全修飾または相対）を使用するには、URLに‘URL:’をプリフィックスとして付けます。たとえば、次のとおりです。

-- 相対URLからの場合

```
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE, 'URL:/forms/java/images/image.png');
```

-- ファイル・システムからの場合

```
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE, '/home/oracle/Pictures/image.png');
```

イメージは、そのサイズに基づいてキャンバスに配置されます。つまり、イメージのサイズがキャンバスのサイズより小さい場合、イメージはキャンバスの一部のみとして表示されます。ただし、イメージを拡大してキャンバス全体に表示されるようにする場合は、イメージを設定する直前にBACKGROUND_IMAGE_ALWAYS_FILLをPROPERTY_TRUEに設定します。たとえば、次のとおりです。

```
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE_ALWAYS_FILL, PROPERTY_TRUE);
```

```
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE, '/home/oracle/Pictures/image.png');
```

ツールバー・キャンバスのアニメーション

ツールバー・キャンバスは、組込みのSET_VIEW_PROPERTYを使用して、表示と非表示を切り替えることができます。この組込みには、アニメーションまたは流動体の動きを使用して、表示と非表示を行う新しい引数が追加されています。このため、ツールバーの表示と非表示が視覚的に突然行われる印象を薄めることができます。これは、ツールバーがMDIウインドウにアタッチされるときにはサポートされません。この新しい引数を使用するには、次のように設定します。

```
SET_VIEW_PROPERTY ('MYTOOLBAR', VISIBLE, PROPERTY_TRUE, 5);
```

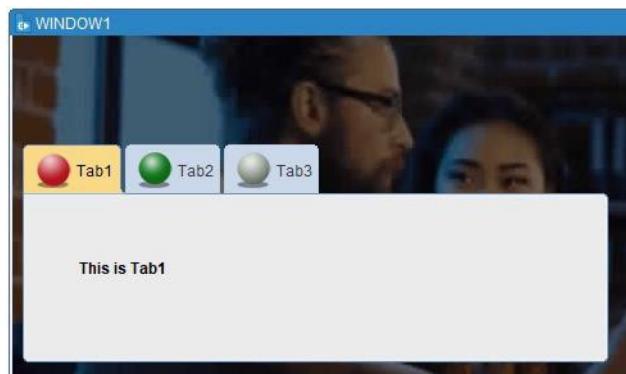
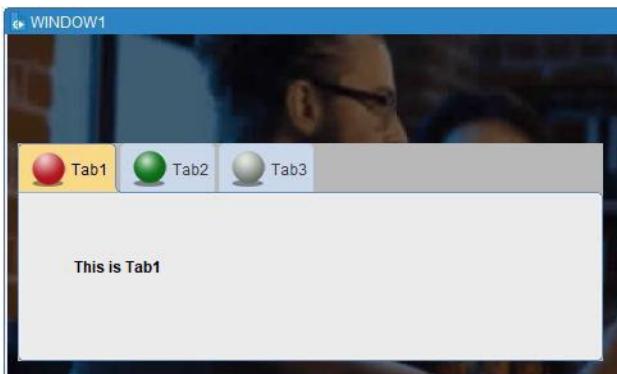
この例の最後の引数はアニメーション率です。これは、次の2つのピクセルの動きの間の遅延時間です。この値はミリ秒単位で、有効な値は0～100です。デフォルト値は0です。この動作の正確なパフォーマンスは、ユーザーのマシンのパフォーマンスに応じて異なる可能性があります。

タブ・バー

透過性

場合によってはそれほど明らかではない可能性がありますが、Formsのタブ・キャンバスには、タブ・キャンバスの幅（垂直方向の場合は高さ）いっぱいまで伸ばされたタブ・バーが含まれます。下の左側のイメージでは、タブ・バーがタブ・キャンバスの幅いっぱいまで表示されていることが分かります。これはデフォルトの動作です。ただし、tabBar.transparentをTRUEに設定すると、右側のイメージのように、タブ・バーは透過的になり、タブ・バーの後のコンテンツが見えるようになります。たとえば、次のとおりです。

```
default.tabBar.transparent=true
```



選択したタブの色

現在選択しているタブを明確に特定することが難しい場合があります。Registry.datで選択している/選択していないタブの背景または前景の色を設定することで、現在選択しているタブをより明確に特定できるようになります。有効な値は、RGB値を使用して特定した色または有効なOLAFの色です。たとえば、上の2つのイメージでは、現在選択しているタブがオレンジ色の背景を使用して表示されており、現在選択していないタブにはデフォルトの色が使用されています。

外観の設定は次のとおりです。

```
default.tabLabel.selectedFgColor=
default.tabLabel.unselectedFgColor=
default.tabLabel.selectedBgColor=OLAF2
default.tabLabel.unselectedBgColor=
```

ウィンドウ

MDIタイトルバーを非表示

Web構成パラメータshowMDITitleBarを使用すると、Formsの親ウィンドウのタイトル・バーの表示と非表示を切り替えることができます。これは、キオスクに似たモードが必要な場合に役に立ちます。たとえば、アプリケーションの親ウィンドウ（FORMS_MDI_WINDOW）が起動時にプログラムによって最大化され、Web構成にshowMDITitleBar=falseおよびalwaysOnTop=trueが含まれる場合、その結果はキオスクに似たモードの表示になります。つまり、ユーザーのディスプレイにはFormsアプリケーション以外は何も表示されなくなります。

タイトルバー・アイコン

フォーム・ウィンドウのタイトルバー・アイコンは、Icon File名を文字列<none>に設定することで、非表示にすることができます。



alwaysOnTop

alwaysOnTop Web構成パラメータを使用すると、ユーザーのデスクトップ上で実行されている他のすべてのアプリケーションの上にアプリケーションのMDIウィンドウが表示されたままになるよう強制できます。これは、実行時にSET_WINDOW_PROPERTYとプロパティ ALWAYS_ON_TOPを使用して設定できます。

```
SET_WINDOW_PROPERTY (FORMS_MDI_WINDOW, ALWAYS_ON_TOP, PROPERTY_TRUE);
```

Windowsのタスクマネージャーなどのオペレーティング・システム・ウィンドウもalwaysOnTopに設定されている場合、このウィンドウがFormsアプリケーション・ウィンドウより優先される可能性があります。

ウィンドウの中央揃え

フォーム・ウィンドウとアプリケーションMDIウィンドウは両方とも、ディスプレイの中央に配置できます。組込みのSET_WINDOW_PROPERTYを使用して、新しいプロパティWINDOW_CENTEREDを使用してプログラムによってウィンドウを中央に設定します。フォーム・ウィンドウの場合、代わりにプロパティ・パレットでプロパティCenter on Startupを設定することもできます。一方、Web構成設定centerOnStartup=trueを設定することで、MDIを中央に設定することもできます。

```
SET_WINDOW_PROPERTY ('WINDOW1', WINDOW_CENTERED, PROPERTY_TRUE);
```

```
SET_WINDOW_PROPERTY (FORMS_MDI_WINDOW, WINDOW_CENTERED, PROPERTY_TRUE);
```

MDI ウィンドウのサイズ変更イベント

MDI ウィンドウのサイズが変更されたかどうか/いつ変更されたかを把握できることが必要な場合があります。MDI ウィンドウのサイズ変更をWHEN-WINDOW-RESIZEトリガーから監視することは技術的に不可能であるため、システム・イベント内のSYSTEM_MDI_WINDOW_RESIZEを使用して、MDI ウィンドウのサイズ変更を監視できるようになりました。これを実現するには、必要なコードを使用して、このイベントのシステム・イベントWHEN-EVENT-RAISEDトリガーを作成します。例：

```
DECLARE
    time VARCHAR2(20);
BEGIN
    If :SYSTEM.LAST_EVENT = 'SYSTEM_MDI_WINDOW_RESIZE' Then
        time := :System.Current_Datetime;
        message ('Resized at: ' || SUBSTR( time, instr(time,' ')), status);
    End If;
END;
```

ウィンドウ・メニュー項目を非表示

メニューを使用していないときに、“ウィンドウ”メニュー項目を表示しないことを好むことは珍しくありません。Web構成parameter hideWindowMenuItem=TRUEに設定することで、モジュールでメニューが指定されていない場合、“ウィンドウ”メニュー項目は表示されなくなります。ロゴが使用されている場合、この設定ではロゴは保持されます。

キャンバス・ウィンドウのスクロールバー

垂直および水平ウィンドウのスクロールバーは通常、自動的に制御されます。しかし、SET_WINDOW_PROPERTYのSHOW_HSCROLLBARまたはSHOW_VSCROLLBARプロパティを設定することで、アプリケーション開発者は、これらのスクロールバーを表示するかどうかを制御できます。これは、キャンバス・ウィンドウのスクロールバーに対してのみ適用され、MDIのスクロールバーには適用されません。

```
SET_WINDOW_PROPERTY ('WINDOW1', SHOW_HSCROLLBAR, PROPERTY_FALSE);
```

この値は、Formsにコントロールを返すAUTO_DISPLAYに設定することもできます。ただし、これをAUTO_DISPLAYに設定すると、必要ないと考えられるときにスクロールバーが表示される可能性があります。

デフォルトのメニューとスマートバー

デフォルトのスマートバー・アイコンが更新されました。古いアイコンを使用したい場合は、Registry.dat（Font and Icon Mapping）で default.icons.style=legacyを設定します。

Page Setup

Printオプションと連携して機能するようにPage SetupメニューとSmartbarエントリが導入されました。

スマートバーのカスタム配色

スマートバーの色は、次のFont and Icon Mapping（Registry.dat）エントリとともに、customColorSchemeを使用して設定できます。

```
colorScheme.sample.smartBar=<hex value color>
```

マルチレコード・ブロック

行バンディング

行バンディングは、Forms 12cで用意されている機能です。この機能を使用すると、マルチレコード・ブロック内の各行に交互に色をつけることができました。これらの色は、実行時配色（またはカスタム配色）で“pinstripe1”および“pinstripe2”色として識別された色に制限されていました。これらの値が、実行時に変更できるようになりました。たとえば、奇数行を緑ファミリーで表示し、偶数行を青ファミリーで表示する場合、アプリケーション・コードで次を実行します。

```
SET_BLOCK_PROPERTY ('EMP', PINSTRIPE1_COLOR, 'r56g248b67'); SET_BLOCK_PROPERTY ('EMP',
PINSTRIPE2_COLOR, 'r146g220b255');
```

このコード例の結果、次のイメージのようになります。モジュールでRow Bandingプロパティが適切に有効化されていることを確認してください。この縦縞色は、フォーム、ブロック、または項目レベルで設定できます。

Empno	Ename	Job	Hiredate
7566	JONES	MANAGER	02-APR-1981
7902	FORD	ANALYST	03-DEC-1981
7369	SMITH	CLERK	17-DEC-1980
7788	SCOTT	ANALYST	09-DEC-1982
7876	ADAM	CLERK	12-JAN-1983
7171	MCDONALD	ANALYST	30-JAN-2022

自動サイズ・ブロック

自動サイズ・ブロックが有効化されている場合、マルチレコード/行ブロックには、設計時に特定された行数を最大数として、返されたレコードを表示するのに必要な行数のみが表示されます。この結果、空の行は表示されなくなります。

自動サイズ・ブロックとともに新しいトリガーWHEN-BLOCK-AUTOSIZEDを使用すると、その他のUIコンポーネントを制御できます。たとえば、表示されている数のレコードを収容するためにキャンバスまたはウインドウを高くするか低くすることができます。

自動サイズ・ブロックが有効化されている場合、項目レベルのNumber of Records Displayedプロパティは無視されることに注意してください。また、ブロックにスクロールバーが含まれる場合、その高さは行の追加または削除に応じて自動的には調整されません。

ただし、スクロールバーの長さは、新しいブロック・プロパティBLOCKSCROLLBAR_LENGTHを使用して変更できます。

自動サイズ無効	自動サイズ有効																																																												
<table border="1"> <thead> <tr> <th>Empno</th><th>Ename</th><th>Job</th></tr> </thead> <tbody> <tr><td>7566</td><td>JONES</td><td>MANAGER</td></tr> <tr><td>7902</td><td>FORD</td><td>ANALYST</td></tr> <tr><td>7369</td><td>SMITH</td><td>CLERK</td></tr> <tr><td>7788</td><td>SCOTT</td><td>ANALYST</td></tr> <tr><td>7876</td><td>ADAM</td><td>CLERK</td></tr> <tr><td>7171</td><td>MCDONALD</td><td>ANALYST</td></tr> <tr><td>4122</td><td>SAL</td><td>MARK</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </tbody> </table>	Empno	Ename	Job	7566	JONES	MANAGER	7902	FORD	ANALYST	7369	SMITH	CLERK	7788	SCOTT	ANALYST	7876	ADAM	CLERK	7171	MCDONALD	ANALYST	4122	SAL	MARK													<table border="1"> <thead> <tr> <th>Empno</th><th>Ename</th><th>Job</th></tr> </thead> <tbody> <tr><td>7566</td><td>JONES</td><td>MANAGER</td></tr> <tr><td>7902</td><td>FORD</td><td>ANALYST</td></tr> <tr><td>7369</td><td>SMITH</td><td>CLERK</td></tr> <tr><td>7788</td><td>SCOTT</td><td>ANALYST</td></tr> <tr><td>7876</td><td>ADAM</td><td>CLERK</td></tr> <tr><td>7171</td><td>MCDONALD</td><td>ANALYST</td></tr> <tr><td>4122</td><td>SAL</td><td>MARK</td></tr> </tbody> </table>	Empno	Ename	Job	7566	JONES	MANAGER	7902	FORD	ANALYST	7369	SMITH	CLERK	7788	SCOTT	ANALYST	7876	ADAM	CLERK	7171	MCDONALD	ANALYST	4122	SAL	MARK
Empno	Ename	Job																																																											
7566	JONES	MANAGER																																																											
7902	FORD	ANALYST																																																											
7369	SMITH	CLERK																																																											
7788	SCOTT	ANALYST																																																											
7876	ADAM	CLERK																																																											
7171	MCDONALD	ANALYST																																																											
4122	SAL	MARK																																																											
Empno	Ename	Job																																																											
7566	JONES	MANAGER																																																											
7902	FORD	ANALYST																																																											
7369	SMITH	CLERK																																																											
7788	SCOTT	ANALYST																																																											
7876	ADAM	CLERK																																																											
7171	MCDONALD	ANALYST																																																											
4122	SAL	MARK																																																											

動的サイズ・ブロック

場合によっては、自動サイズ・ブロック機能を使用しても、正確に必要な動作が得られない可能性があります。たとえば、10個のレコードが返されたとしても、10個すべてを表示することは望ましくない場合があります。この場合、代わりに5個のレコードのみを表示する方が望ましいこともあります。新しいRECORDS_DISPLAYEDプロパティを使用すると、必要に応じて、設計時に定義されたMaximum Records Displayedの数まで行を追加または削除できます。例：

```
SET_BLOCK_PROPERTY ('EMP', RECORDS_DISPLAYED, 5);
```

ブロックにスクロールバーが含まれる場合、その高さは行の追加または削除に応じて自動的には調整されません。ただし、スクロールバーの長さは、新しいブロック・プロパティBLOCKSCROLLBAR_LENGTHを使用して変更できます。

レコード間の距離

マルチレコード・ブロックでは、項目のDistance Between Recordsプロパティは、多くの場合、0に設定されています。この結果、行間のスペースなしで1つの行が前の行の直後に表示されるグリッドに似たパターンになります。ただし、これらの行を垂直に分けた方が望ましい場合もあります。これは、たとえば、フィールドのサイズが実行時にプログラムによって変更される場合です。レコード間の距離は、組込みのSET_ITEM_PROPERTYの次の例を使用して変更できます。この数値は、モジュールのCoordinate System設定に定義されている単位で測定されたレコード間の距離です。

```
SET_ITEM_PROPERTY ('EMPNO', RECORD_DISTANCE, <NUMBER>)
```

タイリング

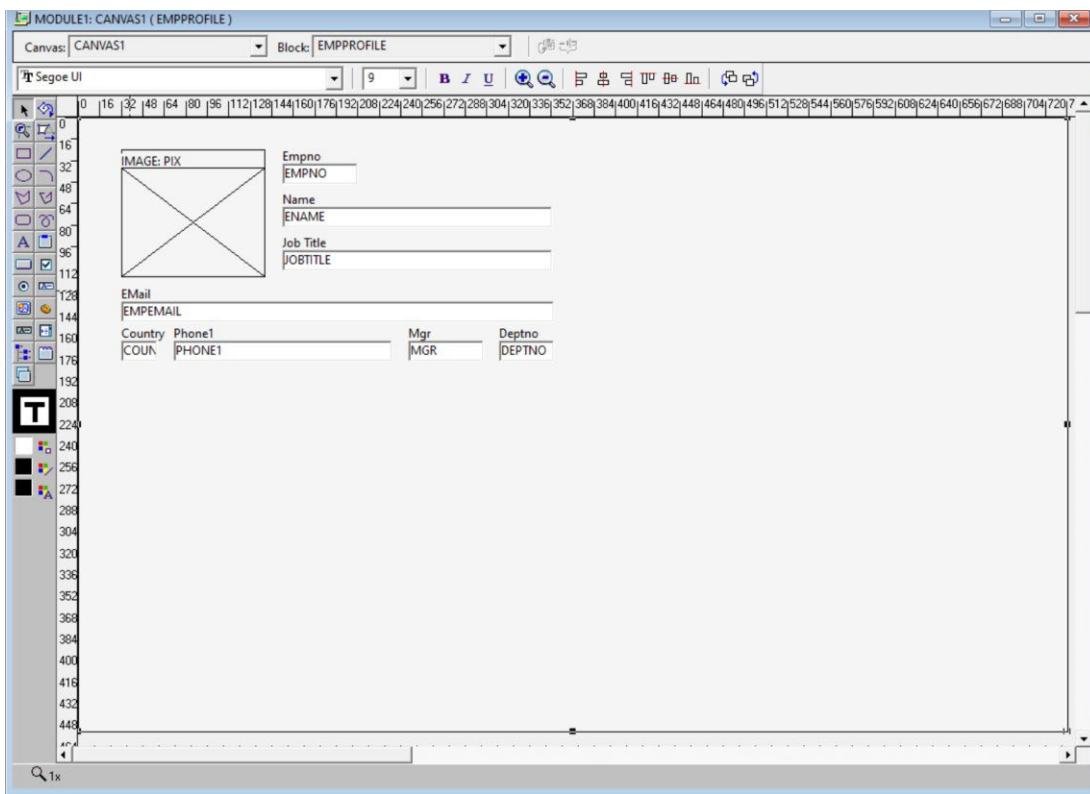
タイリングは、マルチレコード・ブロックの各レコードを視覚的に配置したりグループ化したりできる概念です。これは、“カード”と呼ばれる場合もあります。Formsでは、レコードを垂直向きまたは水平向きにすることができます。この配列により、フィールドを通じたナビゲーション・フローが決まります。タイリングの場合、垂直向きを使用することが推奨されますが、必須ではありません。

ブロック内で一連のタイル・レコードを作成するには、Number of Records Displayedが1に設定されているブロックから開始します。レイアウト・ウィザードを使用して新しいブロックから新しいレイアウトを作成する場合、表形式レイアウトを選択しますが、表示されるレコードの数を1に設定します。

タイルが作成される場合、フレームは自動的には作成されないことに注意してください。各レコード・タイルをフレーム化する方が望ましい場合は、タイルが作成された後にこれらのフレームを手動で作成する必要があります。

レイアウト・エディタ内に单一のレコードが表示されたら、必要に応じて各項目をビジュアル・コレクション内に配置します。

キャンバスとウィンドウのサイズが、最終的に表示される一連のタイルを収容するのに十分な大きさであることを確認してください。次のイメージは、单一のレコードまたはタイルの場合に初期レイアウトがどのように表示されるかを表しています。

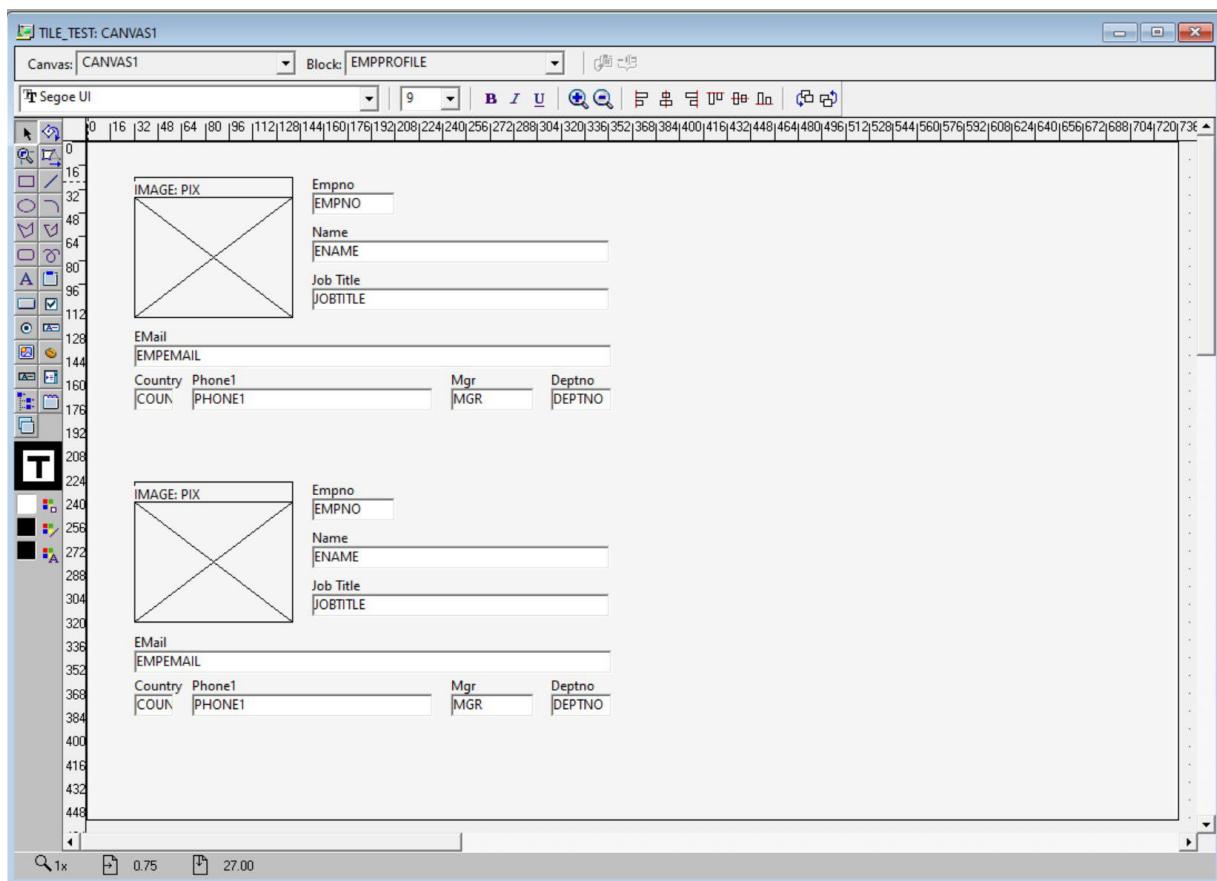
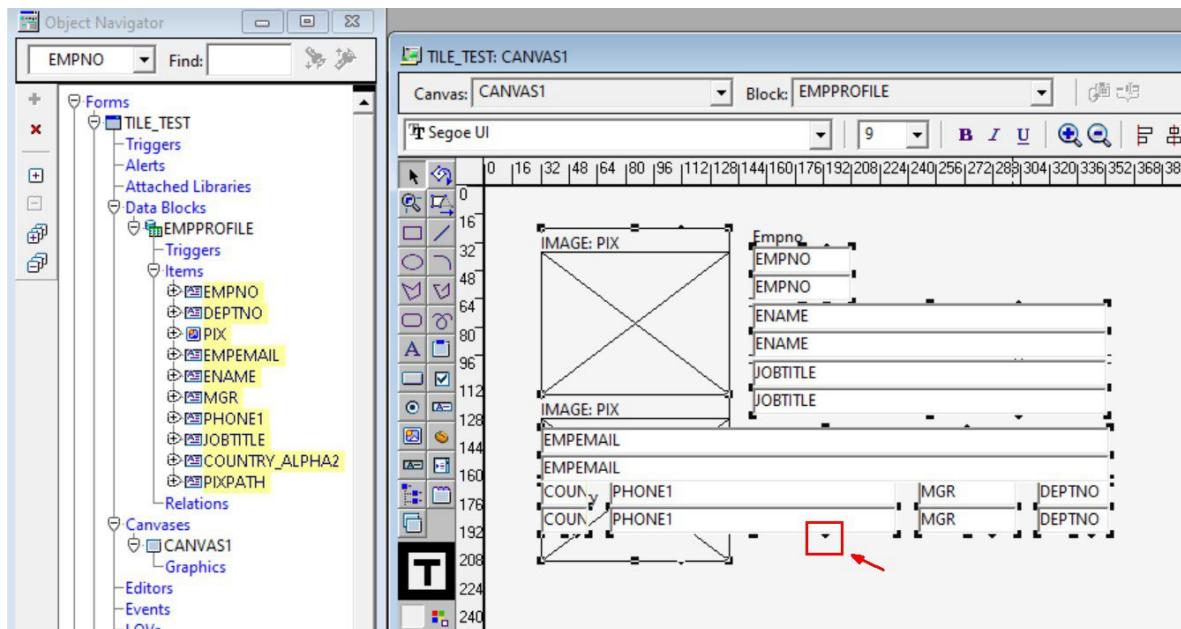


この例では、2つのレコードの2つのグループを表示することが目的であることを前提としています。このブロックは（デフォルトで）レコードを垂直向きに設定するよう構成されているため、グループ1は左側に表示される2つのレコードになり、グループ2は右側の2つのレコードになります。

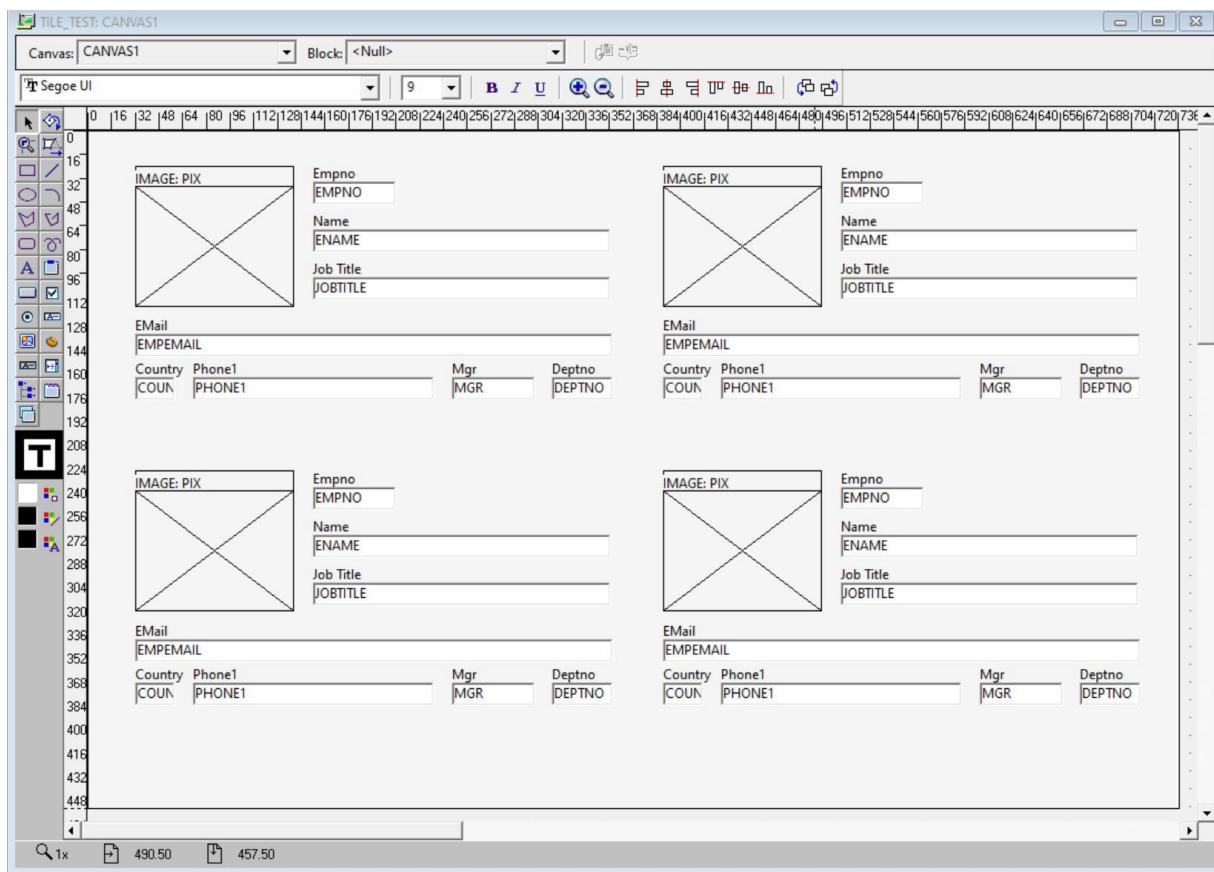
単一レコードのレイアウトが満足できるものになったら、ブロックのNumber of Records Displayedを2に変更し、Records Per Tile Groupを2に変更します。これにより、項目の第2レコード・セットが第1セットの上に表示されるようになります。各項目のDistance Between Recordsプロパティでは、各項目が第1レコードの下の目的の位置に配置されるまで値を大きくします。または、オブジェクト・ナビゲータすべてのレコード項目を選択してから、Distance Between Recordsハンドルを使用して、すべての項目の初期調整を同時に作成できます。このハンドルは、中央右側の選択ボックスの下部にあります。これは、中央のサイズ設定ハンドルの右側に三角形として表示され

す

。表示されている任意の三角形を使用して、マウスを使用してクリックしたまま下にドラッグし、第2レコードの項目の位置を調整します。同じ高さの項目は、調整に応じて同じ距離だけ移動します。異なる高さの項目の場合、プロパティ・パレットでDistance Between Recordsプロパティを手動で調整するか、その項目のDistance Between Recordsハンドルを使用して、マウスを使用して調整することが必要になる可能性があります。



これで、2個のレコードが適切に配置されたので、ブロックのNumber of Records Displayedを4に設定します。Tile Group Distanceを調整して、上記で作成した最初の2個のレコードの横に残りの2個のレコードを配置します。この第2グループのレコードが目的の位置に表示されるまで、Tile Group Distanceの値を大きくします。



WINDOW1

Empno 100002 Name Christine Cooper Job Title Senior Principal Escalation Manager EMail christine.cooper@example.com Country Phone1 Mgr Deptno US 1-999-999-9999 2	Empno 100003 Name Chris Benalamore Job Title Area Business Operations Director EMEA & JAP EMail chris.benalamore@example.com Country Phone1 Mgr Deptno IN 1-999-999-9999 100001 3
Empno 100001 Name Christ Black Job Title Oracle Cloud Infrastructure GTM Channel Direct EMail chris.black@example.com Country Phone1 Mgr Deptno UK 1-999-999-9999 1	Empno 100004 Name Christopher Johnson Job Title Vice-President HCM Application Development EMail christopher.johnson@example.com Country Phone1 Mgr Deptno DE 1-999-999-9999 100002 4

実行時の結果は、上に表示されているイメージのようになります。4個のレコードが表示されていますが、各レコード・セットは、タイルまたはカードに似た構成内に配置されていることに注意してください。フレームまたはその他のグラフィックを使用して各タイルを視覚的に分けることができますが、タイル・セットの作成時にこのようなオブジェクトは自動的には作成されないため、前述のとおり、タイルが作成された後にこのレベルの詳細を追加する必要があります。

この機能の使用方法の詳細については、Builderヘルプ内の特に“About Record Tile Group”というタイトルのページを参照してください。

データ・アクセス、監視、操作

REST

Representational State Transfer (REST) は、インターネットを介したコンピュータ・システム間の相互運用性を実現する手段です。RESTfulサービスを定義することで、基礎となるデータ・ストアに直接アクセスする必要なく、データの問合せと操作が可能になります。

新しいForm BuilderのRESTパッケージ・デザイナ (RPD) を使用して、特別なFormsプログラム・ユニットが生成されます。これを使用すると、アプリケーション開発者は、Javaまたはその他の任意の言語で外部コードを作成する必要なく、Forms PL/SQLからRESTサービスにアクセスできます。また、新しいForms標準パッケージが導入されたため、アプリケーション開発者は、RESTサービスに対するアプリケーションのコールをきめ細かく制御できます。また、これらの新しい組込み機能を使用すると、RESTコールから返されるJSONの解析と操作が可能になります。新しいパッケージの詳細については、Builderヘルプのコンテンツを参照してください。

- FSCRATCHPAD
- FJSON
- FHTTP

The screenshot shows the USPS.com address verification interface. At the top, the USPS logo and "United States Postal Service" are displayed. Below this, there is an address entry form with fields for Street address, Street address line 2, City, State, Zipcode, and Zipcode+4. There are also buttons for "Verify address", "Get zipcode", "Get city/state", and a dropdown menu for "Urbanization (applies to Puerto Rico only)". To the right of the form is a map showing the area around "2300 ORACLE WAY, AUSTIN, Texas". The map includes labels for "Mobile Bicycle Workshop", "ORACLE ATX23 EXEC GARAGE E2", "Lakeshore Park", and "Azul Lakeshore". At the bottom of the interface, a JSON object is displayed in a scrollable box:

```
{"firm":null,"address":{"streetAddress":"2300 ORACLE WAY","streetAddressAbbreviation":null,"secondaryAddress":null,"city":"AUSTIN","cityAbbreviation":null,"state":"TX","postalCode":null,"province":null,"ZIPCode":"78741","ZIPPlus4":"1400","urbanization":null,"country":null,"countryISOCode":null}}
```

Continuous Query Notification

Continuous Query Notification (CQN) をサポートするための新しいFormsデータベース・イベントタイプを使用すると、アプリケーションは、オブジェクト変更通知または問合せ結果変更通知のためにデータベースにFormsのブロック問合せを登録できます。

オブジェクト変更通知 (OCN) 用としてフォームのブロック問合せが登録されている場合、問合せ結果が変更されたかどうかとは関係なく、問合せが参照およびコミットするオブジェクト（表など）がトランザクションによって変更されるたびに、データベースはアプリケーションに通知します。

問合せ結果変更通知 (QRCN) 用として問合せが登録されている場合、問合せとコミットの結果がトランザクションによって変更されるたびに、データベースはアプリケーションに通知します。

OCNまたはQRCNに適したFormsデータベース・イベントおよび対応するWHEN-EVENT-RAISEDトリガーを作成することで、アプリケーションは特定のブロックの問合せに対応できるようになります。オブジェクト/表に対して変更が行われた (OCN) 、または以前に実行された問合せから受け取ったデータセットが変更された (QRCN) という通知をデータベースから受信すると、対応するイベントのWHEN-EVENT-RAISEDトリガーが起動します。

これは、Oracle Databaseオブジェクト/表に関連付けられたFormsデータ・ブロックに対してのみサポートされています。この機能を使用するには、DBMS_CQ_NOTIFICATIONとCHANGE NOTIFICATIONに対する“実行”権限がアプリケーション・ユーザーに付与されている必要があります。

WHEN-EVENT-RAISEDトリガーに、次の1つ以上の新しいプロパティとともに組込みのGET_EVENT_OBJECT_PROPERTYを含めることで、通知に関する次の詳細を収集できます。

- TOTAL_ROWS_AFFECTED
- TOTAL_ROWS_UPDATED
- TOTAL_ROWS_INSERTED
- TOTAL_ROWS_DELETED
- ROWS_INSERTED
- ROWS_DELETED
- ROWS_UPDATED
- TABLE_ALTERED
- TABLE_DROPPED

これらの各プロパティが返す内容に関する詳細については、Builderヘルプを参照してください。

ブロックのソート

任意のブロック（データベースに関連付けられたブロックまたは関連付けられていないブロック）に対してブロックのソートを使用して、同じブロック内の項目別に情報をソートできます。このソート・タスクは中間層で全体的に実行されるため、データベース（データベース・ブロック）に対して再度問合せを行う必要がなくなります。また、アプリケーションでソート・コードを作成する必要もなくなります。コードの單一行を使用して、次の任意の属性を使用してブロックをソートできます。

- ASCENDING/DESCENDING
- NULLS_FIRST/NULLS_LAST
- CASE_SENSITIVE/CASE_INSENSITIVE
- BINARY_ORDER/LINGUISTIC_ORDER

次に例を示します。

```
SORT_BLOCK (:SYSTEM.CURSOR_ITEM, ASCENDING, CASE_SENSITIVE);
```

ソートは、Formsから現在受信したデータを使用して行われます。問合せのすべてのレコードを即時ソートする必要がある場合は、ブロック・プロパティQuery All RecordsをYesに設定します。または、必要に応じてsort_blockコマンドを再発行することもできます。

ソート用としてサポートされているデータ型は、Number、VARCHAR2、またはDateです。

その他の実行時機能

SYSTEM.CURSOR_ROW

Formsのシステム変数SYSTEM.CURSOR_ROWは、マルチレコード・ブロック内のカーソルの可視行番号を返します。このため、レコード番号やその位置とは関係なく、CURSOR_ROWは常に、カーソルの現在の位置を返します。

イメージのコピー（クリップボードへ）

既存の組込みのCOPY_REGIONを使用すると、イメージ項目からシステムのクリップボードへイメージをコピーできるようになりました。クリップボードにコピーされたイメージは、ドキュメントまたはイメージ・エディタに貼り付けることができます。Formsアプリケーション内からの切取りと貼付けは、サポートされていません。この機能を使用するには、目的のイメージ項目に移動し、SELECT_ALL、COPY_REGIONの順に実行します。移動は、ユーザーまたはプログラムによって実行できます。たとえば、次のとおりです。

```
GO_ITEM ('IMAGE_ITEM1');
SELECT_ALL;
COPY_REGION;
```

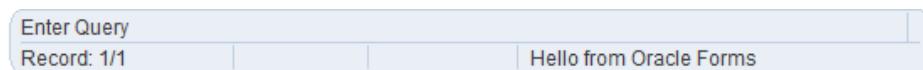
組込みのMESSAGE

MESSAGE引数“JAVA_CONSOLE”を使用することで、メッセージ・テキストをJavaコンソールまたはコマンド・シェルに送信できます（Forms Standalone Launcherを使用している場合）。例：

```
MESSAGE ('Hello from Oracle Forms', JAVA_CONSOLE);
```

MESSAGE引数“STATUS”を使用することで、メッセージ・テキストをメッセージ・バーに送信し、このコマンドが再実行されるまでメッセージ・テキストを永続的に維持できます。例：

```
MESSAGE ('Hello from Oracle Forms', STATUS);
```



ユーザーのJavaバージョンの取得

ユーザーのマシン上でどのJavaバージョンが使用されているかを把握すると、役に立つ場合があります。次を実行すると、アプリケーションを開始するために使用されたJavaバージョンが返されます。

```
GET_APPLICATION_PROPERTY (CLIENT_JAVA_VERSION);
```

クライアント構成の取得

アプリケーションを実行するためにどのFormsクライアント構成が使用されているかを把握すると、役に立つ場合があります。次を実行すると、どの構成が使用されているかを示す、次のいずれかの値が返されます。

- STANDALONE
- WEBSTART
- EMBEDDED

次に例を示します。

```
GET_APPLICATION_PROPERTY(CLIENT_DEPLOYMENT);
```

開かれたモジュールの記録

Forms環境変数FORMS_LOG_MODULE_OPEN=1を設定することで、実行時に開かれたモジュールがForms ODL（Oracle Diagnostic Log）のformsapp-diagnostic.logに記録されます。開かれた各モジュールのエントリは、次の例のように表示されます。この機能を使用すると、過剰なログ書き込みが行われる場合があるため、この機能はテストのみを目的として使用する必要があります。この機能は本番環境で使用できますが、ファイル・サイズが大きくなりすぎないようにログ・サイズを監視する必要があります。

```
[2025-01-15T22:30:04.341+00:00] [WLS_FORMS] [NOTIFICATION] [FRM-91940] [oracle.forms.servlet]
[tid:107] [userId: <anonymous>] [ecid:<EDID HERE>,0] [APP: formsapp#14.1.2]
[FORMS_SESSION_ID:WLS_FORMS.formsapp.77] [SRC_CLASS: oracle.forms.servlet.RunformProcess]
[SRC_METHOD: fromFrmwebToODL] module /home/oracle/forms/MODULE4.fmx loaded by Forms runtime
process
```

WebUtil

WEBUTIL_FILE_TRANSFER.URL_TO_CLIENT

URL_TO_CLIENTの転送レートが大幅に改善しました。

WEBUTIL_FILE.FILE_MULTI_SELECTION_DIALOG

FILE_MULTI_SELECTION_DIALOGから返される文字列の最大長が4000から32767に増加しました。この結果、マルチ選択ファイル・ダイアログで、より多くのファイル名を選択できるようになりました。

CLIENT_TEXT_IO

CLIENT_TEXT_IOの最大行サイズが4000から32767に増加しました。

クライアント構成

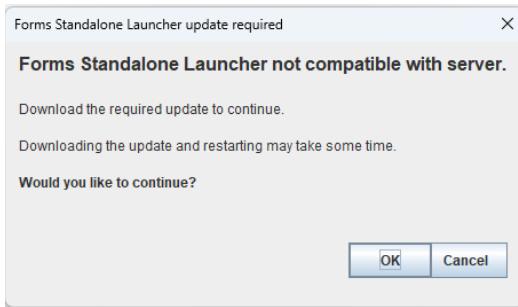
Forms Standalone Launcher

自動更新

Forms Standalone Launcher (frmsal.jar) は、サーバーのパッチ・レベルと一致しています。ランチャとサーバーが互いに互換性がない場合、この状態についてユーザーに警告が通知されます。ユーザーには、ランチャの更新を続行するか終了するかが尋ねられます。この機能は現在、同じメジャー・バージョンからのみサポートされています。つまり、パッチが適用されたサーバー更新のみをこのプロセスを介して提供できます。メジャー・バージョンのアップグレードは、手動で実行する必要があります。また、デジタル署名証明書が更新されていない場合にのみ制限されています。オラクルでは、3年ごとに1回、証明書を更新しています。

自動更新のプロンプトは、次のWeb構成設定を設定することで、無効化できます。

`fsalEnableAutoUpdate=FALSE`

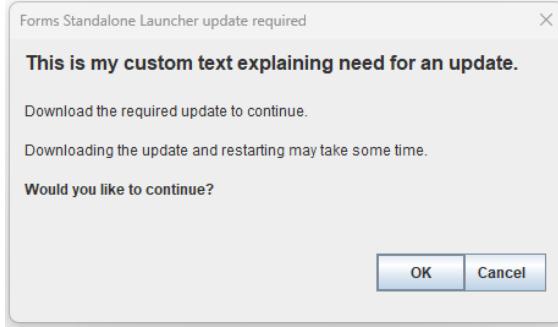


`fsalEnableAutoUpdate=FALSE`を設定することで、v12の動作を体験できます。つまり、管理者は、更新されたfrmsal.jarファイルをすべてのユーザーに提供する必要があります。

更新ダイアログのテキスト

場合によっては、ダイアログに独自のメッセージを追加する方が望ましい場合もあります。ダイアログ内の最初の行は、最大128文字の独自のテキストを使用してカスタマイズできます。カスタム・テキストを表示するには、Web構成パラメータ`fsalUpdateDialogText`を設定し、目的のテキストを追加します。たとえば、次のとおりです。

`fsalUpdateDialogText= This is my custom text explaining need for an update.`



キャッシュのクリア

FSALによって以前にダウンロードされたキャッシュ・ファイルをクリアするために使用できる、新しいコマンドライン引数が導入されました。この引数をTRUEに設定すると、キャッシュはクリアされます。この引数が有効化されている場合、その他すべての引数が無視されます。たとえば、次のとおりです。

```
java -jar frmsal.jar clearCache TRUE
```

ホスト名の検証の迂回

テスト中にホスト名の検証を迂回する機能が必要になる場合があります。例として、オラクルが提供するテスト用の“デモ”証明書の使用を試みると、この証明書はこの目的で作成されたものではないため、SSLを使用した実行の試みは失敗します。サーバーによって提示された証明書からホスト名を検証できない場合、デフォルトでは、次のようなダイアログが表示されます。



Web Start

JNLPの自動削除

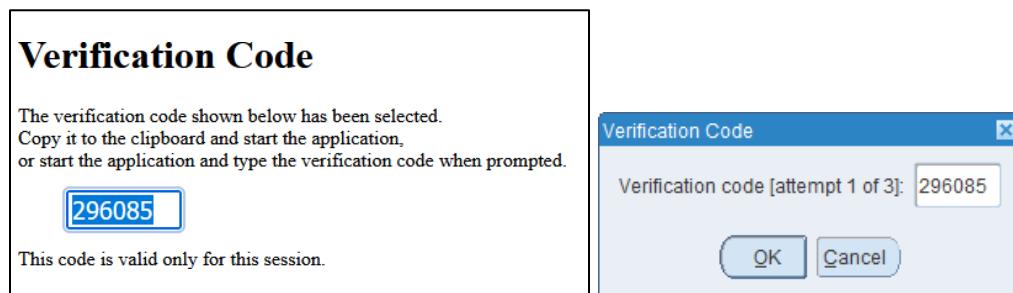
Java Web Startを使用して起動したアプリケーションに対してユーザーが最新の構成を使用するよう徹底するために、ダウンロードされたJNLPファイルは、使用された後、自動的に削除されます。アプリケーションには、Formsサーブレット（frm servlet）にアクセスすることで、常にリクエストが行われる必要があります。以前にダウンロードされたJNLPファイルが使用された後に管理者がアプリケーションの構成を変更している可能性があるため、以前にダウンロードされて使用されたJNLPファイルの再実行を試みることは危険である可能性があります。この機能が望ましくない場合、アプリケーションによって使用されているFormsテンプレート（base.jnlpなど）を編集することで、この機能を無効化できます。デフォルト値は“TRUE”です。これを“FALSE”に変更し、JNLPファイルの自動削除を無効化します。ただし、これは推奨されていません。

```
<property name="jnlp.delete.jnlp.file" value="false"/>
```

2要素認証

ダウンロードされたJNLPファイルをユーザーがその他のユーザーと共有しないよう徹底し、前述のJNLPの自動削除に加えて、Java Web Startを使用して実行されているFormsアプリケーションの全体的なセキュリティを強化するには、Java Web Startを使用して実行されているFormsアプリケーションに対して2要素認証（2FA）を有効化します。この実装は完全な2FAソリューションではありませんが、ダウンロードされたJNLPファイルが再使用されることを防止できます。

有効化すると、JNLPファイルがダウンロードされるときに同時に、ブラウザ内でユーザーに検証コードが提供されます。JNLPファイルが起動されると、検証コードをリクエストするダイアログが表示されます。正しいコードが入力されない場合、アプリケーションは起動しません。検証コードとJNLPは、両方とも同じユーザーのマシンに送信されるため、このコードの共有も行われていない限り、JNLPを別のマシンにコピーする試みは失敗します。しかし、12.2.1.19に導入された2つのWeb構成設定jnlpMatchIPおよびjnlpTimeoutを有効化することで、これも回避できます。



Fusion Middleware Control

User Sessionsページ

FormsのUser Sessionsページで2つの新しい列が使用できるようになりました。SSO UsernameとEvent Listener Portです。これらの列

デフォルトでは表示されませんが、表の左上にあるViewリストから選択することで、簡単に追加できます。

セッションで認証用としてシングル・サインオンが使用されている場合、SSO Usernameは移入されます。Event Listener Portは、Advanced QueueingとContinuous Query Notificationによって使用されます。

View ▾ Stop Enable Tracing... Disable Tracing Notify...						
Process ID	Database	CPU Usage	Private Memory (KB)	DB Username	SSO Username	Event Listener Port
4103174	orcl19	0	53376	scott		40139

Advanced Configurationページ

Advanced Configurationページには、構成ファイルを編集するための完全なテキスト・エディタ・アプローチが用意されています。もっとも一般的に編集されるForms構成ファイルの一部がこのページからアクセスできるようになったため、構成をより簡単に変更できるようになりました。

Advanced Configuration - WLS_FORMS

View or edit a Forms configuration or template file. Use the Select Category and Select File lists to select a file, then edit.

Create Like Delete Select Key Binding Files

Select Category Client Templates Environment Font/Icon Mapping Miscellaneous

Select File formsweb.cfg ftrace.cfg jvmcontrollers.cfg webutil.cfg

リモート・アクセス記述子

リモート・アクセス記述子（RAD）は、シングル・サインオンを介した認証のプロセスを支援するために使用されます。多くのユーザー用としてRADエントリを一度にバルク・アップロードできると、システム間の移動に要する管理上の労力を軽減しやすくなります。バルク・アップロードに必要な構文の詳細については、『Working with Oracle Forms』ガイドを参照してください。

Upload RADs

Overwrite duplicate RADs

Find Go to Line

```
appName="sales" isGroup="true" comments="foo" dbUsername="scott" dbPassword="tiger" dbName="orcl"
2 appName="finance" isGroup="false" comments="foo" ssouserName="john.doe@example.com" dbUsername="scott" dbPassword="tiger" dbName="orcl"
3 |
```

Apply Cancel

Builder

ログへのFSAL出力の公開

Builderからフォームを起動する機能は、新しい概念ではありません。しかし、12cでは、Forms Standalone Launcherを使用して、現在開いているフォームを起動する機能が導入されました。これに関する問題の1つは、エラー、警告、一般情報メッセージなどのシェル・メッセージがマスクされることでした。これらのメッセージを表示する方法がなかったため、FSALを使用したOneButtonRun Builderオプションの使用に関する問題を簡単にトラブルシューティングする方法がありました。

新しい環境変数FORMS_BUILDER_FSAL_LOGGINGを有効化すると、システムの一時ディレクトリ内にログ・ファイルが作成されます。このファイル名はfsal_<random string>です。このファイルには、FSALをコマンドラインから直接実行するときに、一般的に公開されるすべての詳細情報が含まれます。有効な値は次のとおりです。

- 0 : 追加情報を何も表示しないことを示します。
- 1 : リソースのロード元ロケーションに関する詳細情報を表示します。
- 2 : 証明書情報がFSALトラストストアに含まれない場合に、SSL/TLS関連情報に関する詳細情報を表示します。
- 99 : すべての情報を表示します。

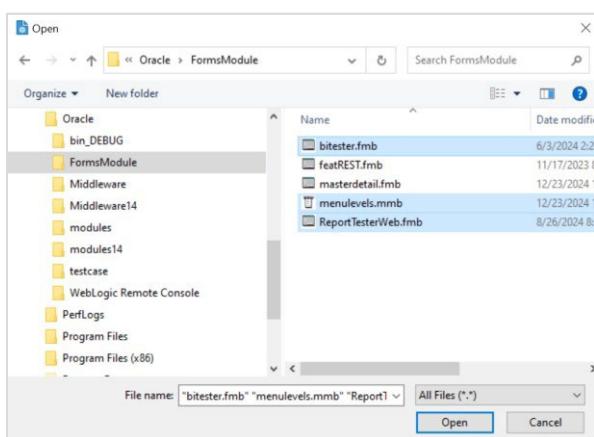
FSALが起動するたびに新しいログ・ファイルが作成されるため、この機能を有効化するのはトラブルシューティング時のみにする必要があります。

起動時のパフォーマンスの改善

MS Windowsでは、Builderが起動して使用可能になるまでに非常に時間がかかる場合があります。これは、Builderが、マシンにインストールされているすべてのフォントを評価し、使用できるフォントとその使用方法を確認しようと試みることが原因です。このため、多くのフォントを備えたマシンの場合、起動時間が長くなります。Windowsレジストリで環境変数TK2_DELAY_WIDTH_CALCを有効化することで、Builderは起動時にフォント確認をスキップし、代わりに必要に応じた確認のみを行います。

複数のファイルを開けるダイアログ

一度に複数のモジュールを選択して開くことができる機能は、プロジェクトで作業を素早く開始できる便利な方法です。マウスのクリックとともに[CTRL]キーまたは[SHIFT]キーを使用することで、任意の数のモジュールを選択してBuilderにロードできます。



メッセージ・バー内のデータベース名

メッセージ・バーにデータベース名が表示されるため、どのデータベースが使用されているかを簡単に確認できるようになりました。

フォーム・プロパティ – Application Name

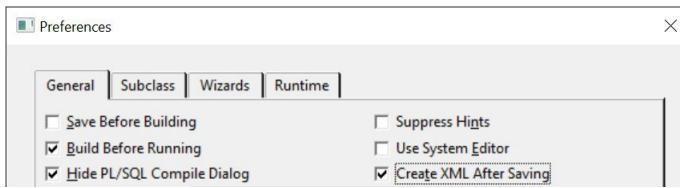
新しいフォーム・レベルのプロパティApplication Nameを使用してモジュールを分類し、後でJava Development API（JDAPI）を使用して、これらを特定できます。このプロパティは、Builderでのみ設定でき、任意の文字列値を受け入れます。ただし、このモジュールが適用されるアプリケーションに名前を提供するために使用することを推奨します。モジュールが複数のアプリケーションで使用される可能性がある場合、カンマ区切りのリストを推奨します。例：sales, hr, inventory

このプロパティを使用すると、このモジュールを処理すべきかどうか、およびその処理方法に関するインテリジェントな決定がJDAPIコードで下されます。Forms 14cのJDAPI JavaDocは、Oracle Forms製品のホーム・ページからダウンロードできます。

Create XML After Save

Builderによって使用されるFormsモジュール・タイプは、人間が読めないバイナリ・ファイルです。人間が読めないバイナリ・ファイルにより、バージョン管理、コード比較、その他の開発タスクや管理タスクなどを達成するのが不可能ではないが困難になります。モジュールをXMLなどのその他の形式に変更する機能は長い間、使用可能でした。しかし、これを行うには、変換を行うための追加手順が必要でした。

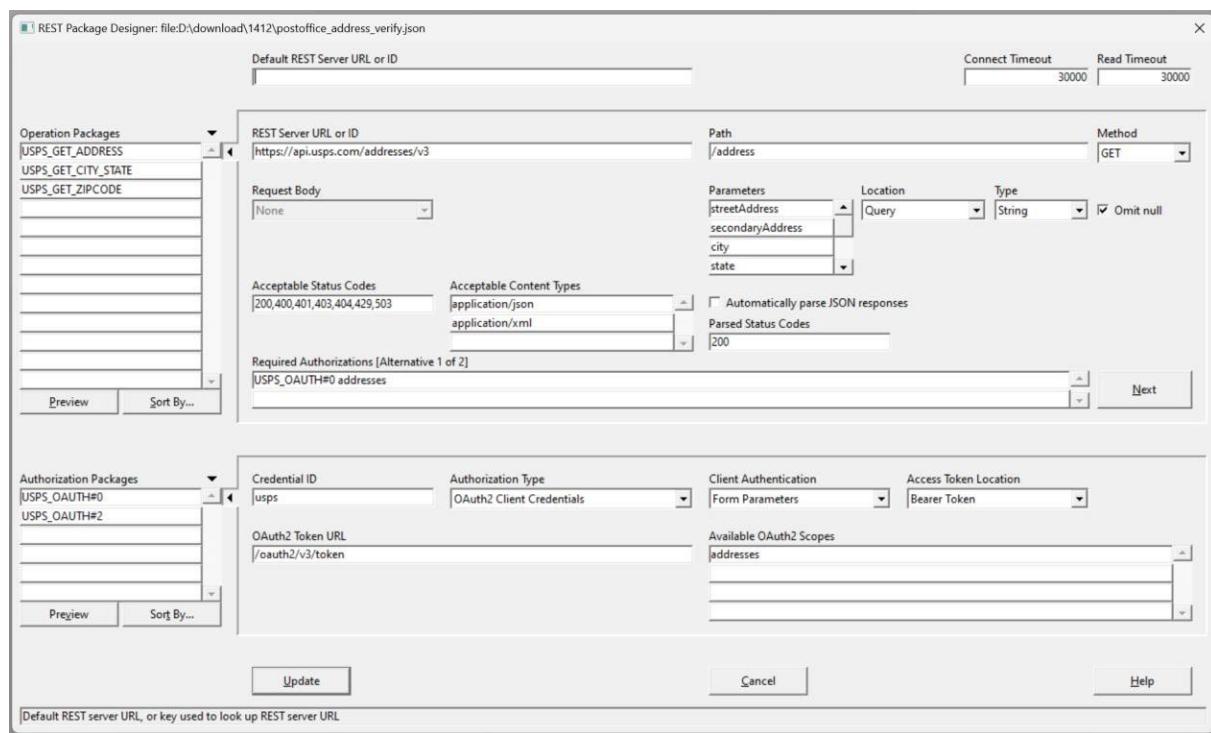
新しいプリファレンスCreate XML After Savingを設定することで、Builderは、変更の保存時にBuilderで開かれたFMB、MMB、またはOLBファイルのXMLバージョンの作成を試みます。開いているモジュールへの変更の保存を試みる場合、モジュールのバイナリ・ファイルが最初に保存された後、XML Converterを使用して、そのバイナリ・モジュールのXMLバージョンが作成されます。この変換の完了を示すダイアログが表示されます。このダイアログを表示することが望ましくなく、よりサイレントなアプローチが必要な場合、FORMS_BUILDER_SILENT_CONVERT=0を設定します。ただし、サイレント変換を実行する場合、失敗メッセージが表示されないことを理解してください。



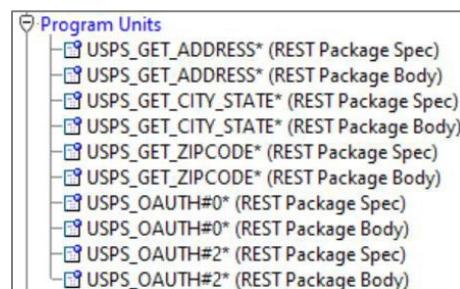
RESTパッケージ・デザイナ (RPD)

RESTパッケージ・デザイナ (RPD) は、データにアクセスして操作するためのRESTエンドポイントへのコールを行うことができるアプリケーション・コードをForms開発者が作成することを可能にする主要コンポーネントです。RPDを使用して、RESTと連携して使用することを目的とした特別なFormsプログラム・ユニット (PU) を作成します。これらのプログラム・ユニットは、RPDに提供された情報に基づいて自動的に作成されます。RESTサービスがOpenAPIまたはSwaggerドキュメントを提供しない場合、RPDを使用して、このサービスに関する詳細情報を手動で提供できます。これらの詳細情報が提供されたら、RPDによって特別なForms PUが自動的に作成され、これらに対してコーディングできるようになります。このサービスが互換性のあるOpenAPIまたはSwaggerドキュメントを提供する場合、このドキュメントを使用して、必要な情報の大半をRPDに自動的に移入できます。とは言え、初期情報がどのように入力されたか（自動的または手動）とは関係なく、RPDで編集を行うことは常に可能です。

OpenAPIまたはSwagger 2.xまたは3.xのみがサポートされており、ドキュメントはJSON形式である必要があります。YAMLとその他の形式はサポートされていません。YAMLなどのその他の形式を使用するには、YAMLからJSONへのコンバータを使用して、互換性のあるドキュメントを作成できます。このタスクを実行できる多くの変換ツールをオンラインで入手できます。



パッケージが作成されたら、パッケージにRESTパッケージ仕様とボディ識別子が組み込まれます。これらのパッケージのコンテンツは、手動で編集してはなりません。これらは、RPDを再入力することによってのみ編集する必要があります。



まとめ

Oracle Forms 14.1.2には、多くの新機能が同梱されています。改善からユーザー・インターフェース、RESTサービス・コールからデータに簡単にアクセスする機能、Formsアプリケーションの最新化までのタスクが大幅に簡単になりました。本文書では多くの新機能が説明されていますが、すべてが含まれているわけではありません。一部の新機能は大幅に軽微な機能であり、それらは説明していません。本文書で説明されている機能やその他の機能とともに使用できる新しい設定の多くが[Working with Oracle Forms](#)ガイドの「References」セクションに含まれるため、このセクションを必ず参照してください。

Connect with us

+1.800.ORACLE1までご連絡いただぐか、oracle.comをご覧ください。北米以外の地域では、oracle.com/contactで最寄りの営業所をご確認いただけます。

 blogs.oracle.com  facebook.com/oracle  twitter.com/oracle

Copyright © 2023, Oracle and/or its affiliates.本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による默示的保証を含め、商品性ないし特定目的適合性に関する默示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle、Java、MySQLおよびNetSuiteは、Oracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。