

## **4D Splash Screen**

---

By Thomas Fitch, Technical Services Team Member, 4D Inc.

Technical Note 09-14

## 概要

---

4D Splash Screen テクニカルノートは、あなたのデータベースにカスタマイズされたスプラッシュスクリーンを挿入するために使用できる 4D コンポーネントを紹介します。スプラッシュスクリーンの設定はウィザードタイプのフォームまたは XML 環境設定ファイルを直接編集することで行うことができます。

## はじめに

---

このテクニカルノートは 4DSplash コンポーネントデータベースを紹介します。このコンポーネントを使用して、データベースにカスタムスプラッシュスクリーンをインストールできます。XML 環境設定ファイルを編集するだけで、同じコンポーネントをあなたのすべてのデータベースで使用できます。テクニカルノートの最初ではコンポーネントの利用方法を説明します。コンポーネントデータベースをインストールして、設定を行います。設定はウィザードスタイルのフォームを使用してプログラマ的に行うか、XML 環境設定ファイルを直接編集します。コンポーネントの紹介の後は、ソースコードを見ることにします。

## 4DSplash コンポーネントのインストール

---

4D v11 SQL に慣れ親しんでいる開発者であれば、コンポーネントのインストールは簡単です。データベースフォルダの中 (ストラクチャファイルと同階層) に "Components" フォルダを作成し、コンパイルされたコンポーネント 4DSplash.4dbase を配置します。作成済みのコンポーネントはサンプルデータベースの Components フォルダにあります。またコンポーネントをカスタマイズし、独自のビルドを作成することもできます。さらにコンパイルモードではなく、インタプリタモードでデータベースを実行することもできます。

データベースの Components フォルダに 4DSplash コンポーネントを配置したら、あとは On Startup データベースメソッドに 4DSP\_SplashScreen コンポーネントメソッドの呼び出しを記述するだけです。これにより、起動時にカスタムスプラッシュスクリーンが表示されます。

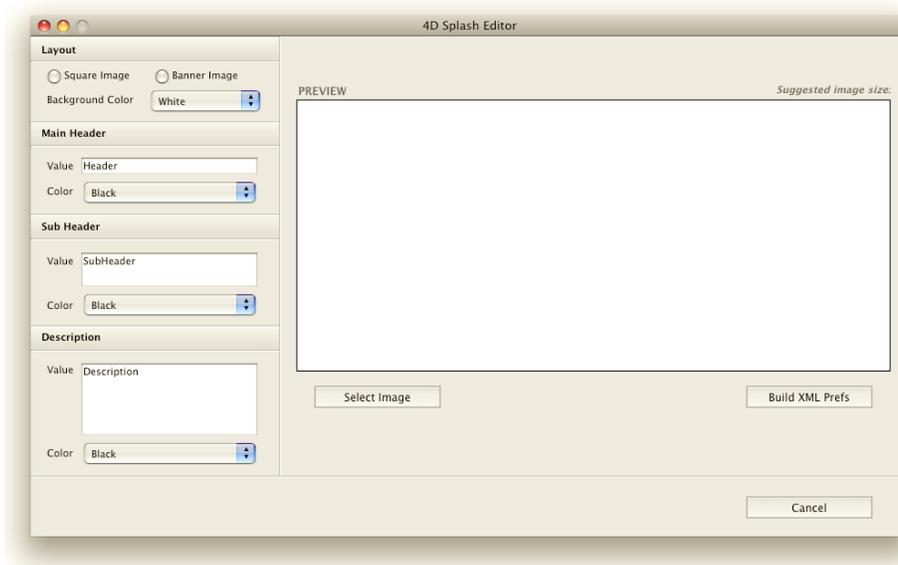
## スプラッシュスクリーンのアピランス設定

---

カスタムスプラッシュスクリーンを設定する方法は3つあります。4DSplash Chooser フォームを使用するか、4DSP\_SplashScreen メソッドを使用するか、demoinfo.xml ファイルを編集するかです。

### 4DSplash Chooser フォームを使用する

4DSplash Chooser フォームは 4DSP\_SplashChooser メソッドを呼び出すことで開くことができます。デフォルトのテンプレートの例題は以下のとおりです:

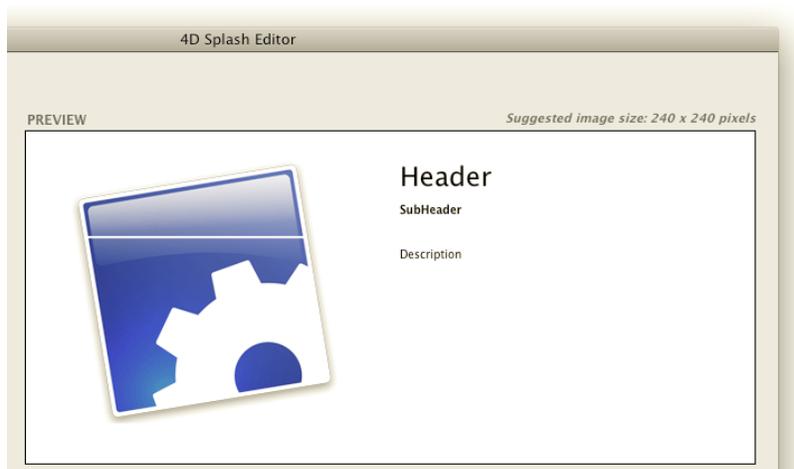


設定可能なオプションがすべて表示されています。それぞれの項目について説明します。

2つのテンプレートオプションがあります: ひとつは四角アイコン画像で、もうひとつはバナースタイルのアイコン画像です。テンプレートはラジオボタンで選択します。同じエリアで、ドロップダウンリストを使用して背景色を選択できます。SET COLOR コマンドで使用可能なカラーをリストから選択できます。これが "Layout" オプションです:



テンプレートが選択されると、スプラッシュスクリーンのサンプルがフォームの右側に表示されます。オプションを変更するたびにこのサンプルは更新されます。以下の例題は四角画像のデフォルトテンプレートです:



テキストエリアにヘッダテキストを入力します。その下にはヘッダテキストのカラーを設定するためのエリアがあります。同様にしてサブヘッダと説明も設定できます:



テンプレートごとに推奨される画像のサイズは、ここで表示されるサンプルスプラッシュスクリーンで与えられたものです。これがフォーム上のピクチャ変数のサイズであり、ゆえに最もクリアなアイコンを表示するにはこのサイズに合わせます。ピクチャはスケーリング (中央合わせ・プロポーショナル) に設定されているため、小さいあるいは大きな画像も使用することはできます。画像の背景は透過にすることをお勧めします。サンプルデータベースでは、四角画像の背景は透過になっています。この場合、背景色に何を選択しても見栄えは良くなります。バナー画像はフルサイズのブロックイメージで、選択された背景色に関係なく、黒い背景色になっています。

"Select Image" ボタンをクリックするとファイルを選択ダイアログが表示され、画像を選択できます。その画像は Resources フォルダにコピーされ、カスタムスプラッシュスクリーンのアイコンとして設定されます。

以下の例題ではヘッダー、サブヘッダー、説明が設定されています。テンプレートはバナーに変更され、背景に黒が選択されています。テキストカラーは白が選択されています:



希望通りのスプラッシュスクリーンが出来上がったら、"Build XML Prefs"ボタンをクリックして環境設定ドキュメントを保存できます。次にデータベースが開かれるとき、このカスタムスプラッシュスクリーンが表示されます。

## プログラムでアピランスを設定する

4DSP\_SplashScreen メソッドを使用して、プログラムからアピランスを設定できます。9つの引数を渡してこのコマンドを呼び出すと、XML 環境設定ファイルは無視され、引数に基づいてスプラッシュスクリーンが表示されます。引数値の説明は以下のとおりです:

\$1	テキスト	Resources フォルダ内に配置された、ロゴピクチャの名前。このピクチャはコンポーネントデータベースの Resources フォルダに置かれていなければなりません。 四角イメージアイコンの推奨サイズは 240 x 240 ピクセル。 バナーイメージアイコンの推奨サイズは 550 x 81 ピクセルです。
\$2	テキスト	ヘッダーとして表示されるテキスト。
\$3	テキスト	サブヘッダーとして表示されるテキスト。
\$4	テキスト	説明として表示されるテキスト。
\$5	倍長整数	ヘッダテキストのカラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
\$6	倍長整数	サブヘッダテキストのカラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
\$7	倍長整数	説明テキストのカラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
\$8	倍長整数	フォームの背景色カラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
\$9	倍長整数	開くフォームページ番号。四角アイコンの場合 1 を、バナーアイコンの場合 2 を使用。

4DSP\_SplashScreen メソッドに引数を指定して呼び出すと、カスタム表示設定を使用してスプラッシュスクリーンウィンドウが開かれます。

## demoinfo.xml ファイルを編集

XML 環境設定ファイルを直接編集して、スプラッシュスクリーンのアピランスを設定することもできます。これはプログラムでアピランスを設定するのと同じです。

4DSP\_SplashScreen メソッドに引数を渡す代わりに、XML 環境設定ファイルを編集します。これを行うにはコンポーネントデータベースの Resources フォルダ内にある "demoinfo.xml" ファイルを開きます。編集すべき XML ノードは以下のとおりです:

/demo/picture	Resources フォルダ内に配置された、ロゴピクチャの名前。 このピクチャはコンポーネントデータベースの Resources フォルダに置かれていなければなりません。 四角イメージアイコンの推奨サイズは 240 x 240 ピクセル。 バナーイメージアイコンの推奨サイズは 550 x 81 ピクセルです。
/demo/header	ヘッダーとして表示されるテキスト。
/demo/subheader	サブヘッダーとして表示されるテキスト。
/demo/description	説明として表示されるテキスト。
/demo/background	ヘッダテキストのカラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
/demo/headercolor	サブヘッダテキストのカラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
/demo/subheadercolor	説明テキストのカラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
/demo/descriptioncolor	フォームの背景色カラー。4D の SET COLOR コマンドのドキュメントで説明されている 0 から 15 の値を渡すことができます。
/demo/page	開くフォームページ番号。四角アイコンの場合 1 を、バナーアイコンの場合 2 を使用。

"demoinfo.xml" ファイル編集後は、4DSP\_SplashScreen メソッドを引数なしで使用できます。

## 4DSplash のソースコード

---

4DSplash コンポーネントの2つの関数はスプラッシュスクリーンやスプラッシュスクリーン選択ウィザードを表示するために使用されます。2つの関数は次のような同様の状況や同様のタスクを処理しなければなりません:

- スプラッシュスクリーンの表示 (4DSplash Chooser 内では、スクリーンはビルドされるサンプルとして表示されます。)
- XML 環境設定ファイルの読み込み
- 更新された設定を XML 環境設定ファイルに書き込む (実際にスプラッシュスクリーンを表示するとき、環境設定ファイルが存在しなければ作成されます。)

それぞれの状況はスプラッシュスクリーン表示または 4DSplash Chooser スクリーン表示のコード例題で個別に説明します。

### スプラッシュスクリーンの表示

以下は 4DSP\_SplashScreen コンポーネントメソッドのコードです。いくつかのリテラル文字列は読みやすくするために変更してあります:

```
C_TEXT($1;$2;$3;$4)
C_LONGINT($5;$6;$7;$8;$9)
C_LONGINT($p1)
Case of
  : (Count parameters>=9)
    $p1:=New Process("Load";1024*1024;"Proc";$1;$2;$3;$4;$5;$6;$7;$8;$9;*)
  Else
    $p1:=New Process("Load";1024*1024;"Proc";*)
End case
```

このコードはスプラッシュスクリーンプロセスを開始するために使用されます。このコードはスプラッシュスクリーンとスプラッシュ Chooser で同じです。“プログラムでアピランスを設定する”で説明したように、定義済みの設定またはプログラムで指定した設定を使用してスプラッシュスクリーンを表示できます。引数を渡さない場合、4DSP\_Load\_SplashScreen メソッドが呼ばれ、XML 環境設定ファイルを使用してスプラッシュスクリーンを表示します。この

メソッドに引数が渡されると、その引数は 4DSP\_Load\_SplashScreen に渡され、スプラッシュスクリーンの表示に使用されます。

以下は 4DSP\_Load\_SplashScreen メソッドのコードの一部です:

```
If (Count parameters>=9)
  4DSPlogo_t:=$1
  4DSP1_t:=$2
  4DSP2_t:=$3
  4DSP3_t:=$4
  4DSP1fg_l:=$5
  4DSP2fg_l:=$6
  4DSP3fg_l:=$7
  4DSPbg_l:=$8
  4DSPpage_l:=$9
Else
```

ここでは 4DSP\_Load\_SplashScreen メソッドに渡された引数に基づいて、表示設定を初期化しています。引数の意味は以下のとおりです:

4DSPlogo\_t, \$1: Resources フォルダ内のロゴピクチャ名

4DSP1\_t, \$2: ヘッダーとして表示されるテキスト

4DSP2\_t, \$3: サブヘッダーとして表示されるテキスト

4DSP3\_t, \$4: 説明として表示されるテキスト

4DSP1fg\_l, \$5: ヘッダーテキストの色

4DSP2fg\_l, \$6: サブヘッダーテキストの色

4DSP3fg\_l, \$7: 説明テキストの色

4DSPbg\_l, \$8: フォームの背景色

4DSPpage\_l, \$9: フォームを開くページ番号

このコードのすべてのカラーは SET COLOR コマンドで割り当てられ、ゆえに 4D カラー定数に対応する 0 から 15 の数でなければなりません。スプラッシュスクリーンのフォームページは、それぞれ異なるテンプレート設定を持っています。4DSplash コンポーネント内では、ページ 1 は四角アイコンを表示し、ページ 2 はバナーアイコンを表示します。これは独自のテンプレートを実装する際に変更することが可能です。この例題では、表示設定は入力された引数に基づいて設定されます。または XML 環境設定ファイルに基づいて設定することもできます。この点については “XML 環境設定ファイルを読み込む” で説明します。

このメソッドの次の部分は、実際のスプラッシュスクリーンを表示します:

```
$P:=Open form window("4DSplash";Pop up window ;Horizontally Centered ;Vertically Centered )  
DIALOG("S")  
CLOSE WINDOW($P)
```

このコードはウィンドウを開き、4DSplash フォームを表示します。これは 4DSP\_Load\_SplashScreen メソッドのごく一部です。コードの残りの部分は XML 環境設定から設定を読み込んだり書き込んだりします。

4DSplash フォームメソッドにも重要なスプラッシュスクリーンコードが記述されています:

```
: (Form event=On Load )  
C_PICTURE(4DSPlogo_p)  
READ PICTURE FILE(Get 4D folder(Current Resources folder ) + logot; logop)  
SET COLOR(*;"Rectangle1";-(4DSPbg_l+(256*4DSPbg_l)))  
SET COLOR(*;"4DSP1_t@";-(4DSP1fg_l+(256*4DSP1fg_l)))  
SET COLOR(*;"4DSP2_t@";-(4DSP2fg_l+(256*4DSP2fg_l)))  
SET COLOR(*;"4DSP3_t@";-(4DSP3fg_l+(256*4DSP3fg_l)))  
GOTO PAGE(4DSPpage_l)  
SET TIMER(60*10)  
: (Form event=On Timer )  
SET TIMER(0)  
CANCEL
```

ここで **SET COLOR** コマンドはスプラッシュスクリーン上のテキストと背景のカラーを処理します。注目していただきたいのは **SET TIMER** です。これは 10 秒後にスプラッシュスクリーンをキャンセルするために使用されます。またスプラッシュスクリーンの最前面オブジェクトは **CANCEL** コードが記述された非表示ボタンです。ユーザがフォーム上をクリックすると、フォームはキャンセルされます。

## XML 環境設定ファイルへの書き込み

以下のコードは 4DSplashChooser コンポーネントフォームの Build XML Prefs ボタンオブジェクトメソッドから一部取り出したものです。

```
$infopath_t:=Get 4D folder(Current Resources folder )+"demoinfo.xml"  
If (Test path name($infopath_t)#Is a document )  
  $xmlref_t:=DOM Create XML Ref("demo")  
  If (OK=1)  
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/picture")
```

```

DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSPlogo_t)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/header")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSP1_t)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/subheader")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSP2_t)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/description")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSP3_t)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/headercolor")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSP1fg_l)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/subheadercolor")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSP2fg_l)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/descriptioncolor")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSP3fg_l)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/backgroundcolor")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSPbg_l)
$xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/page")
DOM SET XML ELEMENT VALUE ($xmlsubref_t;4DSPpage_l)
DOM EXPORT TO FILE($xmlref_t;$infopath_t)
DOM CLOSE XML($xmlref_t)
End if

```

このコードは XML 環境設定ドキュメントが存在しなければそれを作成し、書き込みを行います。ドキュメントパスは **DOM EXPORT TO FILE** コマンドに示されているとおり、\$infopath 変数に収められています。XML ツリーが **DOM Create XML Ref** コマンドで作成され、それぞれの設定が XML ドキュメントのノードに格納されます。以下の設定があります:

- 4DSPlogo\_t: /demo/picture
- 4DSP1\_t: /demo/header
- 4DSP2\_t: /demo/subheader
- 4DSP3\_t: /demo/description
- 4DSP1fg\_l: /demo/headercolor
- 4DSP2fg\_l: /demo/subheadercolor
- 4DSP3fg\_l: /demo/descriptioncolor
- 4DSPbg\_l: /demo/backgroundcolor
- 4DSPpage\_l: /demo/page

XML 環境設定ファイルが既に存在すれば、そのファイルが開かれ、以下のコードが実行されます:

```

$xmlref_t:=DOM Parse XML source($infopath_t)
if (OK=1)
  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/picture")
  if (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/picture")

```

```

End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSPlogo_t)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/header")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/header")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSP1_t)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/subheader")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/subheader")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSP2_t)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/description")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/description")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSP3_t)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/headercolor")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/headercolor")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSP1fg_1)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/subheadercolor")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/subheadercolor")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSP2fg_1)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/descriptioncolor")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/descriptioncolor")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSP3fg_1)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/backgroundcolor")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/backgroundcolor")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSPbg_1)

$xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/page")
If (OK#1)
    $xmlsubref_t:=DOM Create XML element($xmlref_t;"/demo/page")
End if
DOM SET XML ELEMENT VALUE($xmlsubref_t;4DSPpage_1)

DOM EXPORT TO FILE($xmlref_t;$infopath_t)
DOM CLOSE XML($xmlref_t)
End if

```

このコード片では、DOM Parse XML source コマンドで既存ファイルのツリーを解析して、XML ツリーを構築しています。その後、それぞれの XML ノードを DOM Find XML element

コマンドを使用して検索します。ノードが存在すれば値が更新されます。存在しなければ作成され、対応する変数の値が割り当てられます。最後に、XML ツリーは\$infopath パスのファイルに書き出され、新しい設定で XML 環境設定が上書きされます。

## XML 環境設定ファイルの読み込み

以下のコードは 4DSP\_Load\_SplashScreen コンポーネントメソッドです:

```
$xmlref_t:=DOM Parse XML source($infopath_t)
if (OK=1)
  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/picture")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSPlogo_t)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/header")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSP1_t)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/subheader")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSP2_t)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/description")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSP3_t)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/headercolor")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSP1fg_1)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/subheadercolor")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSP2fg_1)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/descriptioncolor")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSP3fg_1)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/backgroundcolor")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSPbg_1)
  End if

  $xmlsubref_t:=DOM Find XML element($xmlref_t;"/demo/page")
  if (OK=1)
    DOM GET XML ELEMENT VALUE($xmlsubref_t;4DSPpage_1)
```

```
End if
DOM CLOSE XML($xmlref_t)
End if
```

このコードは XML 環境設定ファイルが既に存在する場合のみ実行されます。このコードはファイルが存在する場合に XML 環境設定を上書きする動作とよく似ています。違いは、それぞれの値を書き込む代わりに、**DOM GET XML ELEMENT VALUE** コマンドで値を読み込んでいくことです。この方法で、スプラッシュスクリーン表示に使用されるそれぞれの変数が設定されます。ヘッダー、サブヘッダー、説明などのテキスト変数については何も行う必要はありません。対応するフォームオブジェクトには同じ変数名が割り当てられています。ピクチャファイルは Resources フォルダにあり、**Get 4D folder** に **Get Current Resources Folder** 定数を指定したパスから **READ PICTURE FILE** コマンドを使用して読み込みます。フォームオブジェクトに設定されるすべてのカラーは **SET COLOR** コマンドで設定されます。ページの設定は **GOTO PAGE** で処理します。このコードはフォームがロードされるときに、対応するフォームメソッドで実行されます。

## まとめ

---

このテクニカルノートでは 4DSplash コンポーネントを紹介しました。コンポーネントを使用してデータベースにカスタムスプラッシュスクリーンを表示できます。機能はこのテクニカルノートで説明しました。さらにコンポーネントデータベースで紹介されているアイデアを、含まれるソースコードを使用して拡張できます。このテクニカルノートドキュメントでは、スプラッシュスクリーンを表示したり設定を保存したりするコードについても説明しました。