

自動生成されたJSPの確認

■自動生成された表示画面 (JSP) の確認

- (5250DSPF レコード様式名)JSP を確認

■以下のようにプロジェクトを展開

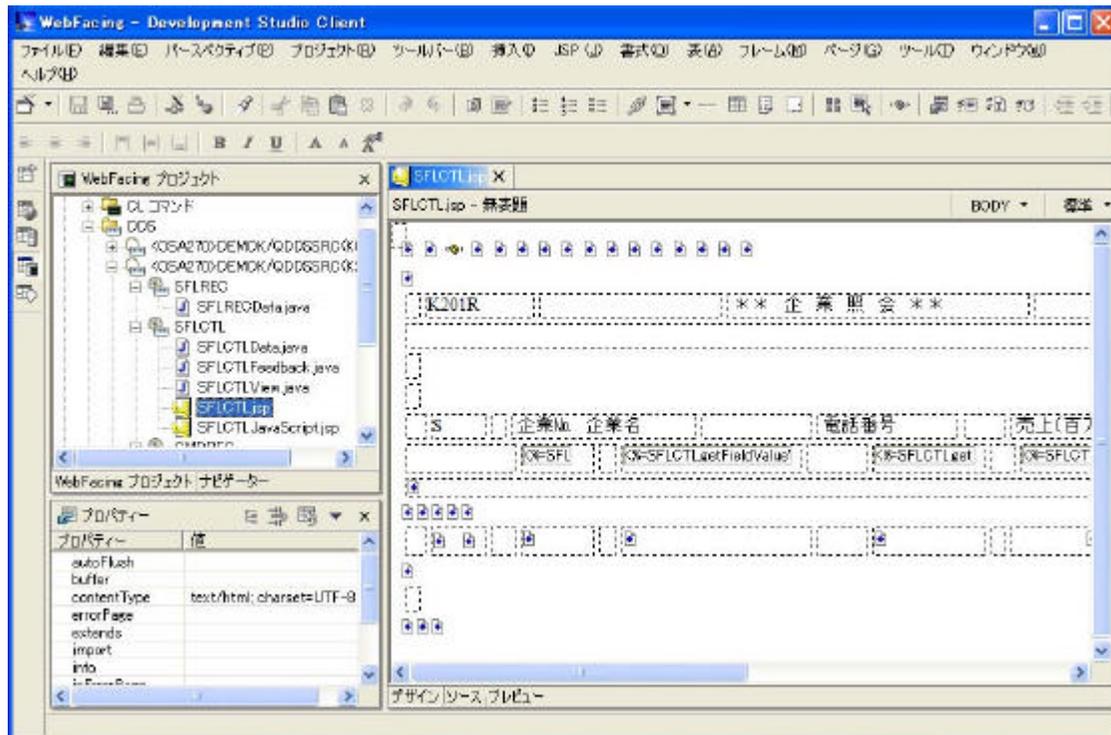
DDS

ソースファイル名
レコード様式名

(レコード様式名)JSP
を右クリック
オープン

■タブ切り替え可能

- デザインタブ
- ソースタブ
- プレビュータブ

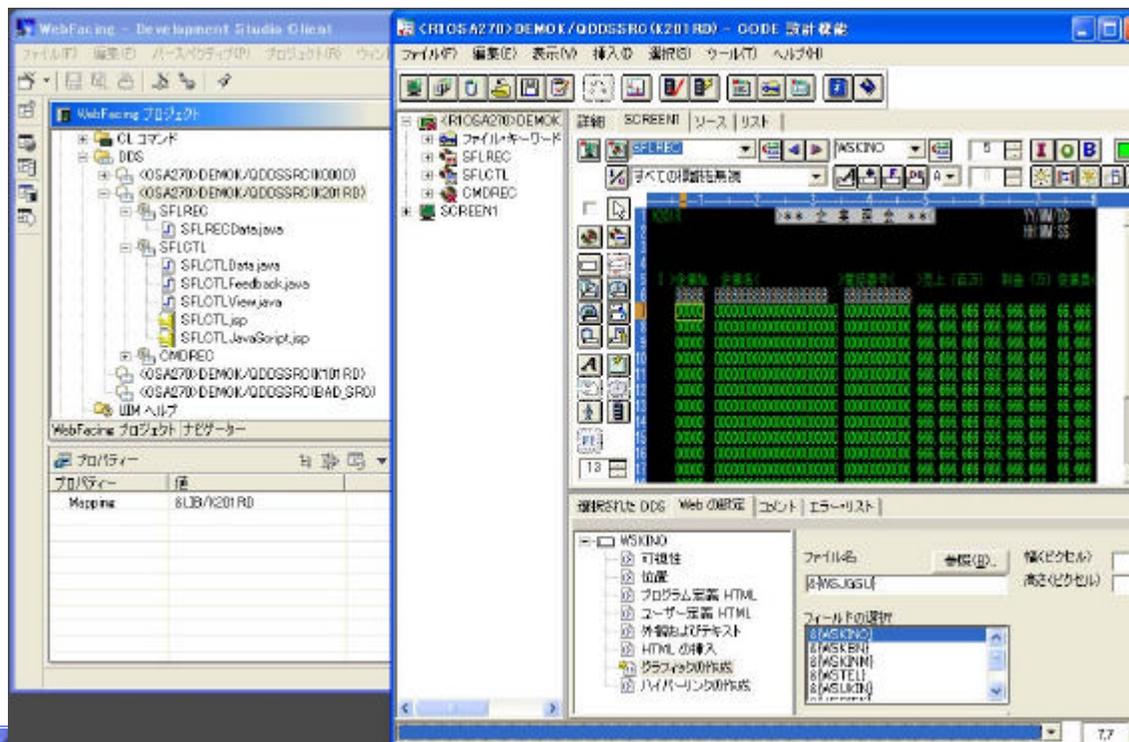


Notes :

1. 自動生成された表示画面 (JSP) のレイアウト等の確認を行う事ができます。
2. 変換で自動生成されたオブジェクト中、(250DSPF レコード様式名)JSP を確認します。
3. 以下のようにプロジェクトを展開します。
DDS ソースファイル名 レコード様式名 (レコード様式名)JSP を右クリック オープン を選択
4. 編集用にページデザイナー (JSPエディター) が起動します。下段のタブ切り替えで表示を選択する事が可能です。
 - ・デザインタブ
 - ・ソースタブ
 - ・プレビュータブ

Notes :

1. 自動生成された表示画面 (JSP) を修正する場合、以下の 2 つのツールを使用する事ができます。
 - CODEエディター : SEUと類似のソースファイルエディターでSeries上のDSPFソースファイルを編集できます。
 - CODE設計機能 : WebFacing専用キーワード(ハイパーリンク、イメージファイル追加、フィールドの非表示等)をiSeries上のDSPFソースメンバーに追加できます。
2. 以下のようにプロジェクトを展開します。
 まず、"WebFacingプロジェクト" タブを開きます (ナビゲータータブを選択していると、以下の操作ができません)
 プロジェクト名 DDS ソースファイル名 右クリック アプリケーションからオープン を選択
 CODEエディター または CODE設計機能 を選択



CODE設計機能の例

(ブランク・ページ)

自動生成される画面関連オブジェクト

■DSPFソースファイルから以下のオブジェクトを生成

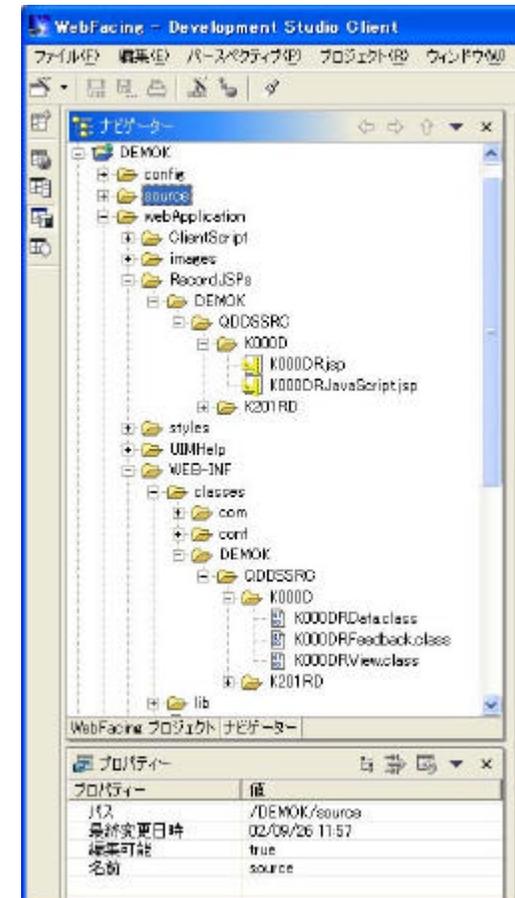
- レコード・フォーマット毎に3つのJavaBeans と2つのJSPを生成

■生成するJavaBeans

- **(DSPFレコード様式名)Data.class**
データBean :JSPへ送るデータを保持
- **(DSPFレコード様式名)View.class**
ビューBean :データ表示用の属性を保持
- **(DSPFレコード様式名)Feedback.class**
フィードバックBean :ブラウザからの
入力データを保持

■生成するJSP

- **(DSPFレコード様式名).jsp**
入出力用ページ
- **(DSPFレコード様式名)Javascript.jsp**
JSPの入力フィールドを制御するJavaScript



Notes :

1. DSPFソースメンバーの変換を実行すると、DSPFソース・メンバーのレコード・フォーマット単位にJavaBeansとJPSが生成されます。

2. 生成するJavaBeans

(DSPFレコード様式名)Data.class : データBean。JSPへ送るデータを保持します。

(DSPFレコード様式名)View.class : ビューBean。データを表示するための属性を保持します。

(DSPFレコード様式名)Feedback.class : フィードバックBean。ブラウザから入力されたデータを保持します。
このデータが5250アプリケーションへ送信されます。

3. 生成するJSP

(DSPFレコード様式名)jsp : 5250のDSPFに替わり入出力を行う表示用画面(ユーザーインターフェース)となります。

(DSPFレコード様式名)Javascript.jsp : JavaScriptです。JSP上の入力フィールドの制御などを行います。

* この他、サーブレットも複数個生成されます。(WFRun.jarファイルに圧縮して生成されます。)

プロジェクトのプロパティ 1.概要

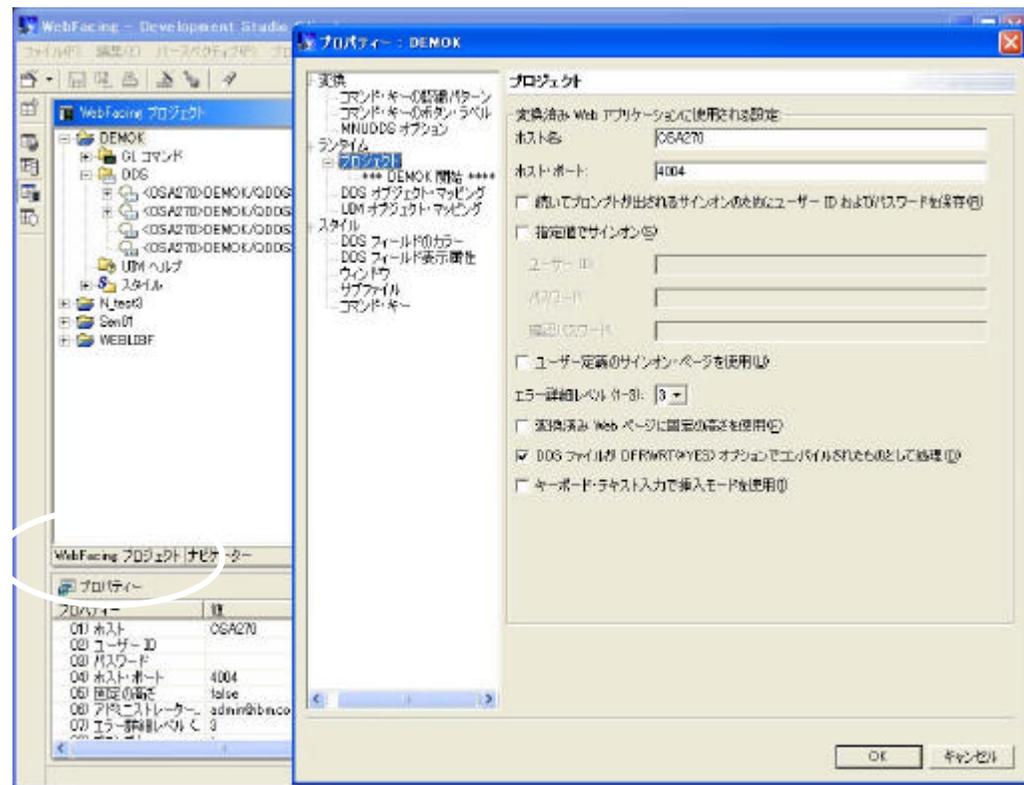
■ プロジェクトのプロパティ

- WebFacingプロジェクトタブ プロジェクト名を右クリック プロパティ

■ 以下の属性を表示 編集可能

- ホスト名
- ユーザーID、パスワード
- ポート番号
- CLコマンドのプロパティ
- CLコマンド単位でホストや
- ユーザーIDを指定
- DDSオブジェクト・マッピング
- WebFacingで変換するソースと
- ホストDSPFのマッピング
- スタイル
- 画面の外観を変更

WebFacingプロジェクト タブを開く



Notes :

1. WebFacingプロジェクトタブを開き、プロジェクト名を右クリック プロパティ を選択すると、各種の設定情報を表示・編集する事が可能です。

以下の属性を表示 編集可能が可能です。

- ・ホスト名 : WebFacing サーバージョブとRPG,COBOL,CLなど5250アプリケーションが実行されるiSeriesの名前
- ・ユーザーID、パスワード : 上記のiSeriesにログオンするためのユーザー名とパスワード
- ・ポート番号 : WASとWebFacing サーバージョブが通信する際使用するポート番号
- ・CLコマンドのプロパティ : WebFacing アプリケーションを開始するためのCLプログラムに関する設定
- ・DDSオブジェクト・マッピング : 5250 DSPFソースファイルのライブラリー、ファイル名のマッピングを個別に変更可能
- ・スタイル : 画面の外観を変更する 等

プロジェクトのプロパティ 2.プロジェクトのプロパティ

■ランタイム プロジェクト を表示

■接続するiSeriesの情報

➤iSeriesの名前

➤ポート番号

➤ログオンの方法

- 指定値で自動ログオン

- logon.htmlより入力

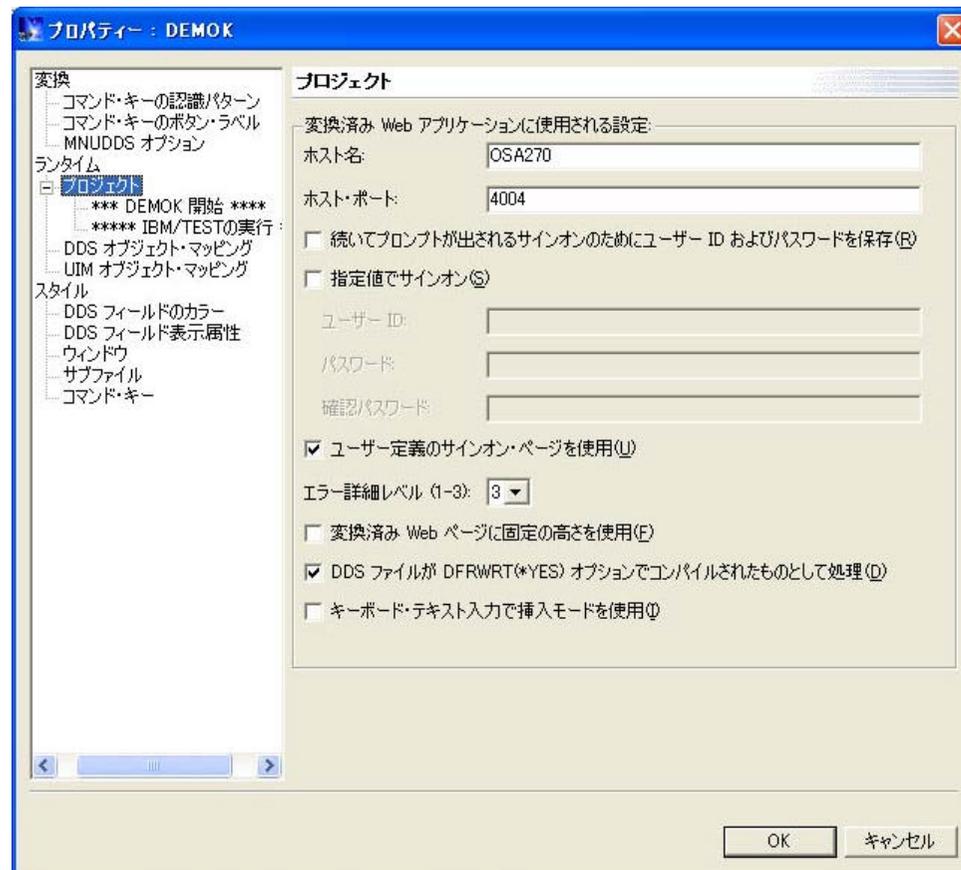
➤カスタマイズした

- logon.htmlの使用

➤JSPへの入力モード指定

- 上書きモード(5250と同様)

- 挿入モード



Notes :

1. WebFacing サーバジョブ、5250アプリケーションを実行するiSeriesの設定はランタイム プロジェクトから表示します。設定値のデフォルトは5250 DSPFソースファイルを取り込んだときの指定値が設定されています。開発環境と実行環境が異なる場合などは、適切な値に設定を変更します。
2. ホスト名 : WebFacingサーバジョブ、5250アプリケーションが実行されるiSeriesの名前を入力します。
3. ポート番号 : WebFacingサーバジョブの使用するポート番号です。デフォルトのポート4004を変更する場合には、ここでの指定の変更を変更する他、iSeries上でWRKSRVTBLEコマンドなどを使用してサービス as-WebFacingを削除し、新しい指定のポート番号でas-WebFacingを作成します。
- 3.指定値でサイン・オン : チェックを入れるとindex.html上のリンクをクリックするだけでWebFacingアプリケーションを実行可能になります。この際iSeries上では以下のユーザー、パスワード欄に指定したID/パスワードが使用されます。チェックを入れない場合は、index.htmlのリンクをクリックするとlogon.html画面が表示され、ここでiSeriesにログオンするためのID/パスワードを入力できます。
4. ユーザー定義のサインオン ページの使用 : カスタマイズしたlogon.htmlを使用したい場合にチェックをつけます。
5. キーボードテキスト入力で挿入モードを使用 : WDS Sc V4.0 SP2以降ではJSP入力フィールドに対する、デフォルトの入力モードは5250画面と同様な上書き入力モードになっています。特に既に値入力済みのフィールドに対する入力効率を向上させていますが、SP1以前と同じような挿入(インサート)モードでの入力にしたい場合は、ここをチェックします。

Notes :

6. 前ページまでのプロジェクトのプロパティ画面での定義は該当のindex.htmlから呼び出されるすべてのプロジェクトのデフォルト値として使用されます。プロジェクト単位で別な値を指定したい場合、ランタイム プロジェクト (CLコマンドラベル)を選択して設定を変更する事ができます。

具体的には、変更したいコマンドラベルをクリックして下の画面を表示します。

”このコマンドのプロジェクト設定の指定を変更”にチェックします。
必要な値について設定を変更します。

右の図ではデフォルトのユーザーIDを指定し、
ログイン画面をバイパスしてWebアプリケーション
を開始するように設定しています。



Notes :

7. CLコマンドを追加する際、プログラム呼び出し用にパラメーターを指定する事も可能です。
例えば CALL DEMOK/K000C PARM('&PARM1')と指定すると、index.html上にパラメーターを入力するためのフィールドが自動的に追加、表示されます。

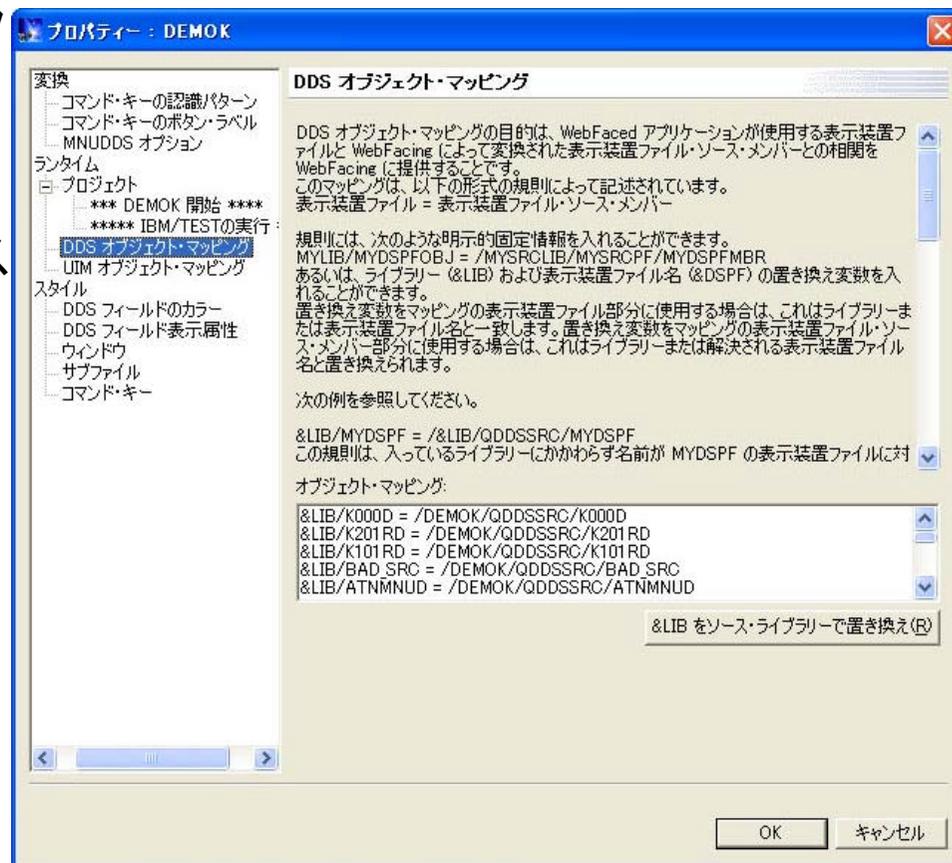


プロジェクトのプロパティ 3.DDSオブジェクトマッピング

■変換に使用するDSPFソースファイルの設定を詳細に指定可能。

■WebFacing プロジェクトタブを選択
プロジェクト名を選択、右クリック
DDSオブジェクトマッピング
を選択

■DSPFオブジェクト・変換対象ソース
ファイルのマッピング情報を
確認・変更



Notes :

1. WebFacingプロジェクトのウィンドウで WebFacingプロジェクト・タブを選択します。次にプロジェクト名を選択して、右クリック、メニューからプロパティを選択します。

DDSオブジェクト・マッピングを選択すると、DSPFオブジェクトと変換対象ソースファイルとのマッピング情報を確認することができます。

2. このフィールドの値はWebFacing ToolでDSPFソースファイルをコンバートする際の、DSPFオブジェクトとソースファイルのマッピングを示しています。

デフォルトは &LIB/ (DSPFオブジェクト名) = / (ソースファイルLIB名) (ソースファイル名) (メンバー名) と指定されます。前ページの画面例では DSPFオブジェクトK000Dがどのライブラリーに存在するかに関わらず DEMOK/QDDSSRC/K000D のソースファイルがコンバートに使用されます。

3. 必要に応じてDSPFオブジェクト名とソースファイルのマッピング情報を変更することができます。("オブジェクト・マッピング" 欄に値を直接入力して、マッピング情報を個別に変更することが可能です。)

例えば、同一オブジェクト名で、ライブラリーの異なるDSPFオブジェクトがある場合を考えてみます。以下の例はDSPFオブジェクトK300Dがライブラリー DEMOK とDEMOK2 に存在する場合です。まずWebFacingウィザードで DEMOK/QDDSSRC/K300D とDEMOK2/QDDSSRC/K300Dの2つのDSPFソースファイルを変換対象ソースファイルに追加して、以下の作業を行います。

DSPFオブジェクトDEMOK / K300D はDEMOK/QDDSSRC/K300D ソースファイルからコンパイルされており

DSPFオブジェクトDEMOK2 / K300D はDEMOK2/QDDSSRC/K300D ソースファイルからコンパイルされて

いる場合は次のように入力します。

DEMOK/K300D = /DEMOK/QDDSSRC/K300D

DEMOK2/K300D = /DEMOK2/QDDSSRC/K300D

または以下のように指定することも可能です。

&LIB/K300D = &LIB/QDDSSRC/K300D

以上をオブジェクト・マッピング フィールドに入力後、

'OK' ボタンを押します。



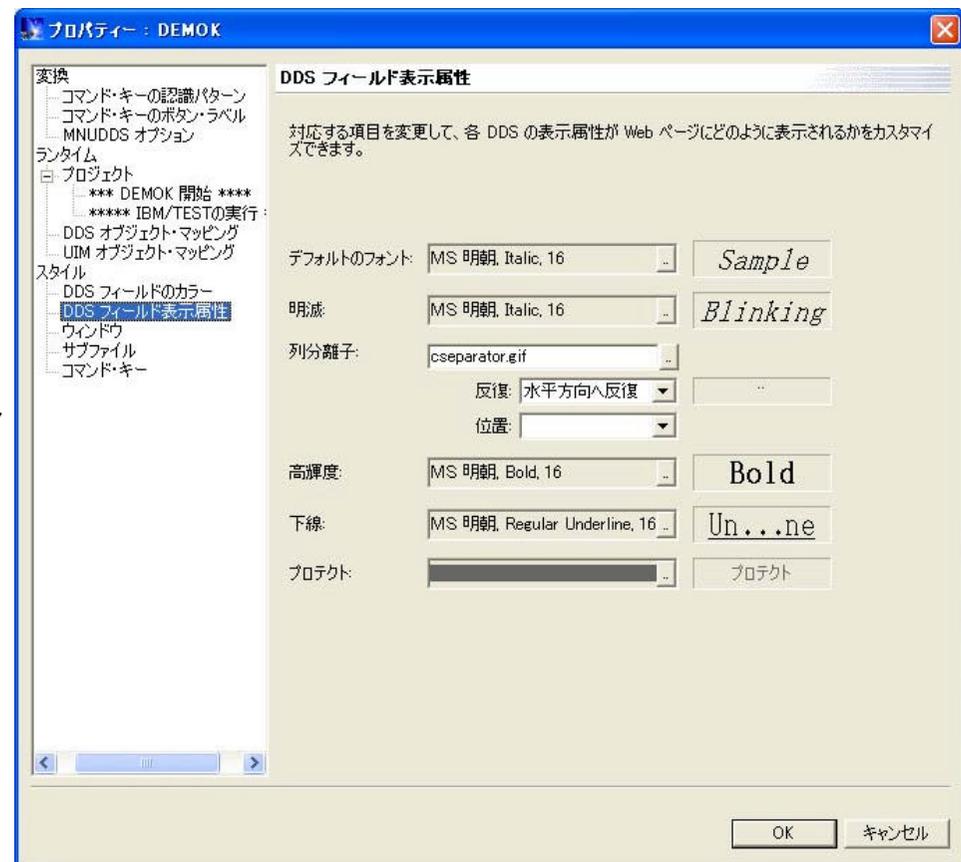
プロジェクトのプロパティ 4.スタイル

■変換時に使用するJSPの表示属性を指定

- フィールドの色
- フォント
- DDSウィンドウの表示属性
- サブファイル画面の表示属性
- ファンクションキーの表示属性

■背景用などに指定されている
GIFファイルを取り外すと
パフォーマンスが向上する場合
もあり。

■設定を変更した場合は
スタイルの保管を実行



Notes :

- 変換時に使用するJSPの表示属性を指定します。指定可能な属性には以下があります。
 - ・フィールドの色 : 5250上のDDSフィールドの色別(デフォルト、緑、白、青、等)に前景色、背景色を指定可能
 - ・フォント : JSPで使用するフォントを指定可能。
 - ・DDSウィンドウの表示属性 : DDSウィンドウを持つDSPFソースファイルを変換した際の属性を指定
 - ・サブファイル画面の表示属性 : サブファイルレコードを変換した際の属性を指定
 - ・ファンクションキーの表示属性 : ファンクションキーを変換した際の属性を指定
- デフォルトでは背景用などにGIFファイルが指定されていますが、これを取り外すとパフォーマンスが向上する場合があります。(JSPの伝送量が低減するため)

- 設定を変更した場合はスタイルの保管を実行します。

プロパティ画面で表示属性を変更

OK ボタンで閉じる

WebFacingパースペクティブのプロジェクト名を

右クリック

スタイル

別名保管 を選択し、名前を指定して保管

- スタイルを変更する場合、WebFacingパースペクティブ

からプロジェクト名を右クリックして

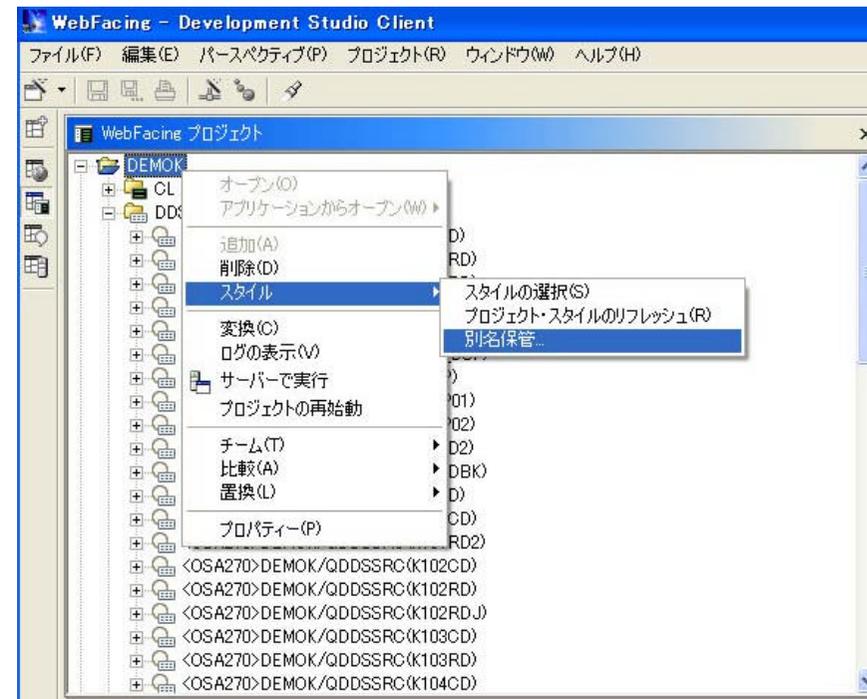
スタイル

スタイルの選択 から該当のスタイル名を選択

再度プロジェクト名を右クリック

スタイル

スタイルのリフレッシュ を実行



プロジェクトのプロパティ 5.コマンド・キーの認識パターン

■自動生成するファンクション・キーのボタン文字列をカスタマイズ可能

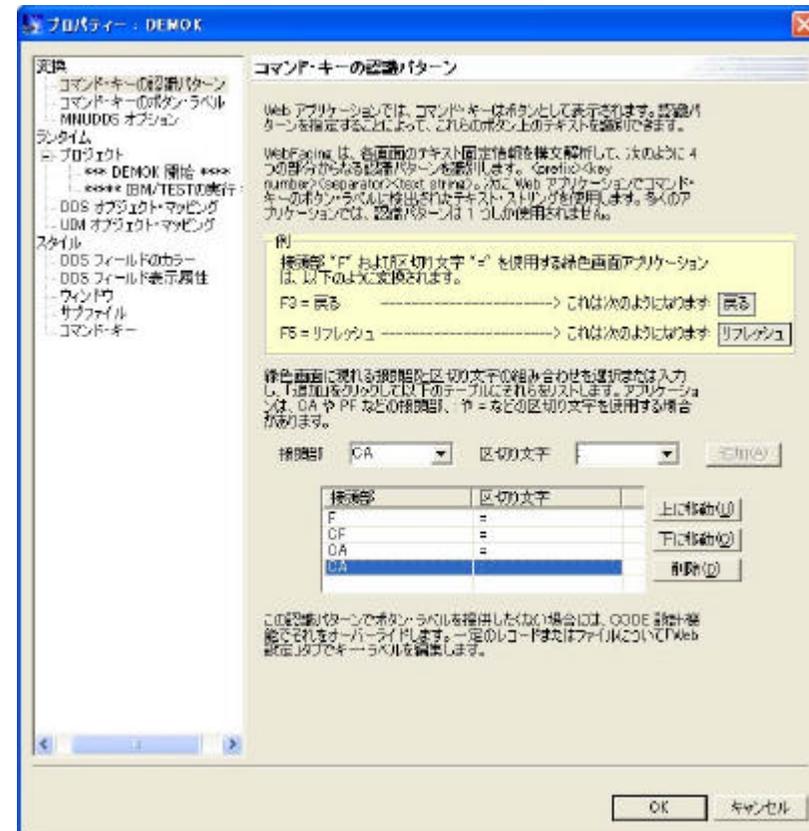
- DSPFソースファイル上の文字列を検索してボタンに表示する。
- 認識する文字列パターンのカスタマイズが可能

■以下の文字列を識別してボタンのラベルに自動変換

- 5250画面上に表示される文字列
(例) F3 = 終了 '終了'をボタンに表示
- 5250DSPFソースファイル上のCA定義
(例) CA07(07 '前頁') '前頁'をボタンに表示

■カスタマイズ例

- 接頭語 : CMD
- セパレーター : =
- CMD03= 終了 とある場合、'終了'をボタン上に表示



Notes :

1. WebFacing Toolでは5250DSPFソースファイル上に定義されているファンクション・キー定義を自動的に判別してファンクションキーボタンを自動生成します。この際、ボタンに表示する文字列も5250DSPFソースファイル上から自動的に取得します。

2. デフォルトでは以下のような文字列を5250DSPFソースファイル上で検索して、合致するものがあれば、ボタンの文字列に使用します。

(例) F01 = ヘルプ
CF03 = 終了
CA12 = 前画面

文字列は接頭部 (上例ではFxx, CFxx, CAxx)とセパレーター (≡)の組み合わせでマッチングが行なわれ、該当する文字列がある場合、セパレーター以降がボタンの文字列に表示されます。

また、パターンが一致した場合、5250DSPFソースファイル上に定義されている該当の文字列はJSP化する際に取り除かれ表示されなくなります。

3. この接頭部とセパレーターは前ページの "コマンド・キーの認識パターン"画面でカスタマイズして、自社で使用しているDSPFソースファイルの文字列にあわせカスタマイズが可能です。

例えば、自社の5250画面ファイル上での表記ルールが CMD03 :終了 となっている場合、接頭部にCMD を、セパレーターに : を追加します。これにより自動的にファンクション03のボタンの文字列に "終了"が設定されます。

4. もし、上記接頭部とセパレーターの組み合わせで合致しないばあい、WebFacing Toolは5250DSPFソースファイル上の Caxx(xx 'YYYYY')という設定を検索し、ファンクションボタンxxの文字列にYYYYを設定します。

5. 前ページの例では 接頭部としてCA、セパレーターとして : を認識するよう設定を追加しています。

Notes :

6. 例として以下のような5250DSPFソースファイルを取り上げます。

```
-----  
A      R SFLCTL          TEXT(' サブファイルCTL ')  
A      SFLCTL(SFLREC)  
A      OVERLAY  
A      CA03(03)  
A      CF07(07 ' 前頁 ')  
A      CF08(08 ' 後頁 ')  
A      CF09(09)  
A      CA12(12)  
      :                  :  
      :                  :  
A      R CMDREC         TEXT(' 明細画面 ')  
A      OVERLAY  
A      22 2 'F3= 終了 '  
A      22 12 'F7 前ペ - ジ'  
A      22 26 'F8= 後ページ '  
A      22 37 'F9 追加 '  
A      22 46 ' F12 前画面 '  
-----
```

Notes :

1. WebFacing Toolにおける、コマンドキーのデフォルトの変換ロジックは下記のようになります。

5250DSPFソースファイル		JSP画面		理由
5250画面上の文字列 (変換時の優先度1)	ソースファイル上のCaxx の指定 (変換時の優先度2)	ボタンの文字列	JSP上の文字列	
F3 = 終了	CA03(03)	終了	(無し)	優先度1の指定よりボタン文字列設定。およびJSP上より文字列の除去。
F7 前ページ	CA07(07 ' 前頁 ')	前頁	F7 前ページ	優先度1はセパレーターが無い ため不一致、優先度2よりボタン文字列を設定。JSP上の文字列は残る。
F8= 後ページ	CA98(08 ' 後頁 ')	後ページ	(無し)	優先度 1の指定が優先される。 (優先度2の指定は無視される。)JSP上の文字列も除去される。

1). まず、5250画面に表示される文字列を検索し、F1=XX、CF03=XX、CA12=XX のような文字列を検索します。いずれかの文字列が発見された場合、XXの部分で生成するボタンの文字列として使用します。さらに変換するJSP画面からはその文字列全体を削除します。(F3キー)

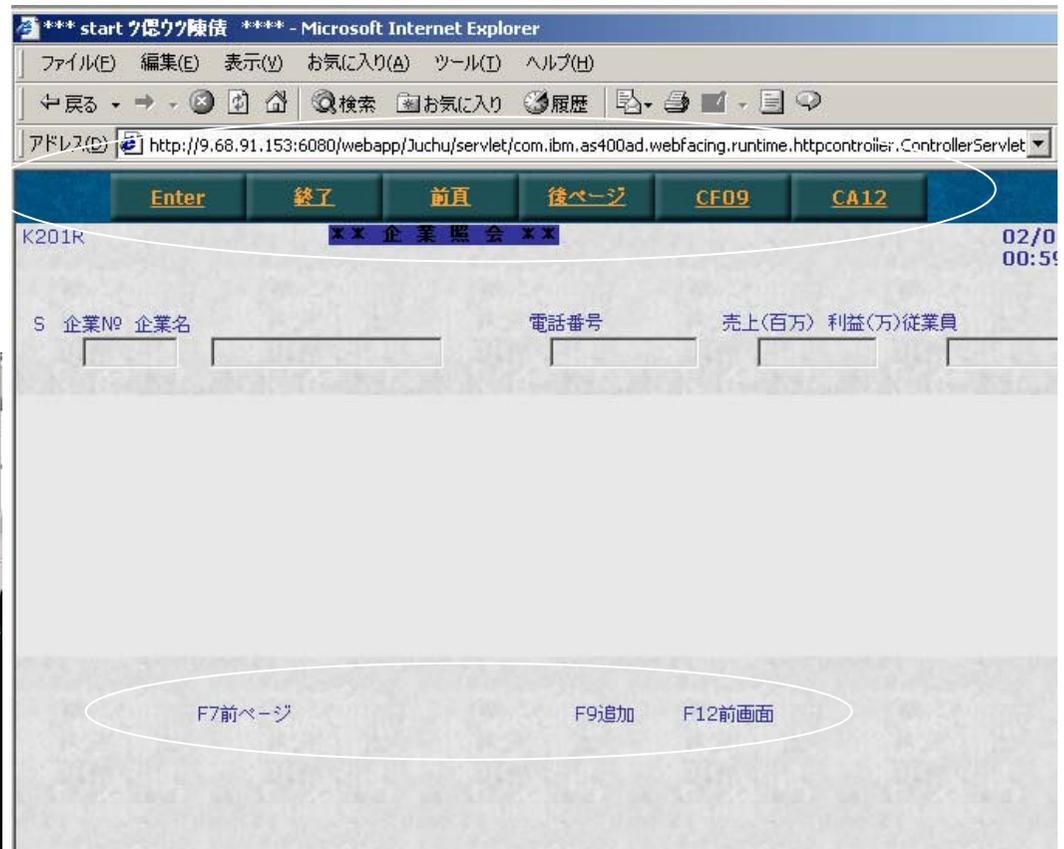
2). 1.で定義した文字列が見つからない場合は、DSPFのソースファイル上でCA_{xx}(_{xx} 'YY') という定義があるか参照します。'YY'に相当する文字列が定義されている場合、この文字列をボタンの文字列として採用します。(F7キー)JSP上には文字列検索ではボタンの文字列へ置換されなかったため、F7 前ページという文字列が残ります。

3). 1.と2.の両方が見つかった場合は1.(5250画面に表示される文字列)を優先します。(F8キー)JSP上はF8キーに関する文字列は削除されます。

Notes :

前ページまでの設定に従って変換した場合の例です。

WebFacing Toolで変換されたJSP



5250画面

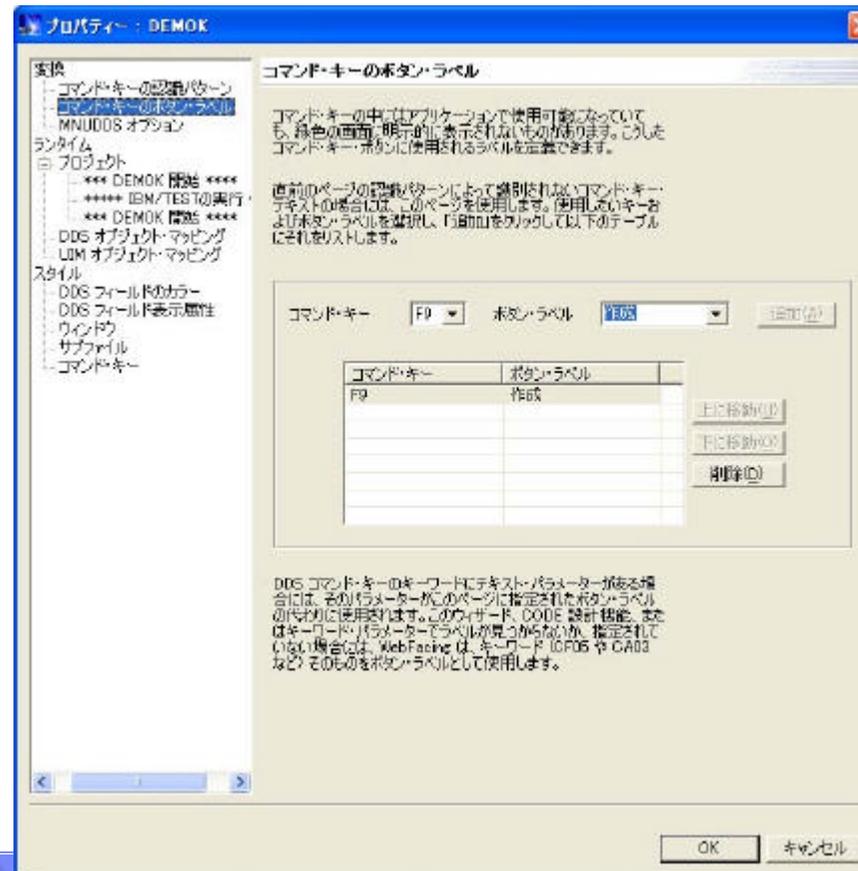


Notes :

7. 前ページまでのコマンド・キーの認識パターンで認識されない場合 (例えばCA03(03 '終了')だけが指定されている) ような場合のデフォルトの変換ルールを設定する事もできます。
これは "コマンド・キーのボタン・ラベル" の画面から指定を行います。

以下の例ではファンクション・キー09に対して "作成" という文字列を割り当てています。

* このページの設定より、前ページまでの "コマンド・キーの認識パターン" で設定した値が優先されます。



開発手順 4.WASへのエクスポート1

■WebFacing Toolで生成したアプリケーションをWASへエクスポート

■WAR ファイル形式でエクスポート
●WAS V3.5, WAS V4.0とも共通

■WebFacingパースペクティブから
ファイル エクスポートを選択

”次へ”ボタンを押す。



Notes :

1. WebFacing Toolでの作業が完了したら、生成したアプリケーションをプロジェクト単位でWASへエクスポートします。
2. WDSからWAS V3.5, WAS V4.0のいずれの場合でもWAR ファイル形式でエクスポートする事をお勧めいたします。
(WebFacing Tool First Editionの場合、WAS V3.5に対するエクスポートはファイルシステム形式でした。)
3. エクスポートを実行するにはWebFacingパースペクティブから以下の操作を実行します。
ファイル エクスポートを選択し、 ”次へ”ボタンを押します。

開発手順 4.WASへのエクスポート2

■エクスポート情報の指定

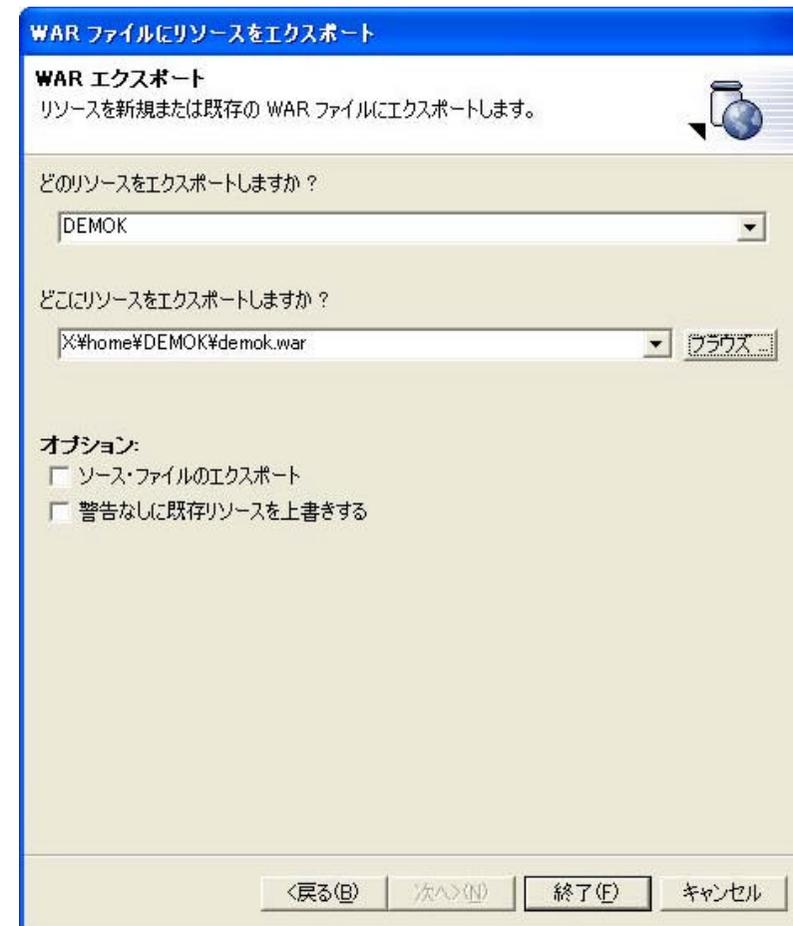
▶リソース

エクスポートするプロジェクト名

▶エクスポート先

エクスポートするIFSパス

”終了”ボタンを押すとエクスポート
される。



Notes :

- 次にエクスポート情報を指定します。リソース欄にはエクスポートするプロジェクト名を指定します。また、エクスポート先にはエクスポートするIFSパスを指定します。”終了”ボタンを押すとエクスポートが開始されます。
- 事前にiSeries上でネットサーバーを開始しておき、エクスポートを実行するPCからネットサーバーの共用ディスクを割り当てておきます。前頁の例ではiSeriesの / (ルート)をXドライブとして事前に割り当てています。結果、iSeries上の /home/DEMOKディレクトリにDEMOK.warをエクスポートする事になります。

```

A - osa270 telnet
ファイル (F) 編集 (E) 配置 (V) 通信 (C) 選択 (A) ウインドウ (W) ヘルプ (H)
オブジェクト・リンクの処理
ディレクトリー . . . : /home/DEMOK
オプションを入力して、実行キーを押してください。
  2= 編集  3= コピー  4= 除去  5= 表示  7= 名前の変更  8= 属性の表示
 11= 現行ディレクトリの変更 ...
OPT  オブジェクト・リンク      タイプ      属性      テキスト
---  ---
..      ..      DIR
demok.war      STMF
終わり
パラメーターまたはコマンド
==>
F3= 終了  F4= プロンプト  F5= 最新表示  F9=コマンドの複写  F12= 取り消し
F17= 位置指定  F22=フィル全体の表示  F23=オプション 続き
Mw 英数 半角 A 12/002
ホスト 23 を使用してリモート・サーバー/ホスト 9.188.153.7 に接続しました

```

iSeries にエクスポートされた
warファイル

/home/DEMOK/demok.war

開発手順 5.WASにWebアプリケーション登録1

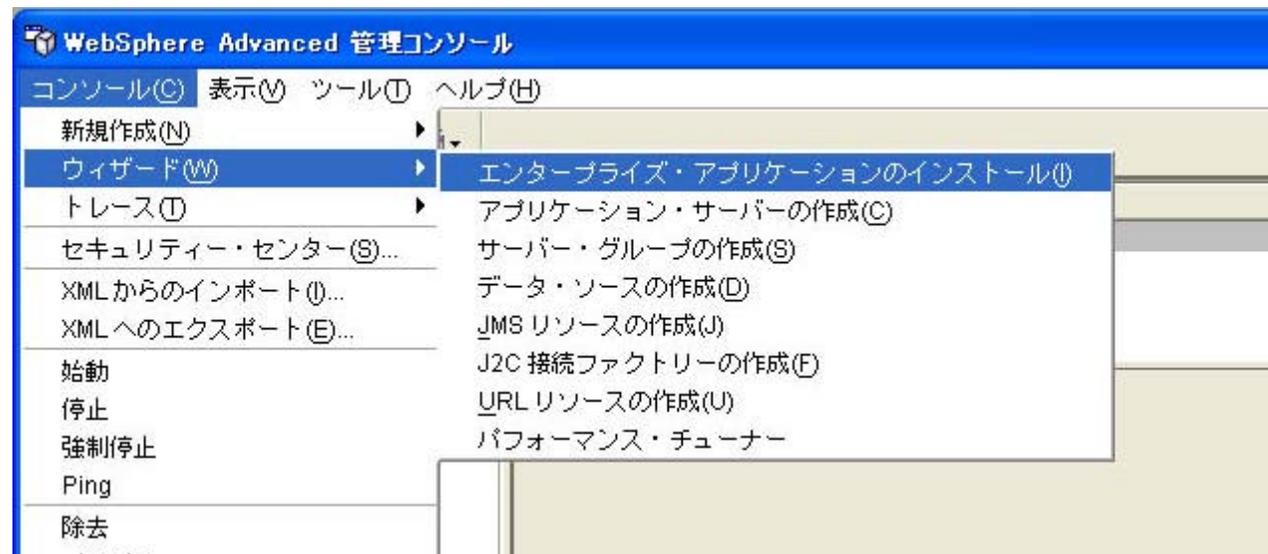
■エクスポートしたwarファイルをWebアプリケーションとしてWASに登録

■WASを事前に開始しておく

- (例) STRSBS QEJBADV4/QEJBADV4

■WASコンソールを起動し、メニューから以下を実行

コンソール ウィザード エンタープライズ・アプリケーションのインストール



Notes :

1. エクスポートしたWebアプリケーション (warファイル)はそのままではWASから使用する事ができません。エクスポートしたWebアプリケーションを実行可能にするには、WASにWebアプリケーションとして登録する必要があります。

2. 手順はまず、WASを事前に開始しておきます。5250画面から実行する場合はSTRSBS コマンドでWASのサブシステムを開始します。

(例)WAS V4.0アドバンス版の場合 STRSBS QEJBADV4/QEJBADV4

3. 次にPC上でWAS管理コンソールを起動します。

メニューから以下を実行します。

コンソール ウィザード エンタープライズ・アプリケーションのインストール

開発手順 5.WASにWebアプリケーション登録2

■以下を指定

➤インストールするスタンドアロン・モジュールを選択

➤パス : エクスポートした*.warファイルを指定

(例) /home/DEMOK/demok.war

➤アプリケーション名 : Webアプリケーションの名前 (任意)を指定

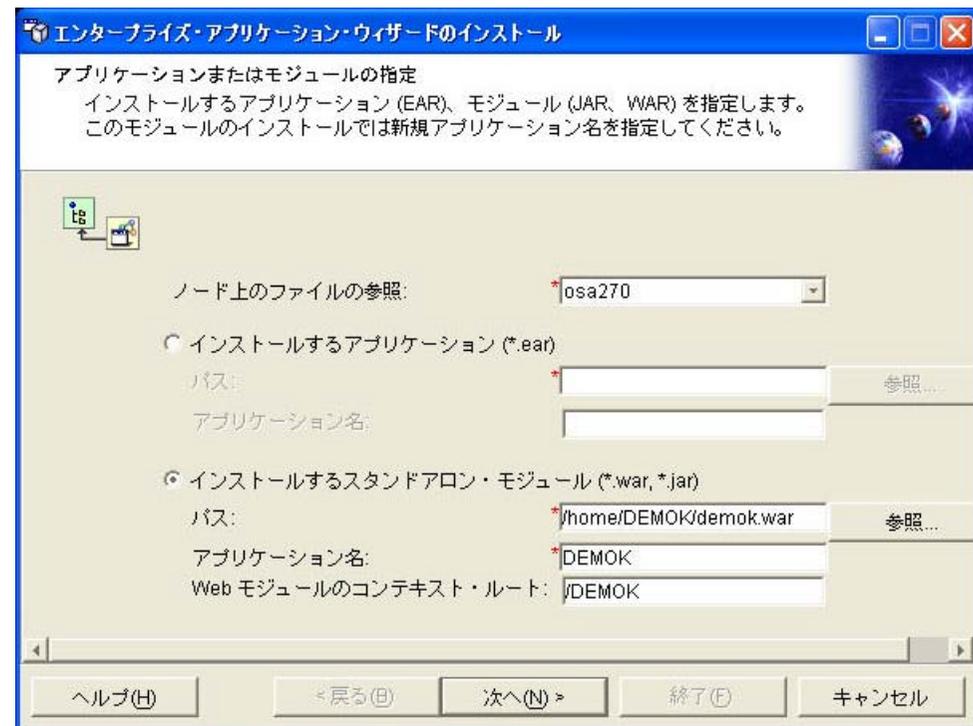
(例) DEMOK

➤Webモジュールのコンテキスト・ルート :

ブラウザからアクセスする際のサーバー名以降のURLを指定

(例) /DEMOK

”次へ” ボタンを押す



Notes :

1. 以下を入力します。
2. ノード上のファイル参照 :
Webアプリケーションを登録するWAS ノード名を指定します。通常CFGTCP オプション12で指定したiSeriesホスト名と同じ名前になります。
3. インストールするスタンドアロン モジュール :
WDSからエクスポートした*.warファイルのパス名、WARファイル名を指定します。'参照'ボタンを押すとiSeries上のディレクトリパスを検索して表示させる事ができます。
例では /home/DEMOK/demok.war を指定しています。
4. アプリケーション名 :
このWebアプリケーションの名前を指定します。任意の名前を指定します。WAS管理コンソール上で表示されるWebアプリケーション名になります。
5. Webモジュールのコンテキスト・ルート
ブラウザからアクセスする際のサーバー名以降のURLを指定として使用されます。前ページの例ではブラウザから初期ページ(index.html)を表示する場合、以下のように指定する事になります。
http:// (ホスト名) : (ポート番号) /DEMOK/index.html
* 太字の/DEMOK がWebモジュールのコンテキスト・ルートと一致します。

すべてを入力したら '次へ' ボタンを押します。

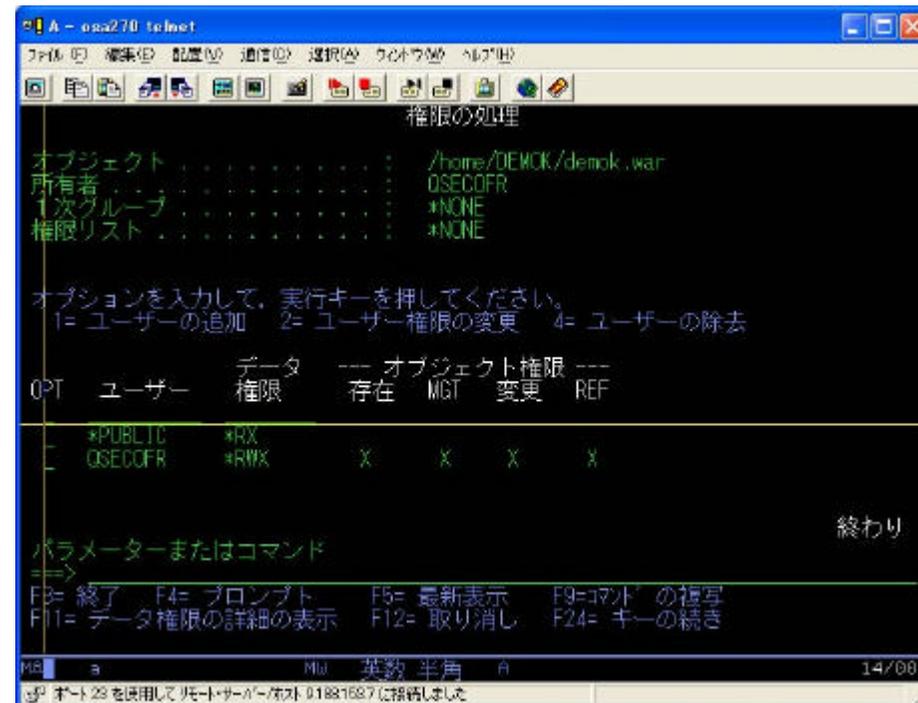
Notes :

6. もし、下図のように "ファイルのロードに失敗しました" というエラーが出る場合は、エクスポートしたwarファイルのアクセス権限をチェックしてください。エクスポートしたwarファイルをWASにエクスポートするには*PUBLIC (またはQEJB, QEJBSVR)ユーザープロフィールに対する *RX権限が対象ファイルに付与されていなければなりません。

7. ファイルに対するアクセス権限を確認するには WRKLNK コマンド等で該当のIFS上のwarファイルを表示し、オプションに "9 = 権限の処理" を選択するか、CHGAUT(WRKAUT)コマンド等で*PUBLIC(またはQEJB, QEJBSVR)に対して *RX権限を付与してください。



*PUBLICに*RX権限がない場合のエラー例



WRKAUTコマンドで*PUBLICに*RX権限を付与

(ブランク・ページ)

開発手順 5.WASにWebアプリケーション登録3

■以下のページは何も変更せずに "次へ" ボタンを押す。

- ユーザーへの役割のマッピング ' 画面 次へ' ボタンを押す
- EJB RunAs役割のユーザーへのマッピング ' 画面 次へ' ボタンを押す
- JNDI名へのEnterprise Beanのバインド ' 画面 次へ' ボタンを押す
- Enterprise Beanのバインド ' 画面 次へ' ボタンを押す
- Enterprise BeanへのEJB参照のマッピング ' 画面 次へ' ボタンを押す
- リソース参照のリソースへのマッピング ' 画面 次へ' ボタンを押す
- EJBモジュールのデフォルトデータソースの作成 ' 画面 次へ' ボタンを押す
- 個々のCMP Beanのデータ・ソースの指定 ' 画面 次へ' ボタンを押す
- ' 個々のCMP Beanのデータ・ソースの指定 ' 画面 次へ' ボタンを押す

Notes :

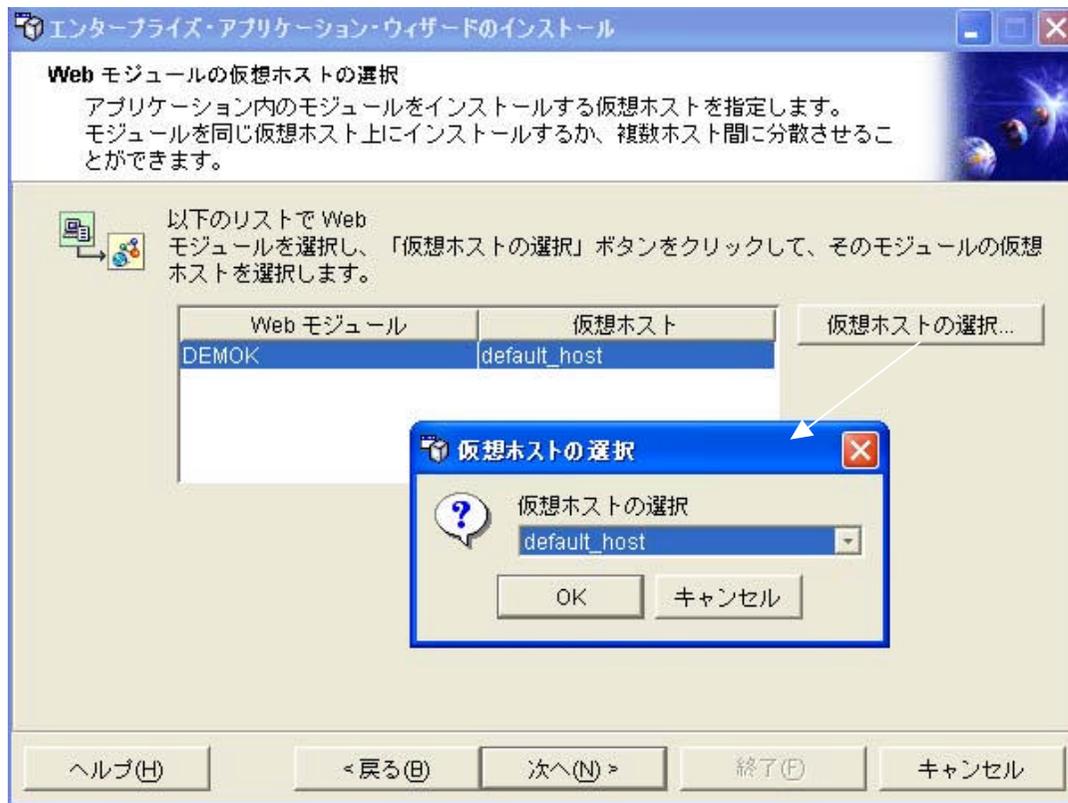
1.以下のページは設定変更をする必要が無いので、何も変更せずに”次へ”ボタンを押します。

- ・ ユーザーへの役割のマッピング ’ 画面
- ・ EJB RunAs役割のユーザーへのマッピング ’ 画面
- ・ JNDI名へのEnterprise Beanのバインド ’ 画面
- ・ Enterprise Beanのバインド ’
- ・ Enterprise BeanへのEJB参照のマッピング ’ 画面
- ・ リソース参照のリソースへのマッピング ’ 画面
- ・ EJBモジュールのデフォルトデータソースの作成 ’ 画面
- ・ 個々のCMP Beanのデータ・ソースの指定 ’ 画面
- ・ 個々のCMP Beanのデータ・ソースの指定 ’画面

開発手順 5.WASにWebアプリケーション登録4

- 'Webモジュールの仮想ホストの選択' 画面
 - '仮想ホストの選択' ボタンを押して仮想ホストを選択
 - デフォルトでは default_host を選択

'次へ' ボタンを押す



Notes :

1. ' Webモジュールの仮想ホストの選択 ' 画面ではWebアプリケーションをインストールする仮想ホストを指定します。
2. デフォルトでは仮想ホストはdefault_hostですが、ユーザーが仮想ホストを追加している場合には'仮想ホストの選択' ボタンを押して仮想ホストを選択します。
3. デフォルトでは、 default_host を選択します。
4. ' 次へ ' ボタンを押します。

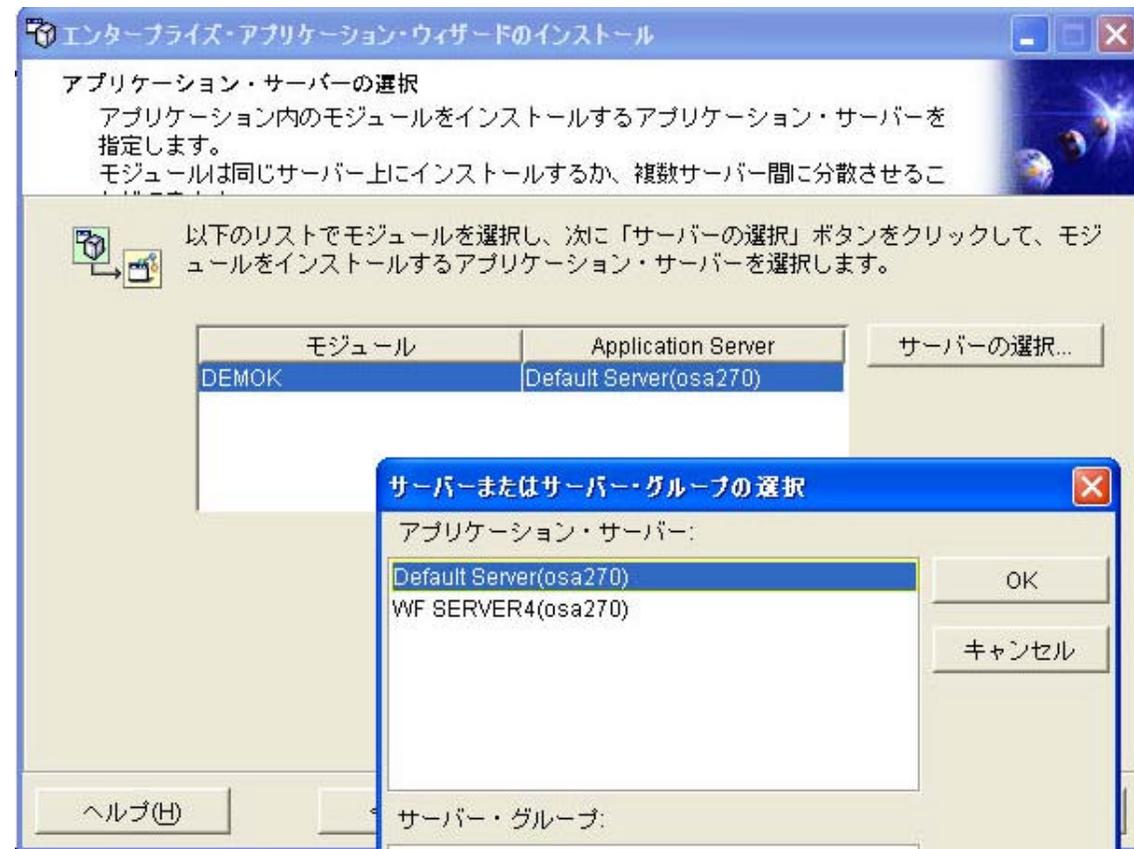
開発手順 5.WASにWebアプリケーション登録5

■ 'アプリケーション・サーバーの選択' 画面

● 'サーバーの選択'ボタンを押してアプリケーションサーバーを選択

➤ デフォルトでは Default Serverを選択

'次へ' ボタンを押す



Notes :

1. 'アプリケーション・サーバーの選択' 画面ではWebアプリケーションを登録するアプリケーションサーバーを選択します。
2. デフォルトではアプリケーションサーバーはひとつ (Default Server)のみですが、ユーザーがアプリケーションサーバーを追加している場合は、'サーバーの選択' ボタンを押してWebアプリケーションを登録するアプリケーションサーバーを選択します。
3. デフォルトでは Default Serverを選択します。
4. '次へ' ボタンを押します。

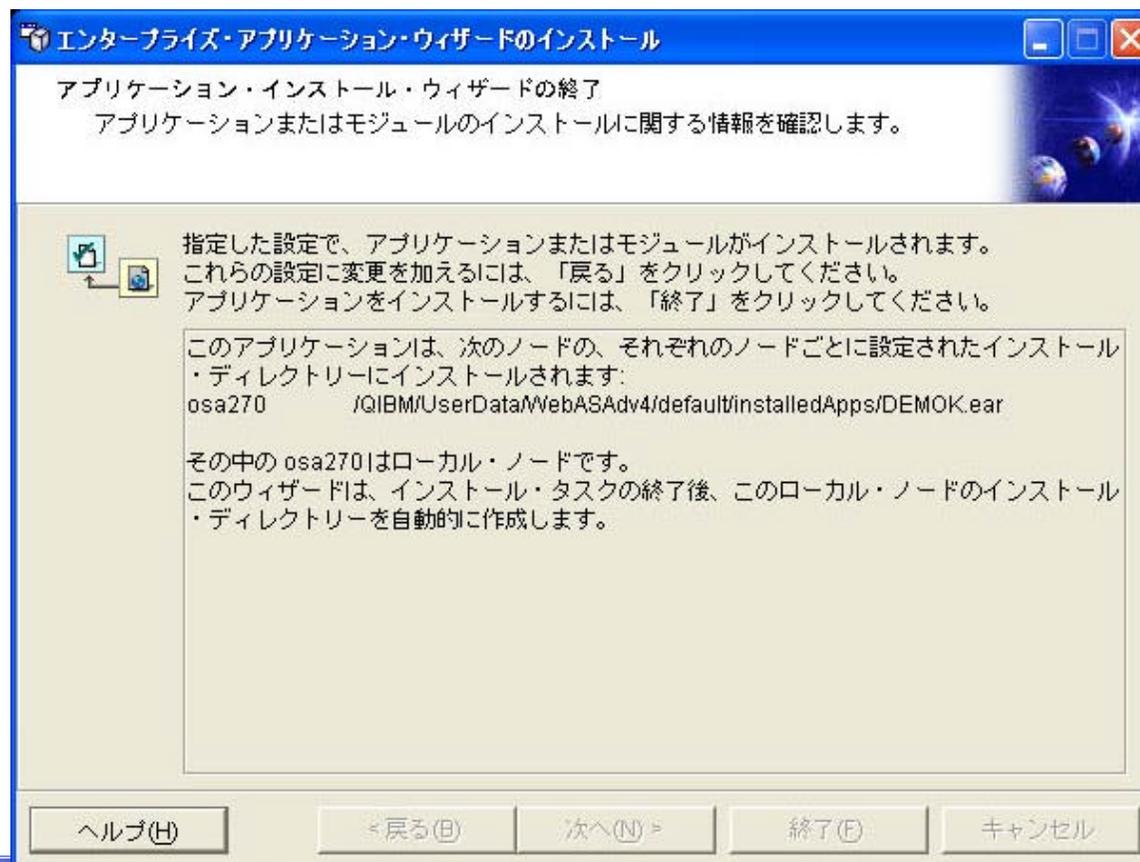
開発手順 5.WASにWebアプリケーション登録6

■' アプリケーションインストールウィザードの終了 ' 画面

●warファイルから展開される、実行モジュール配置先パスが通知

➤例では /QIBM/UserData/WebASAdv4/default/installedApps/DEMOK.ear 以下

' 終了' ボタンを押す



Notes :

1. ' アプリケーションインストールウィザードの終了 ' 画面が表示されます。
WAS V4.0上で実行されるWebアプリケーションは*.warファイルから展開され、実行用パスに配置されます。この画面で実際の実行モジュールの配置先パスが通知されます。

2. 実行されるWebアプリケーション用ファイルは以下のパスに配置されます。

WAS V4.0アドバンスド版の場合

/QIBM/UserData/WebASAdv4/default/installedApps/DEMOK.ear 以下

WAS V4.0 アドバンスド・シングル・サーバー版の場合

/QIBM/UserData/WebASAEs4/default/installedApps/DEMOK.ear 以下

* /DEMOK.ear は 「開発手順 5.WASにWebアプリケーション登録2」のページのアプリケーション名で指定した (アプリケーション名).ear というディレクトリ以下に配置されます。

3. ' 終了 ' ボタンを押します。

以上でWebアプリケーションのインストールが完了します。