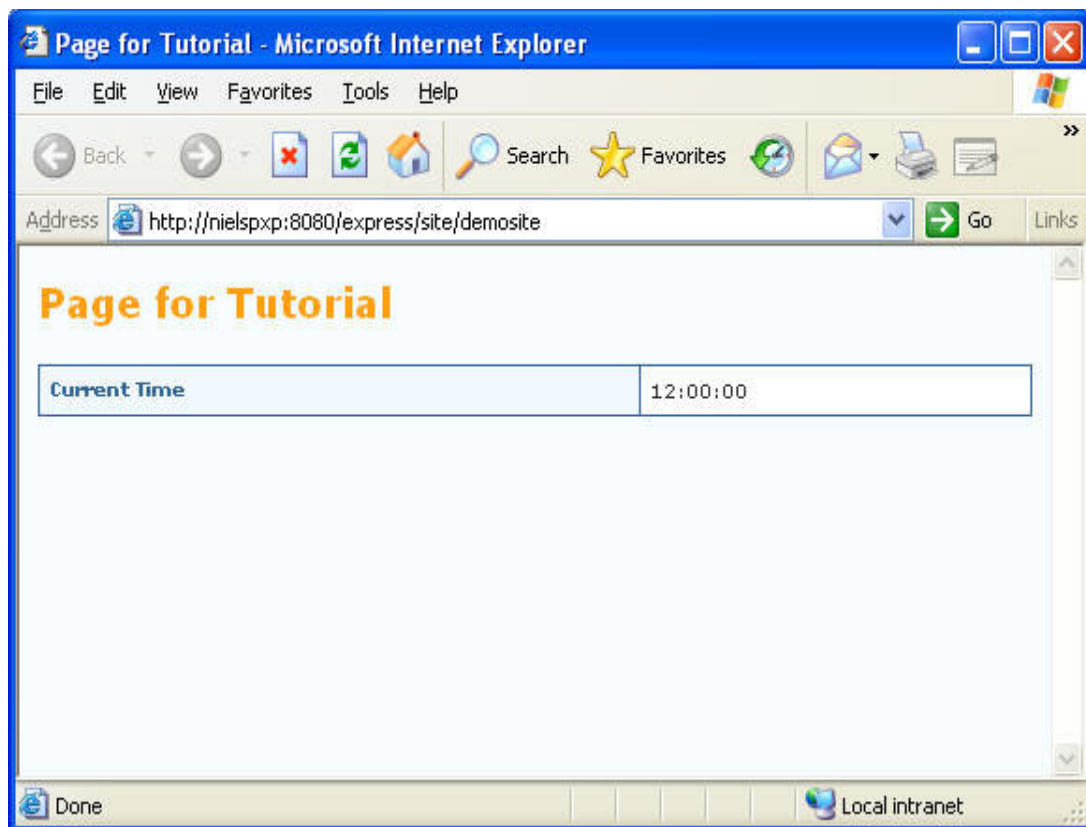




WebsydianExpress

# RPG Developer Tutorial

## Part 1 - Getting Data to the Web



## Table of Contents

---

Table of Contents.....	2
Introduction .....	3
Create template .....	3
Define Page Structure.....	4
Create Template based on the definitions made .....	9
Create ProcessEntryPoint Program .....	11
Source Code Explained .....	11
Compiler directives and includes .....	11
Declarations.....	12
Parameter Interface .....	12
Call PageGenerator .....	12
Create PageGenerator Program.....	12
Source Code Explained .....	13
Constant declarations .....	13
Populate Program Field .....	13
Set Output Values.....	14
Write Page to Browser.....	14
Compile Programs .....	15
Deploy Objects.....	15
Define Business Process in Websyidian Express .....	15
Specify Business Process .....	15
Add Business Process to Site Structure.....	16
Run Process .....	22

## Introduction

---

The first part of the tutorial will show how to create a business process that fetches the current time and displays it on an HTML page in a browser.

Instead of the current time, the data displayed in the browser could have been fetched from a database, being the result of calculations done in the program, or being values returned from other application programs.

## Create template

---

Websyidian Express uses HTML templates to create the HTML page sent to the browser. This means that it is necessary to create a template for the HTML page that will show the current time.

This is done using the Page Modeler found in the Websyidian Express administration interface.

You access the Page Modeler in this way:

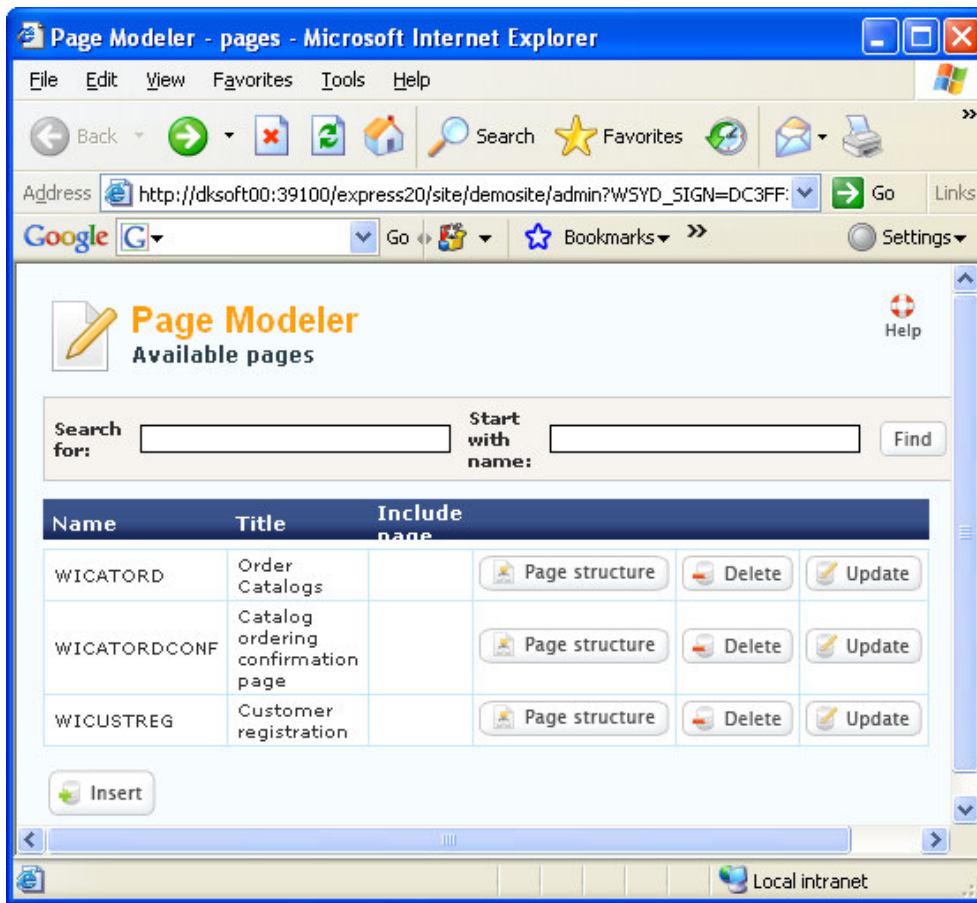
Access the administration interface by entering the URL for the demosite followed by /admin.

Log in to the administration interface. If you are using a default installation, you can access the site using the User WSADMIN, password ADMIN.

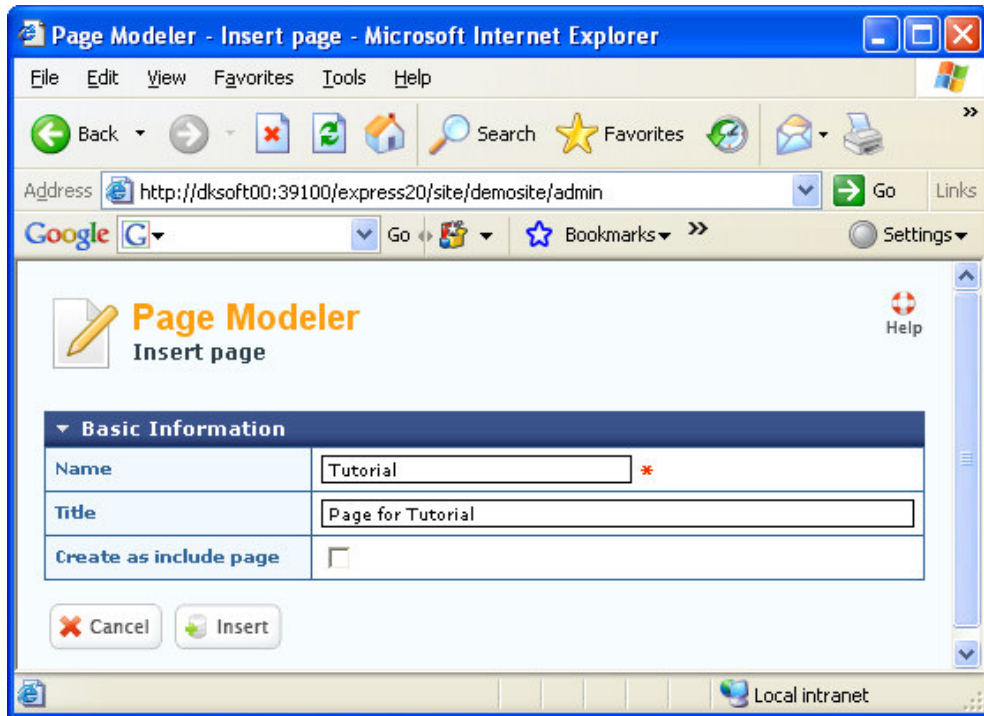
In the left menu, select the Developer tools→Page Modeler menu item.

## Define Page Structure

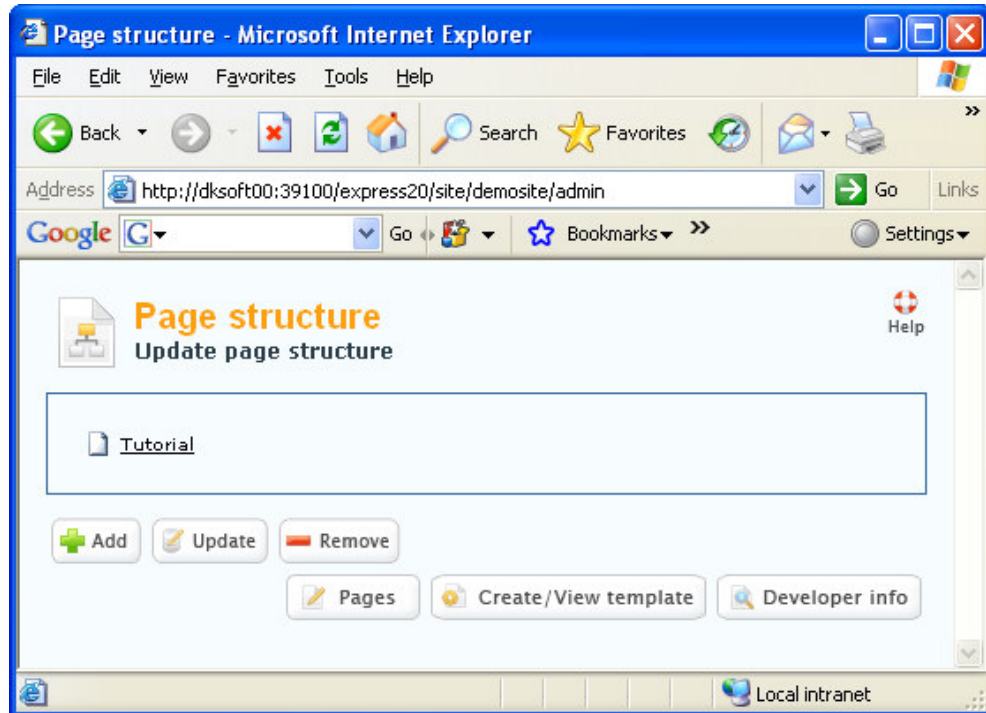
The start page for the Page Modeler shows a list of existing pages.



As a new page is needed, press "Insert".



Enter the values shown on the insert page. Press "Insert".

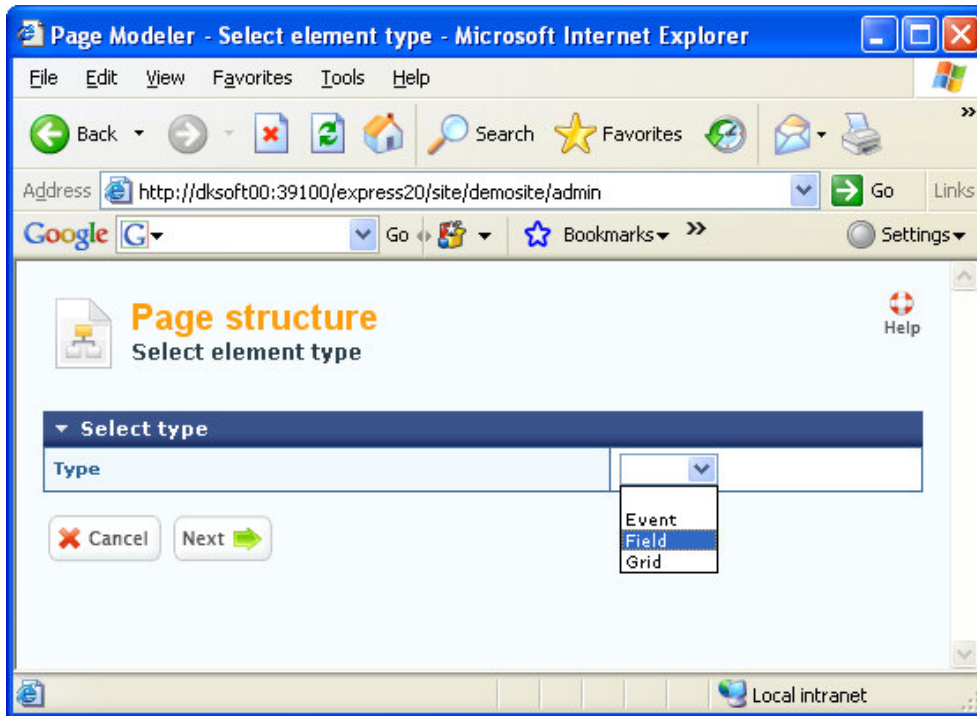


The page shown is a tree structure showing the contents of the Tutorial page.

As the page has only just been created, only the page itself exists.

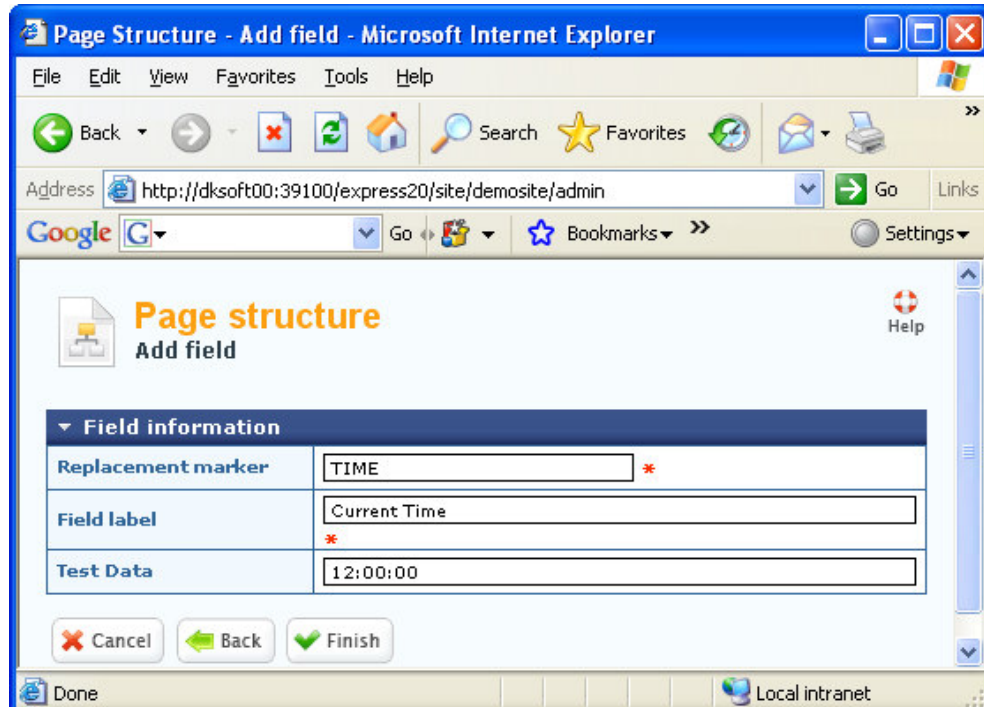
An output field must be added, so the current time can be written to the HTML page at runtime.

Select the Tutorial page with the mouse and press the Add button.



On the page shown, select "Field" in the Type drop down box. This should lead you to the next page.

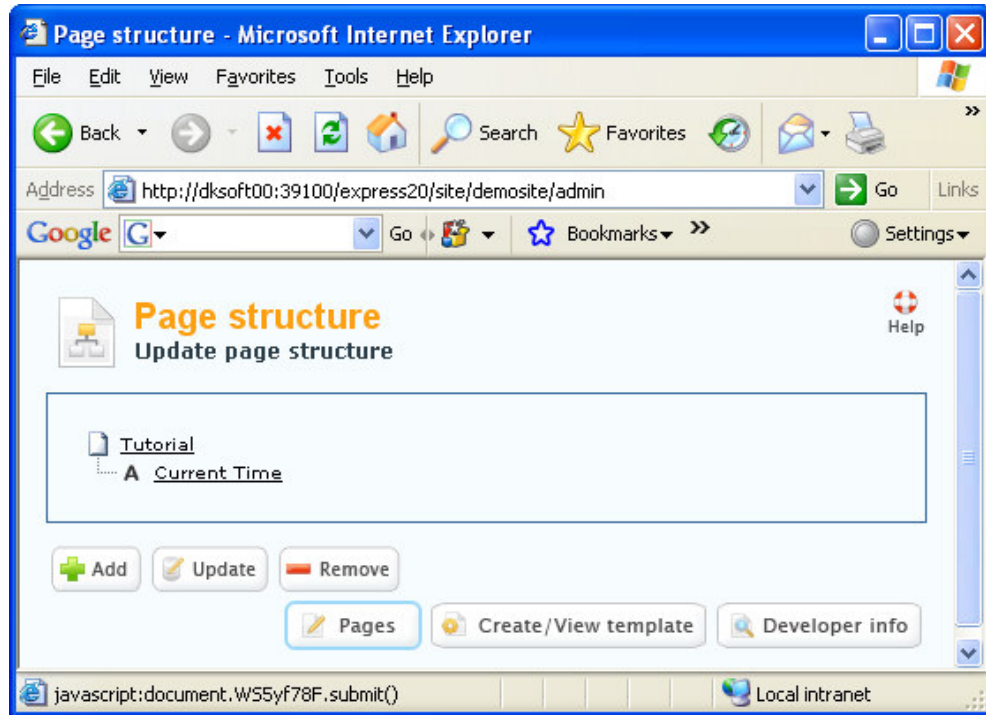
If this does not happen automatically, press "Next".



On this page, the definitions for the field are entered.

Enter the values shown.

Press "Finish" to add the defined field to the page.

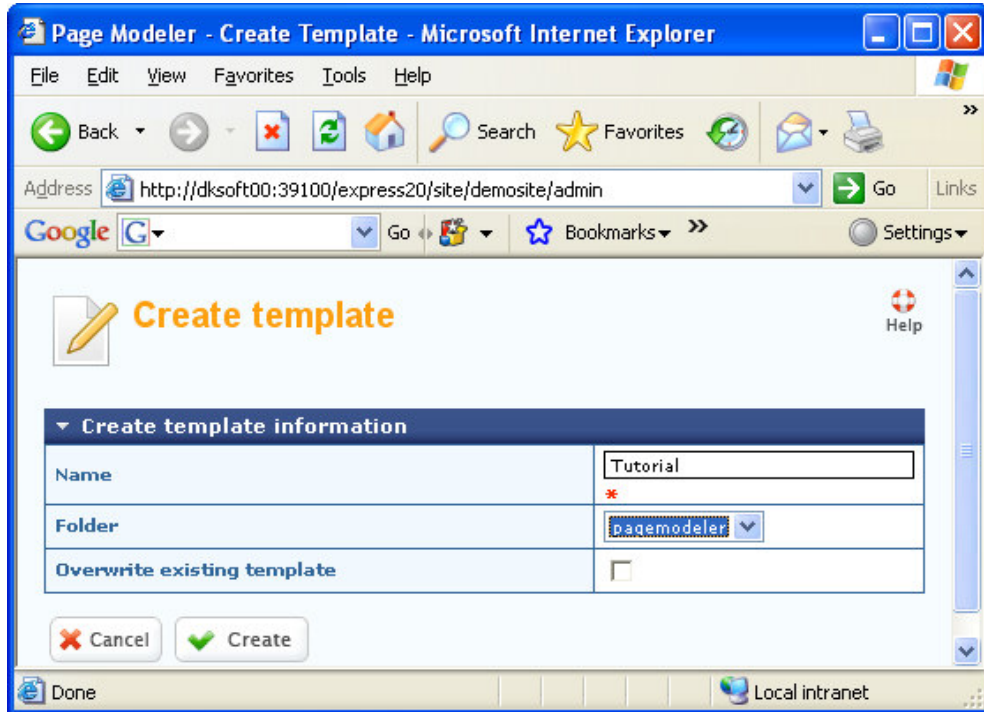


This is all that is needed to define an HTML template with one output field.

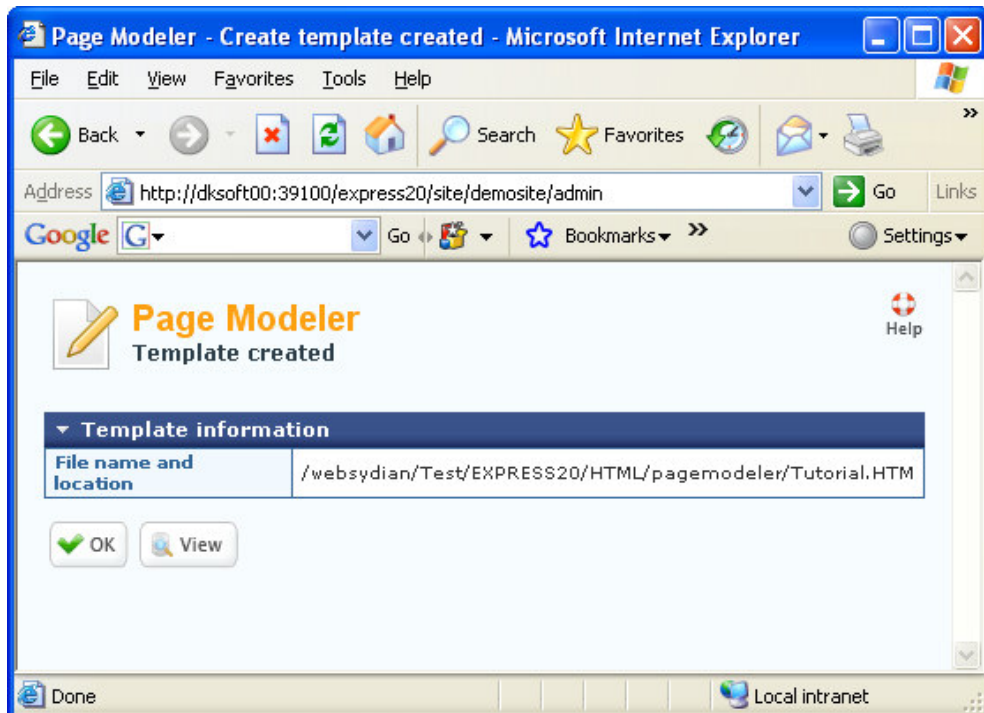


## Create Template based on the definitions made

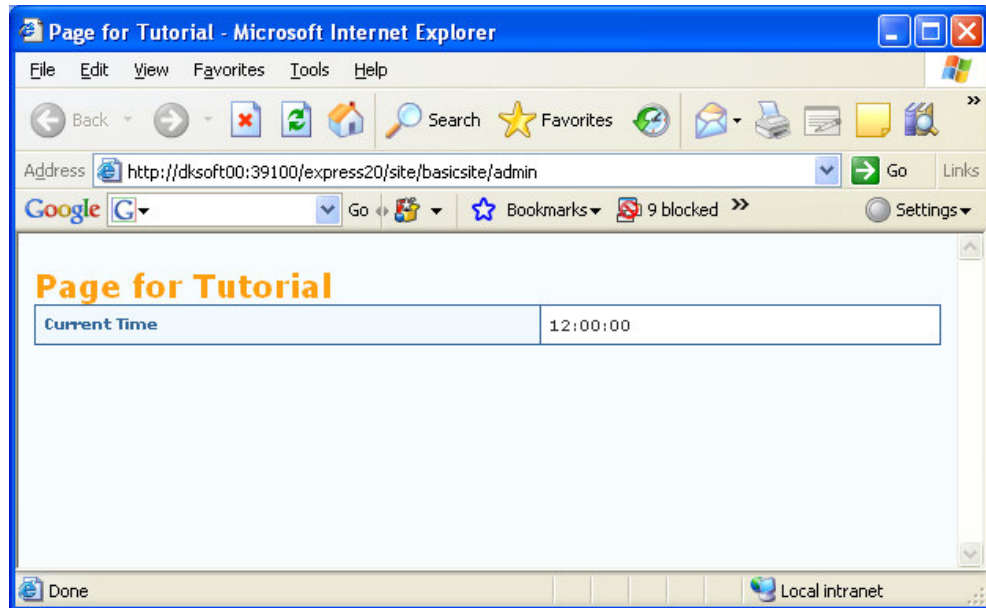
To create the actual template file, press "Create/View template".



Leave the name "Tutorial" as the value of the Name field. Select the folder "pagemodeler". Press "Create".



Press View to show the template with the standard style sheet applied and the test data entered for the field when the field was added to the page.



Now the template has been created. As it is placed in the pagemodeler folder, it can be used by the application to generate the HTML page without any further actions.

## Create ProcessEntryPoint Program

The Websyidian Express runtime calls a program, when a menu item is pressed.

For each business process it is necessary to create one of these programs. These programs are called ProcessEntryPoints.

As ProcessEntryPoints are called dynamically by the runtime, this program MUST have a specific interface (one parameter field - a length 7 character field).

As the ProcessEntryPoint always will be the first program called each time the business process is used, this program can be used as an initialization routine for the process.

The one step all ProcessEntryPoints MUST perform is to call the PageGenerator function for the first page to show in the business process.

Create a new ILE RPG source member named TUTORIAL containing the following source code:

```
H DFTACTGRP(*NO) ACTGRP(*CALLER) BNDDIR('QC2LE':'WIEXPRESS')
  /COPY QRPGLSRC,EXPRESS_H

D RC          S          like(TypRC) inz(*zero)
C   *ENTRY    PLIST
C           PARM          Return          7

* Initialize
C           eval          RC = 0
C           eval          Return = *blank

* Call the Page Generator
C           call          'TUTORIALPG'

C           RETURN
```

### Source Code Explained

In these sections, the source code shown will be explained. Only the statements deemed to have a special significance for developing for Websyidian Express will be explained.

#### Compiler directives and includes

```
H DFTACTGRP(*NO) ACTGRP(*CALLER) BNDDIR('QC2LE':'WIEXPRESS')
  /COPY QRPGLSRC,EXPRESS_H
```

The most important part of the first line is the declaration of WIEXPRESS as a binding directory.

The /COPY statement makes the prototypes defined in the delivered EXPRESS\_H header file available for the program.

## Declarations

```
D RC          S          like(TypRC) inz(*zero)
```

The API functions used returns a return code indicating whether the call to the API was successful. To ensure that the declaration of this field is correct refer to the TypRC abstract field defined in the WICORE\_H header file (the WICORE\_H file is referred in the EXPRESS\_H header file referred in the /COPY statement).

## Parameter Interface

```
C      *ENTRY      PLIST
C      PARM          Return          7
```

ProcessEntryPoints are called dynamically by the runtime. This means that all ProcessEntryPoints MUST have this interface.

## Call PageGenerator

```
C          call      'TUTORIALPG'
```

All ProcessEntryPoints MUST call a PageGenerator function. Otherwise nothing will be sent to the browser.

## Create PageGenerator Program

---

Create a new source member TUTORIALPG with the following contents:

```

H DFTACTGRP(*NO) ACTGRP(*CALLER) BNDDIR('QC2LE':'WIEXPRESS')
  /COPY QRPGLSRC,EXPRESS_H
D TEMPLATE          C          'TUTORIAL'
D FLD_TIME          C          'TIME'
D rc                S          like(TypRC) inz(*zero)
D Time              S          8T

c                    eval      Time = %time()

c                    eval      rc = SetOutput(TEMPLATE :
c                                     FLD_TIME :
c                                     %char(Time))

c                    eval      rc = WritePage(TEMPLATE)

C                    return

```

## Source Code Explained

### Constant declarations

```
D TEMPLATE          C          'TUTORIAL'
```

This assigns the name of the template specified in the PageModeler to a constant. This constant is used whenever a call to an API must transfer information about the template that is being handled.

```
D FLD_TIME          C          'TIME'
```

This assigns the name of the time field specified in the PageModeler to a constant. This constant is used when the value is assigned to the field.

Declaring these constants makes it easier to change the name of a template or a field if this at some point is deemed necessary. Furthermore, having these declarations in the top of the program makes it somewhat easier to compare the definitions in the PageModeler with the definitions in the programs.

### Populate Program Field

```
c                    eval      Time = %time()
```

This is of course just a normal use of a built-in function. The main point to observe here is that it does not really matter how you populate a program field - as soon as you have the value in a field, you will be able to get it written to the web page - as shown in the next steps.

## Set Output Values

```
c          eval          rc = SetOutput(TEMPLATE :  
c          FLD_TIME :  
c          %char(Time))
```

The SetOutput function is one of the delivered API functions, it is prototyped in the WICORE\_H header file, where you can read a short description of the API as well as see the formal declaration of the interface.

The parameters used are:

### 1. Template Name

Specifies the template the value is written to.

### 2. Field Name

Specifies the name of the field the value is specified for.

### 3. Field Value

Specifies the value assigned to the field.

The specified template need to contain a field with the specified name defined for this call to have any effect. If you assign a value to a non-existing field - nothing will happen.

All values set or retrieved are character strings,

This means that is necessary to convert the time value before writing it to the page.

## Write Page to Browser

```
c          eval          rc = WritePage(TEMPLATE)
```

The WritePage API is one of the delivered API functions, it is prototyped in the WICORE\_H header file, where you can read a short description of the API as well as see the formal declaration of the interface.

The parameter used is:

### 1. Template Name

Specifies the template used to generate the page that is to be sent to the browser.

The WritePage API reads the specified template (generated by the PageModeler) and merges this template with the value specified by the SetOutput call. The generated template is sent to the browser.

## Compile Programs

---

The EXPRESS\_H header file is placed in the runtime library specified at the installation (default WXP2OPGM). Add this library to your library list before compiling.

For this tutorial, it is recommended that you create the objects in the library that was specified as the application library during the installation (default WXP2OAPP). Doing this means that the objects are immediately available for the Websyidian Express runtime.

Compile the two programs.

## Deploy Objects

---

If you have chosen not to compile the programs to the application library, you must move the two program objects to this library.

The template was generated to the pagemodeler folder. This means that it is already available for the application.

## Define Business Process in Websyidian Express

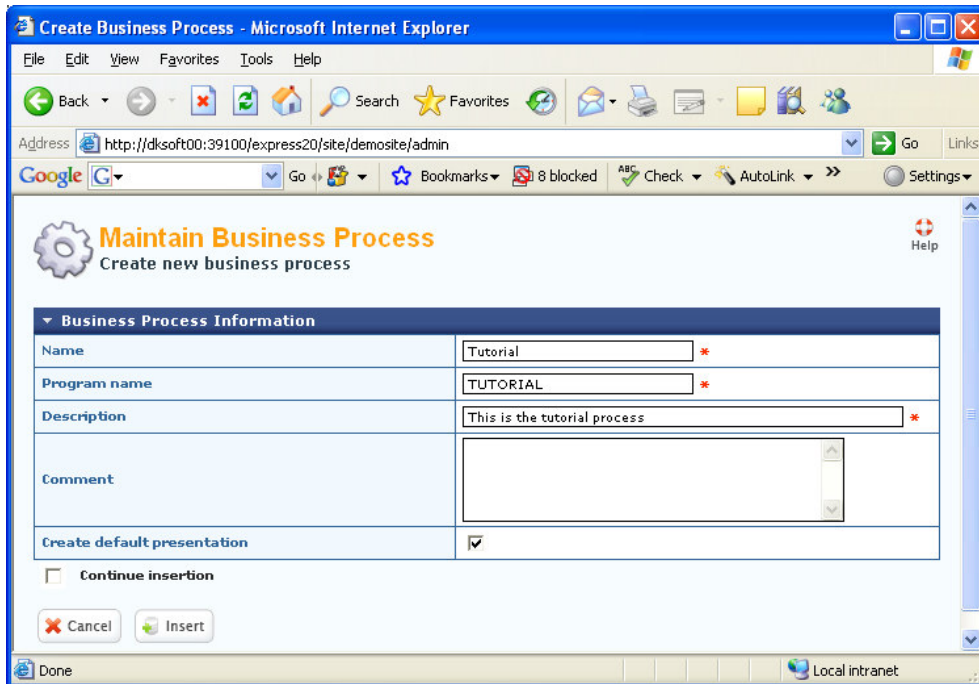
---

### Specify Business Process

Log in to the administration interface for the demo site using the WSADMIN user profile.

In the administration menu to the left, select Content Loaders→Business Processes.

Press the Insert button.



On the "Create new process" page, enter the values shown above and press Insert.

Please note that:

- The Program name **MUST** correspond to the name of the ProcessEntryPoint program object.
- Do **NOT** remove the check mark from the "Create default presentation". Keeping the check mark ensures that a standard presentation for the process will be created.

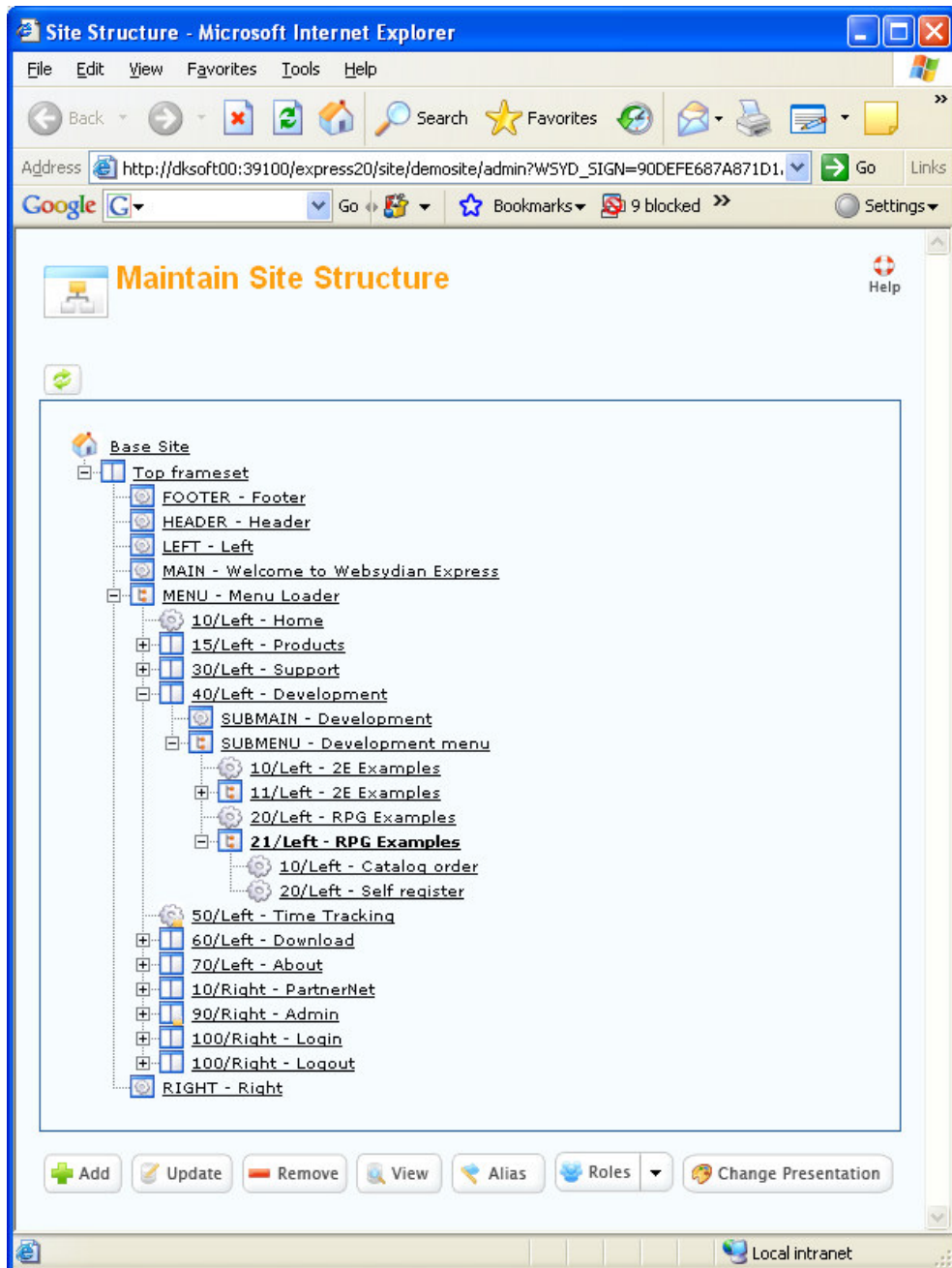
The above definitions make it possible for the Websyidian Express runtime to call the business process.

The business process must also be made available as a menu item. This will be done in the next section.

## Add Business Process to Site Structure

In the administration menu, select Site Structure→Site Structure.

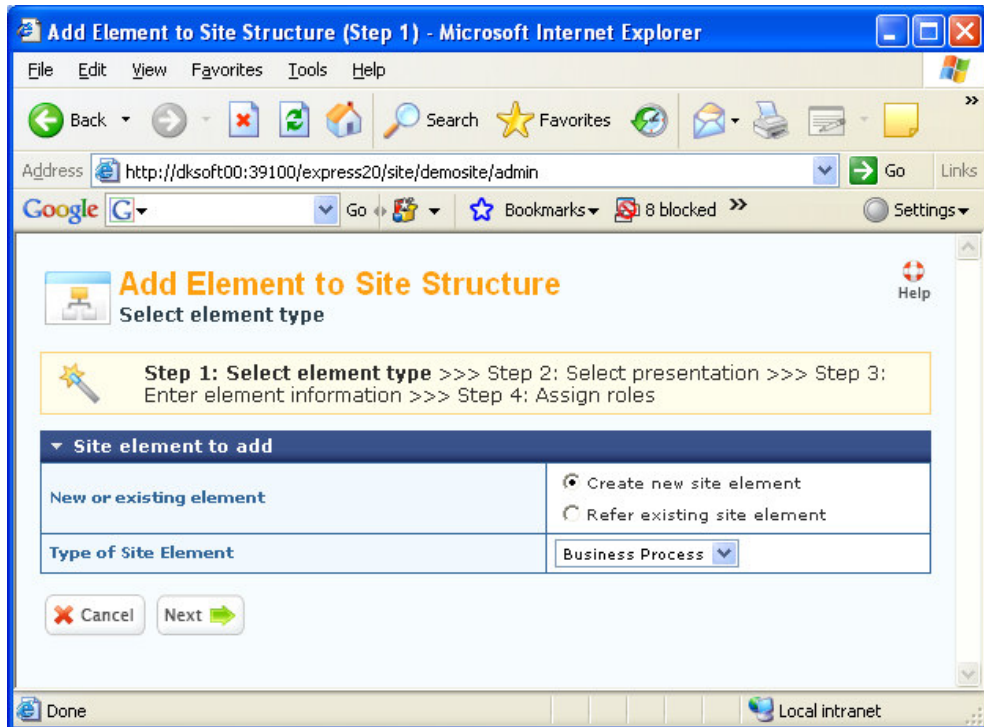




Expand "Top frameset", "MENU - Menu Loader", "40/Left - Development", and "SUBMENU - Development Menu" to access the menu item "21/Left - RPG Examples".

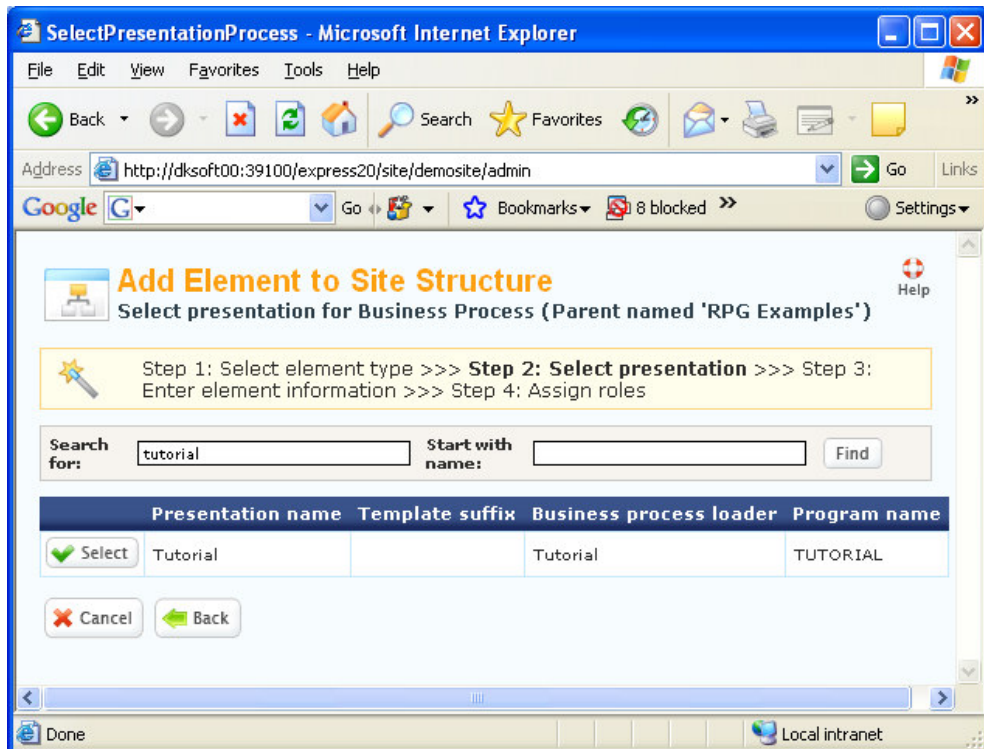
Select this site element and press "Add".

This starts a wizard leading you through the steps needed to add a new menu item to the menu.



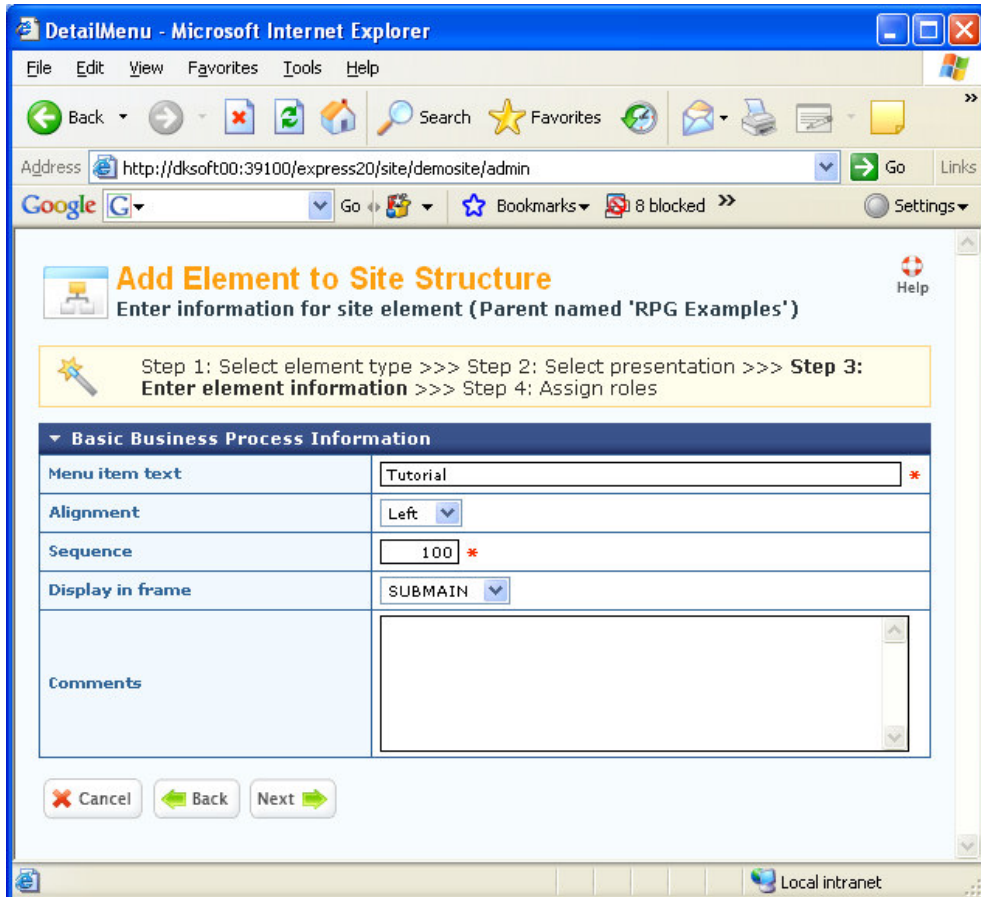
On the first page, keep the selections of "Business Process" and "New". Press "Next".

Select the presentation that was defined when the business process was created.



Enter "tutorial" in the Search for field and press "Find".

Select the tutorial process by pressing the "Select" button.



**Add Element to Site Structure**  
Enter information for site element (Parent named 'RPG Examples')

Step 1: Select element type >>> Step 2: Select presentation >>> **Step 3: Enter element information >>>** Step 4: Assign roles

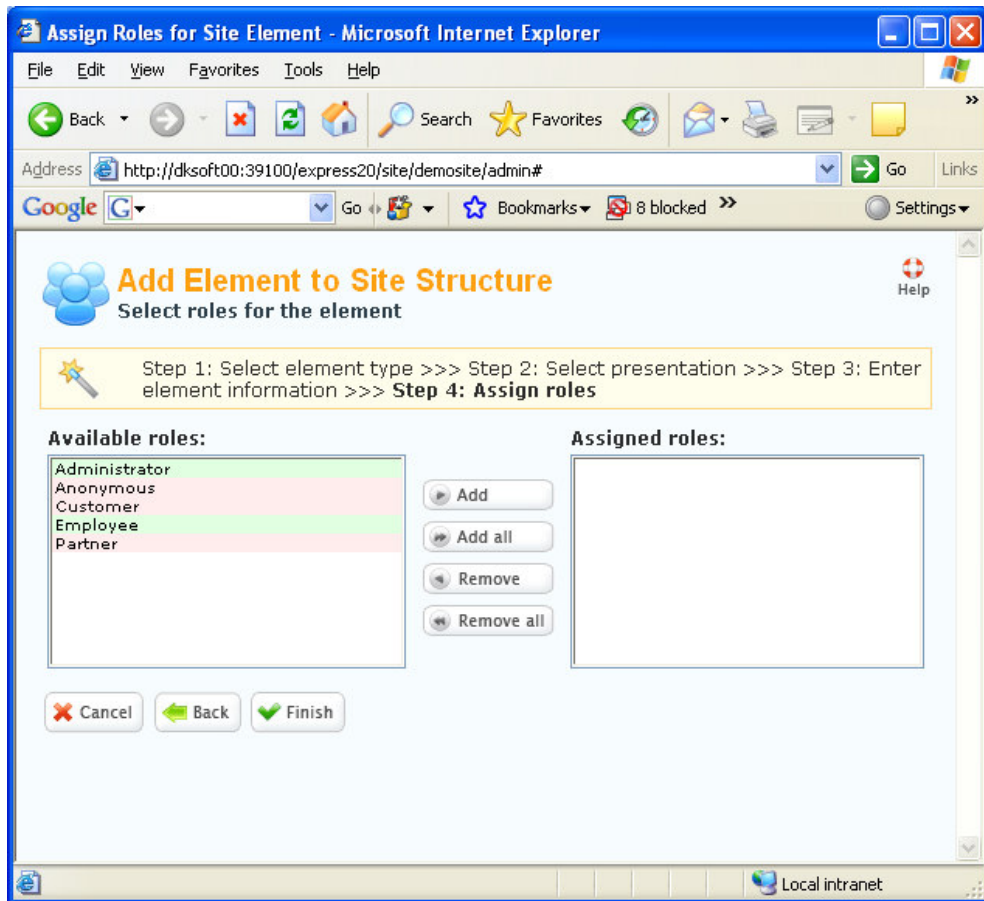
**Basic Business Process Information**

Menu item text	Tutorial *
Alignment	Left
Sequence	100 *
Display in frame	SUBMAIN
Comments	

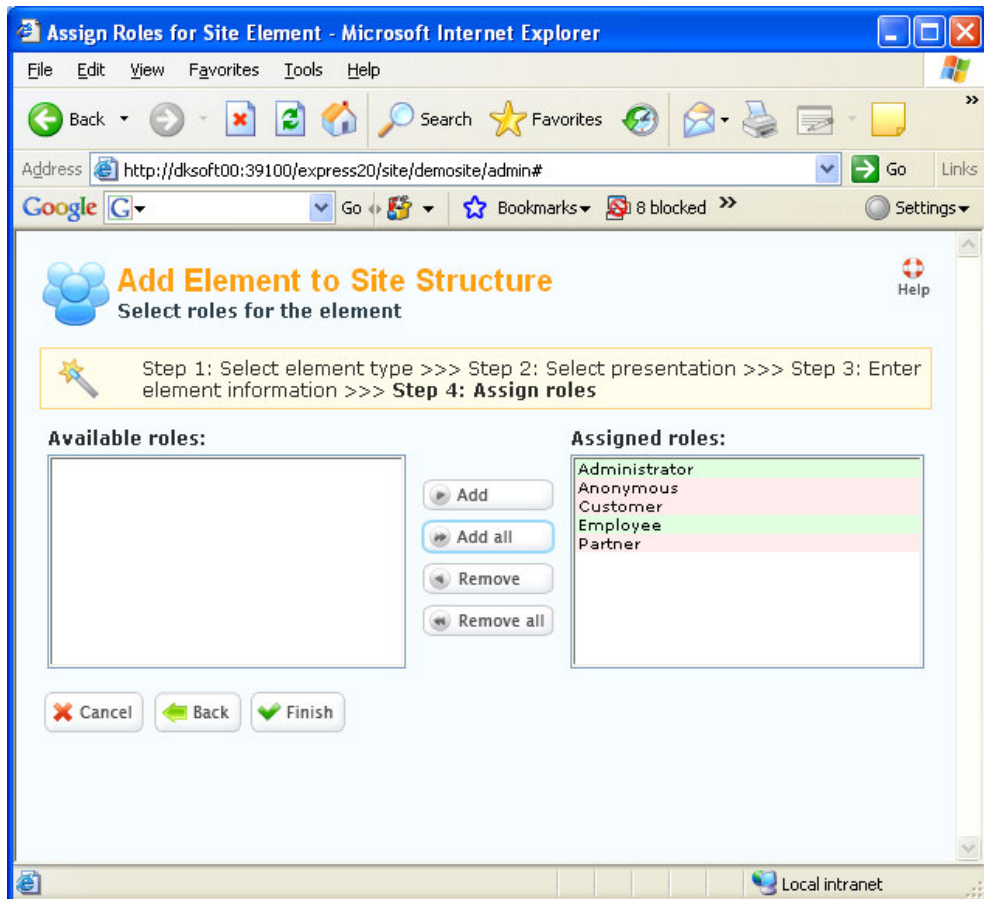
Cancel Back Next

Enter/Select the values shown and press "Next".

- Menu item text  
The text that will be shown in menu
- Alignment  
Selects the part of the menu that the item will be shown in. As this is a vertical menu, the alignments are a bit irrelevant - left is selected for all items.
- Sequence  
Determines the position of the menu item in the menu
- Display in frame  
Determines which frame the business process will populate when the menu item is pressed.

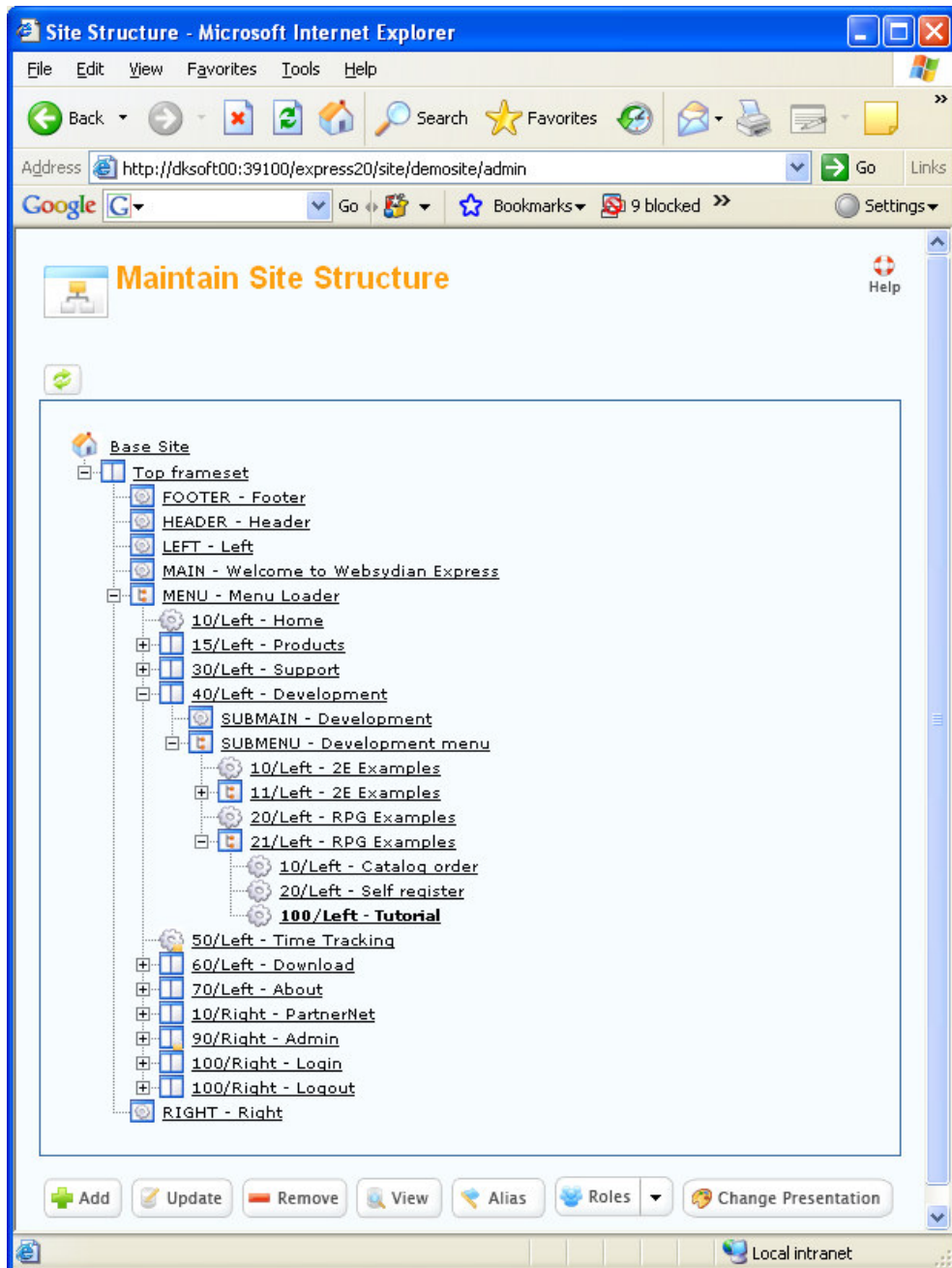


Press "Add all" to give everyone access to the business process.



Press "Finish" to create the new site element and return to the site structure.

This returns to the site structure where you should be able to see that the menu item has been added.



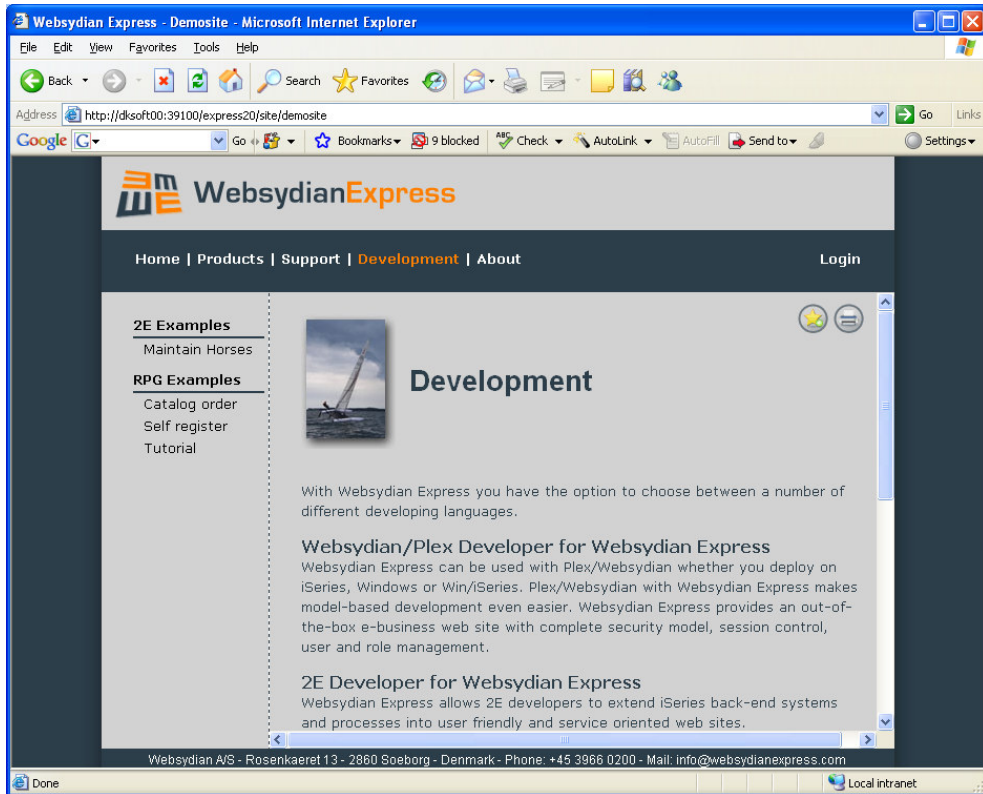
## Run Process

Close the browser, and start the demo site in a new browser instance.

Press the Development menu item in the top menu

The Tutorial menu item will be shown in the left menu.





Press the Tutorial menu item.

The following page should be shown.

