

WEBSYDIAN™

2E Tutorial

Get started with WebsydianExpress Developer for 2E

Revision 2

```
Session A - [24 x 80]
Fil Redigér Vis Kommunikation Funktioner Vindue Hjælp
EDIT ACTION DIAGRAM Edit WXPMDL Websydian Express Example
FIND=> Tutorial Page
I(C,I,S)F=Insert construct I(X,0)F=Insert alternate case
I(A,E,Q,*,+,-,=,=A)F=Insert action IMF=Insert message
---
> Tutorial Page
---
. Set W2E/TemplateName to the template used by the PageGenerator
. WRK.W2E/TemplateName = CND.CHANGE THIS
---
. ...Set values for output fields
. ...Set initial value for input fields
. ...Set parameters for events
---
. WritePage - Websydian Express RPG *
. Exit program - return code PGM.*Return code
---
F3=Prev block F5=User points F6=Cancel pending moves F23=More options
F7=Find F8=Bookmark F9=Parameters F24=More keys
MÅ a MW 08/002
I902 - Sessionen er startet uden fejl I:\WTSERVER0\Olivetti d-Copia 25Aktiv\IP_192.168...
```

Table of Contents

| | |
|---|----|
| Table of Contents | 1 |
| Prerequisites..... | 2 |
| Install WebsyidianExpress instance..... | 2 |
| Import the 2E model objects into your local model | 2 |
| Overview..... | 2 |
| Tutorial | 2 |
| Step 1: Identify Pages..... | 2 |
| Step 2: Identify Events | 2 |
| Step 3: Create functions in 2E..... | 3 |
| Create ProcessEntryPoint | 3 |
| Create PageGenerator | 3 |
| Create EventHandler..... | 3 |
| Step 5: Create template using PageModeler..... | 3 |
| Step 6: Define and code PageGenerator "Tutorial Page" | 13 |
| Make action diagram code for the Pagegenerator | 15 |
| Set values for Output fields | 17 |
| Set initial value for input fields..... | 19 |
| Step 7: Define and code ProcessEntryPoint Tutorial.PE | 20 |
| Step 8: Define and code EventHandler Tutorial.Test..... | 22 |
| Retrieve User Input | 23 |
| Step 9: Generate and build functions | 26 |
| Step 10: Deploy objects | 27 |
| Step 11: Define Process in the administration interface | 27 |
| Define business process | 27 |
| Add business process to site structure | 29 |
| Step 11: Add pagemodeler folder to Anonymous folder list..... | 34 |
| Step 12: Test | 37 |

Prerequisites

To complete the tutorial, you must have an active installation of WebsyidianExpress (iSeries/iSeries version).

Install WebsyidianExpress instance

You can find guidelines on how to install WebsyidianExpress in the documentation on www.websyidianexpress.com.

Choose Support, Documentation to access the documentation.

Import the 2E model objects into your local model

The import of the 2E model object is described here:

www.websyidian.com/websydiandoc/express/2edeveloper/2EModel.htm.

Overview

This tutorial will show you how to make a simple business process for WebsyidianExpress using 2E Developer for WebsyidianExpress.

The resulting business process will contain one page on which an output field and a button with an input field will be shown.

When the button is pressed, the text entered in the input field is read by the application and used for creating the text for the output field - and for populating the input field again.

The tutorial will follow the method described in the document "How to develop a business process". This document can be found here:

www.websyidian.com/websydiandoc/Express/development/2edeveloper/developbusinessprocess.htm

Tutorial

Step 1: Identify Pages

Before starting to develop a business process, you must have at least a reasonably good idea about which pages you are going to create, what information you want to be able to show on the pages, and which event (buttons/links) you want to have on the pages.

For the tutorial, you will create one page (Tutorial Page). This page will have one output field.

Step 2: Identify Events

The events provide the user with the possibility to interact with the business process. Events will typically be presented to the user as buttons or links on the pages.

For each event you will need to know what information the user must provide for the application (and thereby the necessary input fields for the events) and what page you want to show when the event is activated.

The tutorial will have one event - a button (Test) placed on the tutorial page that has one inputfield.

Step 3: Create functions in 2E

In 2E, all functions must be scoped by a file. Normally, you would want the scoping file to be the file containing the primary data for the application.

However, to keep the tutorial simple there are no file interactions.

This means that you can either use an existing file or create a "dummy" file called Websyidian Tutorial to scope the functions.

Three functions must be created for the tutorial:

ProcessEntryPoint (Tutorial.PE)

This is the function that will be called by the runtime, when the business process is called from the menu.

PageGenerator (Tutorial Page)

This is the function that will populate the fields on the page and use these values and the template to generate the final page and send it to the browser.

EventHandler (Tutorial.Test)

This is the function that will be called by the runtime when the Test button on the tutorial page is pressed. The function will read the data entered by the user and call the PageGenerator.

Create ProcessEntryPoint

On the Edit Functions page under the scoping file, press F21 "Copy *Template function".

In the list shown, select W2E/ProcessEntryPoint.

Name the new function "Tutorial.PE" (access path *NONE).

Create PageGenerator

Press F21.

Select W2E/PageGenerator.

Name the new function "Tutorial Page" (access path *NONE).

Create EventHandler

Press F21.

Select W2E/EventHandler.

Name the new function "Tutorial Page.Test" (access path *NONE).

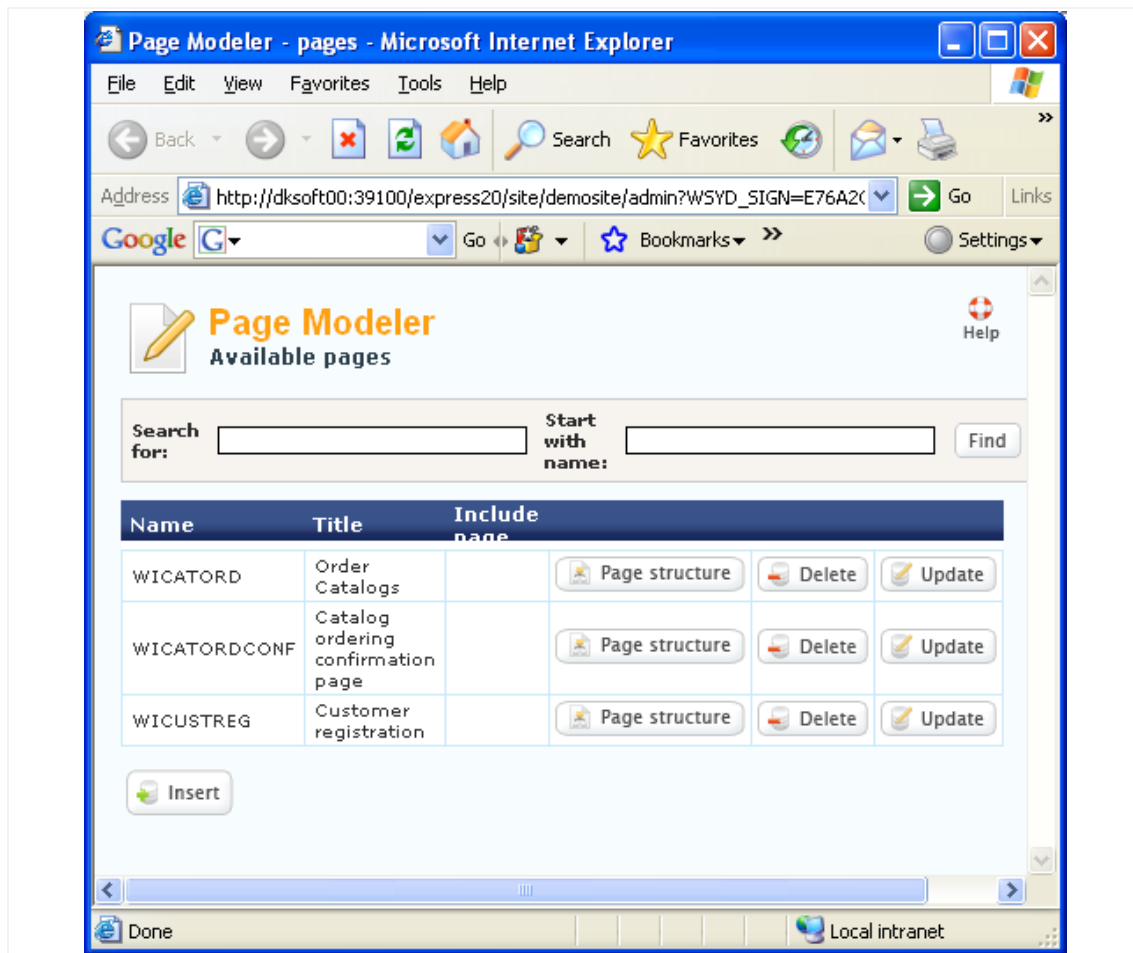
Step 5: Create template using PageModeler

In WebsyidianExpress, web pages are created by merging a template, defining the static content of the page, with values set by the programs.

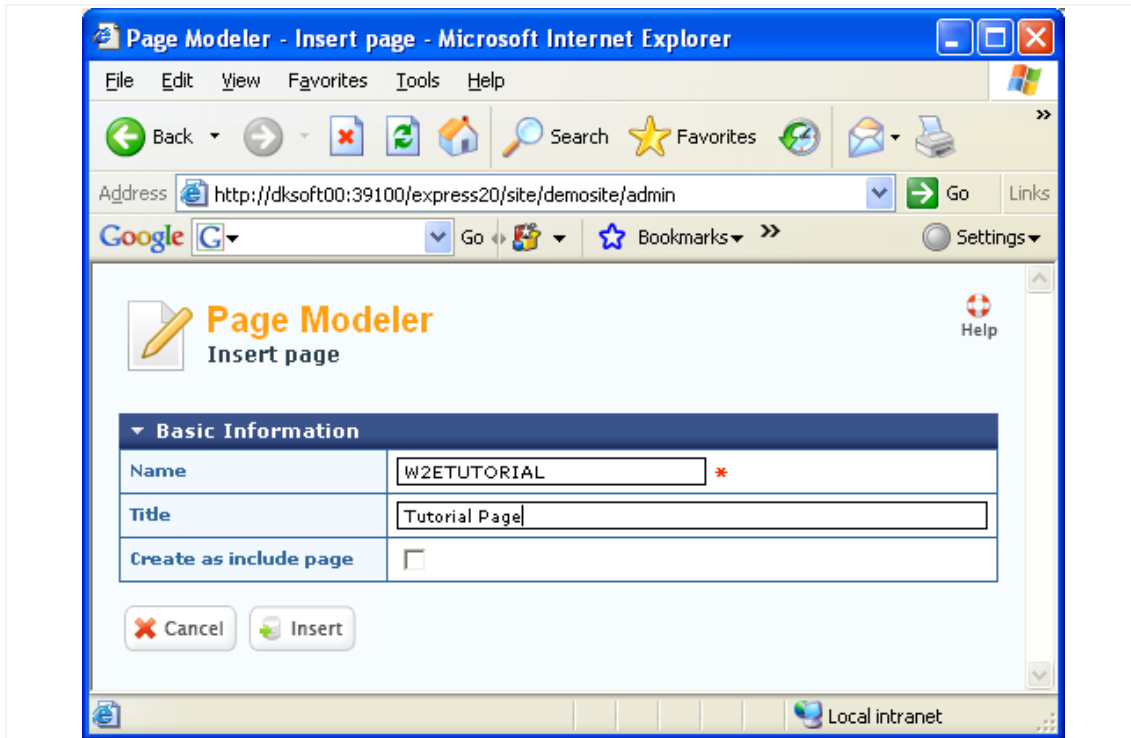
This means that for each page you want to be able to create, you will need to have a template that defines the content of the page. The Page Modeler application provides you with a way to obtain a first cut version of the template.

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

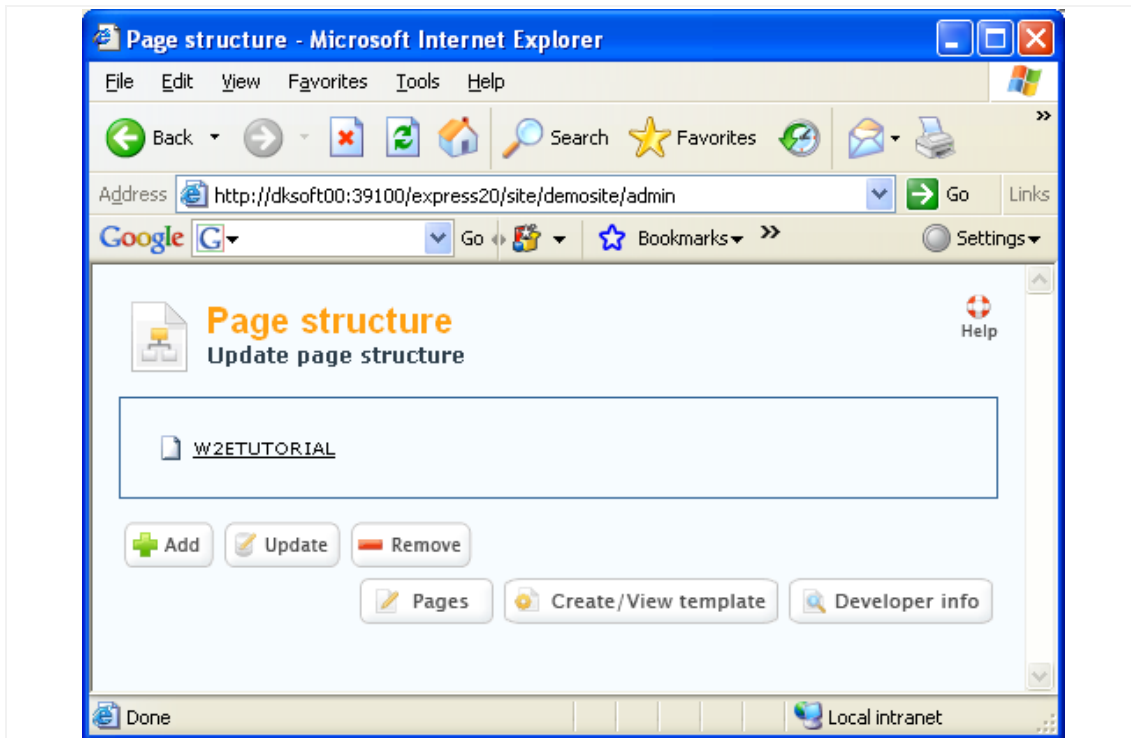
Select the menu item Developer tools → Page Modeler.



Press "Insert".



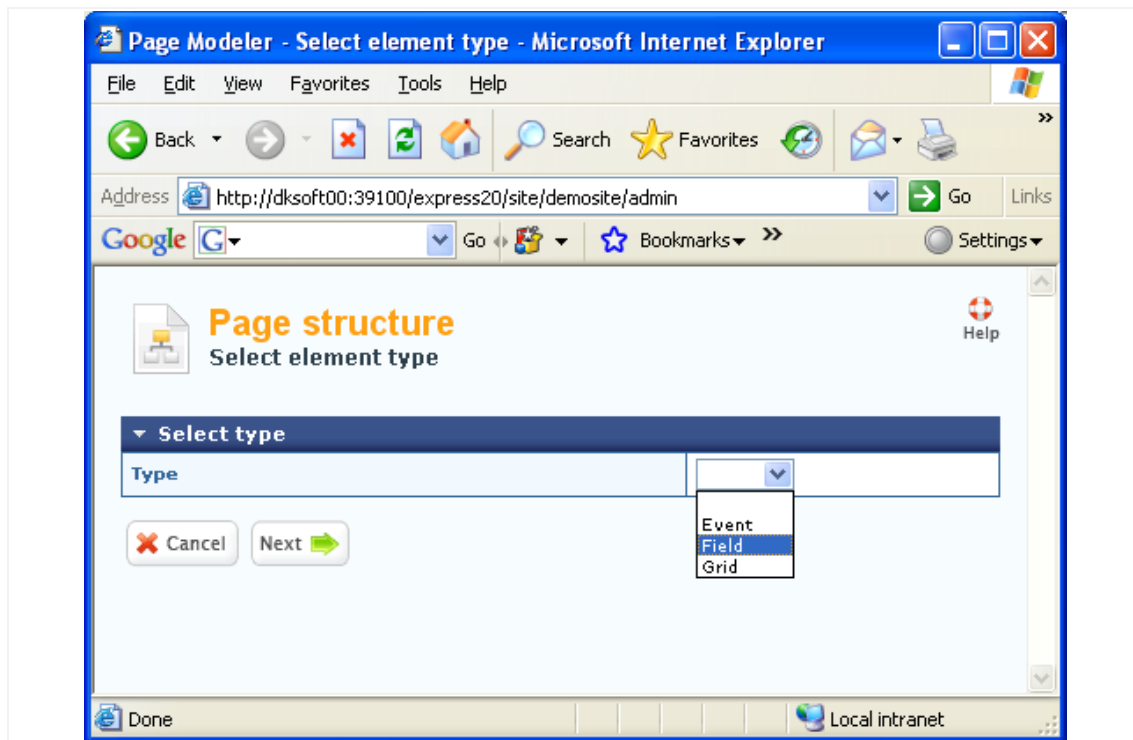
Enter:
Name: W2ETUTORIAL
Title: Tutorial Page
Press "Insert".



The first thing that will be added to the page is the output field.

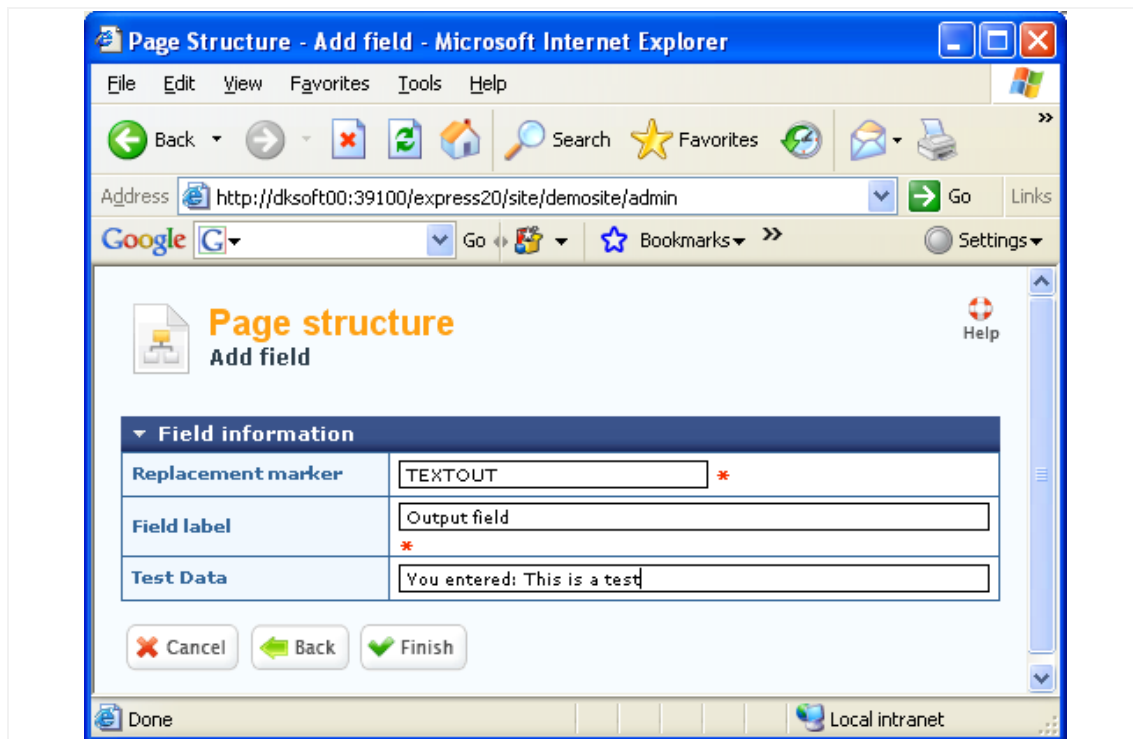
Select W2ETUTORIAL.

Press "Add".



Select Field in the drop down box.

(If the browser does not automatically show the next page, Press "Next").

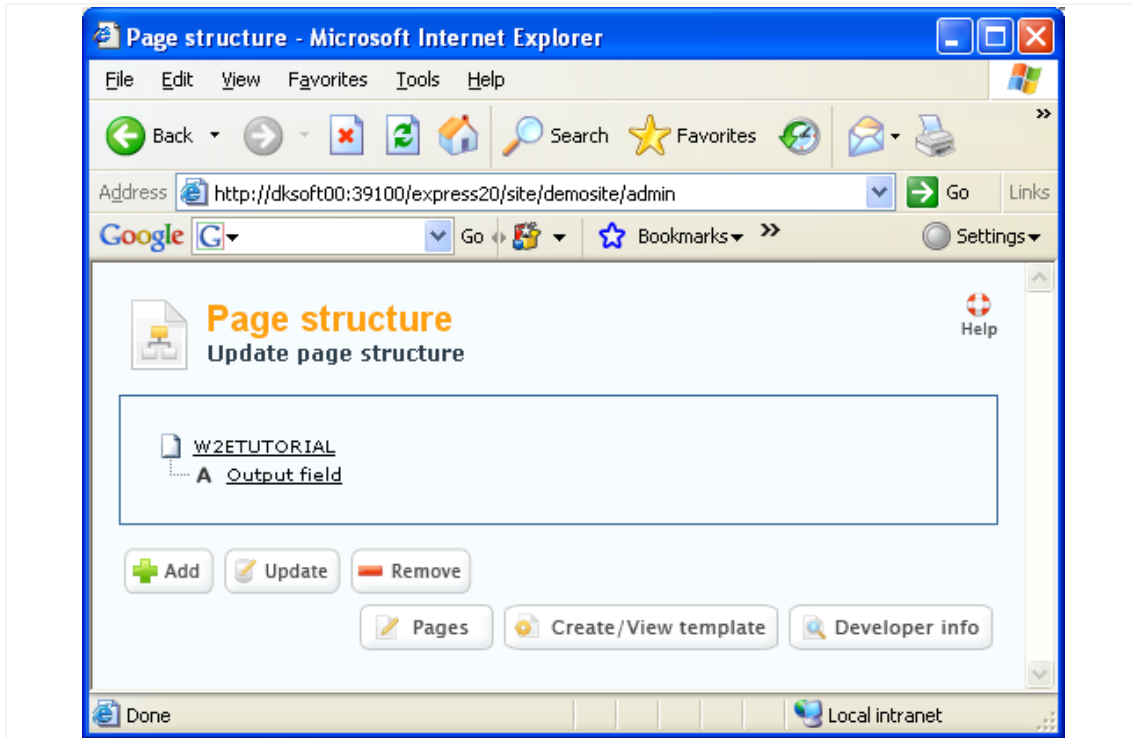


Enter:

Replacement marker: TEXTOUT

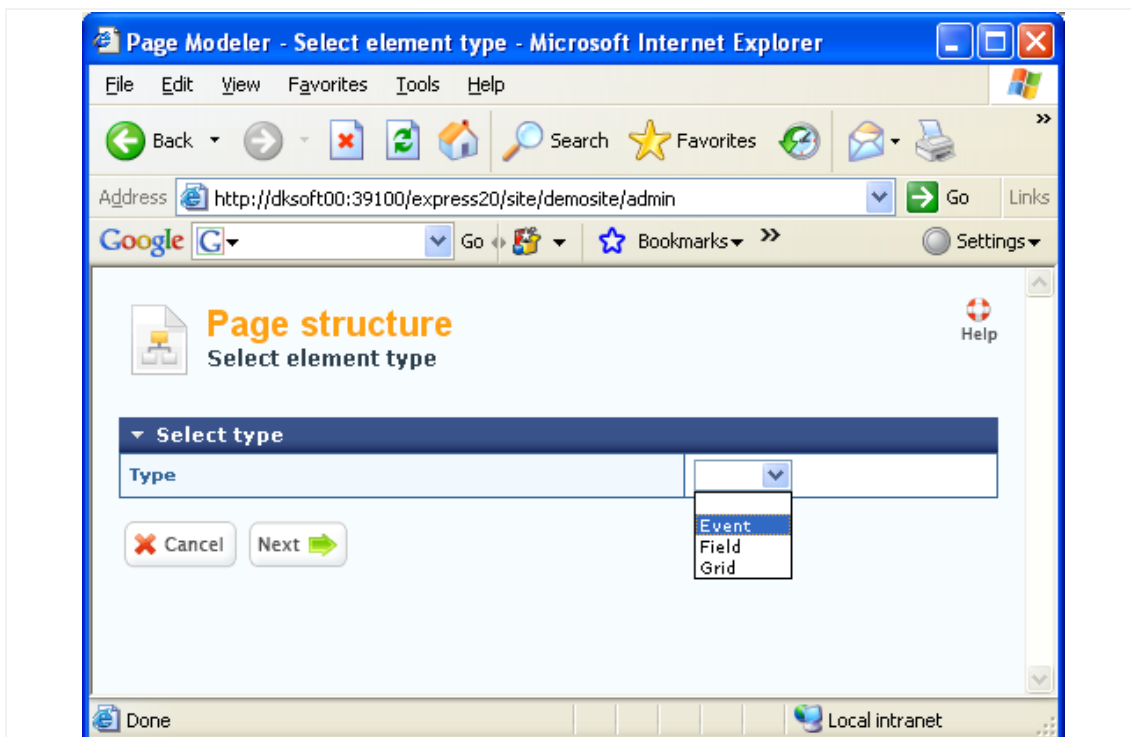
Field label: Output field

Test Data: You entered: This is a test
Press "Finish".



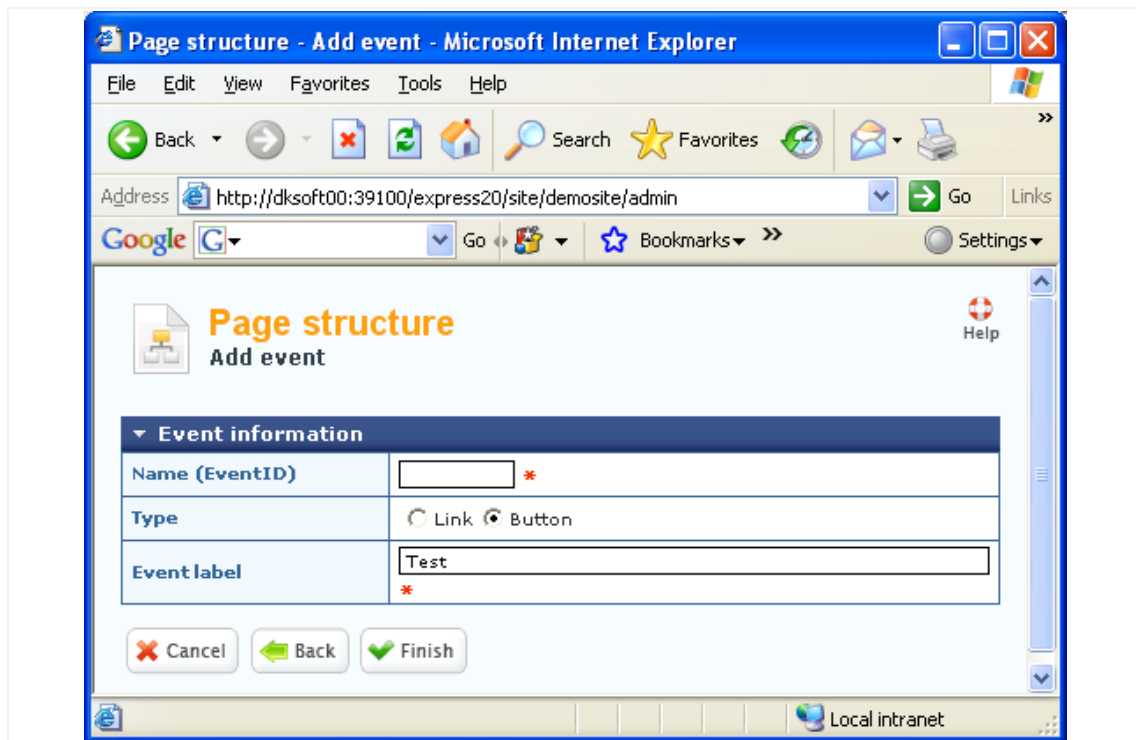
The next thing to add is the event Test (as a button):

Select W2ETUTORIAL.
Press "Add".



Select Event in the drop down box.

(If the browser does not automatically show the next page, Press "Next").



The EventID must be the name of the function that will handle the request that is created when the button is pressed. In this case it is the EventHandler Tutorial.Test.

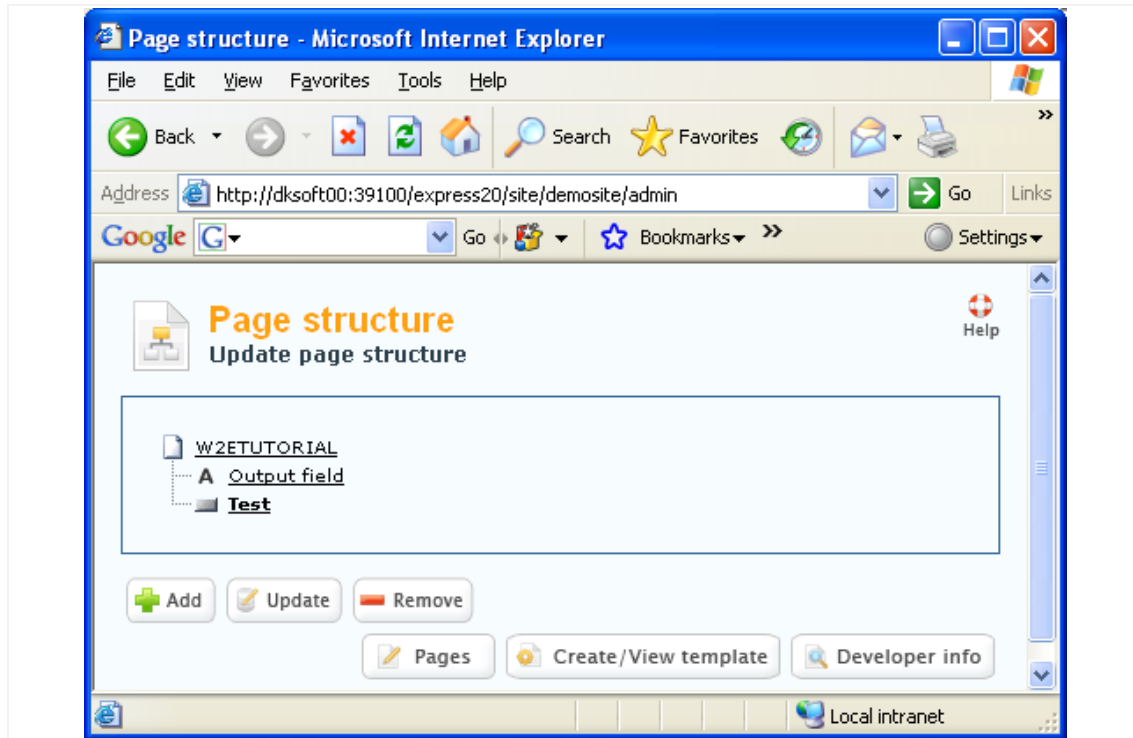
Enter:

Name (EventID): Enter the Source Name of the EventHandler function Tutorial.Test created in step 3.

Type: Keep the selection of "Button".

Event Label: Test

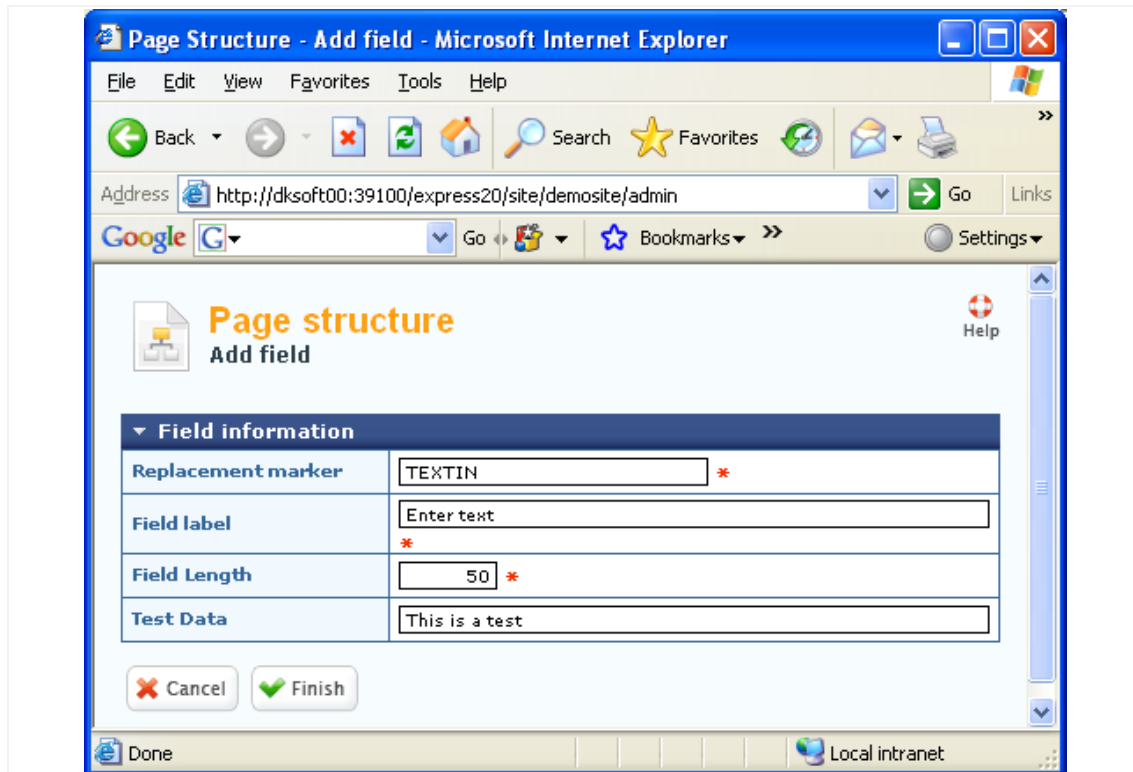
Press "Finish".



The last thing to add is the input field for the event.

Select "Test".

Press "Add".

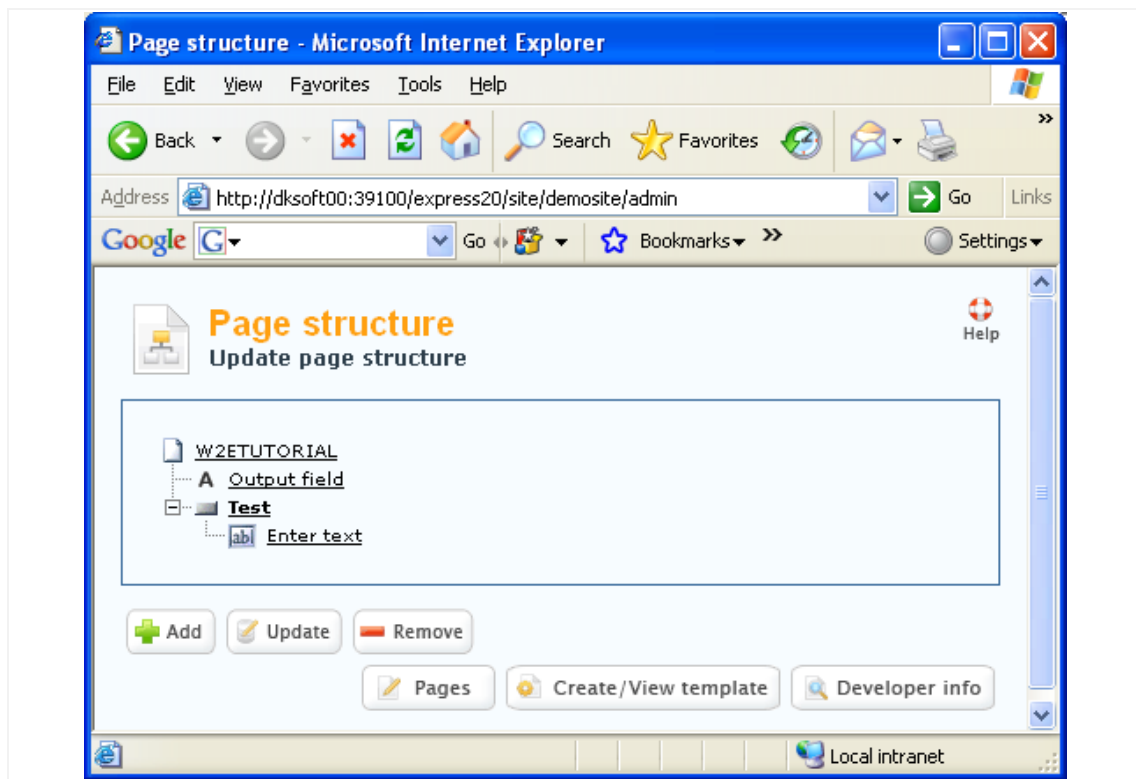


Enter:

Replacement Marker: TEXTIN

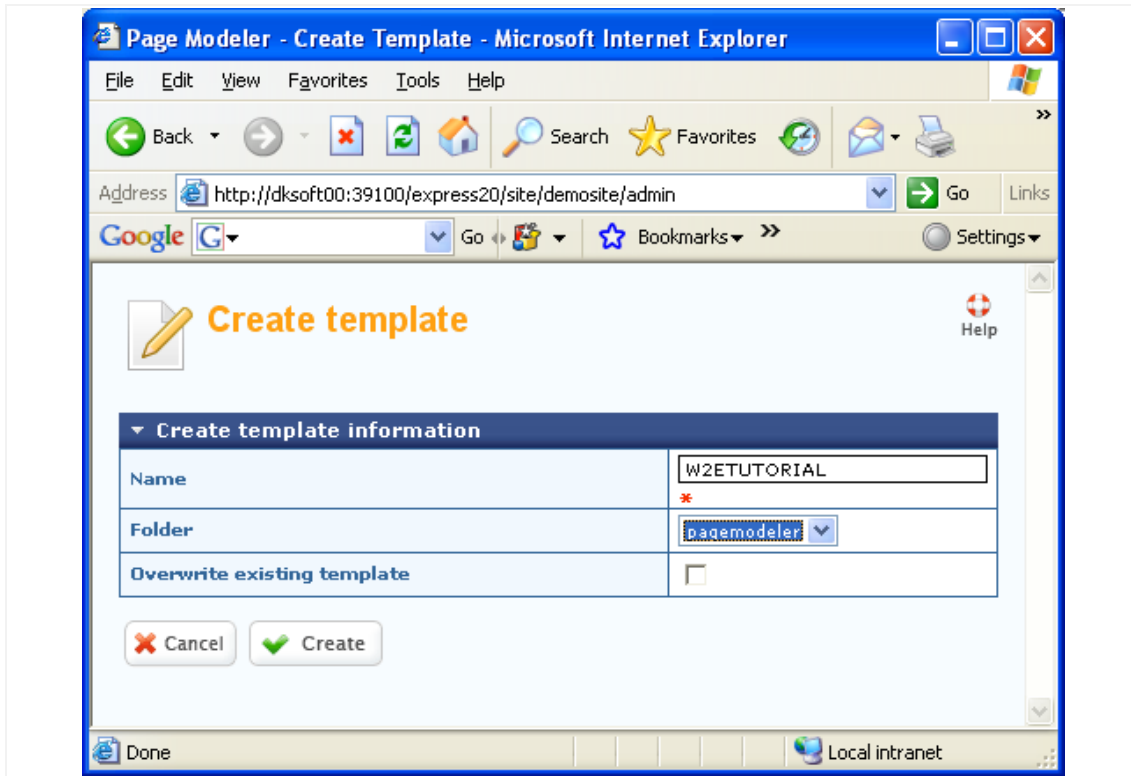
Field label: Enter text

Field Length: 50
Test Data: This is a test
Press "Finish".



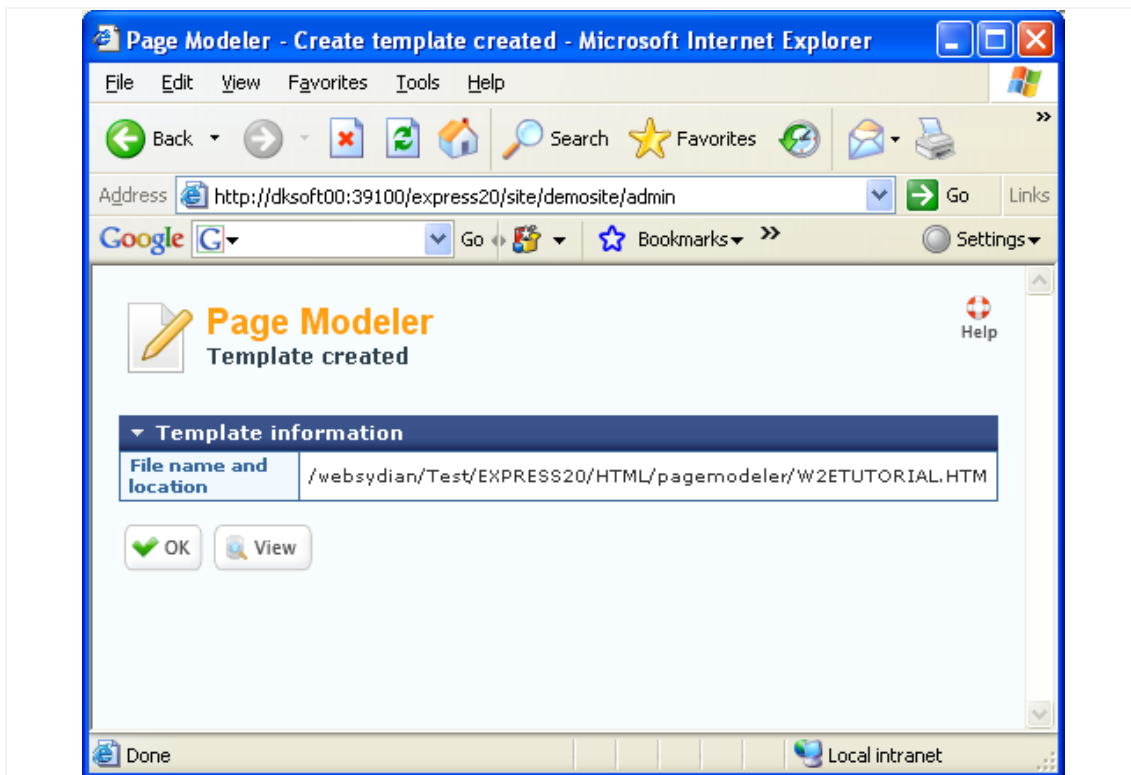
This concludes the definitions for the structure of the page. The next step is to create the template based on these definitions.

Press "Create/View" template.



On this page you can change the name of the template if you want to - and you can specify where the template should be placed.

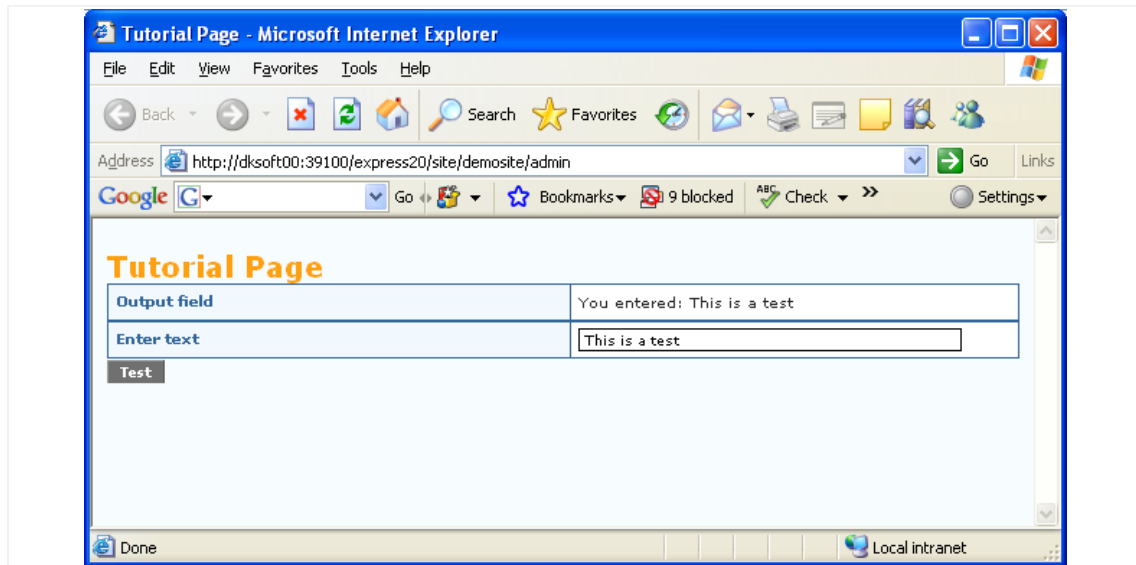
Select "pagemodeler" for Folder.
Press "Create".



The next page shows where the template has been placed in the IFS. (The path will be dependent on the choices made during the installation).

Press "View" if you want to see how the template will look, when loaded with the test data entered.

The resulting template should look like this:

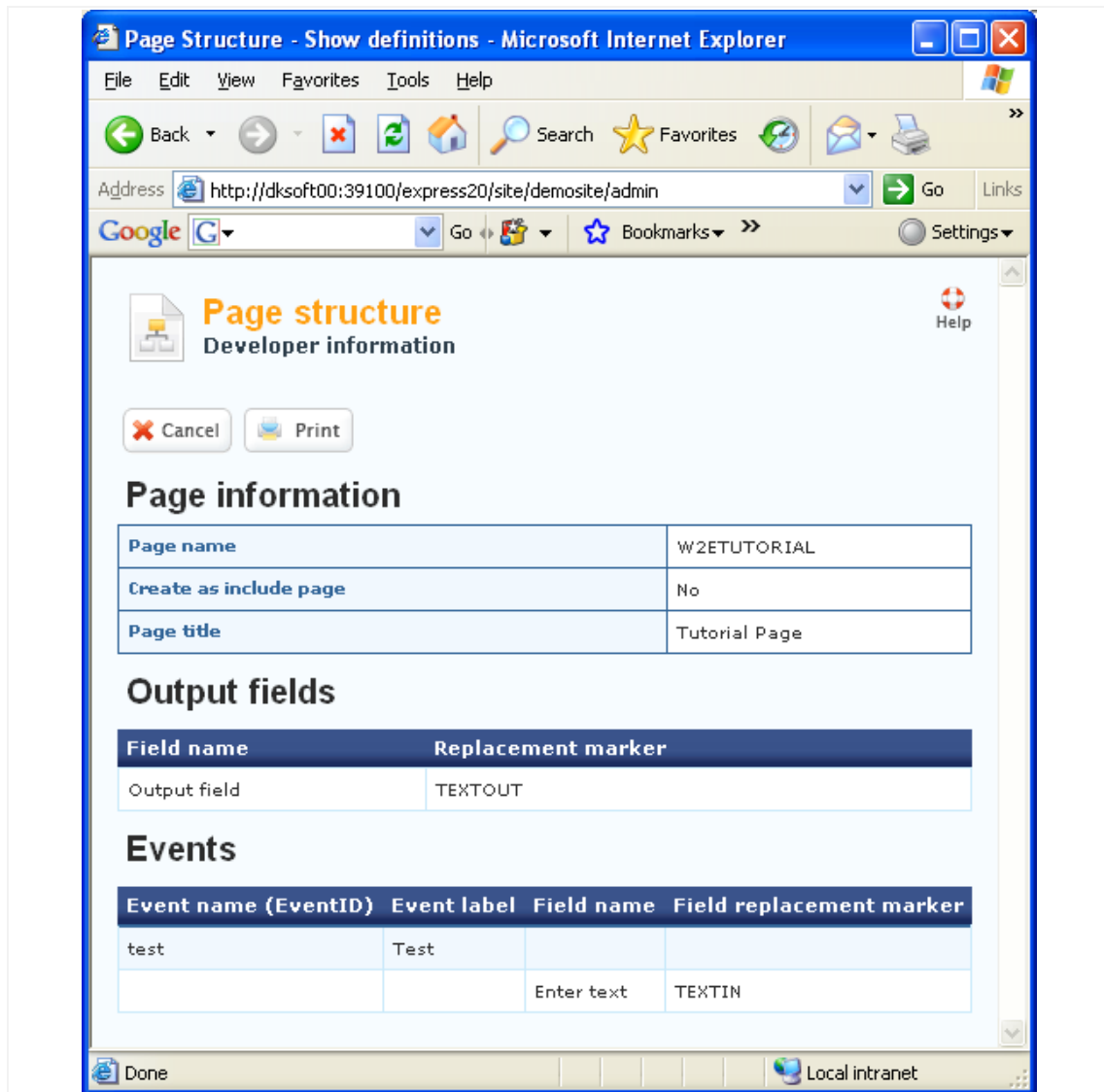


The template is created and ready to use by the application.

Before starting to code the application, it is normally very useful to print an overview of the structure.

Close the browser window showing the example Page. Press "OK" on the "Template Created" page.

Press "Developer Info" on the Page Structure Page.



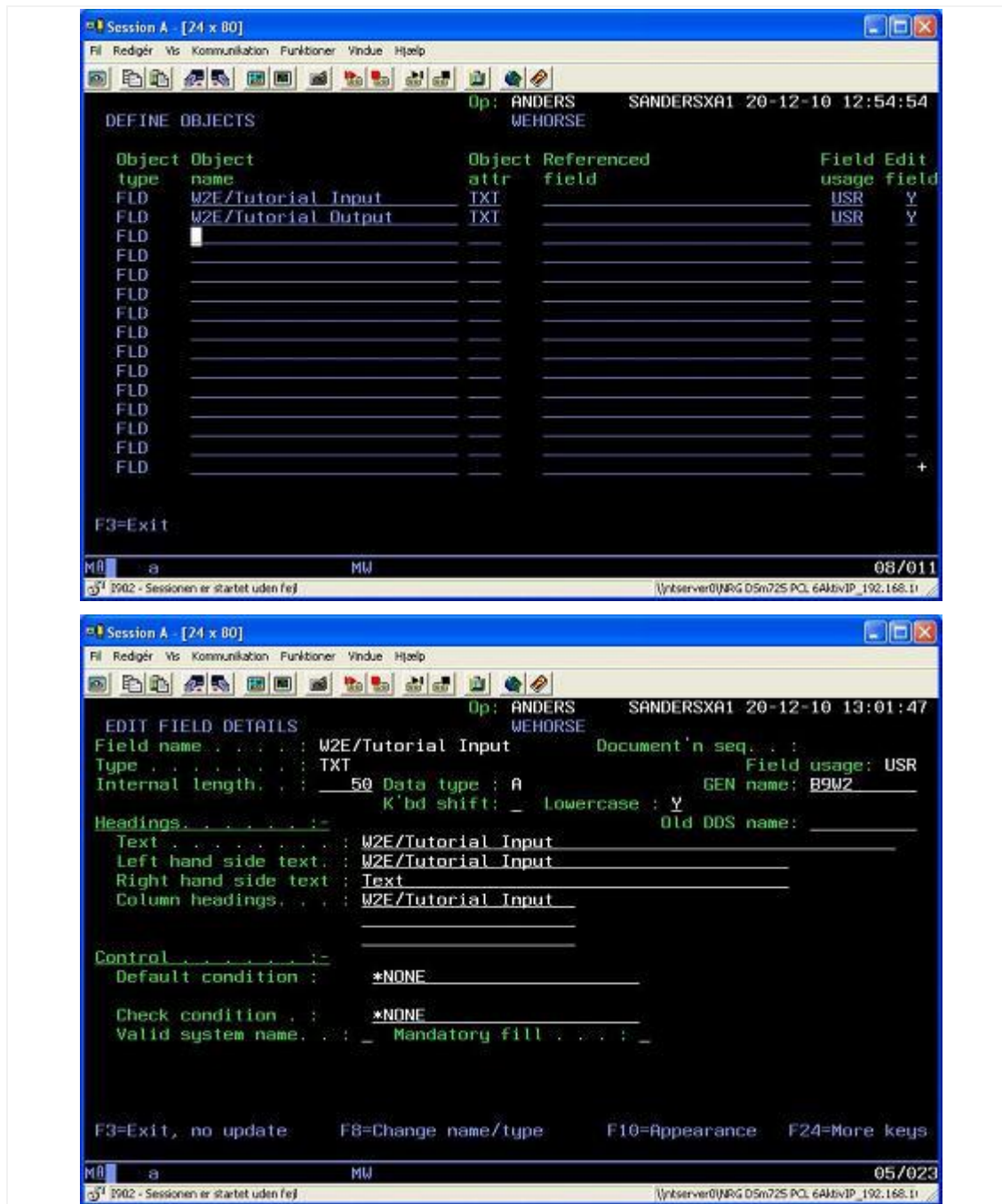
This page shows the information necessary for coding the functions.

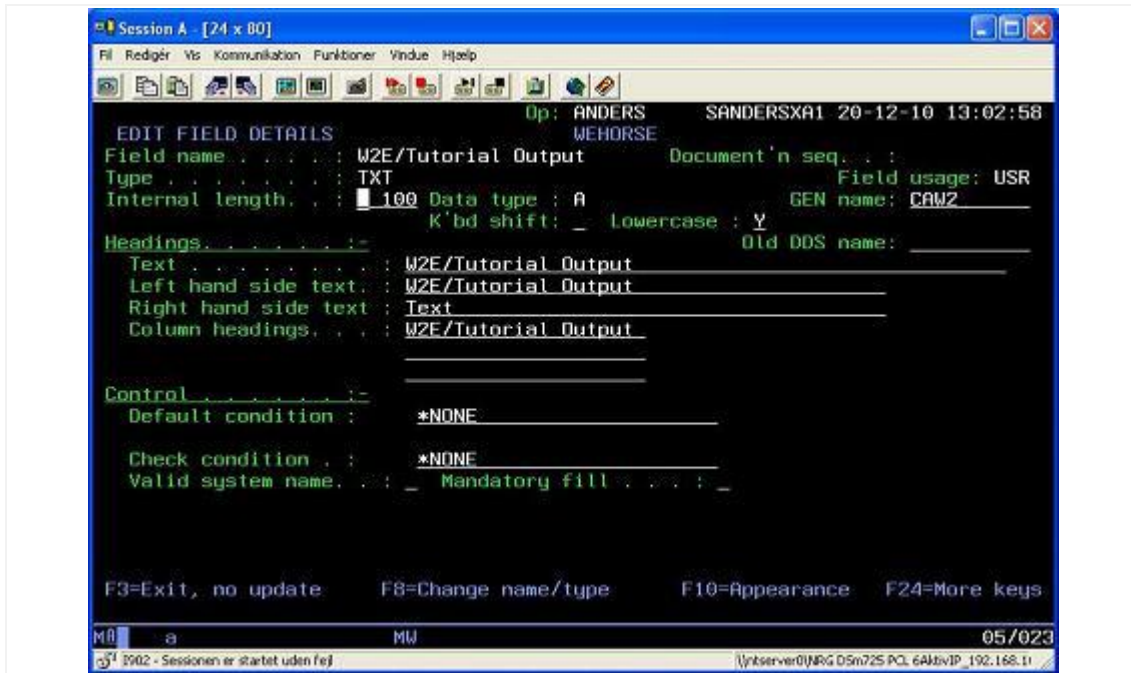
Press "Print" to print the page.

Step 6: Define and code PageGenerator "Tutorial Page"

Before creating the functions, you need to create two text fields:

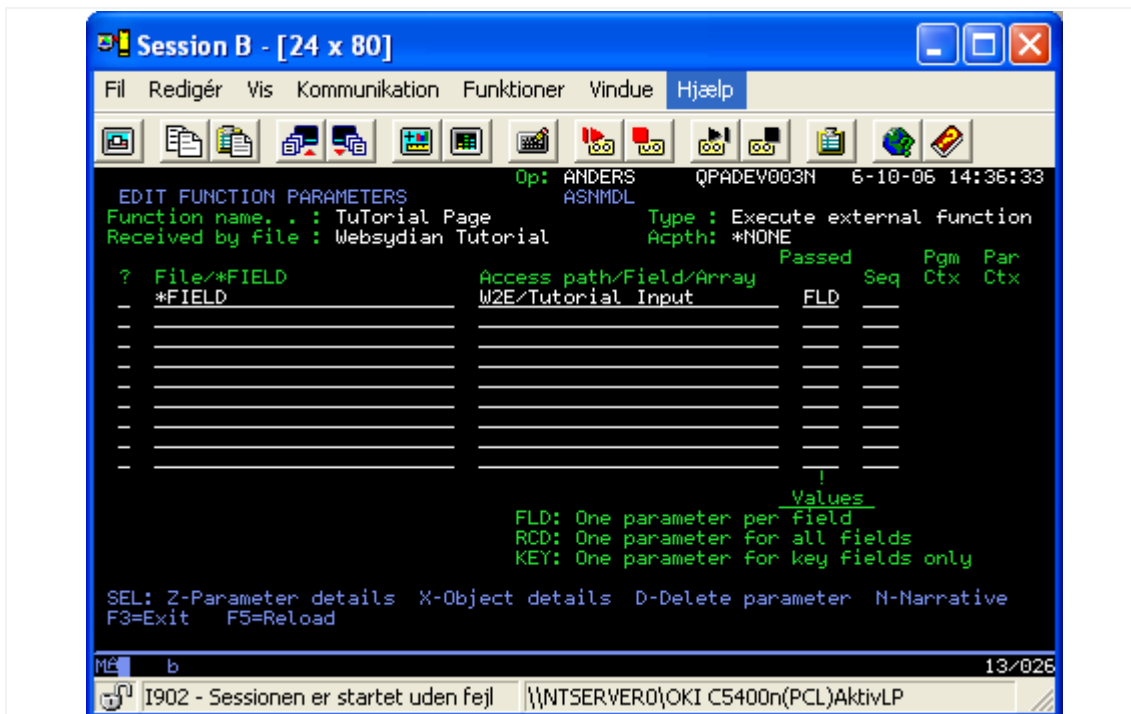
In the 2E model, press F7 to display all fields. Press F10 to define the two new fields:





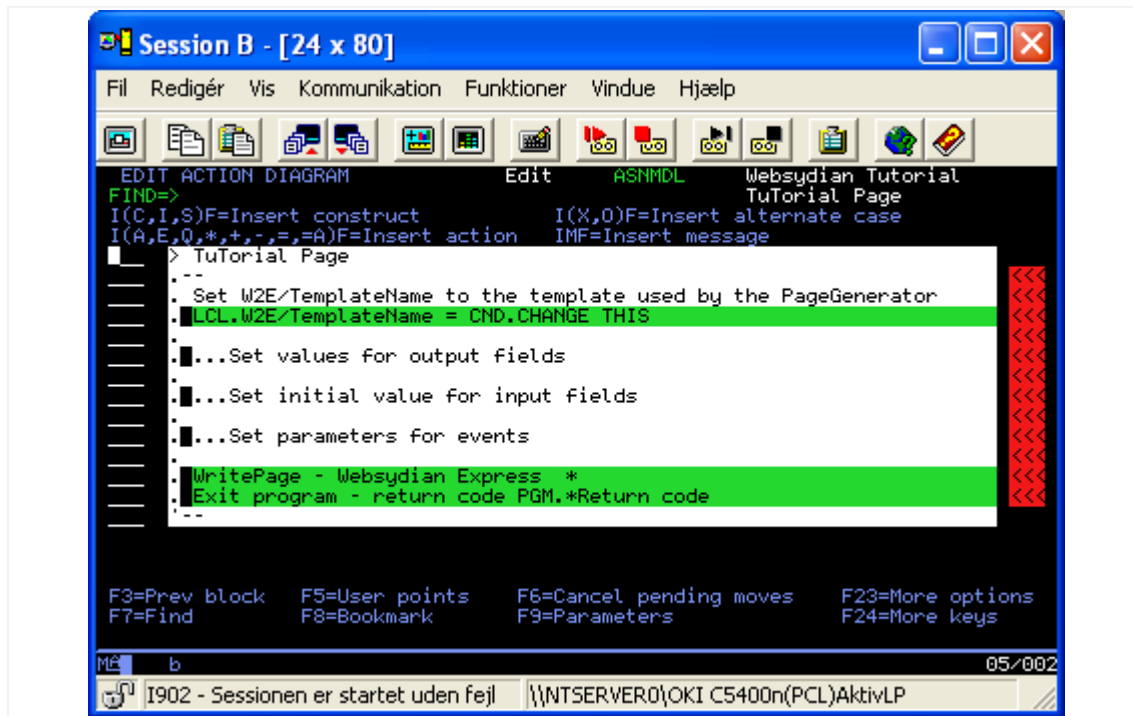
After defining the fields, specify parameters for the PageGenerator

- In the 2E model, open the Edit Functions page for the file, where you scoped the functions.
- Select Parameters for the PageGenerator "Tutorial Page".
- Add the field "W2E/Tutorial Input" as an input field.



Make action diagram code for the Pagegenerator

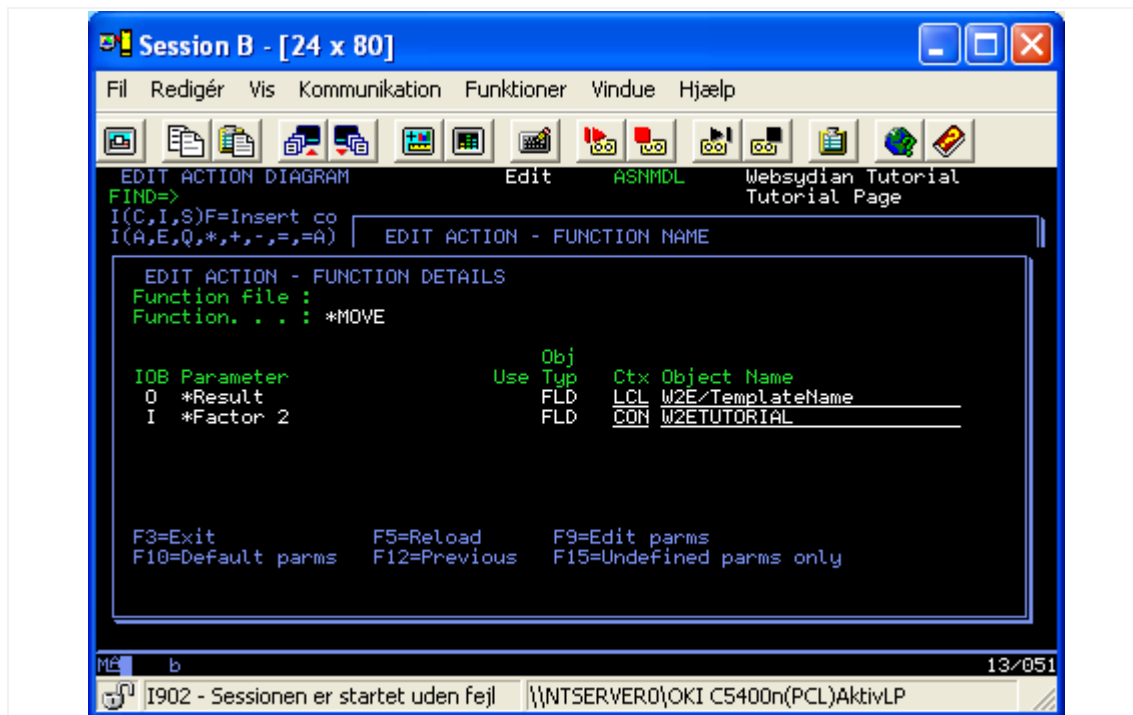
- Open the action diagram for the function Tutorial Page.



The first thing to do is to define the name of the template (as this is used by all the API functions which will be called). The template name must be the one used in the page modeler.

On the printed Developer info page, you can find this information as the "Page name" in the section "Page information".

Change the value assigned to LCL.W2E/TemplateName from CND.CHANGE THIS to CON.W2ETUTORIAL.



As indicated by the predefined sequences, there are basically three things to do in a PageGenerator function:

1. Set the values for output fields

In the tutorial, the value for the "Tutorial Output" field must be specified.

2. Set the initial values for the input fields

In most cases you want to be able to specify a value for the input fields. These values might for example be default values or previously entered values for the input fields.

In the tutorial, we will specify the previously entered value as the initial value for the "Tutorial Input" field.

3. Specify parameters for events

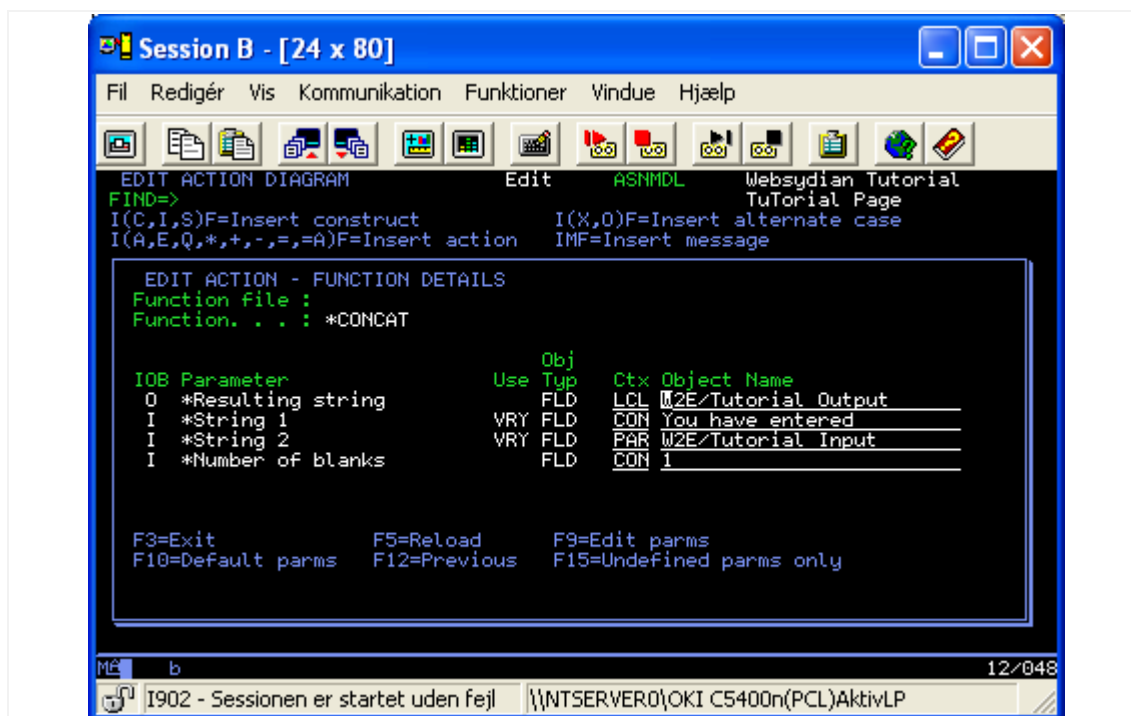
This is done if the PageGenerator has information that has to be transferred to the EventHandler, without giving the user the opportunity to change the value.

The tutorial does not include such parameters.

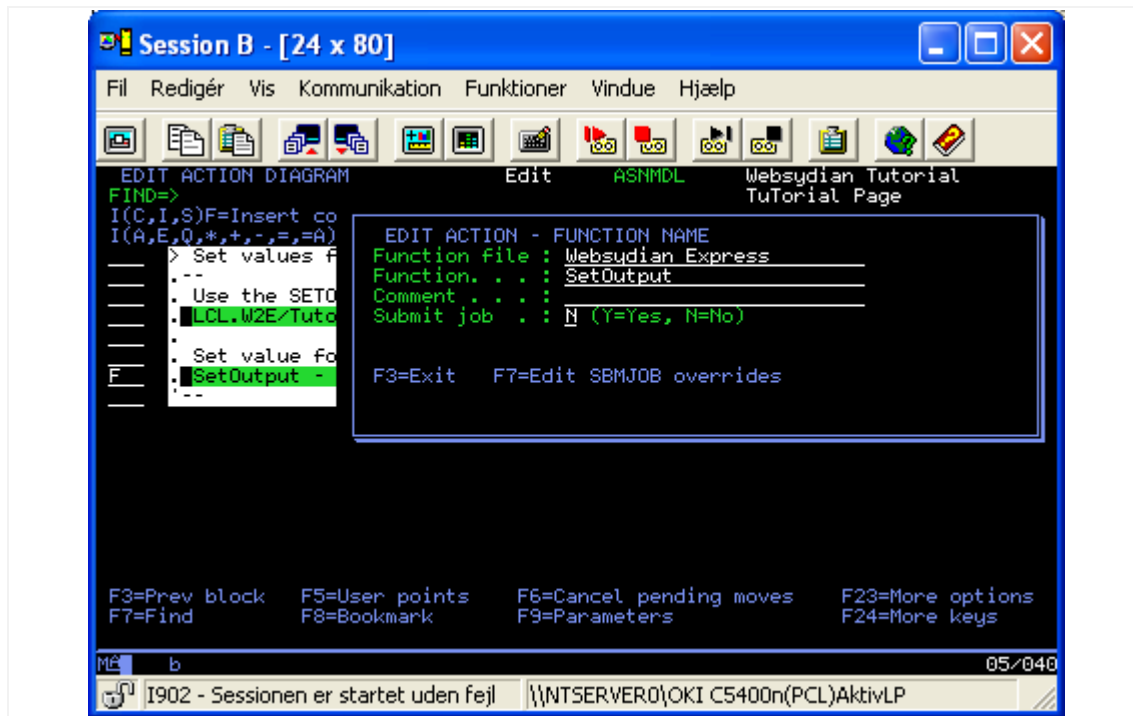
Set values for Output fields

Before setting the value for the output field, the value has to be determined. The value will be the input value, preceded by "You have entered: ".

In the sequence "Set values for output fields", use the built in function *CONCAT to set the value of LCL.W2E/Tutorial Output to "You have entered" concatenated with the value of PAR.W2E/Tutorial Input.



After the concatenation, enter a call to the function SetOutput scoped by the file "WebsydianExpress".



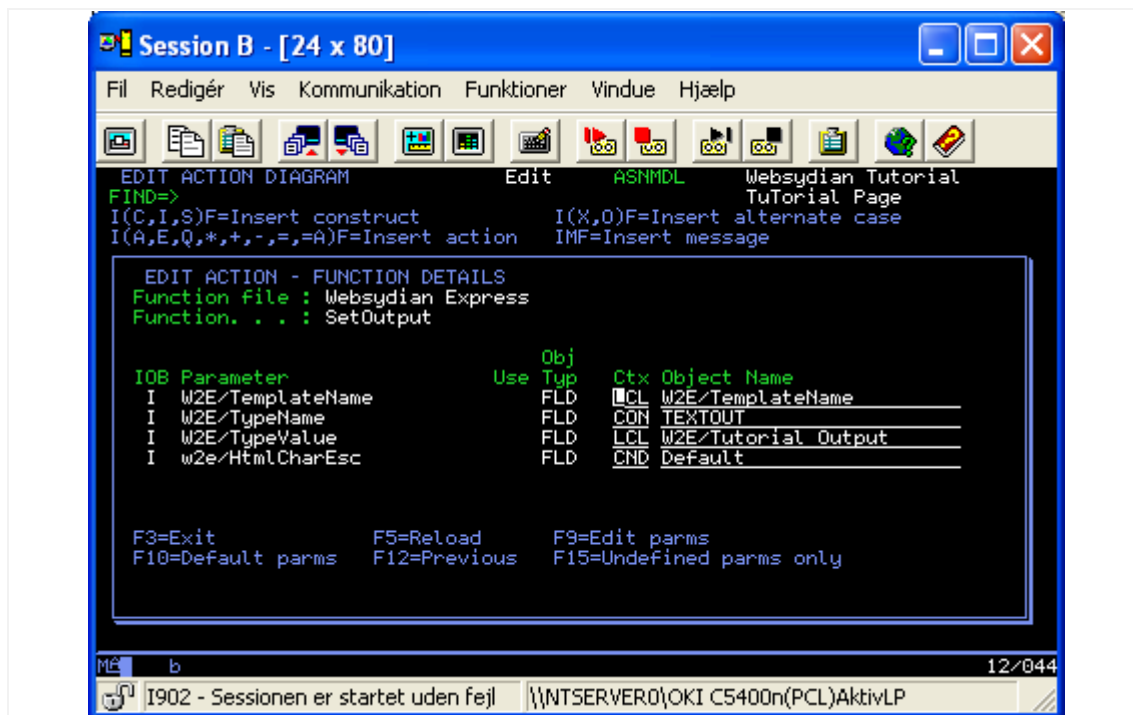
Set the parameters to:

LCL.W2E/TemplateName

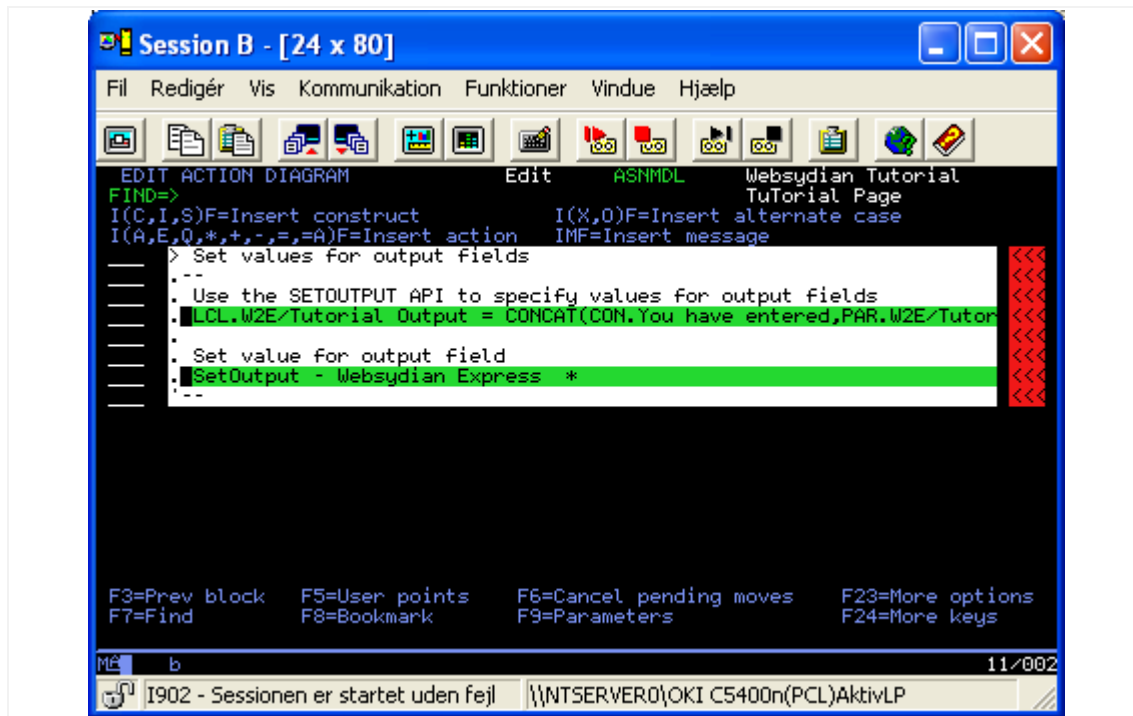
CON.TEXTOUT

WRK.Tutorial Output

CND.*Blank



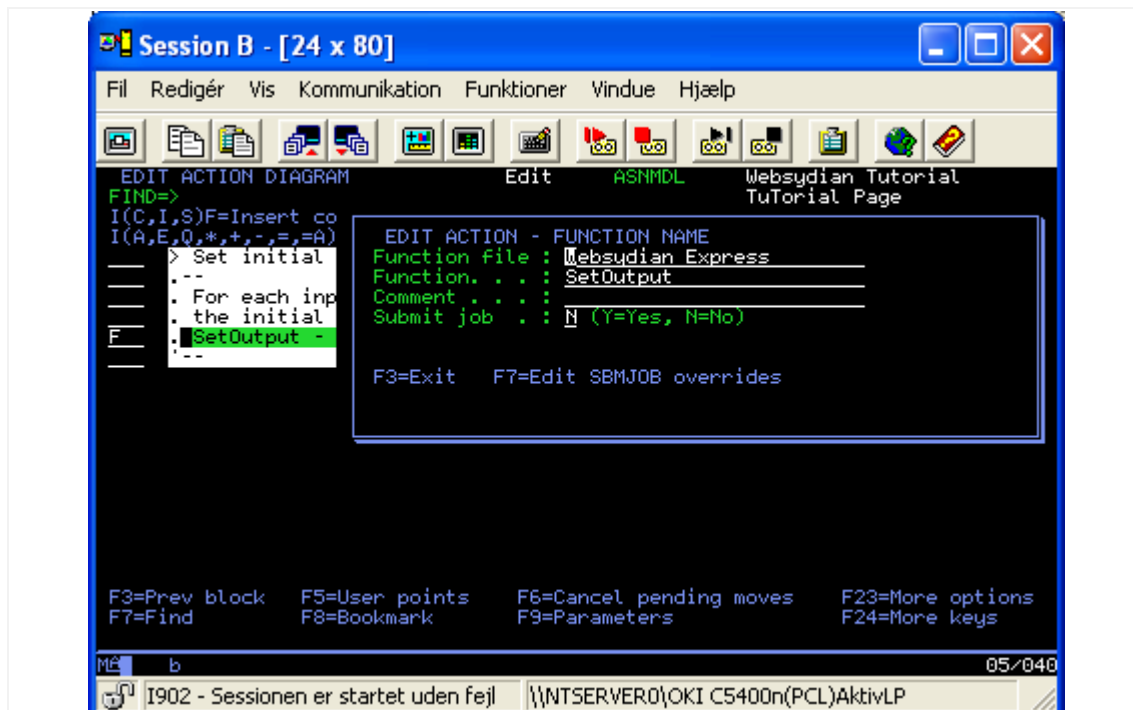
Having done this, you will have the following content of the sequence:



Set initial value for input fields

The value entered by the user should be shown again in the input field.

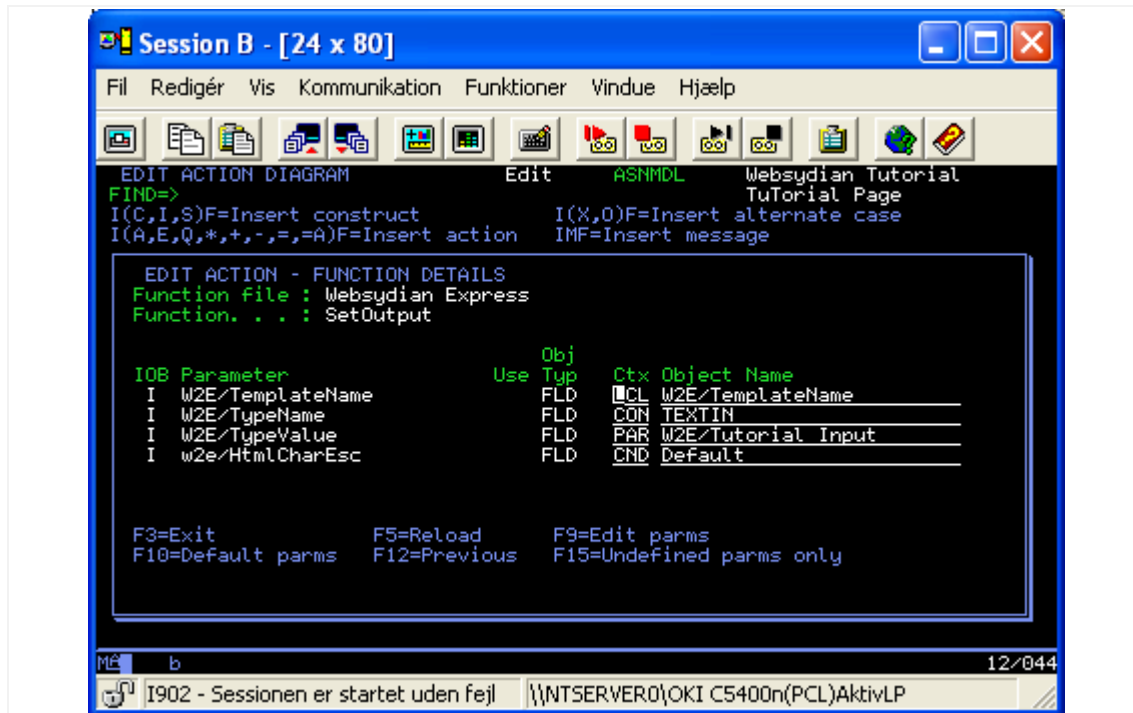
In the sequence "Set initial value for input fields" enter a call to the function SetOutput scoped by the file "WebsydianExpress".



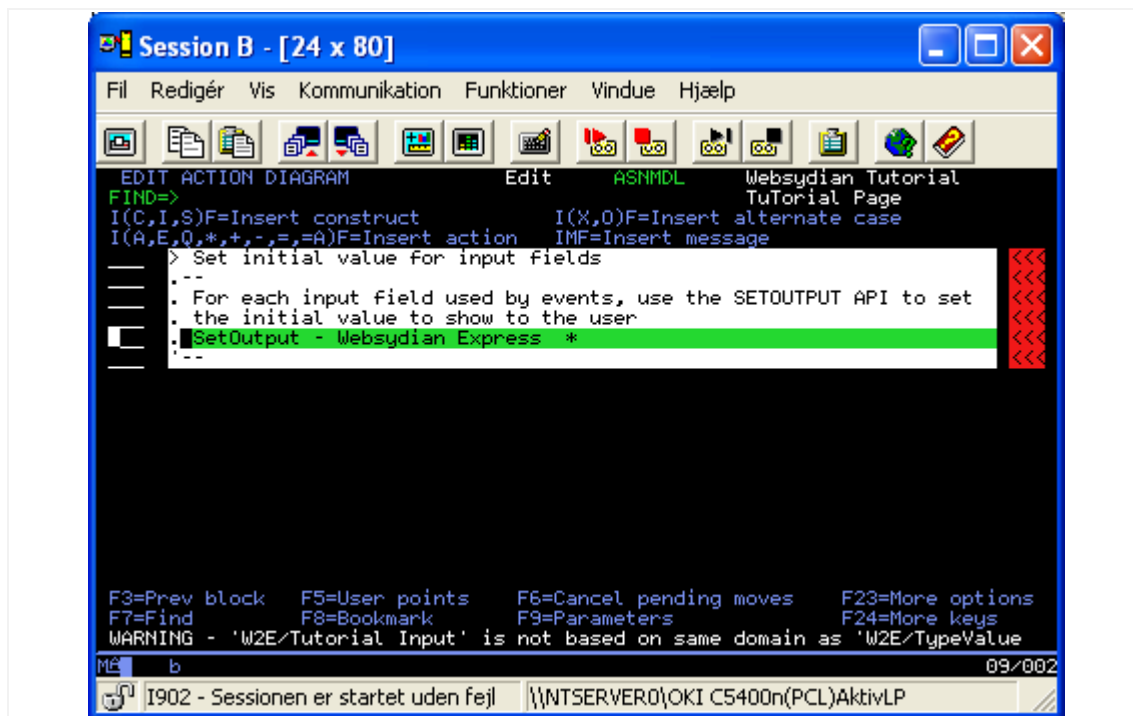
Set the parameters to:
 LCLW2E/TemplateName
 CON.TEXTIN

PAR.Tutorial Input

CND.*Blank



Having done this, you will have the following content of the sequence:



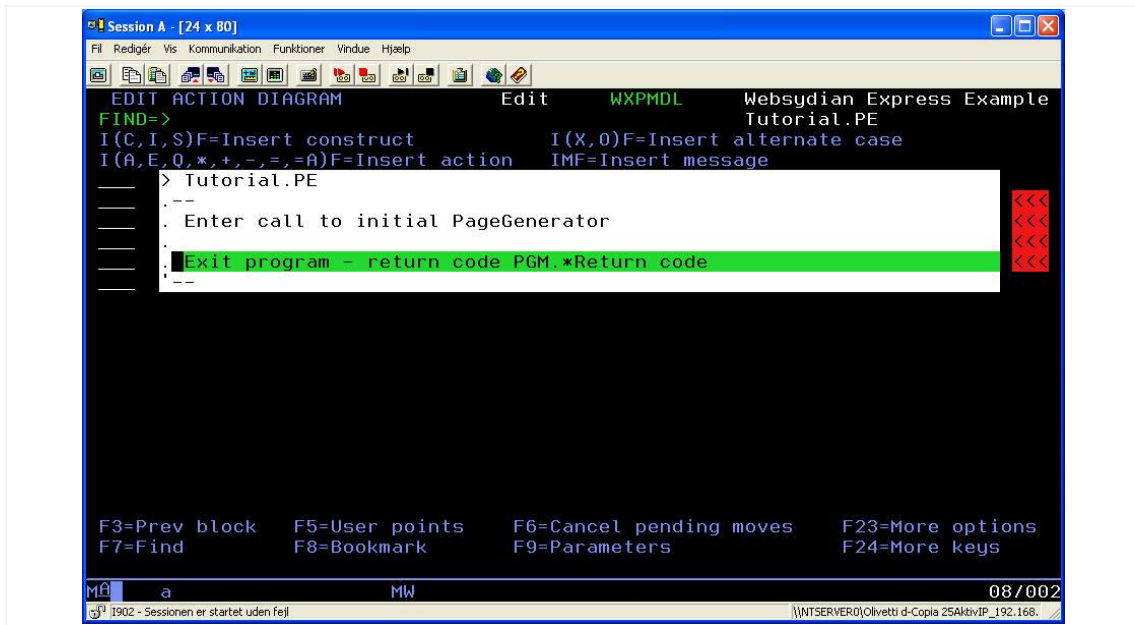
This concludes the coding of the PageGenerator.

Step 7: Define and code ProcessEntryPoint Tutorial.PE

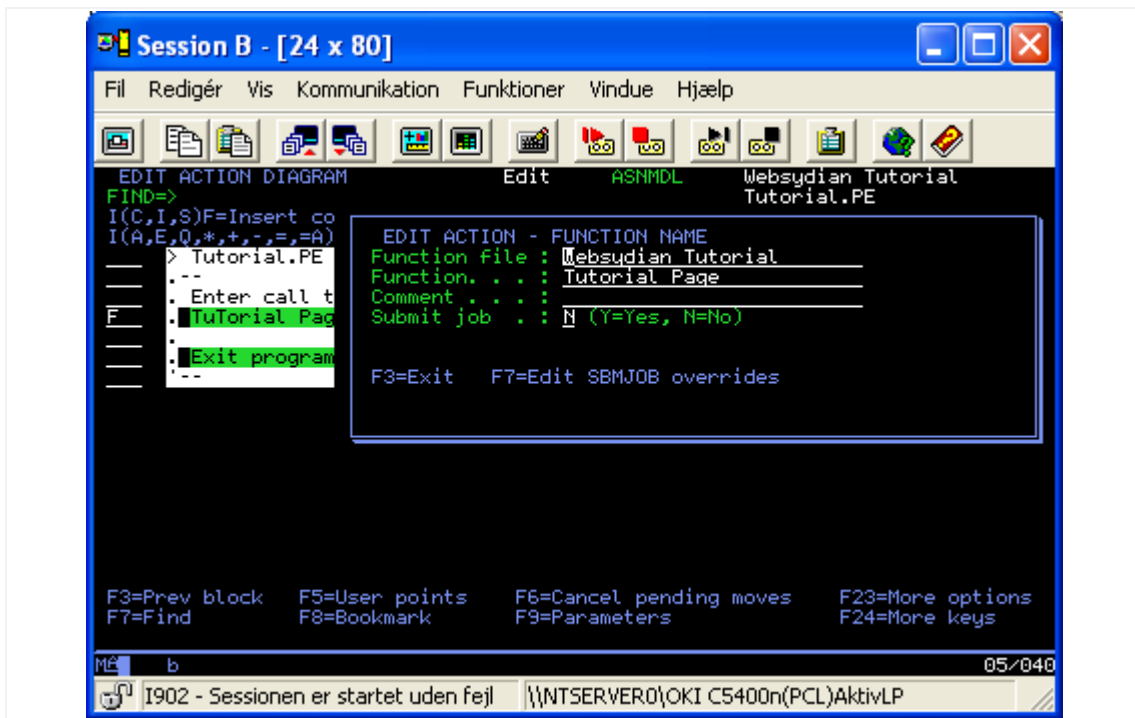
As the ProcessEntryPoint functions are called dynamically by the runtime, you are not allowed to define any parameters for these functions.

In most cases the ProcessEntryPoint will just call the first PageGenerator of the business process.

Open the action diagram for the Tutorial.PE function.

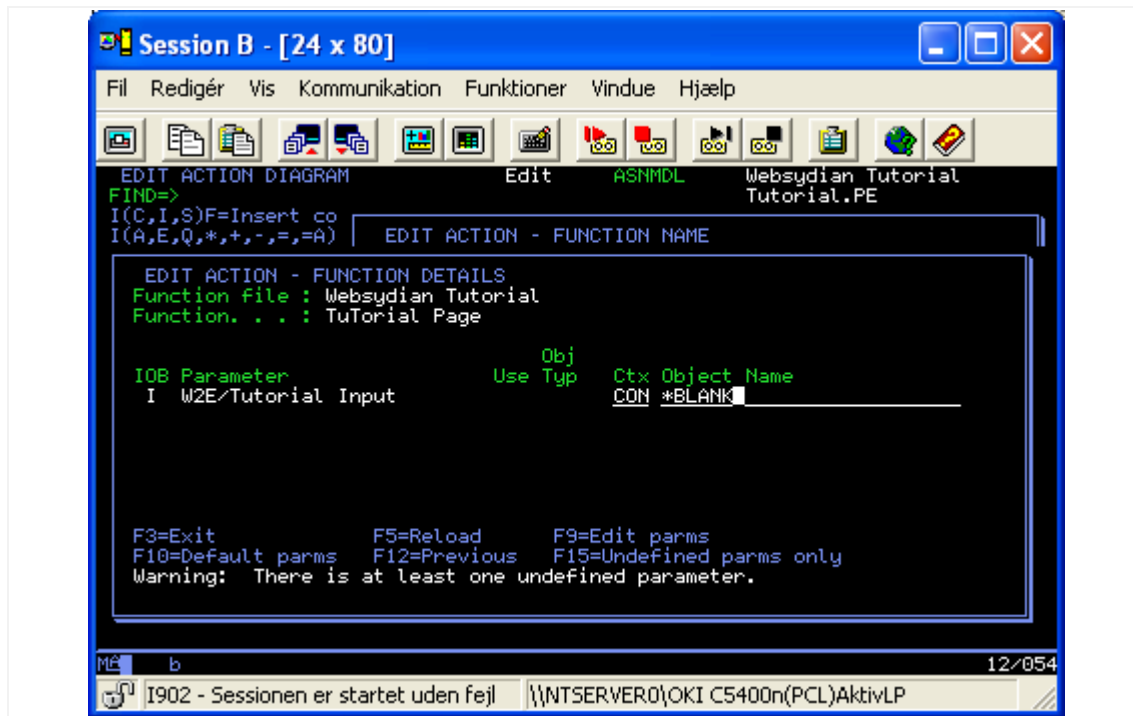


Right after the comment "Enter call to initial PageGenerator", enter a call to the PageGenerator "Tutorial Page".

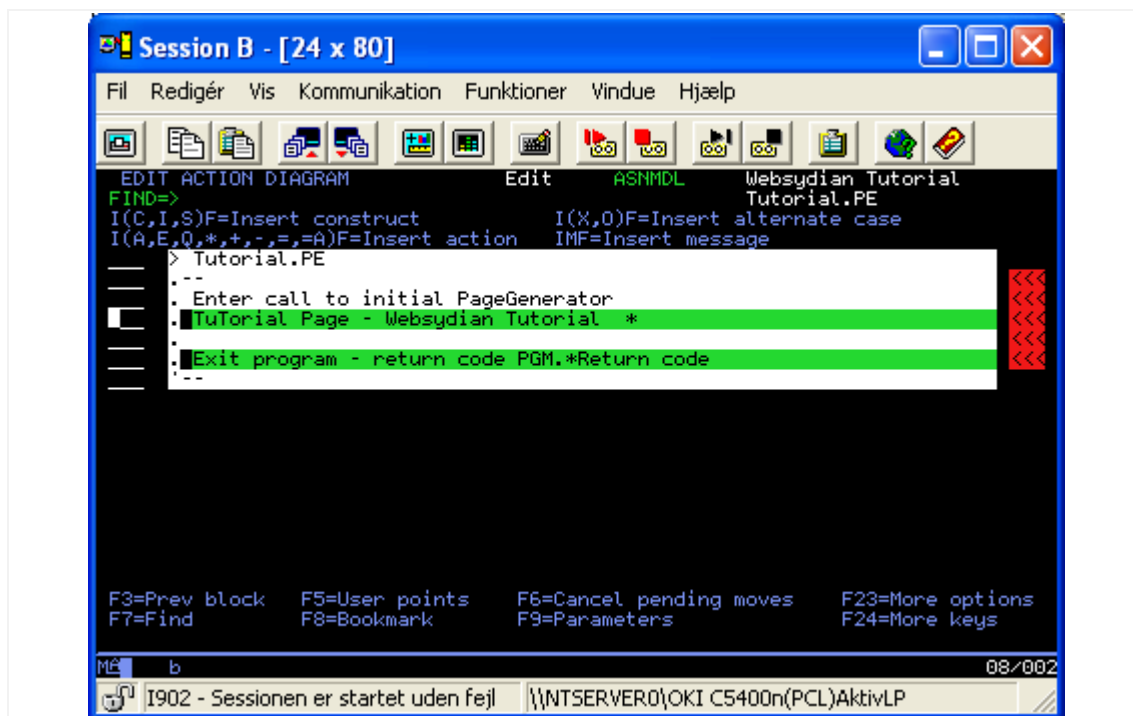


Instead of "Websydian Tutorial" - specify the file you have scoped your own PageGenerator under.

Specify CON.*BLANK for the parameter.



This will give you the following content of the ProcessEntryPoint:

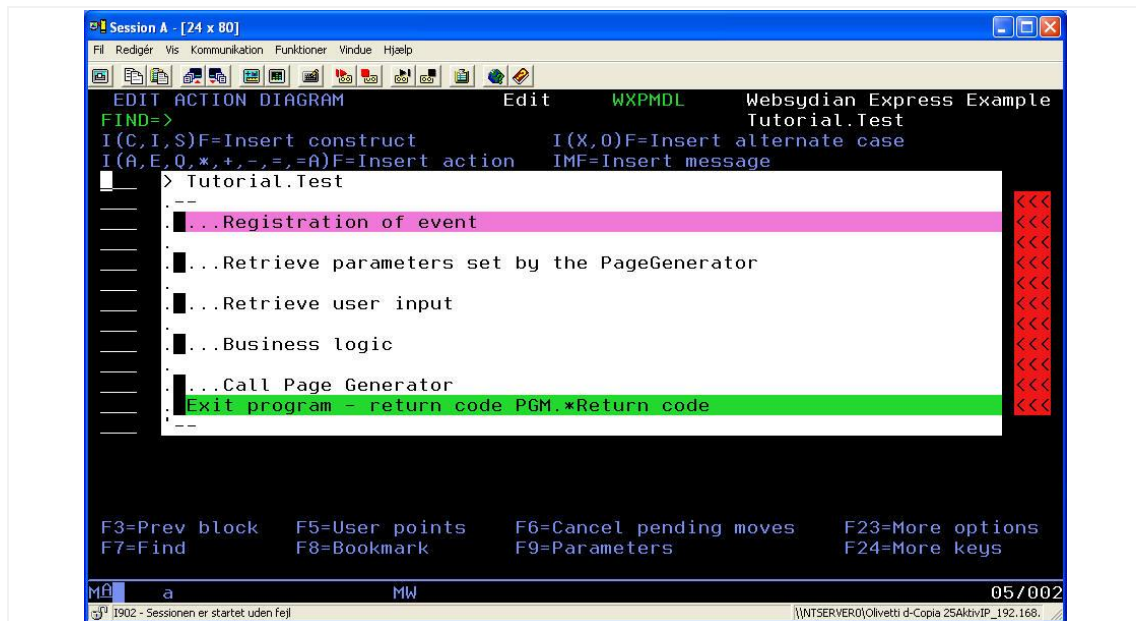


This concludes the coding of the ProcessEntryPoint.

Step 8: Define and code EventHandler Tutorial.Test

The EventHandlers are called dynamically by the runtime. This means that you are not allowed to change the parameters for EventHandler functions.

Open the action diagram for the function Tutorial.Test.



As indicated by the predefined sequences, there are 4 steps that can be relevant for an EventHandler:

1. Retrieve parameters set by the PageGenerator

If any parameters have been set for the event by the PageGenerator, the values of these parameters must be fetched.

In the tutorial, no parameters are transferred.

2. Retrieve User input

The data entered by the user for the event must be retrieved, so that they are available for the program.

In the tutorial, the value for the input field "Tutorial Input" is retrieved.

3. Business logic

In most cases, activating an event should lead to some kind of functionality being executed (E.g. read/update data base, make calculations).

In the tutorial, no business logic is executed.

4. Call Page Generator

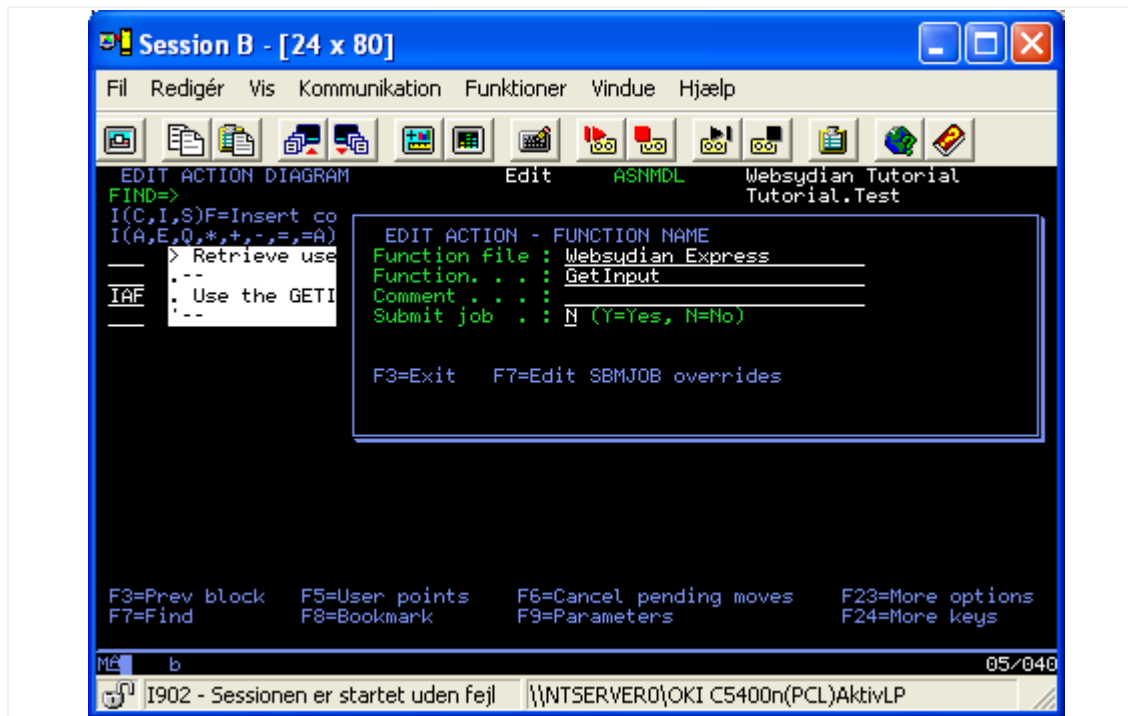
All EventHandlers must end by calling a PageGenerator function. If this is not done, no reply will be sent to the browser, and the browser will be left blank.

In the tutorial, the PageGenerator "Tutorial Page" is called.

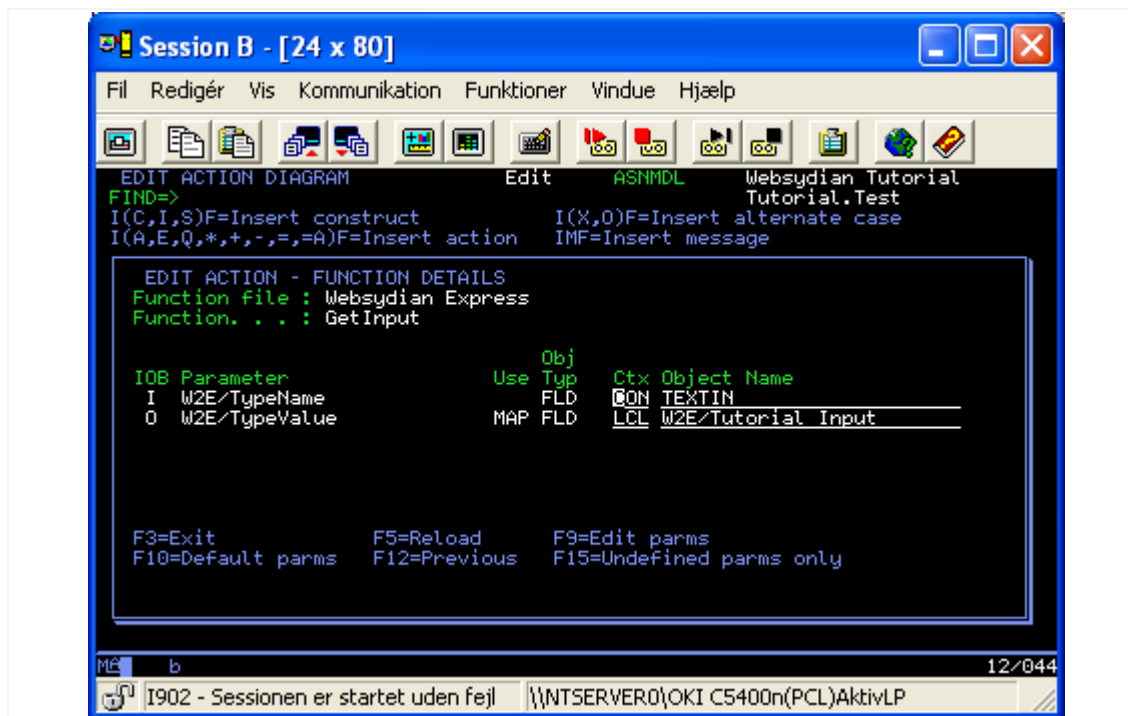
For all EventHandlers the case structure "Registration of event" must just be left as it is - and it must be the first step in the EventHandler.

Retrieve User Input

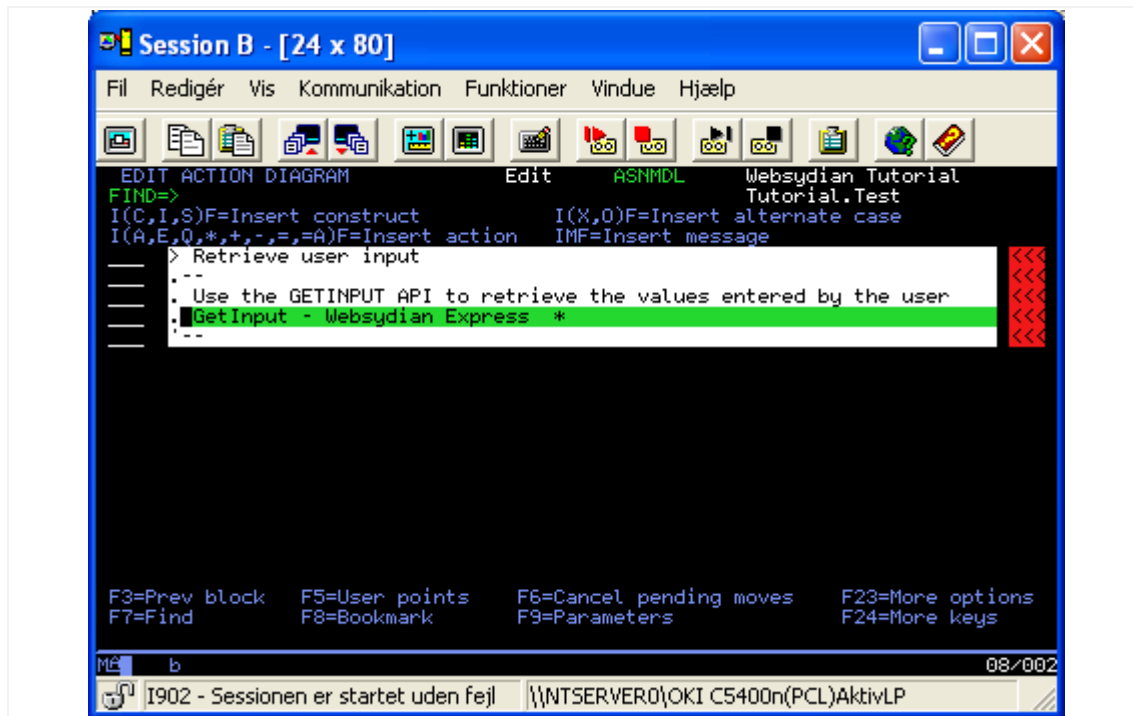
In the sequence "Retrieve user input", add a call to the function GetInput scoped under the file WebsydianExpress.



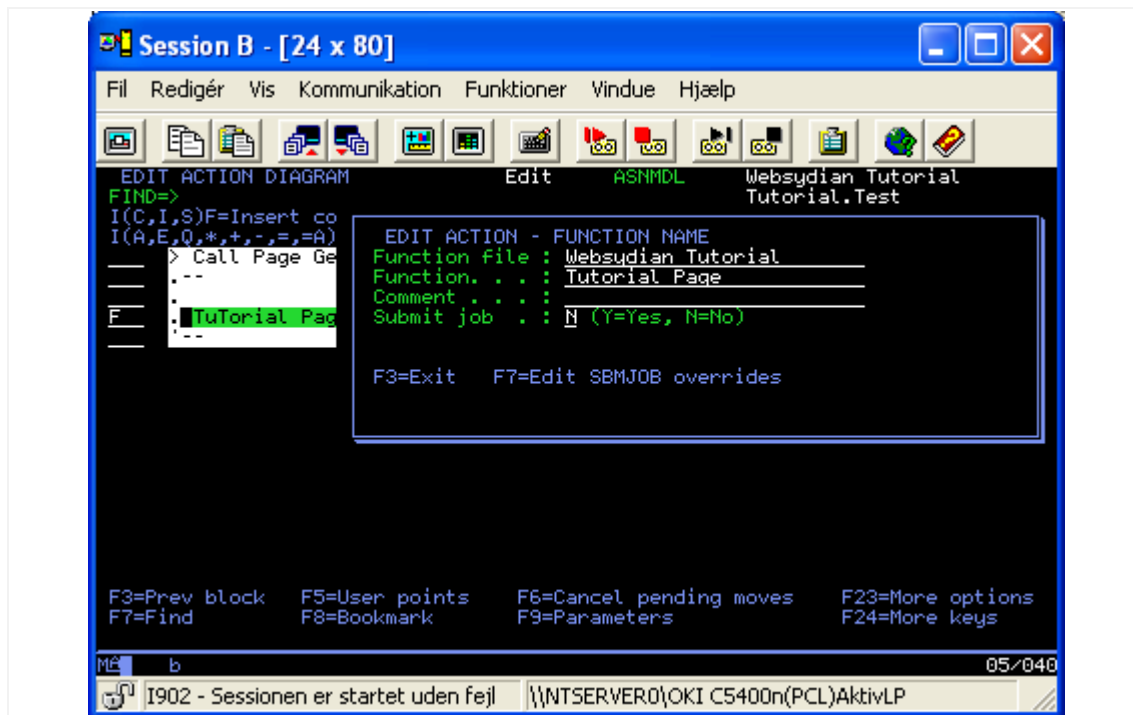
Use the parameters:
 CON.TEXTIN
 LCL.W2E/Tutorial Input



This will be the content of the sequence:

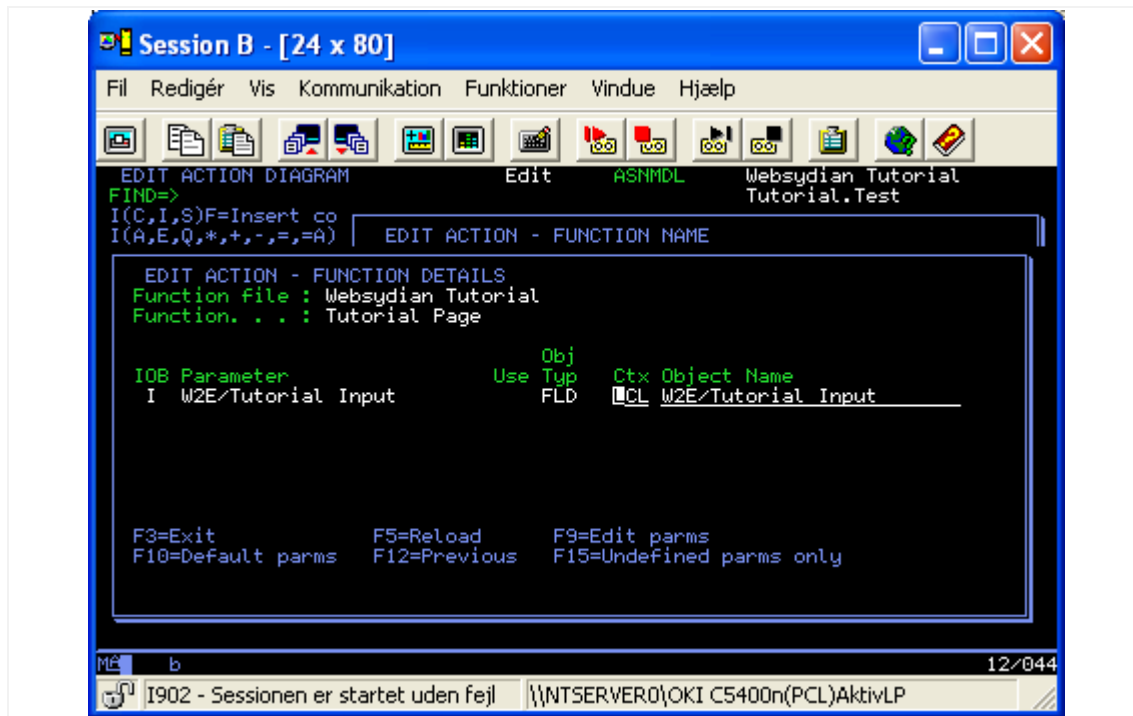


In the sequence "Call Page Generator" enter a call to the PageGenerator "Tutorial Page".

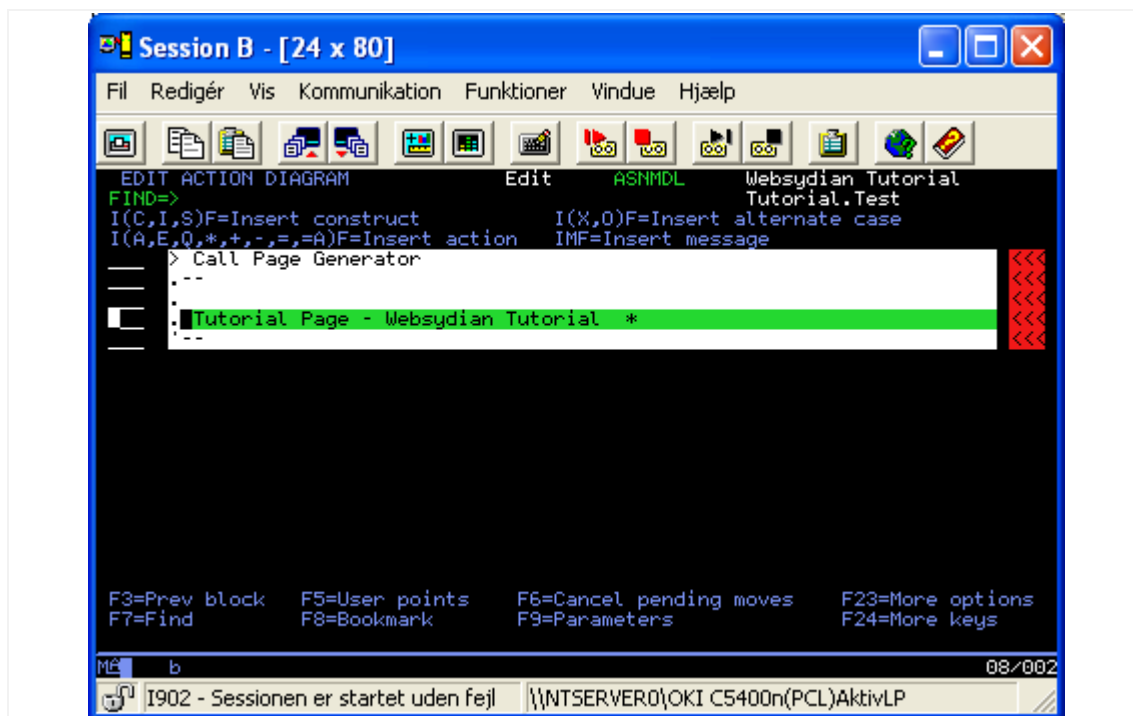


Enter your own file instead of WebsydianExpress example.

Specify the parameter:
LCL.Tutorial Input



This will be the content of the sequence:



This concludes the coding of the EventHandler.

Step 9: Generate and build functions

Generate and build the functions:

- Tutorial Page
- Tutorial.PE
- Tutorial.Test

Step 10: Deploy objects

Instead of moving the objects to e.g. WXP30APP, during development, it is much easier to have your generation library as part of the library list used by the jobs servicing the application.

The 2E runtime must also be added to the library list for the jobs.

Add the generation library to the library list of the job description EXPRESS (found in the database library - default WXP30DB).

Add the 2E runtime to the library list of the job description EXPRESS.

To make the application service jobs use the changed library list, restart the application services using the command RSTAS in the program library (WXP30PGM).

Step 11: Define Process in the administration interface

The development is done - the last part is to make the developed function available in the installed site.

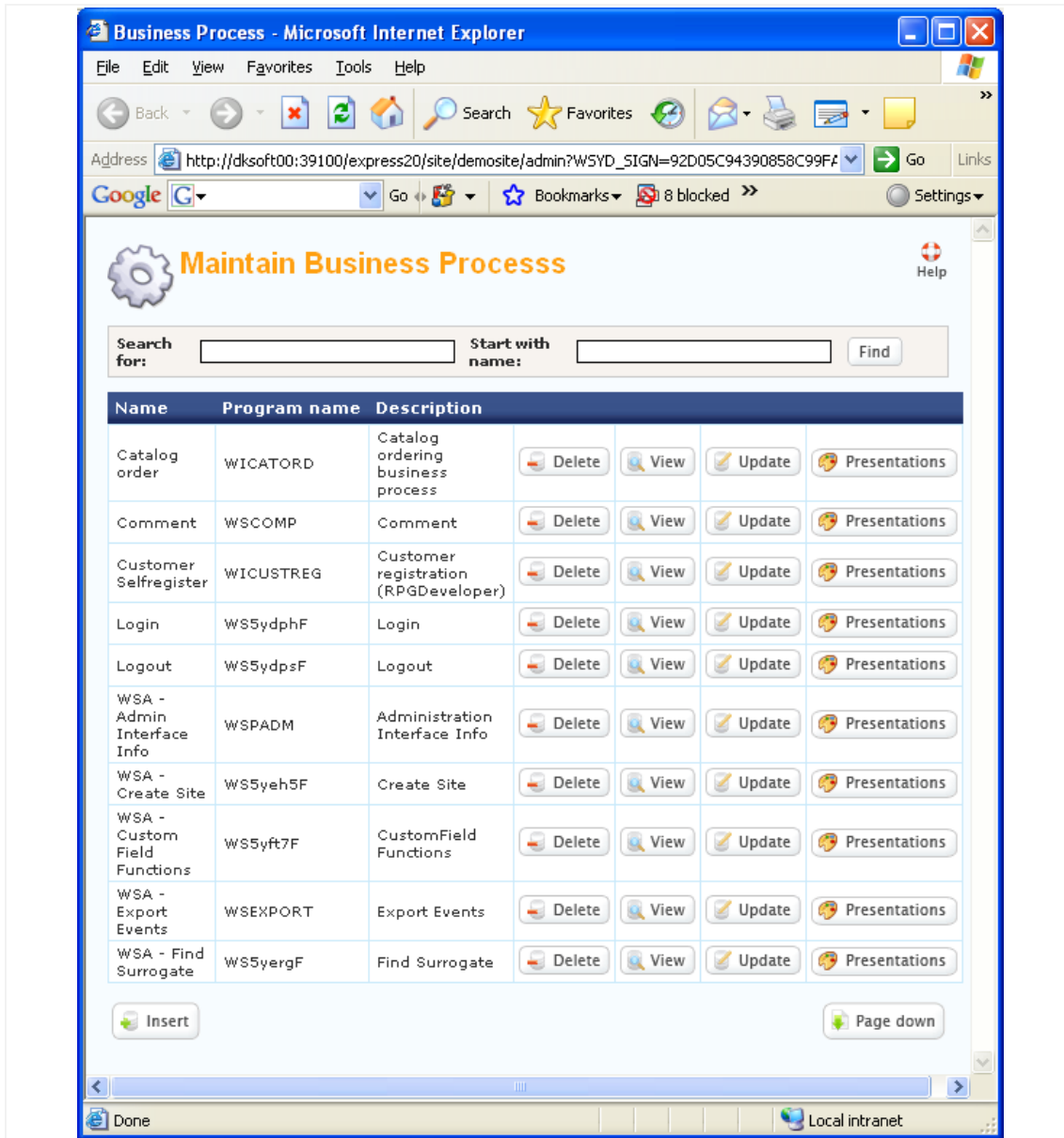
This is done using the administration interface for the site.

Log in to the installed demo site using an administrator profile (default user: WSADMIN, password: ADMIN).

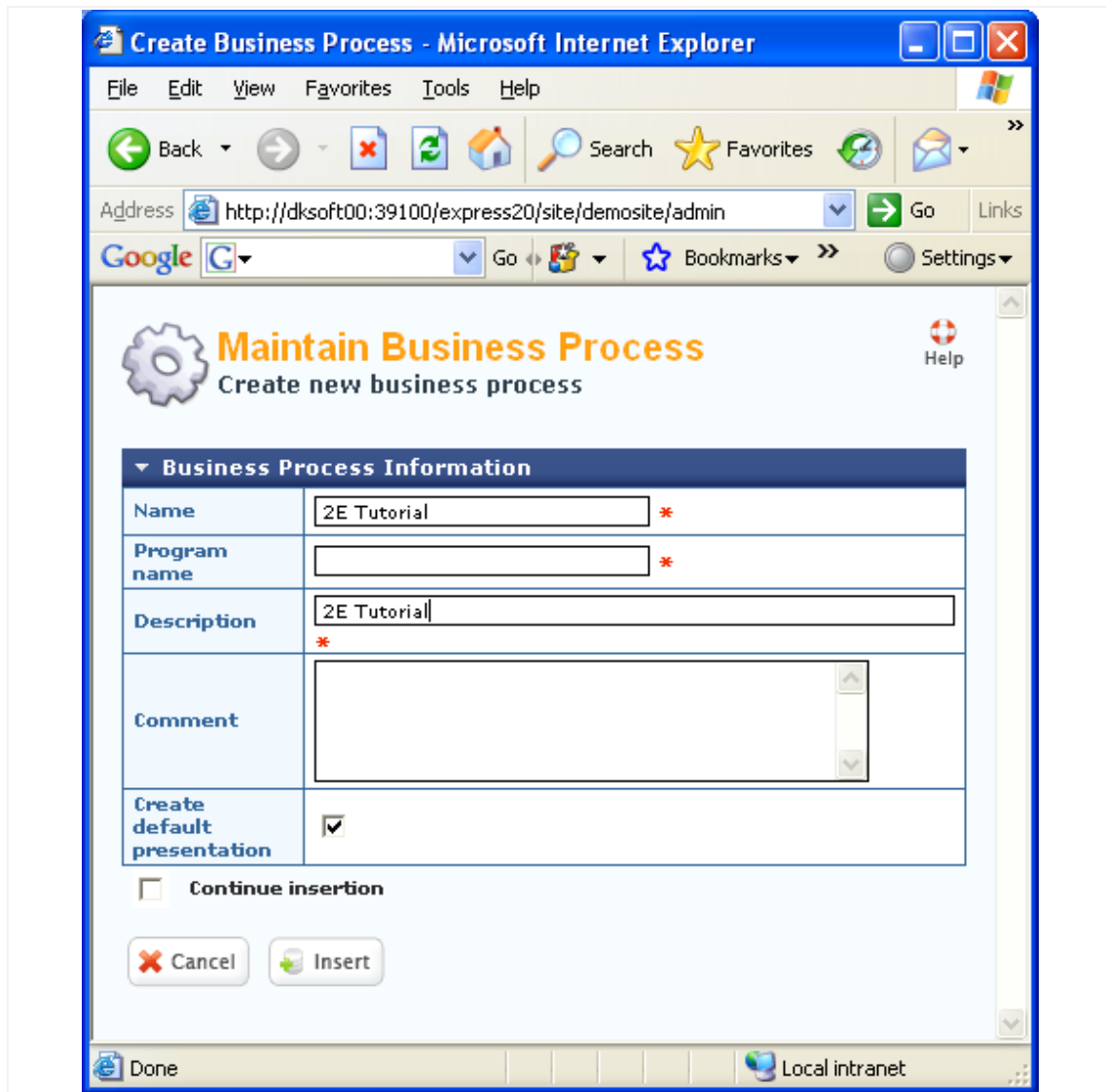
Press the "Admin" menu item in the top menu.

Define business process

Select "Content Loaders→Business Processes".



Press "Insert".



Enter:

Name: 2E Tutorial

Process ID: Enter the source name of the ProcessEntryPoint function Tutorial.PE.

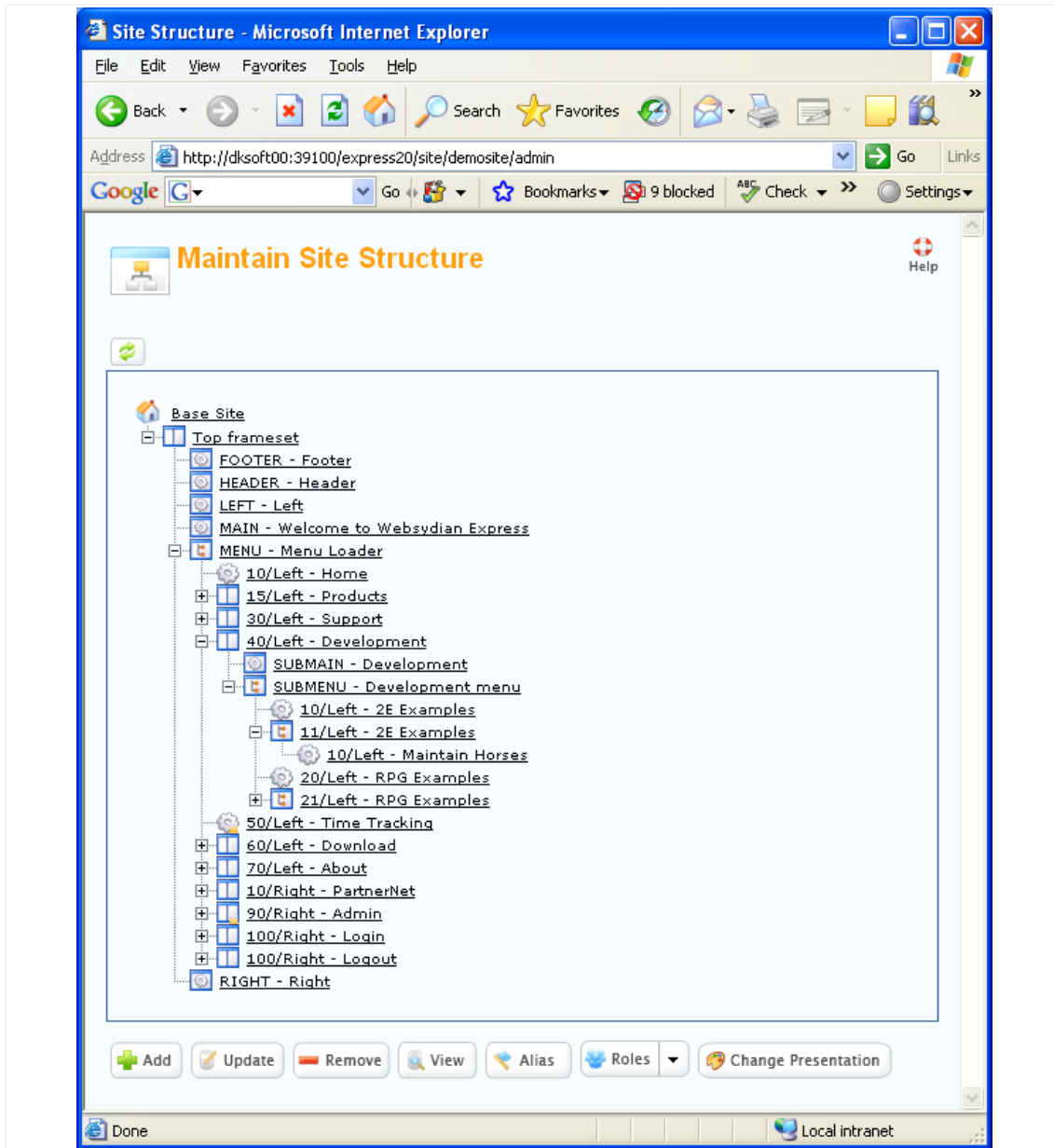
Description: 2E Tutorial

Keep the "Create default presentation" checkbox checked.

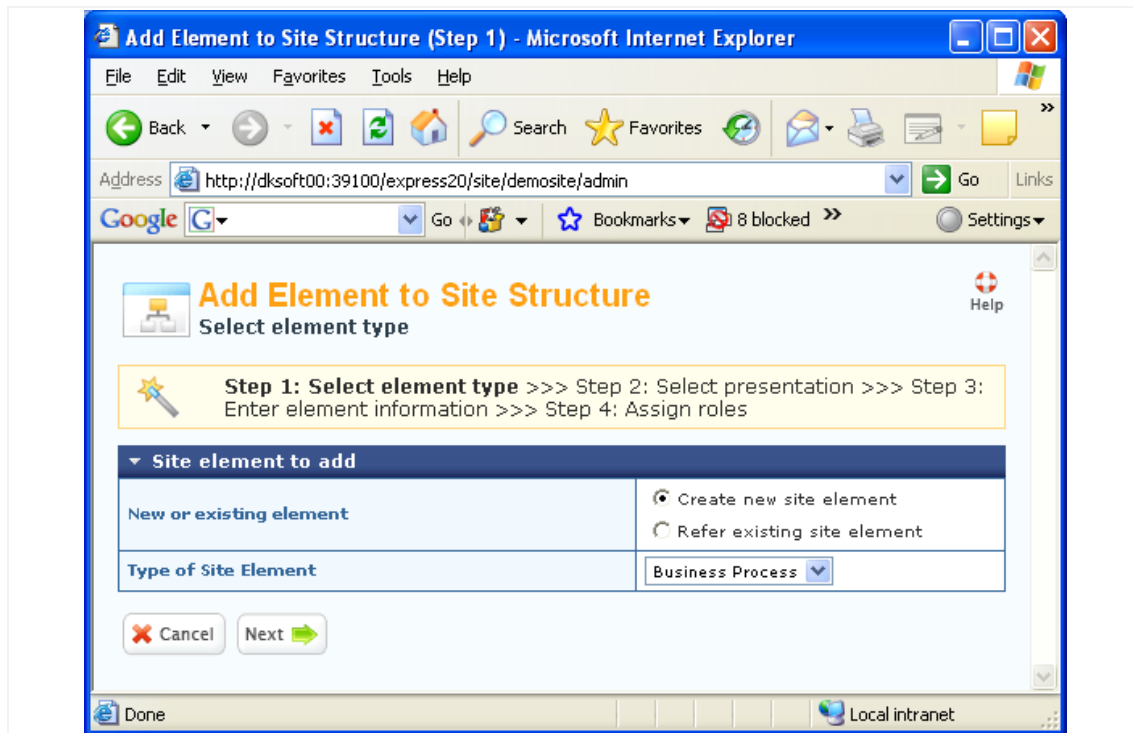
Press "Insert".

Add business process to site structure

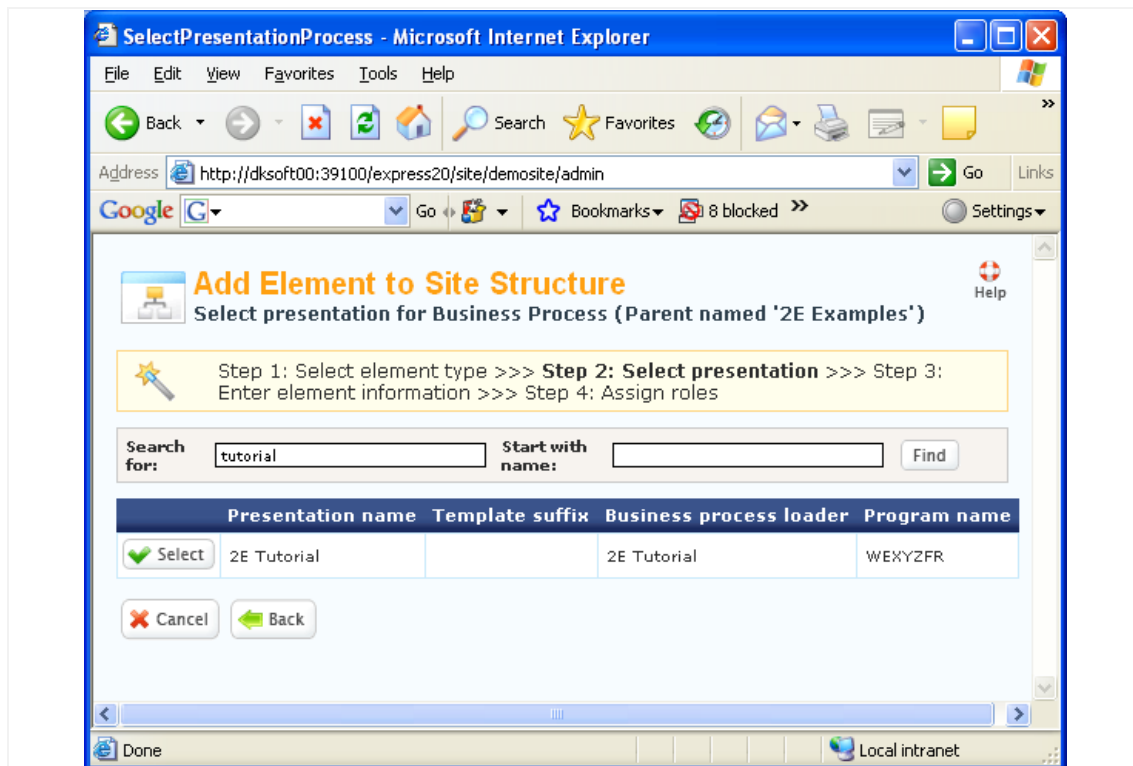
Select the menu item "Site Structure→Site Structure".



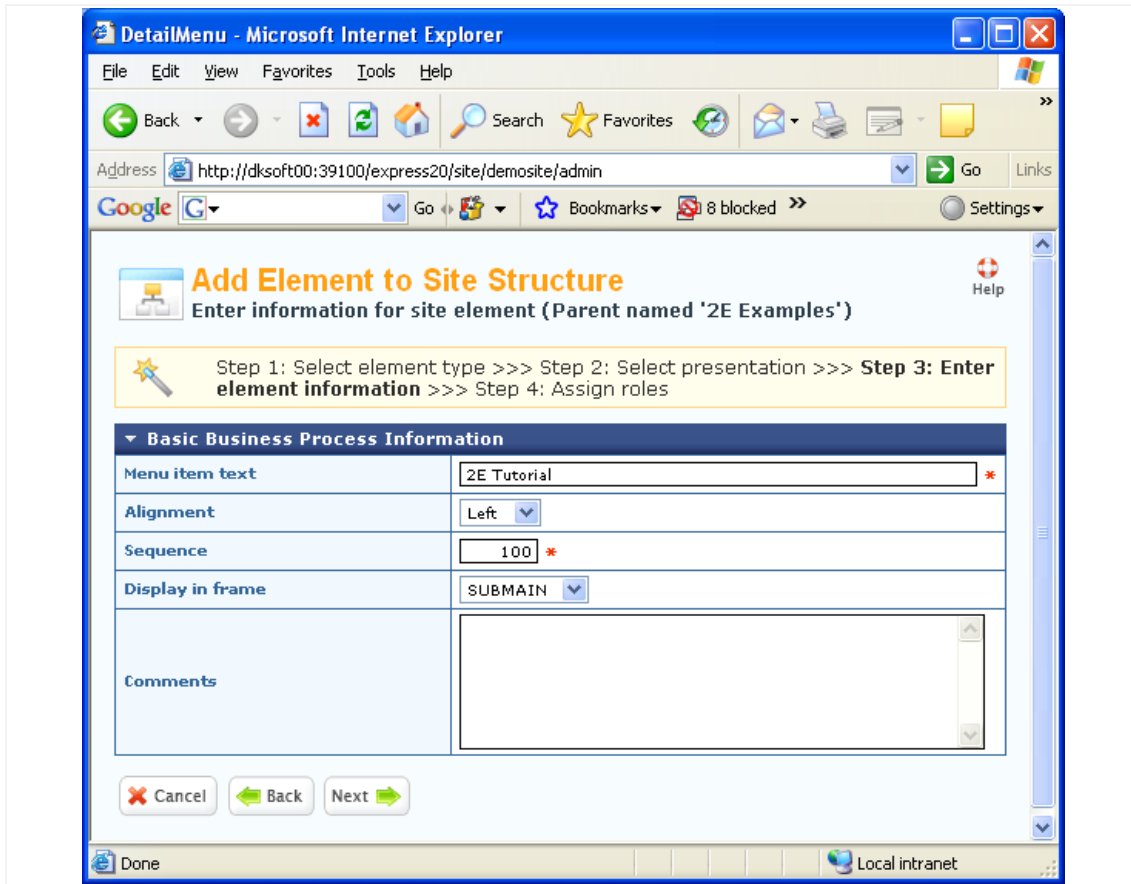
Select the line "11/Left - 2E Examples".
Press "Add".



Press "Next".



Enter "tutorial" in the "Search for" field and press "Find".
Press the "Select" button to select the presentation.



Enter/select:

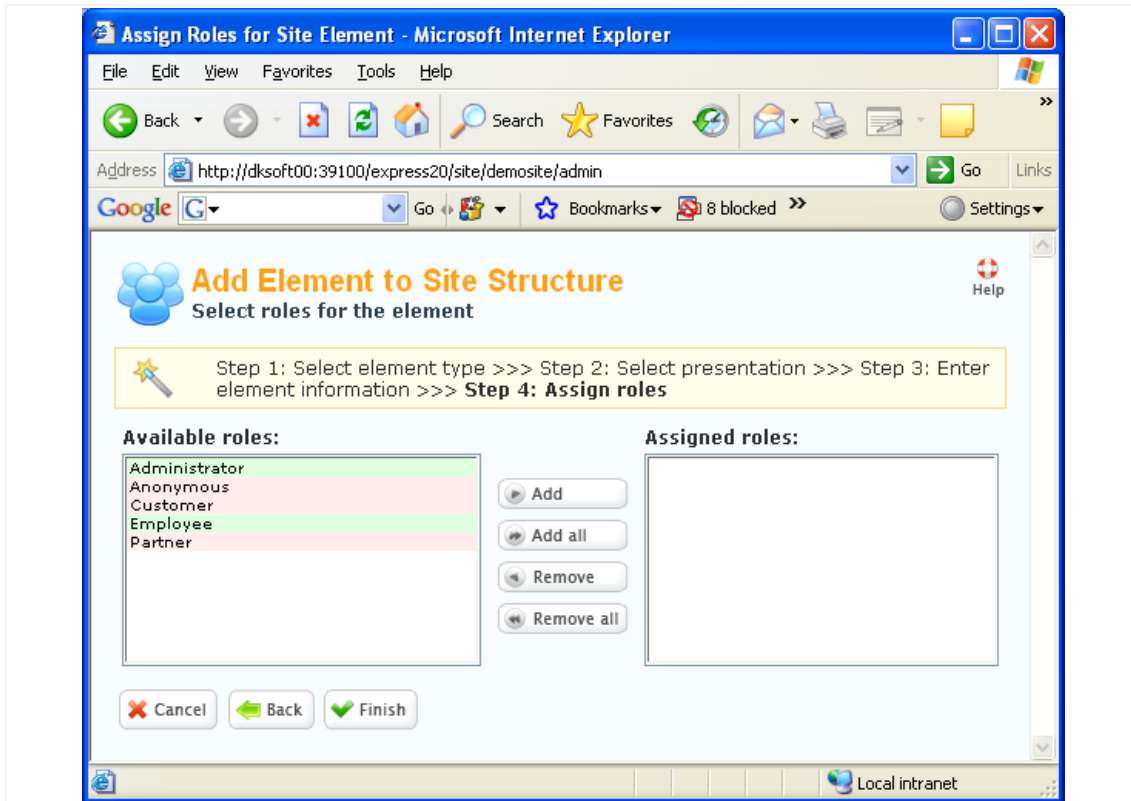
Menu item text: 2E Tutorial

Alignment: Left

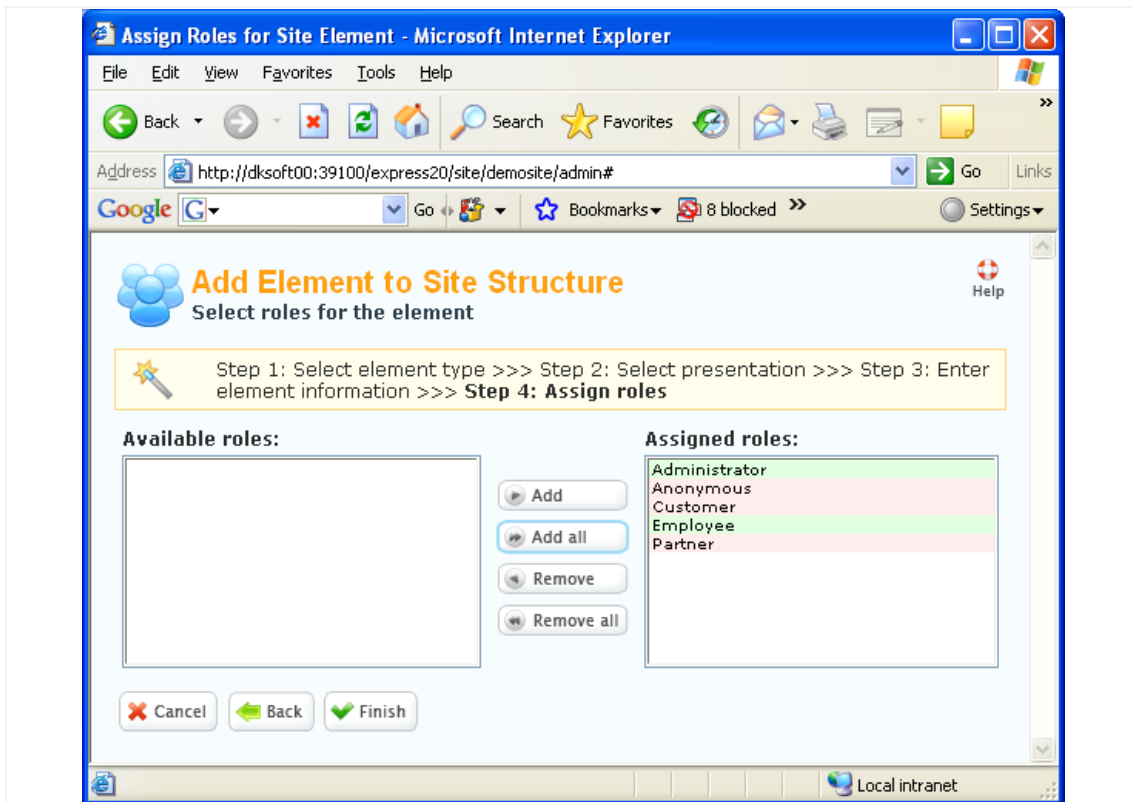
Sequence: 100

Display in frame: SUBMAIN.

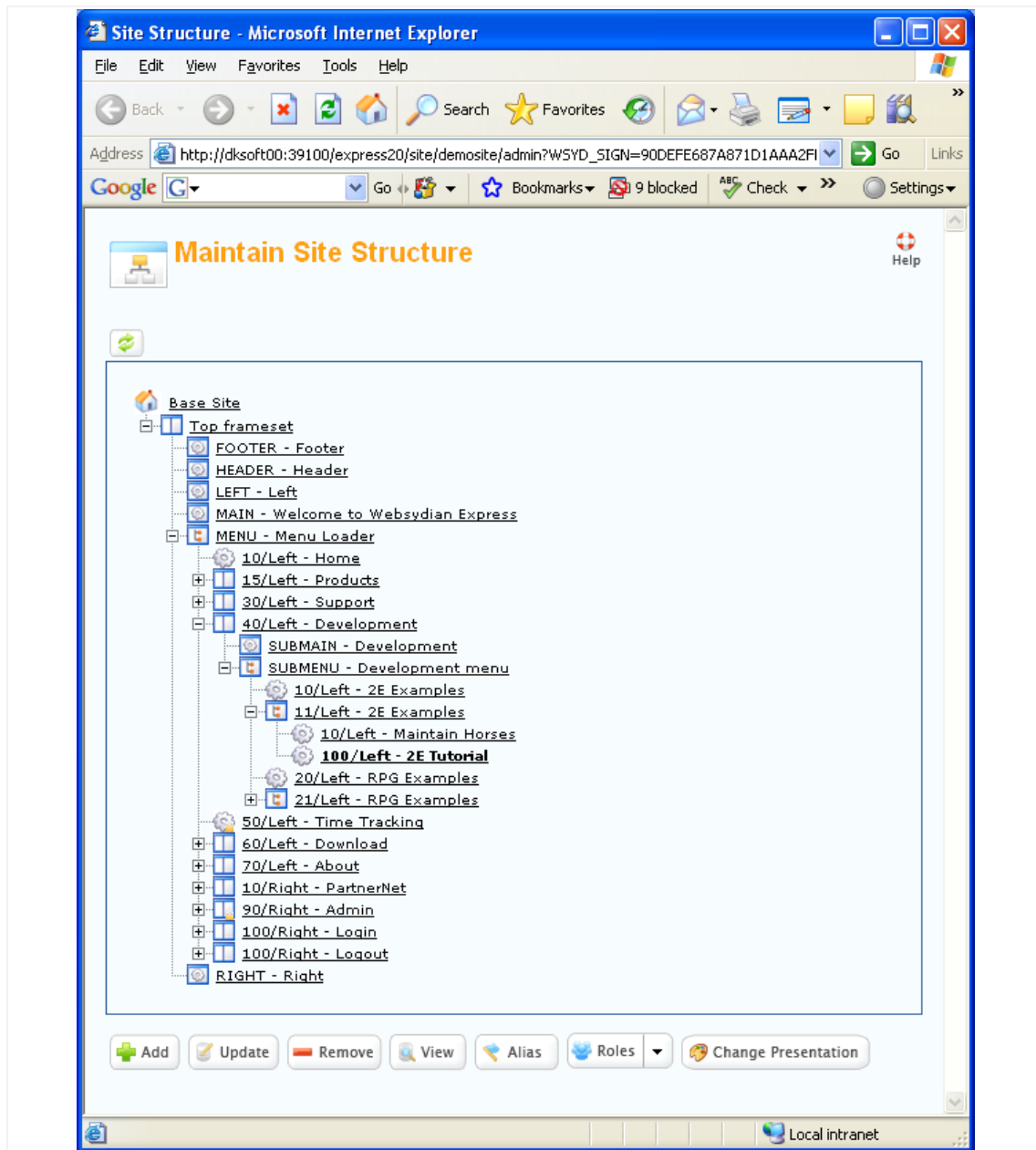
Press "Next"



Press "Add all" to give all roles access to the tutorial process.



Press "Finish".

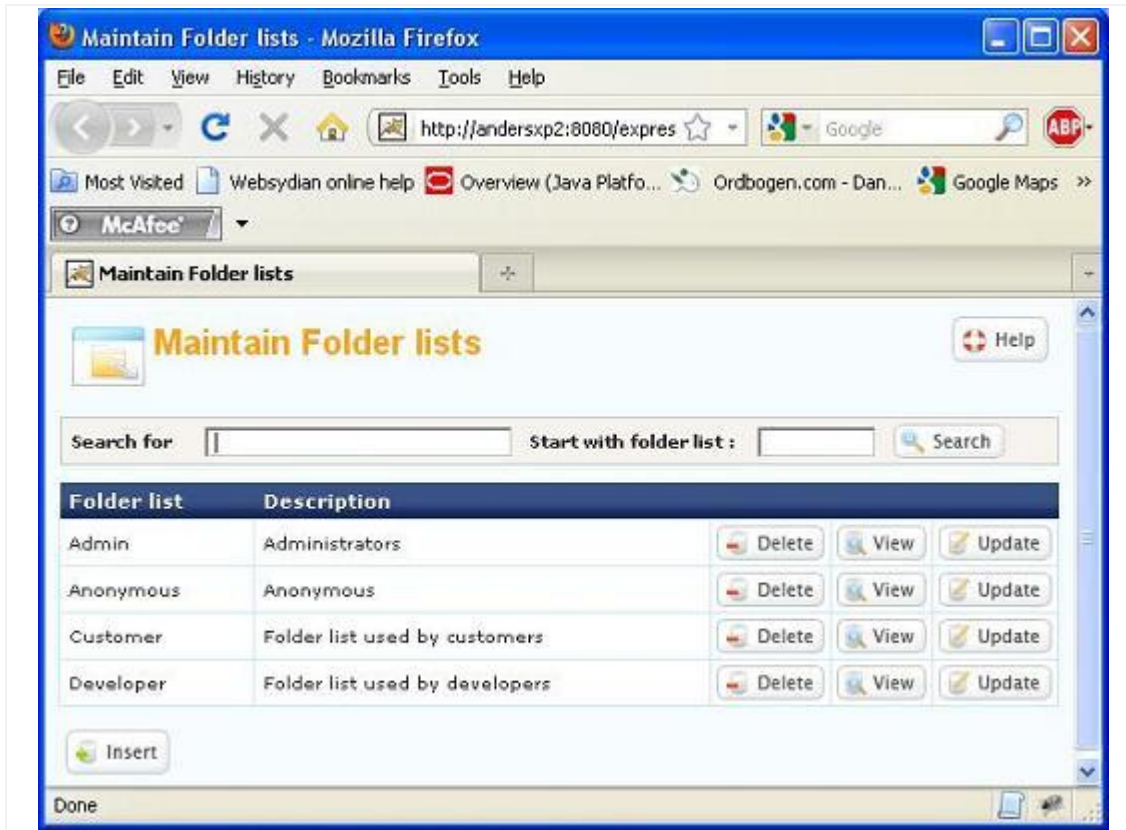


Step 11: Add pagemodeler folder to Anonymous folder list

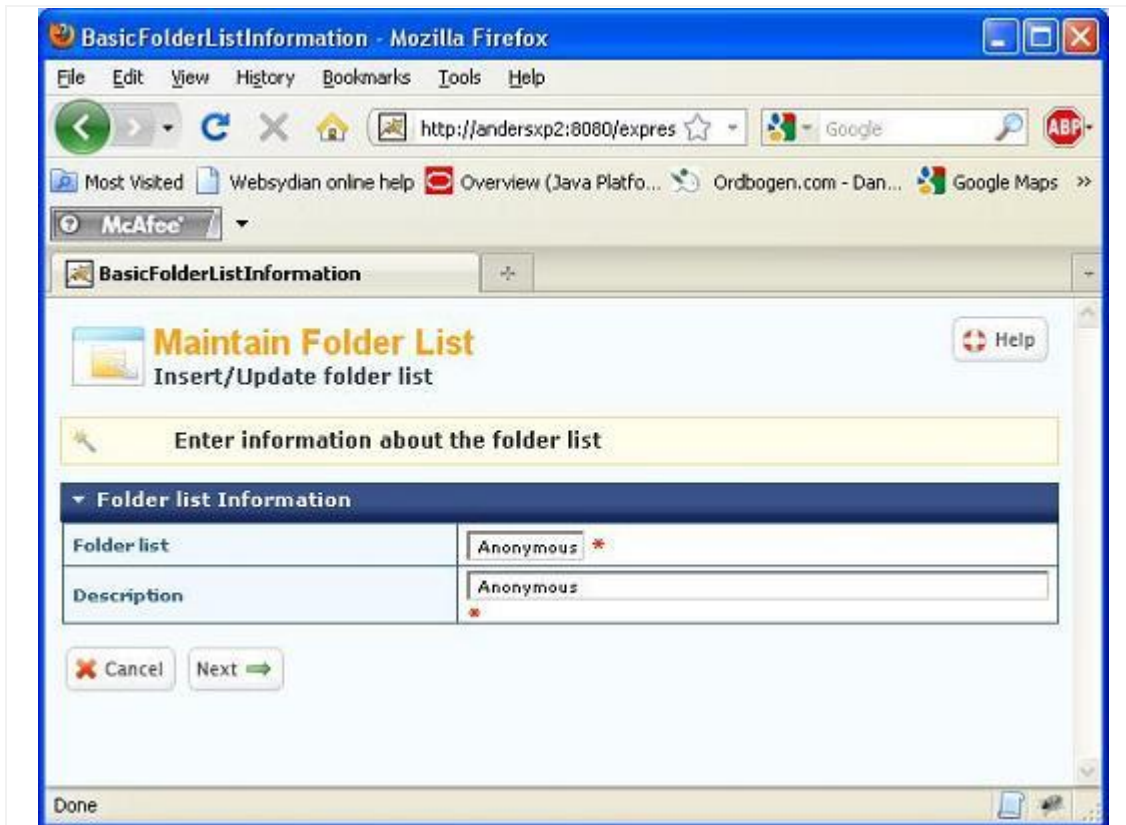
In step 5, the template was saved to the folder "pagemodeler". To avoid having to log in to test the application, you must add this folder to the folder list "Anonymous".

You only have to do this once for your site.

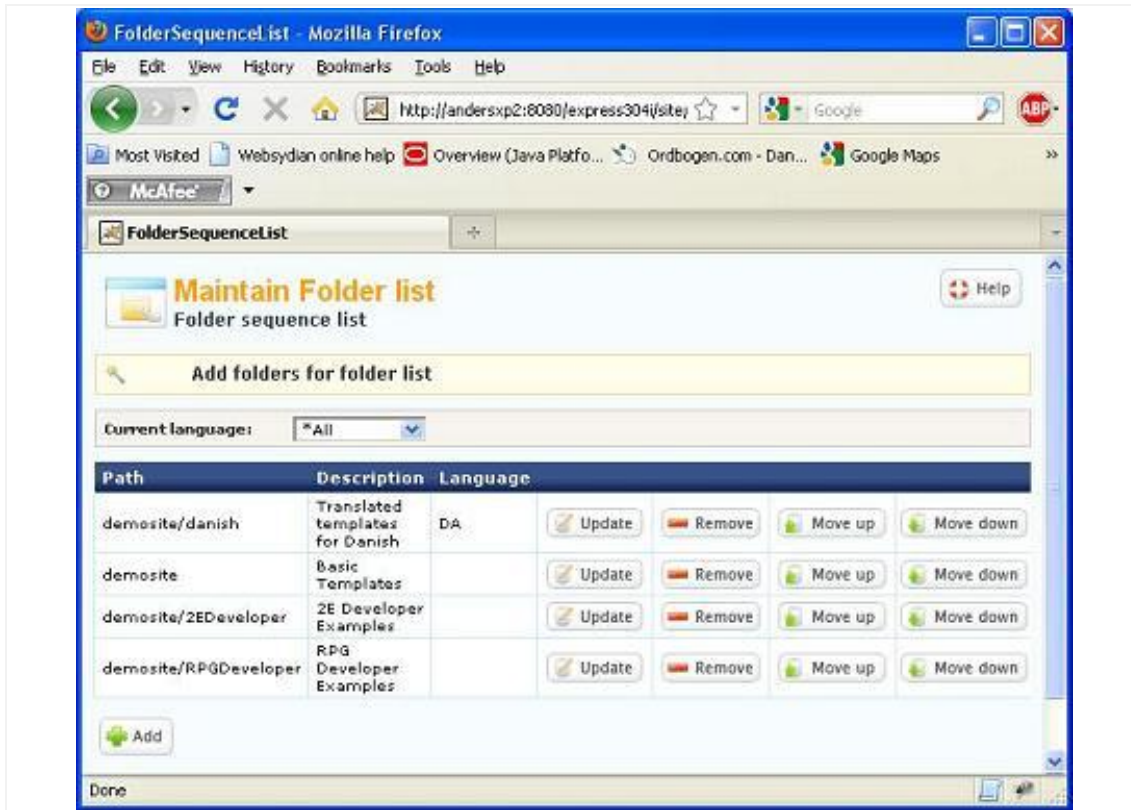
Select the menu item "Templates→Folder Lists".



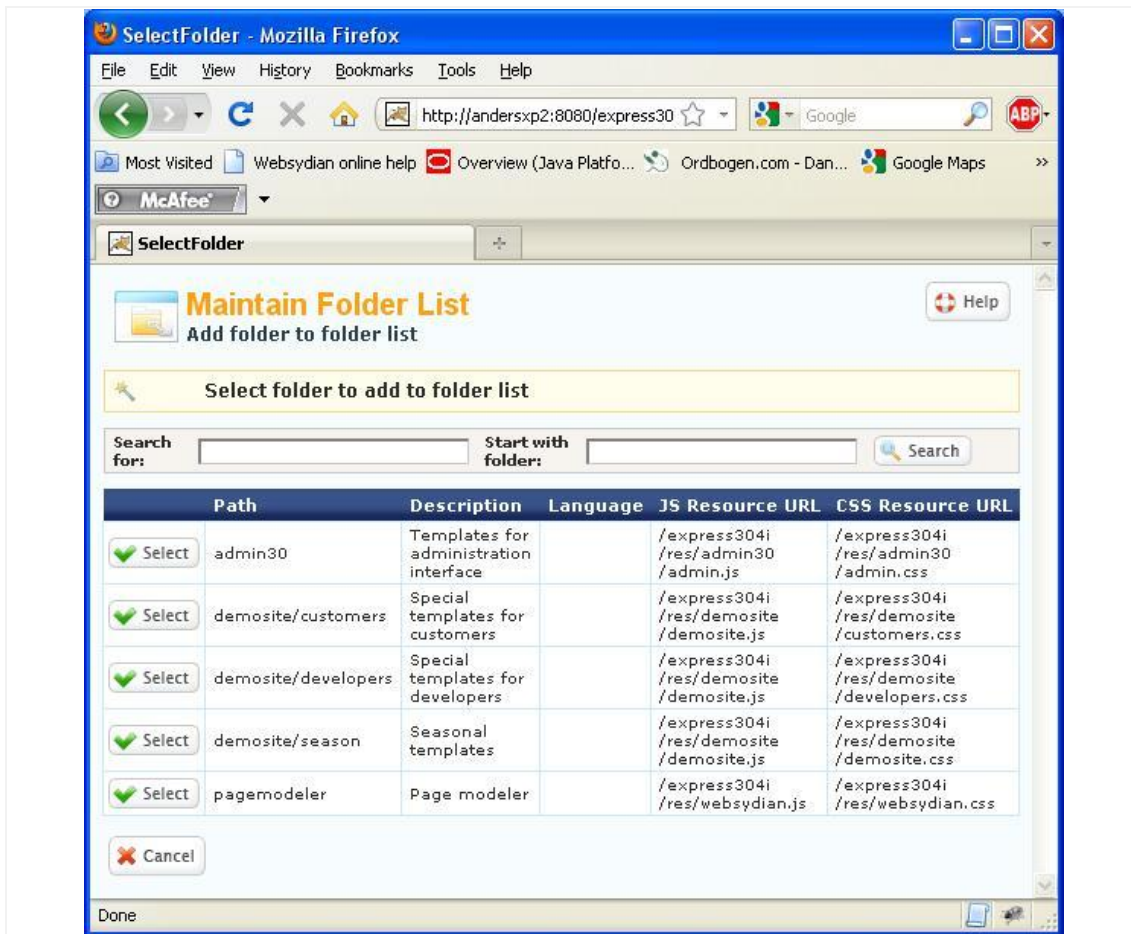
Press "Update" for the folder list "Anonymous".



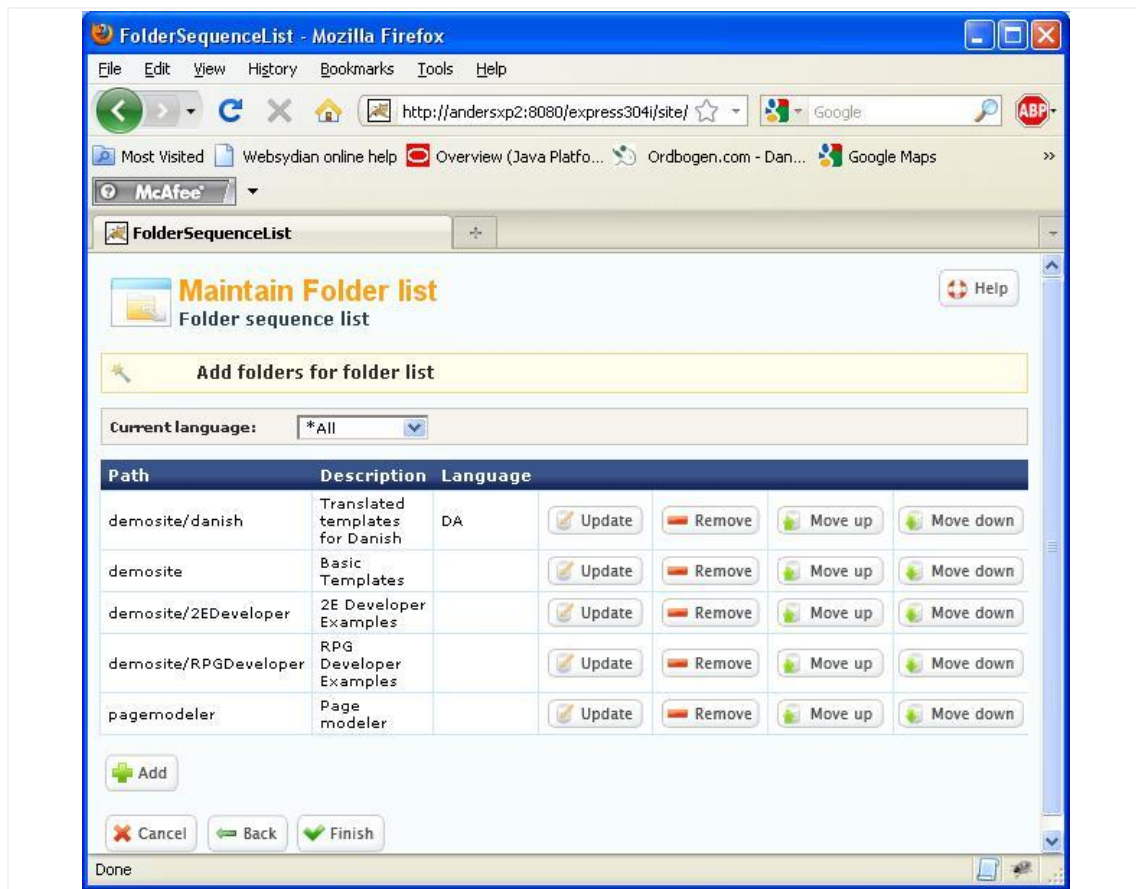
Press "Next".



Press "Add".



Press “Select” for the “pagemodeler” folder.



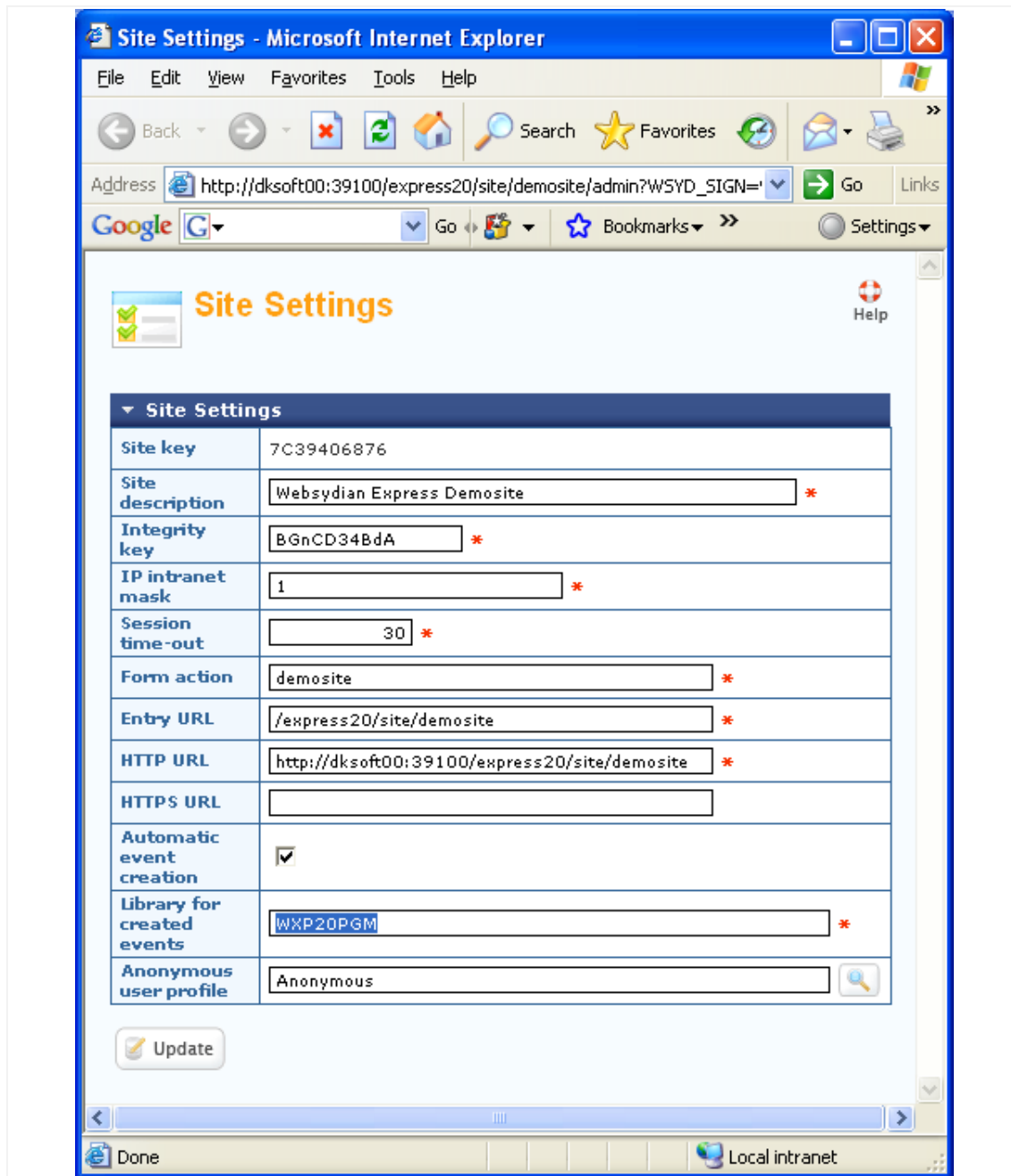
Press “Finish” to finalize the addition of the pagemodeler folder to the folder list.

Note that the change does not take effect until you have pressed “Finish”.

Step 12: Test

Before testing the application, the Site must be set up to accepting new Events placed in the generation library.

Use the Menu item "Site configuration→Site Settings".



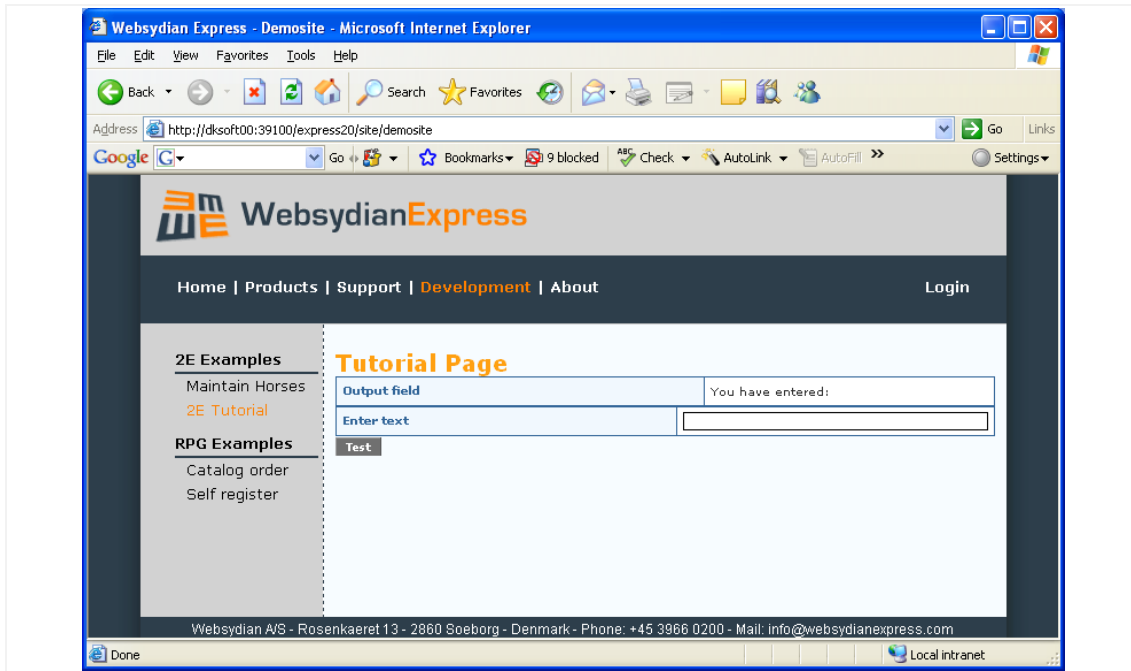
Change the setting "Library for created events" to your generation library.
Press "Update".

Open a new browser window, access the demosite, select Development in the top menu

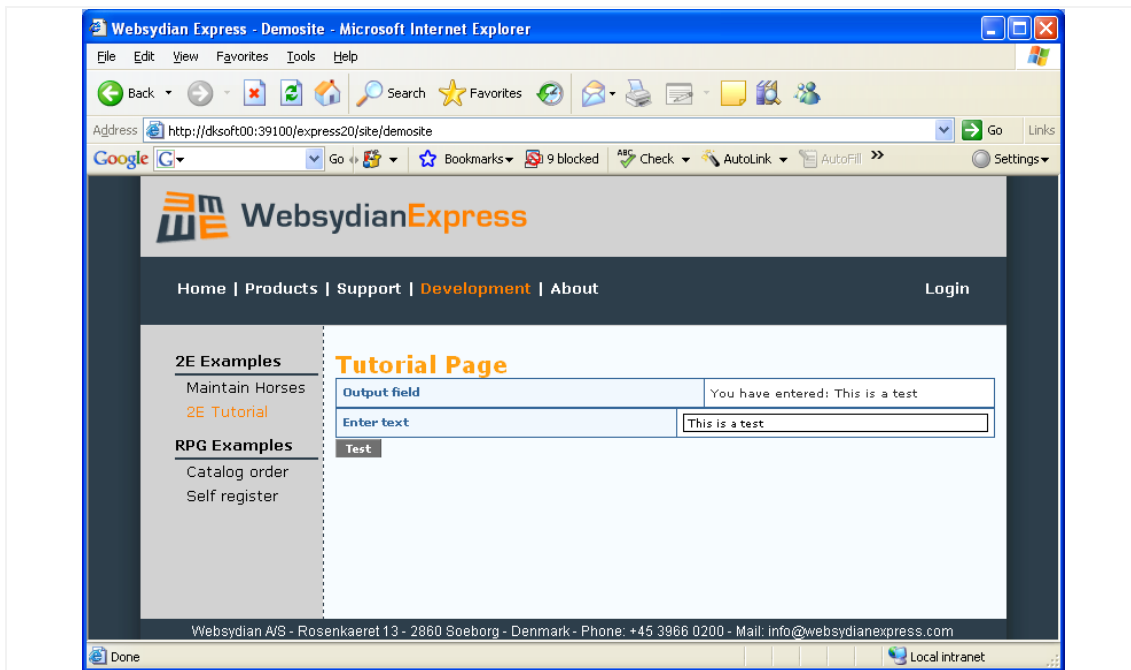
The menu item 2E tutorial should be shown in the left menu under 2E Examples - as shown below.



Press the "2E tutorial" menu item



Enter a text in the input field and press "Test".



If this is the result, the tutorial has been successfully concluded.