

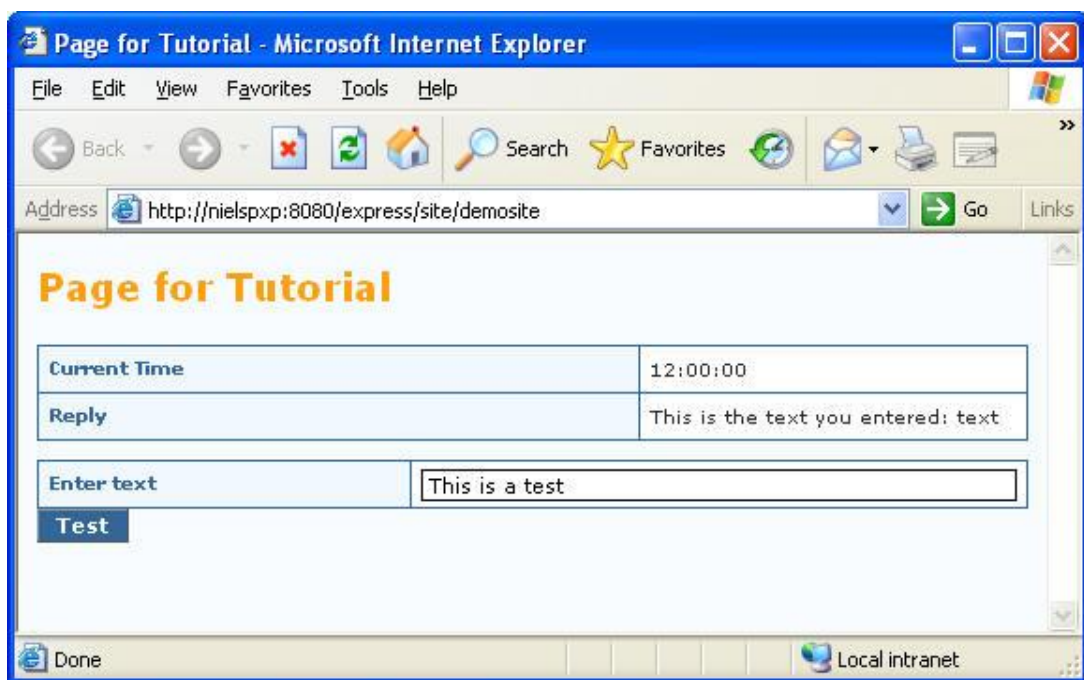
---

# WEBSYDIAN™

## RPG Developer Tutorial

### Part 2 – Retrieving Data from the Web

Revision 1



## Table of Contents

---

Table of Contents .....	1
Introduction .....	2
Change Template .....	2
Create EventHandler .....	9
Parameter Interface .....	10
Handle Registration of Event.....	10
Retrieve User Input .....	10
Call PageGenerator .....	11
Create PageGenerator.....	11
Declarations .....	11
Parameter Interface .....	12
Set output values .....	12
Set initial value for input field.....	12
Change ProcessEntryPoint .....	12
Declarations .....	13
Initialize .....	13
Call to PageGenerator .....	13
Compile Programs .....	13
Deploy Objects .....	13
Restart application .....	14
Run Business Process.....	14

## Introduction

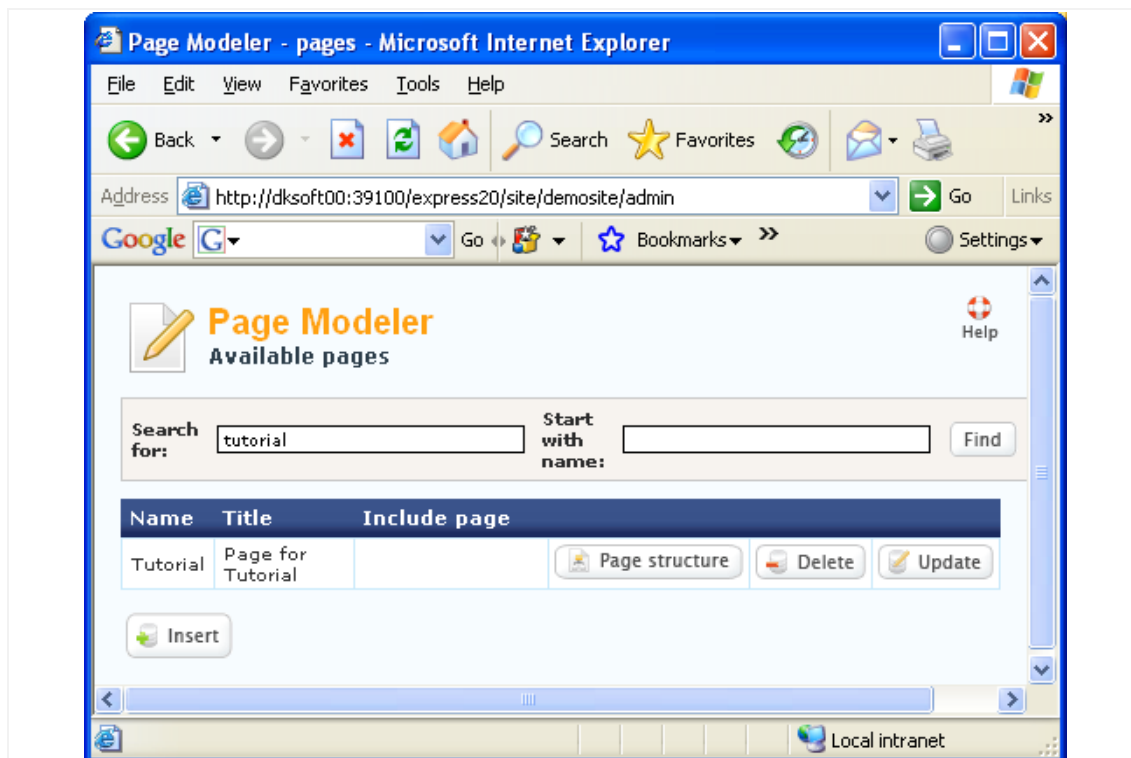
The second part of the tutorial will extend the business process made in the first part of the tutorial. It will show how to add a button and an input field to the HTML page and how to retrieve the value entered in the input field.

The retrieved value is used to create a reply text, which is shown on the HTML page, but it could as well have been used to update a database, been used for calculations in the program or been used as input to other programs.

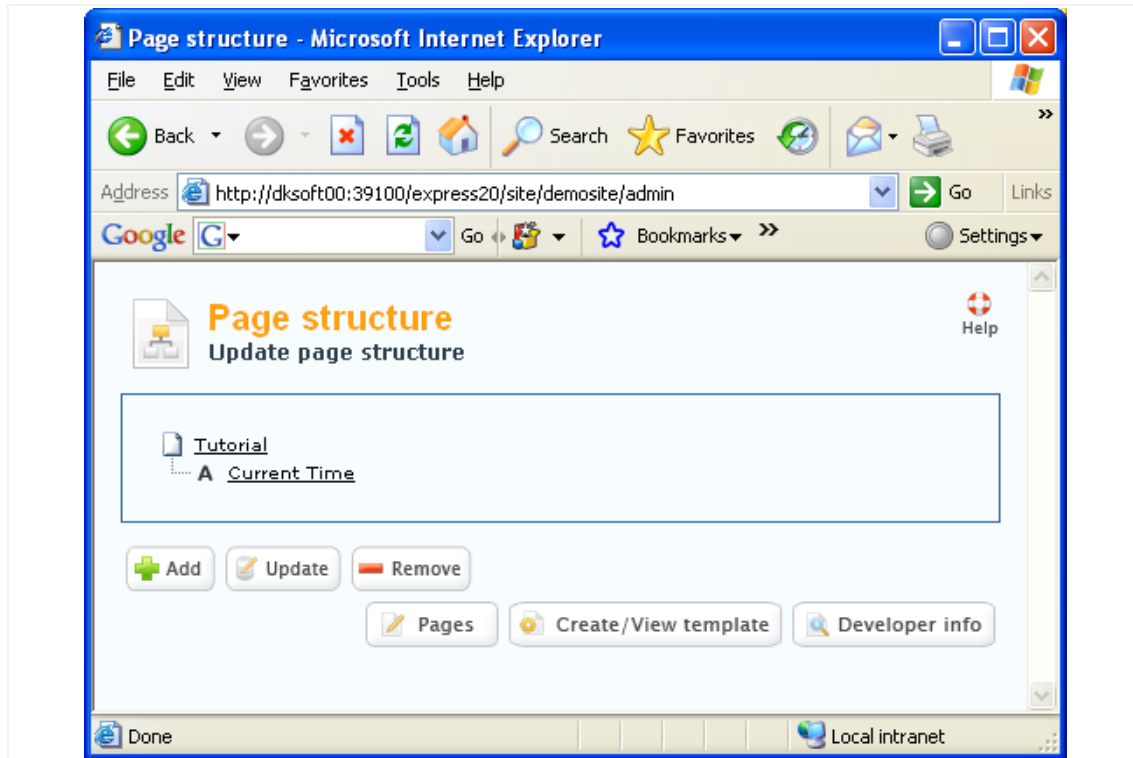
## Change Template

The first step in extending the business process is to add a button and an input field to the HTML template used by the tutorial business process.

Log in to the demo site using the WSADMIN profile, open the administration interface, and select RPGDeveloper→PageModeler.

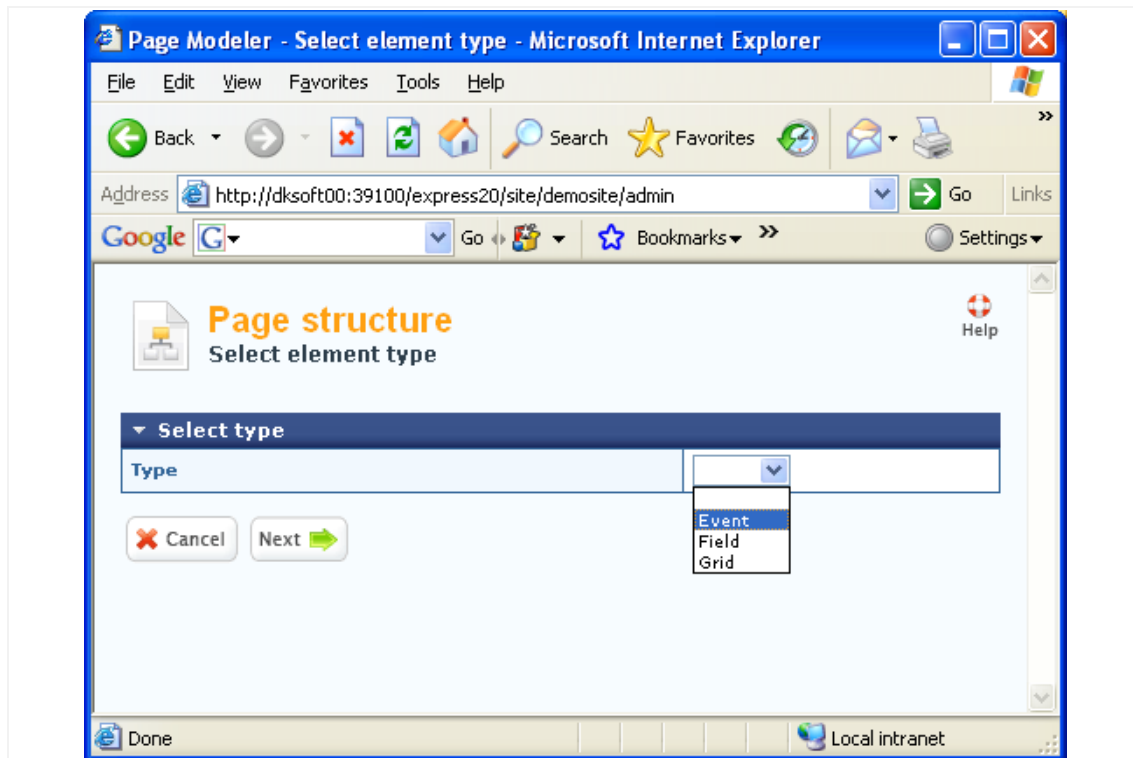


Find the Tutorial page in the list of pages (if necessary search for Tutorial). Press the "Page structure" button.

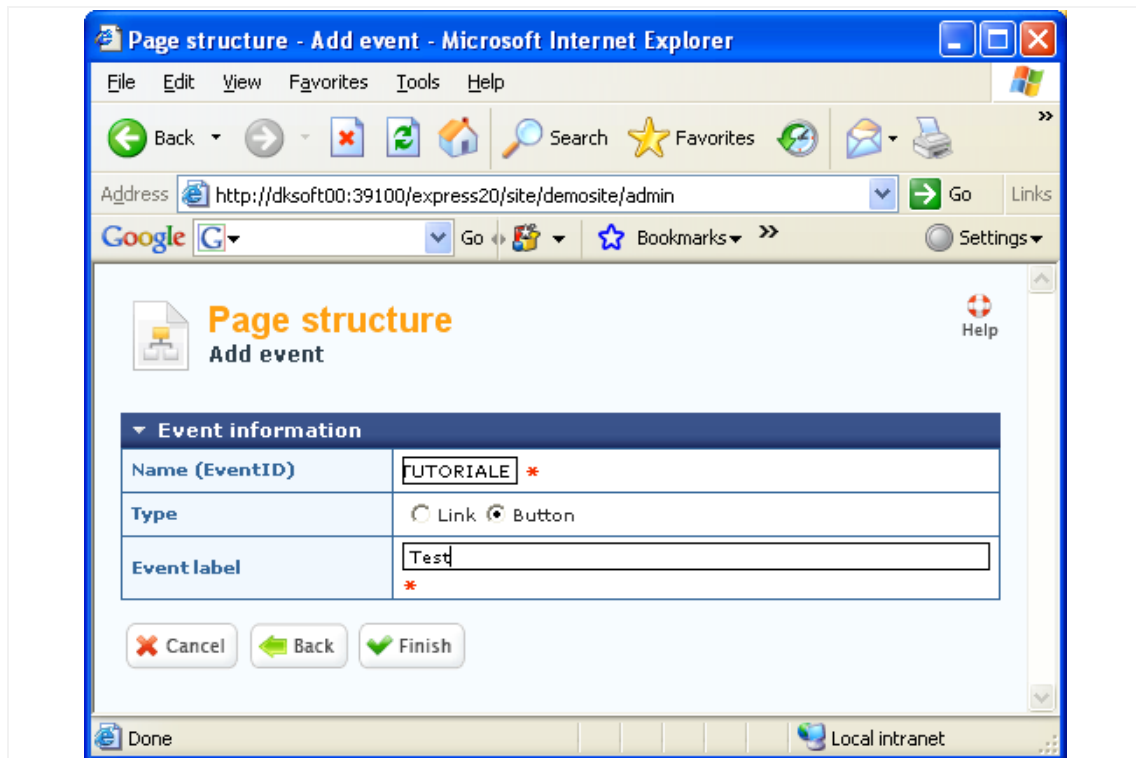


Next step is to add the button to the page.

Select the Tutorial page and press "Add".

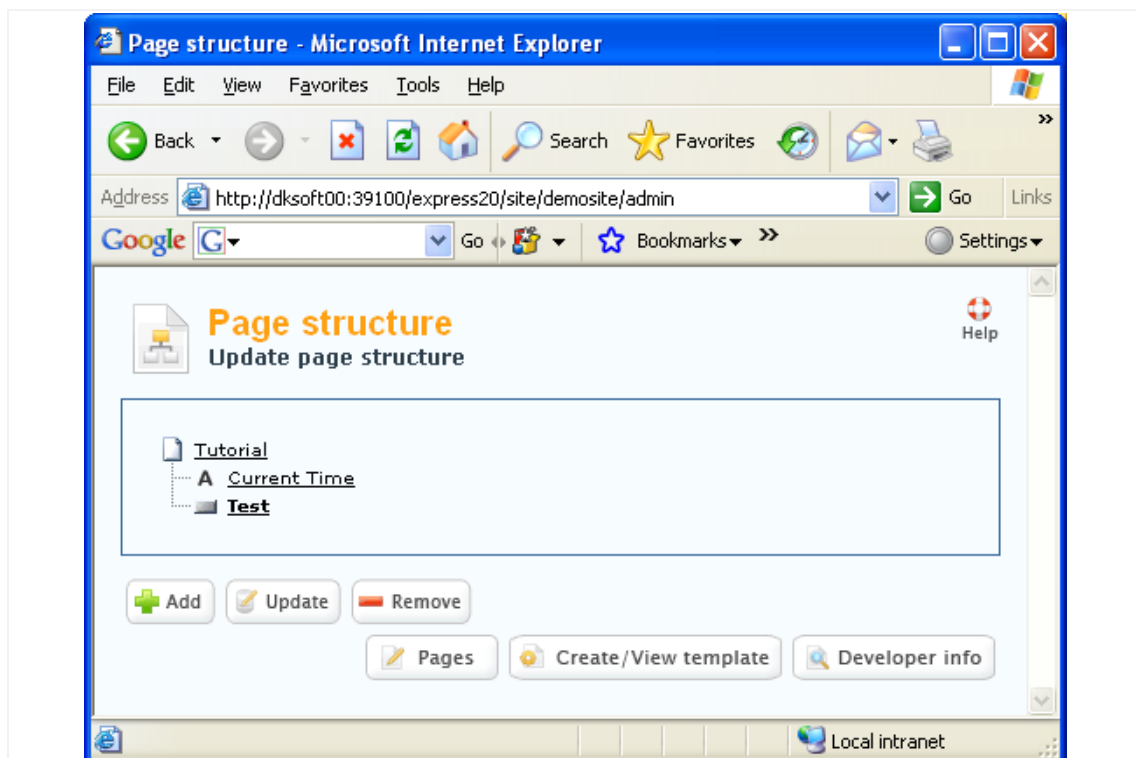


Select Event.

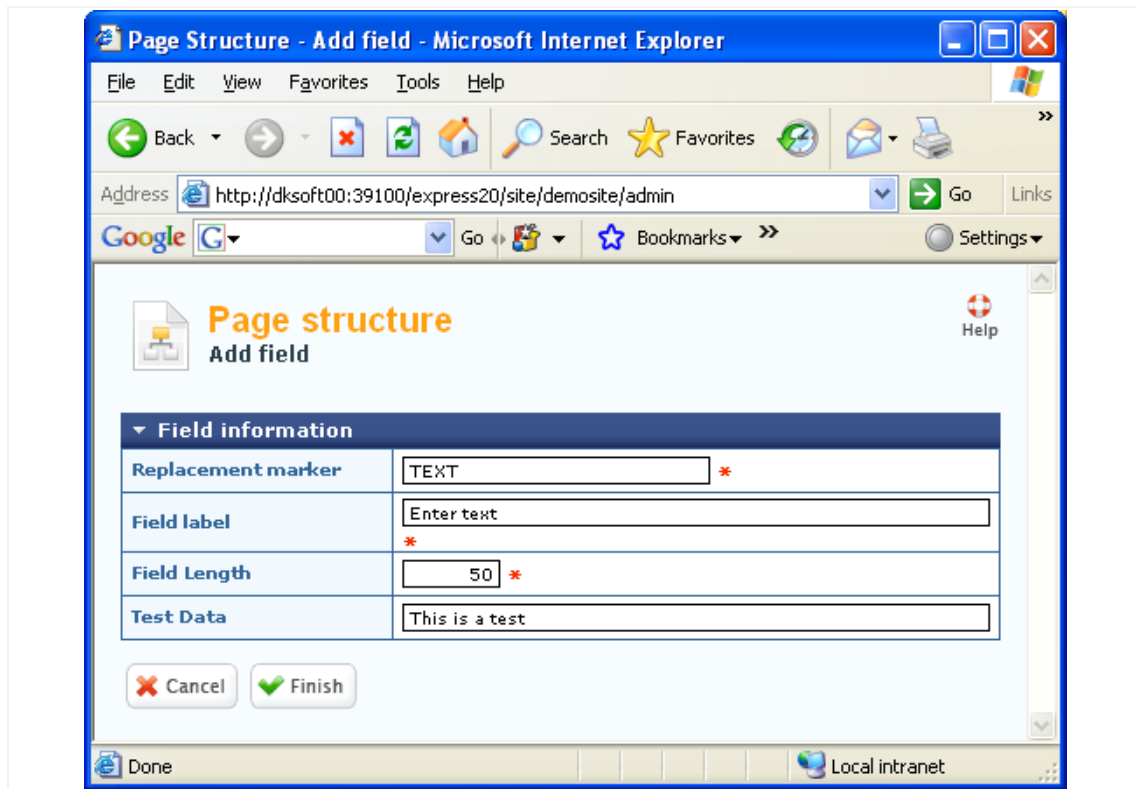


Enter the values shown and press "Finish" (The name is **TUTORIALE**).

This adds an event to the page. The next step is to add an input field to the event.



Select the event "Test" and press the "Add" button.



The above page allows you to define an input field.

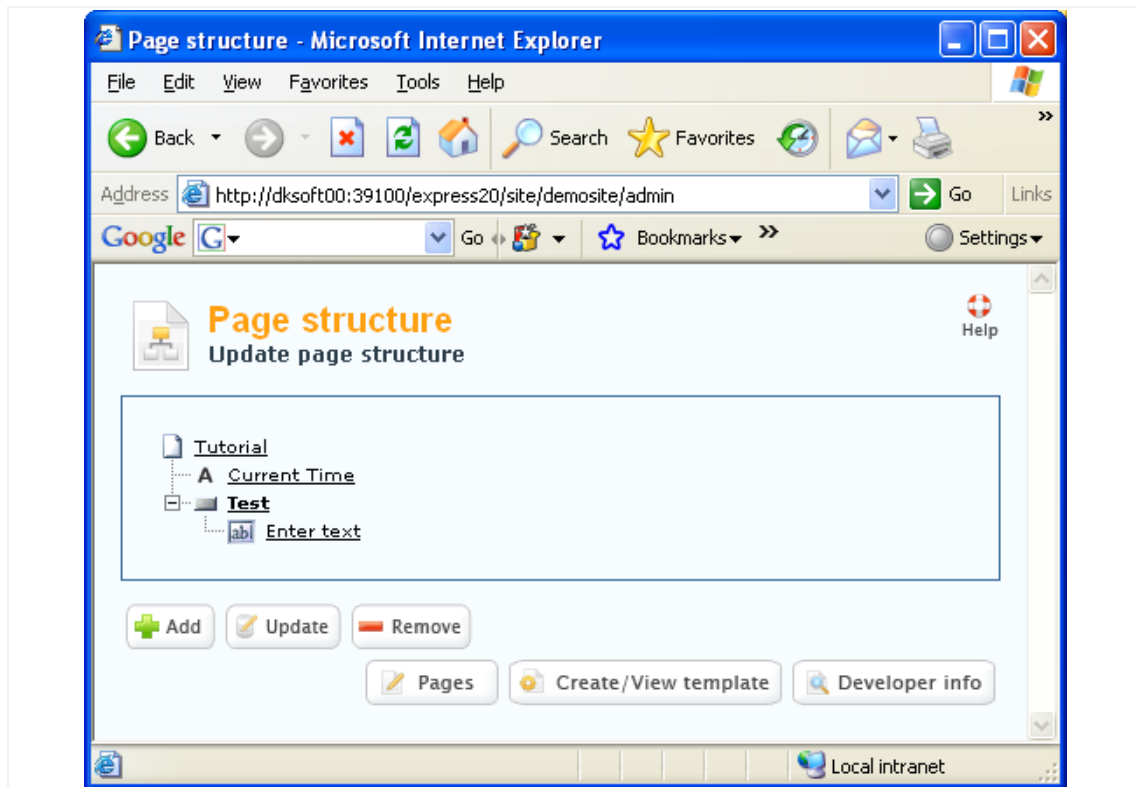
Enter the values shown and press Finish.

The replacement marker is the name used to refer to the field in the program generating the HTML page.

The field label is the label for the input field.

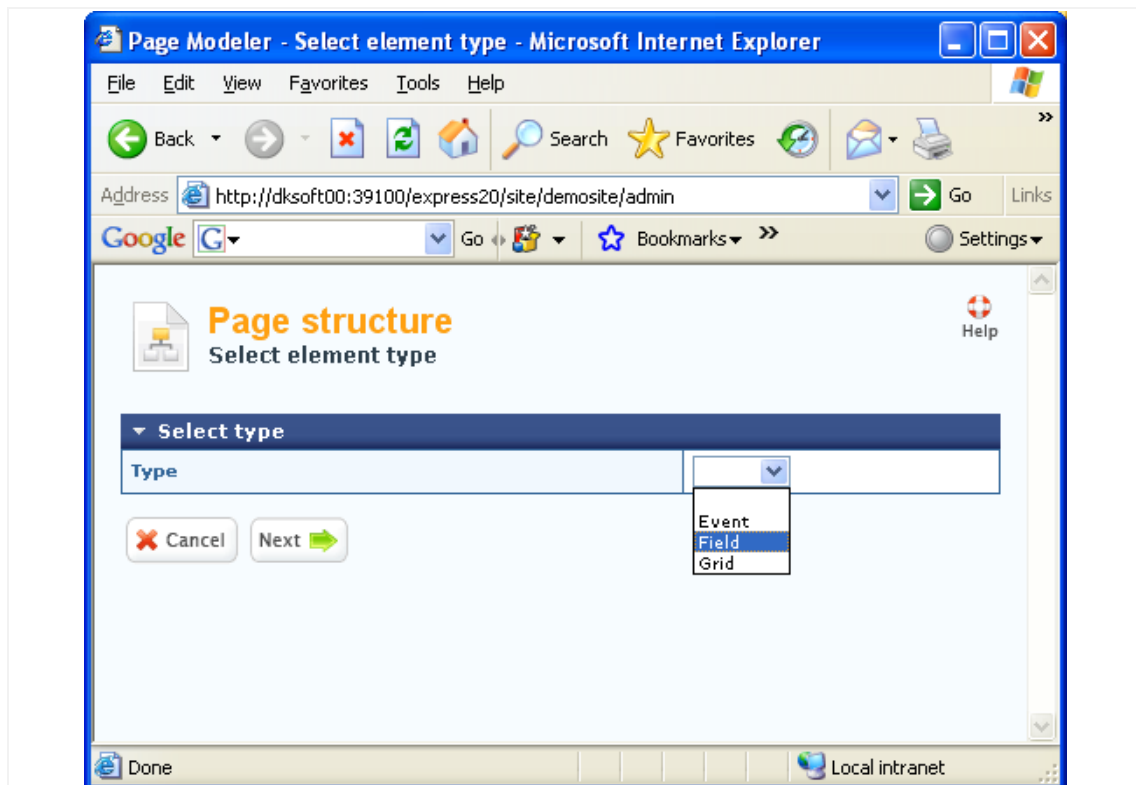
The field length determines the length of the input field.

The test data is used when viewing the generated template.

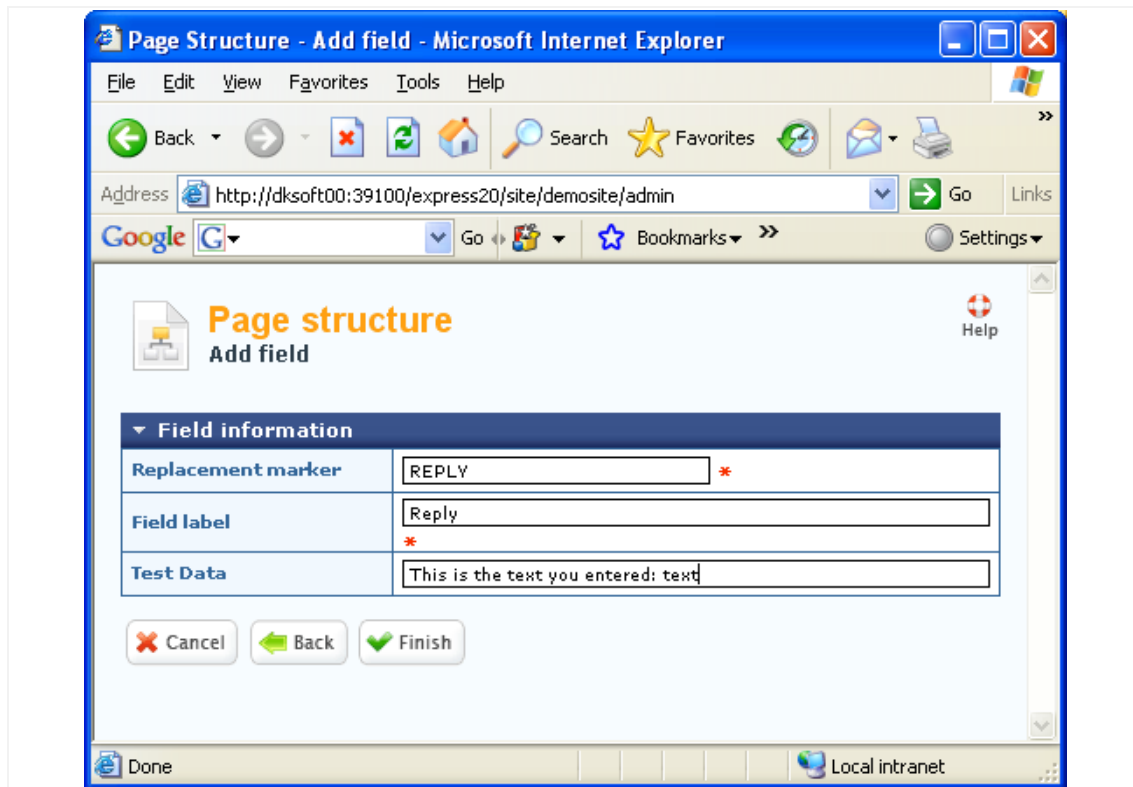


The last thing to add is an output field, which can be populated based on the entered text.

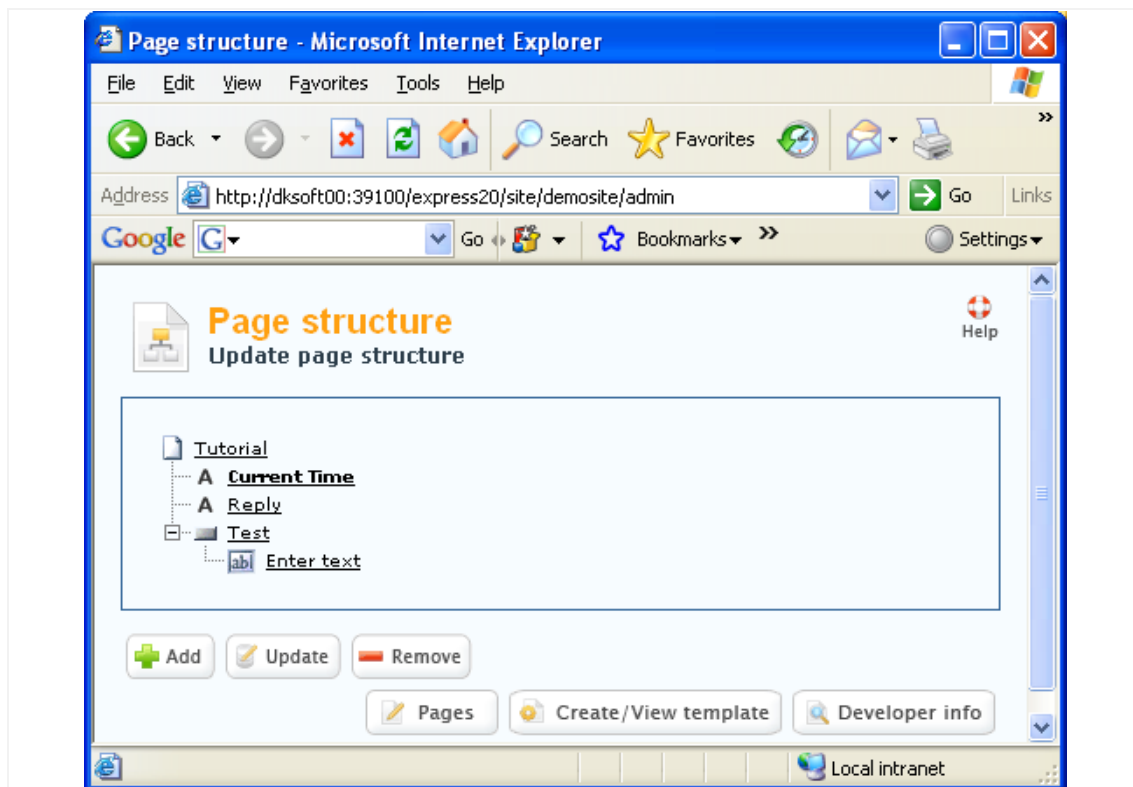
Select the "Tutorial" page and press "Add".



Select "Field".

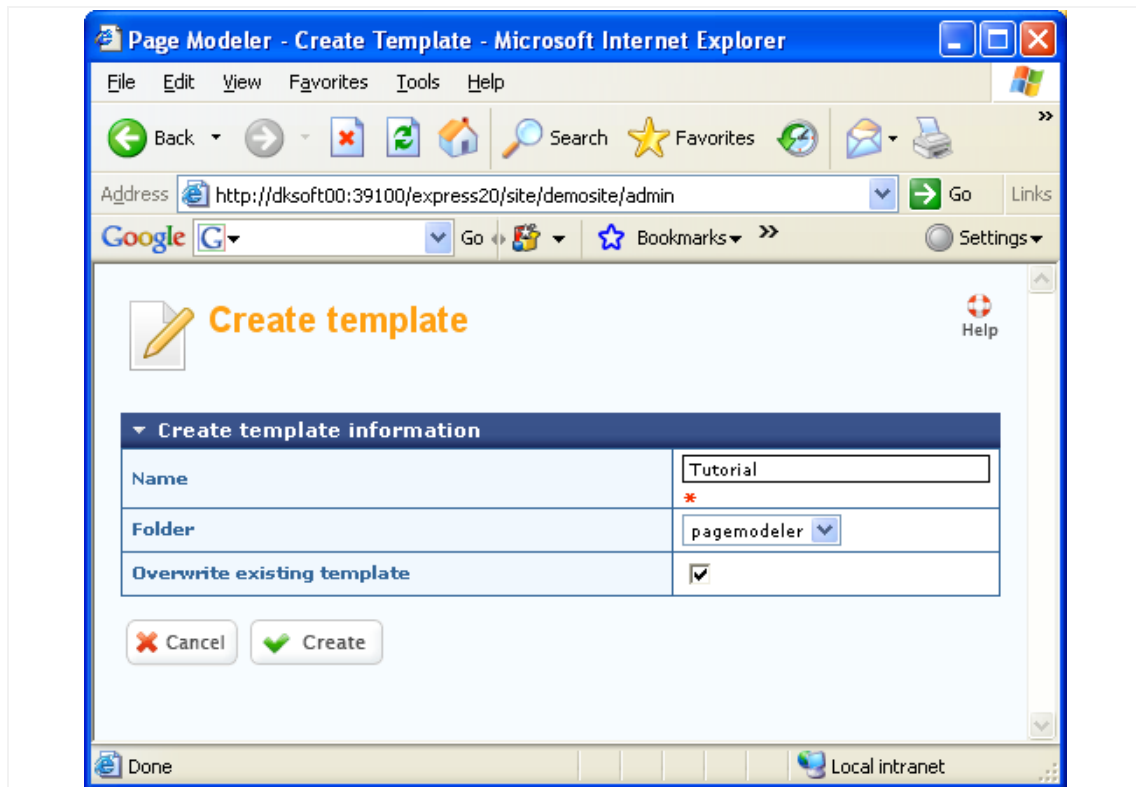


Enter the values shown and press Finish.

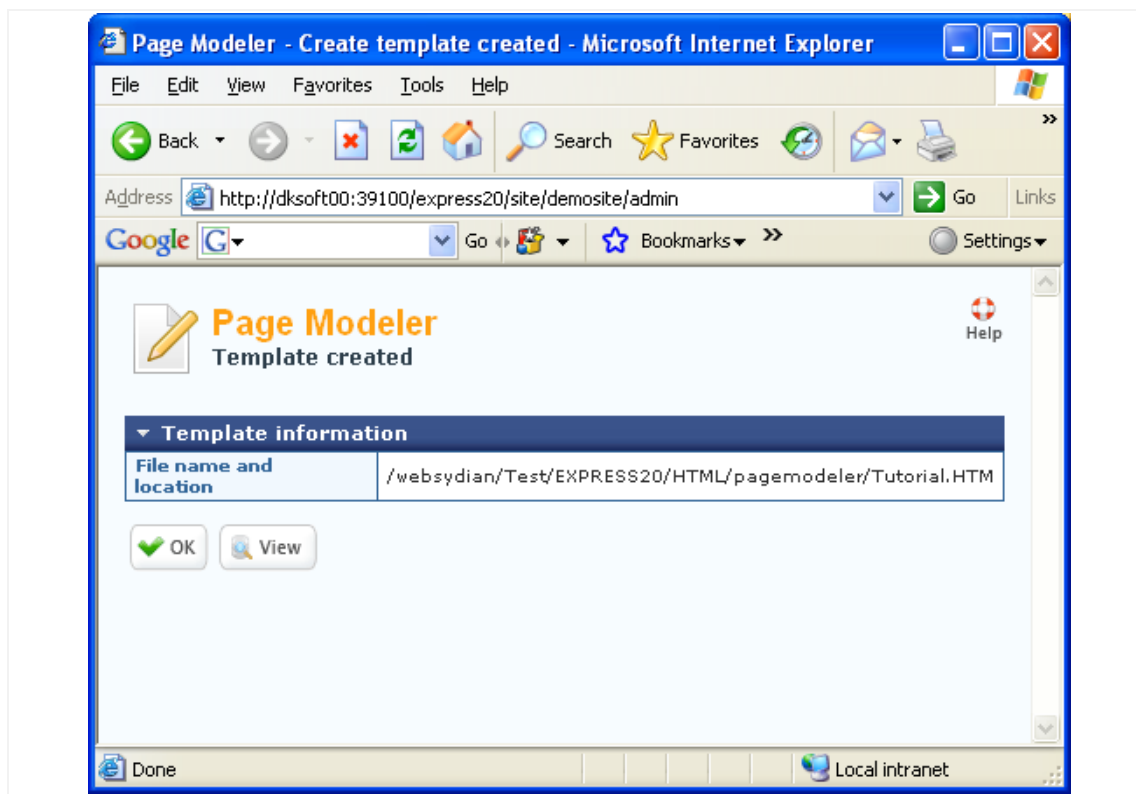


An output field and an event with an input field have been added to the page. To create the HTML template press "Create/View template".

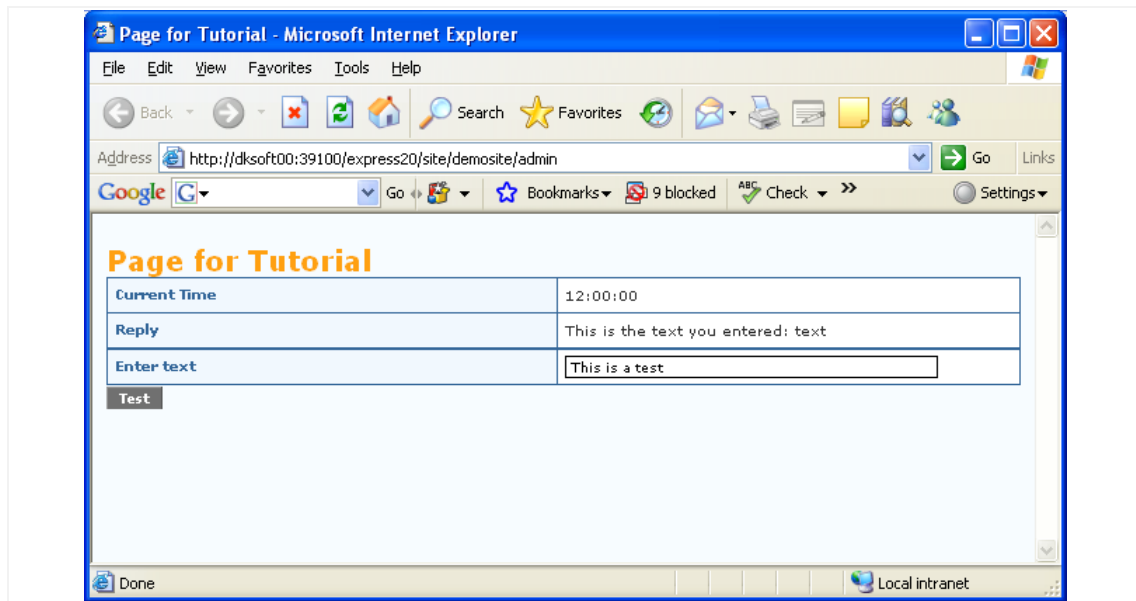




Keep the existing values, but check the "Overwrite existing template" checkbox. Press "Create".



Press View to check that the button and fields have been added.



The template has now been changed and is ready to use.

## Create EventHandler

---

The Websyidian Express runtime must have a program with a specific interface to call when the test button is pressed. Such a program must be made for each new button you add to your business process.

These functions are known as EventHandlers.

Create a new ILE RPG source member named TUTORIALE entering the following source code.

```

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

D FLD_TEXT      C          'TEXT'
D rc            S          like(TypRC) inz(*zero)
D Text         S          256A  inz('') varying

c      *Entry      PLIST
C          PARM          Return          7
C          PARM          Action          1

C          eval      Return = *blank

* Handle Registration of Event
C          if          ( Action = 'R' )
C          Return
C          endif

* Retrieve User Input
C          eval      rc = GetInput(FLD_TEXT:Text)

* Call Page Generator
C          CALL      'TUTORIALPG'
C          parm      Text

C          RETURN

```

Source Code explained

### Parameter Interface

```

C      *Entry      PLIST
C          PARM          Return          7
C          PARM          Action          1

```

All EventHandlers MUST have this interface - otherwise the Websyidian Express runtime simply can't call the function.

### Handle Registration of Event

```

C          if          ( Action = 'R' )
C          Return
C          endif

```

All EventHandlers MUST check for the case where the parameter "Action" is "R" (Register). If this is the case the program MUST terminate (with a blank value in the Return parameter).

### Retrieve User Input

```

C          eval      rc = GetInput(FLD_TEXT:Text)

```

This API call returns the value entered by the user in the input field for the event. The value is returned in the second parameter of the call (in this case, the field "Text").

### Call PageGenerator

C	CALL	'TUTORIALPG'	
C	parm		Text

All EventHandlers MUST call a PageGenerator - otherwise nothing will be sent to the browser and the user will experience a time out.

### Create PageGenerator

---

Change the existing source member TUTORIALPG to add the functionality for populating the additional fields.

After the changes, the source member must have the following content.

```
H DFTACTGRP(*NO) ACTGRP(*CALLER) BNDDIR('QC2LE':'WIEXPRESS')
/COPY QRPGLSRC,EXPRESS_H
D TEMPLATE          C              'TUTORIAL'
D FLD_TIME          C              'TIME'
D FLD_REPLY         C              'REPLY'
D FLD_TEXT          C              'TEXT'
D rc                S              like(TypRC) inz(*zero)
D Time              S              8T
D Text              S              256A  varying
D Reply             S              256A  inz('')
C      *ENTRY          PLIST
C      parm              Text
C      * Set Output
C      eval              Time = %time()
C      eval              rc = SetOutput(TEMPLATE :
C                          FLD_TIME :
C                          %char(Time))
C      * Write Reply Text
C      if                ( Text <> '' )
C      eval              Reply = 'You entered: ' + %trim(Text)
C      eval              rc = SetOutput(TEMPLATE :
C                          FLD_REPLY:
C                          Reply)
C      endif
C      * Set initial values for input field
C      eval              rc = SetOutput(TEMPLATE :
C                          FLD_TEXT:
C                          Text)
C      * Write the page to the browser
C      eval              rc = WritePage(TEMPLATE)
C      return
```

Added Source

### Declarations

D FLD_TIME	C	'TIME'
D FLD_REPLY	C	'REPLY'

Declares the constants for the names of the two new fields.

D Text	S	256A	varying
D Reply	S	256A	inz('')

Declares the program fields used to specify the values of the two fields.

### Parameter Interface

C	*ENTRY	PLIST	
C		parm	Text

As the EventHandler needs to transfer information about the entered text to the PageGenerator, this field is added to the parameter interface.

### Set output values

C	if	( Text <> '' )
C	eval	Reply = 'You entered: ' +
C		%trim(Text)
C	eval	rc = SetOutput(TEMPLATE :
C		FLD_REPLY:
C		Reply)
C	endif	

If the user has entered any text, a reply text is generated and written to the template.

### Set initial value for input field

C	eval	rc = SetOutput(TEMPLATE :
C		FLD_TEXT:
C		Text)

The text entered by the user is shown in the input field.

## Change ProcessEntryPoint

As a parameter was added to the PageGenerator function, and as the ProcessEntryPoint function calls the PageGenerator, a minor change is necessary in the ProcessEntryPoint.

Change the source of the TUTORIAL source member to the following:

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

D Text          S          256A   inz('') varying
D RC            S          like(TypRC) inz(*zero)
C   *ENTRY      PLIST
C           PARM          Return      7

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

* Call the Page Generator
C           call        'TUTORIALPG'
C           PARM          Text
C           RETURN
```

Added Source

### Declarations

```
D Text          S          256A   inz('') varying
```

Declare the work field that will be used for the parameter.

### Initialize

```
C           eval        Text = *blank
```

Initialize the work field.

### Call to PageGenerator

```
C           PARM          Text
```

Add the work field as a parameter on the call to the PageGenerator.

## Compile Programs

---

The EXPRESS\_H header file is placed in the runtime library specified at the installation (default WXP20PGM). 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 WXP20APP). Doing this means that the objects are immediately available for the Websyidian Express runtime.

Compile the three 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.

## Restart application

Use the RSTAS command (on the iSeries in the library with the default name WXP20PGM) to restart the application.

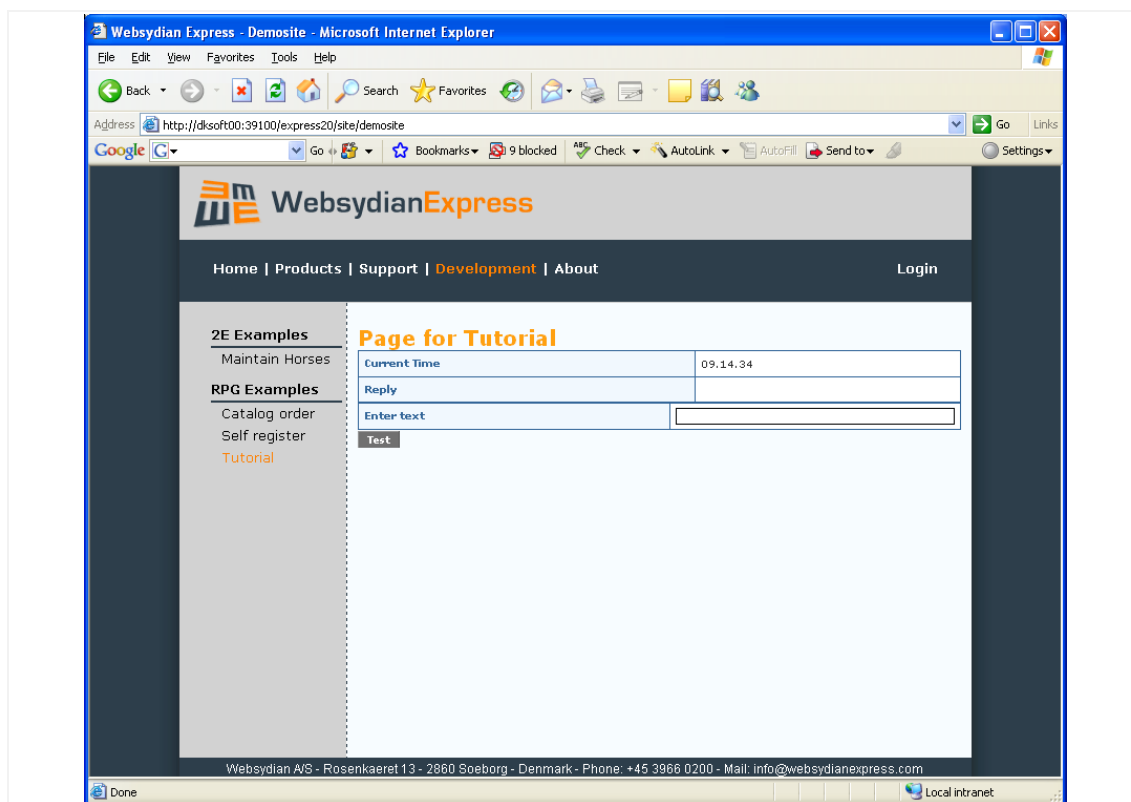
WXP20PGM/RSTAS

This is necessary to make the application use the new program objects.

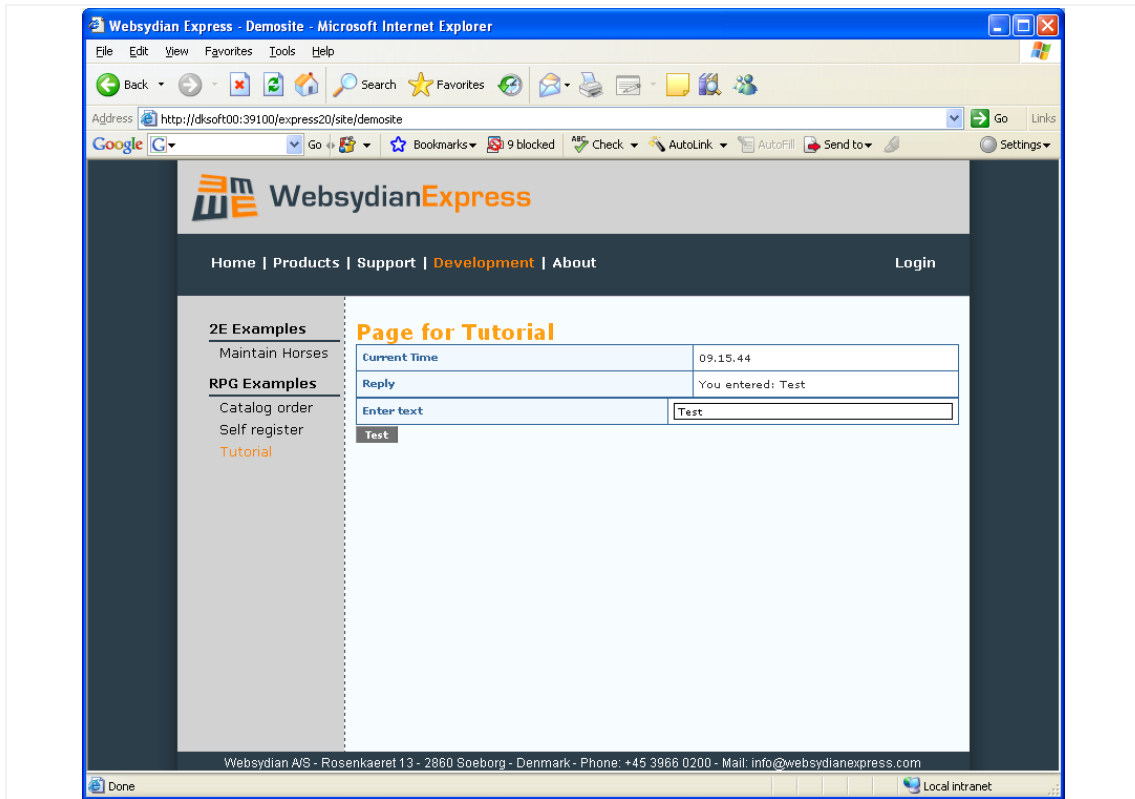
## Run Business Process

Close the browser and start the demo site in a new instance.  
Select the menu item Tutorial.

The following page should be shown.



Enter a text in the input field and press the Test button to test the eventhandler.



If the result is as shown above, the test is successful.