



The most powerful way
to develop
Web applications!



CodeCharge Studio 4
**Examples and
Techniques**



8275 S. Eastern Ave. Suite 200 Las Vegas, NV 89123 USA
Telephone: + 1 (888) 241-7338 Fax: + 1 (866) 312-8049
info@yessoftware.com
<http://www.yessoftware.com>

Copyright © 2000-2008 YesSoftware, Inc.

All rights reserved.

This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by YesSoftware, Inc. YesSoftware, Inc assumes no responsibility or liability for any errors or inaccuracies that may appear in this documentation. Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of YesSoftware, Inc.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to YesSoftware Inc., 6330 S. Eastern Ave. #5, Las Vegas, NV 89119 USA. YesSoftware, the YesSoftware logo, CodeCharge and the CodeCharge Logo are either registered trademarks or trademarks of YesSoftware, Inc in the United States and/or other countries. ASP, IIS, Microsoft, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. JavaScript and JSP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

Examples and Techniques	5
Tips and Tricks	5
Creating a MailTo Hyperlink	5
Publishing Custom Files.....	6
Showing a Grid Record in Two Rows.....	6
Programming	6
Simple Reporting.....	6
Simple Report With Group Headers	6
Simple Report with Simple Grouping.....	10
Simple Report with Total.....	12
Hiding Components.....	15
Hiding a Grid If No Records are Found.....	15
Hiding a Record Form Control	17
Customizing the Data Source.....	20
Dynamically Modify the "List Of Values" of a ListBox	20
Dynamically Modify the ORDER BY Clause	21
Dynamically Modify the WHERE and ORDER BY Clauses of a ListBox	23
Dynamically Modify the WHERE Clause	24
Dynamically Modify the SQL Statement	26
Prevent Operation for Individual Records via Editable Grid.....	28
Retrieve Values From a Multi-Select ListBox.....	31
Working With Databases.....	34
Execute Custom SQL	34
Lookup Single Field Value in a Database.....	36
Retrieve Multiple Field Values from a Database.....	38
Utilize MySQL Enum Field Type.....	42
Validating Data Input.....	45
Using the Control Validation Event	45
Using the Form Validation Event	47
Processing Errors.....	48
Display Custom Error Messages	48
Dynamically Modifying HTML Output	52
Add Custom HTML to a Page.....	52
Change Control Value	54
Changing Table Cell Color.....	57
Changing Text Color.....	60
Working With Pages.....	64
Add URL Parameters to a Page	64
Working with Custom Template Blocks	66
Dynamically Redirect Users to Another Page.....	70
Dynamic Login Field in a Registration Form	72
Working With Panels.....	75
Iterating Panel Controls	82
Placing an Existing Component Within a New Panel.....	84
Restricting SQL Execution for Components Hidden Using a Panel.....	86
Working with Reports	87
Inserting Row Number.....	87
Translating Control Values	88
Using Action to Dynamically Translate Control Value.....	88
Translating Control Values via Custom Code	88
Other	89
Create a Grid Column with a Calculated Value	89

Create Custom Session Variables	92
Dynamically Modify a Hyperlink	94
Dynamically Set the Default Value of a ListBox	97
Examine Form's Mode	99
Replace Control Value Before Inserting.....	101
Working With Component Attributes.....	103
Debugging	105
Displaying Output.....	105
CodeCharge Studio Example Pack	109
CodeCharge Studio Example Solutions.....	111
Common Errors and Warnings	112
Common Errors	112
Could not load type error (.NET)	112
HTTP 500 Internal Server Error	113
Incorrect File Paths	113
Microsoft JET Database Engine (0x80004005) Could not use "; file already in use.	113
Operation must use an updateable query. (Microsoft JET Database Engine).....	114
Page takes forever to load or the IIS web server appears to hang.....	114
Warnings.....	115
Application Builder: Incomplete Data Message.....	115

Examples and Techniques

Tips and Tricks

Creating a MailTo Hyperlink

Description

This example shows how to create a MailTo hyperlink and run a client email system.

1. Click on the [Link](#) or [Image Link](#) icon in the Forms tab of the ToolBox to add the control.
2. Click on the [Control Source](#) property and select a database column to specify the visible text
3. Click on the [Href Source](#) property and select a database column to specify the hyperlink.
4. Modify the HTML code as shown below:

```
<a href="mailto:{email_Src}">{email}</a>
```

See [how to configure a hyperlink](#), if you want to modify a hyperlink dynamically.

Mailto syntax

Syntax

```
<a href="mailto:<address>[&subject=<subject text>][&body=<body text>][&cc=<cc address>][&bcc=<bcc address>]">
```

Possible Values

Keyword	Description
Address	Required. Specify one or more valid e-mail addresses separated by the semicolon using only the Internet-safe characters, for example '%20' for the space character.
Subject	Optional. Specify the text to appear in the subject of the email message.
Body	Optional. Specify the body of the email message.
CC	Optional. Specify the email addresses to be entered into the 'CC' (Carbon Copy) section of the email message.
BCC	Optional. Specify the email addresses to be entered in the 'BCC' (Blind Carbon Copy) section of the email message.

Example

The example below shows how to build a link to send an email message.

```
<A HREF="mailto:{email_Src}&subject=Feedback">{mail}</A>
```

See also

How to [dynamically Modify a Hyperlink](#)

Publishing Custom Files

This example shows how to publish the dynamically used project files for example images.

1. Select 'Project -> Settings' from the top menu.
2. Select the 'Publishing' tab.
3. Select the 'All files with extensions' or 'All files excluding extensions' option, for which you can modify the list of extensions.
4. Click 'OK'.

Showing a Grid Record in Two Rows

This example describes how to show a grid in two rows. For example, the 'Tasks' grid contains a 'Description' field that will be shown in the next row.

Modify the 'HTML' code as shown below:

```
<!-- BEGIN Row -->
<tr>
  <td>{Task_id}</td>
  <td>{TaskName}</td>
  <td>{Priority}</td>
  <td>{Status}</td>
</tr>
</tr>
  <td colspan="4">{Description}</td>
</tr>
<!-- END Row -->
```

Programming

Simple Reporting

Simple Report With Group Headers

This example shows how to create a simple report. The Grid form in this example contains a list of employees whereby each employee belongs to a department. The Grid lists the employees (: *ID*, *Name* and *Title*) in such a manner that a *Header* row indicates the department of the employee records appearing in the rows below it.

ID	Name	Title
Finance		
2	John Smith	Manager
5	Rob McDonald	Manager
Software		
1	Bob Kuch	Sr. Developer

3	Pablo Sanchez	Developer
7	David Snyder	Developer
Support		
6	Stefan Fey	Support

For ASP, PHP, Perl, ColdFusion

Make sure that you are in HTML mode then:

1. Modify the HTML content as shown below. In this case, the code in [blue](#) has been added to the existing HTML code for the Grid.

```

<!-- BEGIN Row -->
  <!-- BEGIN Label department_name -->
  <tr>
    <td bgcolor="#336699" colspan="3"><font
color="White">{department_name}</font></td>
  </tr>
  <!-- END Label department_name -->
  <tr>
    <td class="ClearDataTD">{Label1} </td>
    <td class="ClearDataTD">{emp_name} </td>
    <td class="ClearDataTD">{title} </td>
  </tr>
<!-- END Row --

```

2. In the [Before Show Row](#) event of the grid, add the following code:

ASP

```

Dim DepName
Function departments_BeforeShowRow(Sender)

  If DepName = departments.department_name.Value Then
    departments.department_name.Visible = False
  Else
    departments.department_name.Visible = True
    DepName = departments.department_name.Value
  End If

End Function

```

PHP

```

function departments_BeforeShowRow(& $sender) {
  global $DepName;
  global $departments;

  if ($DepName == $departments->department_name->GetValue()) {
    $departments->department_name->Visible = False;
  } else {
    $departments->department_name->Visible = True;
  }
}

```

```

    $DepName = $departments->department_name->GetValue();
}
}

```

Perl

```

sub departments_BeforeShowRow() {

    if ($DepName eq $departments->{department_name}->GetValue()) {
        $departments->{department_name}->{Visible} = 0;
    } else {
        $departments->{department_name}->{Visible} = 1;
        $DepName = $departments->{department_name}->GetValue();
    }
}
}

```

ColdFusion

```

<!--departments_BeforeShowRow --->
<CFPARAM Name="DepName" Default="">
<CFIF DepName EQ flddepartment name>
    <CFSET hideDepartment name= True>
<CFELSE>
    <CFSET hideDepartment name=False>
    <CFSET DepName=flddepartment name>
</CFIF>

```

Java

```

//'departments_BeforeShowRow

String depName = (String)
e.getPage().getRequest().getAttribute("departmentname");
if ( depName == null ) {
    depName = "";
}
if
(depName.equals(e.getGrid().getControl("department_name").getValue())
) {
    e.getGrid().getControl("department_name").setVisible(false);
} else {
    e.getGrid().getControl("department_name").setVisible(true);
    e.getPage().getRequest().setAttribute("departmentname",
e.getGrid().getControl("department_name").getValue());
}
}

```

.NET

1. If you are using C# or VB.Net, modify the HTML as follows:

```

<!-- BEGIN Row -->
  <tr id="departmentsNameTR" runat="server">
    <td bgcolor="#336699" colspan="3"><font
color="White">{department_name}</font></td>
  </tr>
  <tr>
    <td class="ClearDataTD">{Label1} </td>
    <td class="ClearDataTD">{emp_name} </td>
    <td class="ClearDataTD">{title} </td>
  </tr>
<!-- END Row -->

```

- Then within the code for the page, add the variable called *DepName* below the section that ends with the text *'End Forms Definition*

VB.Net

```

'End Forms Definition
  Dim DepName As String

```

C#

```

//End Forms Definition
  protected String DepName;

```

- Then add the following code in the [Before Show Row](#) event of the grid.

VB.Net

```

'departments_BeforeShowRow

  If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =
ListItemType.AlternatingItem Then
    Dim departmentsNameTR As Control =
e.Item.FindControl("departmentsNameTR")
    If DepName = DataItem.department_name.Value Then
      departmentsNameTR.Visible = False
    Else
      departmentsNameTR.Visible = True
      DepName = DataItem.department_name.Value
    End If
  End If

```

C#

```

//departments_BeforeShowRow

  if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem) {
    Control departmentsNameTR =
e.Item.FindControl("departmentsNameTR");

```

```

if (DepName == DataItem.department_name.Value.ToString()) {
    departmentsNameTR.Visible = false;
} else {
    departmentsNameTR.Visible = true;
    DepName = DataItem.department_name.Value.ToString();
}
}

```

See also:

[Before Show Row](#) event, [Grid](#) form

Simple Report with Simple Grouping

This example shows how to hide duplicate values on a Grid form. In this case, the Grid has a column for *Department* and another for *Employee*. The records in the Grid are sorted by department so records with the same department appear in consecutive rows. Instead of having the department name appear in each row, this example shows how to suppress the department name if it already appeared in a higher row.

Department	Employee
Administration	John Smith
	Rob McDonald
Software	Bob Kuch
	David Snyder

1. Add the [Before Show Row](#) event for the [Grid](#) form.
2. Within the event, add the code below:

ASP

```

Dim PrevDepartment

Function Employees_BeforeShowRow(Sender)

    If PrevDepartment = Employees.Department.Value Then
        Employees.Department.Value = ""
    Else
        PrevDepartment = Employees.Department.Value
    end if

End Function

```

PHP

```

function Employees_BeforeShowRow(& $sender) {
    global $Employees;
    global $PrevDepartment;

    if ($PrevDepartment == $Employees->department->GetValue()) {

```

```

    $Employees->department->SetValue("");
} else {
    $PrevDepartment = $Employees->department->GetValue();
}
}

```

Perl

```

sub Employees_BeforeShowRow() {

    if ($PrevDepartment eq $Employees->{department}->GetValue()) {
        $Employees->{department}->SetValue("");
    } else {
        $PrevDepartment = $Employees->{department}->GetValue();
    }

}

```

ColdFusion

```

<!--Employees_BeforeShowRow --->
<CFPARAM Name="PrevDepartment" Default="">
<CFIF PrevDepartment EQ fldDepartment>
    <CFSET fldDepartment="">
<CFELSE>
    <CFSET PrevDepartment = fldDepartment>
</CFIF>

```

VB.Net

1. Add a variable called *PrevDepartment* below the *End Forms Definition* section
2. Add the [Before Show Row](#) event for the grid.
3. Within the event, add the code below:

```

Dim PrevDepartment As String

'Employees_BeforeShowRow
If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =
ListItemType.AlternatingItem Then
    Dim EmployeesDepartment As System.Web.UI.WebControls.Literal =
DirectCast(e.Item.FindControl("EmployeesDepartment"), System.Web.UI.Web
Controls.Literal)
    If PrevDepartment = DataItem.Department.Value Then
        EmployeesDepartment.Text = ""
    Else
        PrevDepartment = DataItem.Department.Value
    End If
End If

```

C#

1. Add a variable called *PrevDepartment* below the *End Forms Definition* section
2. Add the [Before Show Row](#) event for the grid.
3. Within the event, add the code below:

```
protected string PrevDepartment;

//Employees_BeforeShowRow
if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem) {
    System.Web.UI.WebControls.Literal EmployeesDepartment =
(System.Web.UI.WebControls.Literal)e.Item.FindControl("EmployeesDepart
ment");
    if (PrevDepartment == (DataItem.Department.Value).ToString()) {
        EmployeesDepartment.Text = "";
    } else {
        PrevDepartment = (DataItem.Department.Value).ToString();
    }
}
```

Java

1. Add the [Before Show Row](#) event for the [Grid](#) form.
2. Within the event, add the code below:

```
//departments_BeforeShowRow

String depName = (String)
e.getPage().getRequest().getAttribute("departmentname");
if ( depName == null ) {
    depName = "";
}
if (depName.equals(e.getGrid().getControl("Department").getValue())
) {
    e.getGrid().getControl("Department").setValue("");
} else {
    e.getPage().getRequest().setAttribute("departmentname",
e.getGrid().getControl("Department").getValue());
}
```

See also:

[Before Show Row](#) event, [Grid](#) form

Simple Report with Total

This example shows how to create a simple report with a *Total* row at the bottom of the grid. The Grid has a column for the Employee name and another for the number of hours spent by the employee. The bottom row sums up the spent hours for all the employees listed in the grid.

Employee	Spent Hours
John Smith	10

Rob McDonald	20
Bob Kuch	15
Total:	45

1. Switch to HTML mode and add the HTML code in [blue](#) below the *Row* block.

```

2.      <!-- BEGIN Row -->
3.      <tr>
4.          <td class="ClearDataTD">{Employee}</td>
5.          <td class="ClearDataTD">{SpentHours}</td>
6.      </tr>
7.      <!-- END Row -->
8.      <tr>
9.          <td class="ClearDataTD"><b>Total</b></td>
10.         <td class="ClearDataTD"></td>
11.     </tr>

```

12. Add a **Summ** [Label](#) control into the last cell of the added code. (Note that we use 'Summ' and not 'Sum' because 'Sum' is a reserved word in some of the languages including SQL).

```

13.     <!-- BEGIN Row -->
14.     <tr>
15.         <td class="ClearDataTD">{Employee}</td>
16.         <td class="ClearDataTD">{SpentHours}</td>
17.     </tr>
18.     <!-- END Row -->
19.     <tr>
20.         <td class="ClearDataTD"><b>Total</b></td>
21.         <td class="ClearDataTD">{Summ}</td>
22.     </tr>

```

23. Set the [Data Type](#) property to *Integer* or *Float* for the *SpentHours* and *Summ* Label controls.

24. In the [Before Show Row](#) event of the grid, add the code below:

ASP

```

Function Employees_BeforeShowRow(Sender)

    Employees.Summ.Value = Employees.Summ.Value +
    Employees.SpentHours.Value

End Function

```

PHP

```

function Employees_BeforeShowRow(& $sender) {
    global $Employees;

    $Employees->Summ->SetValue ($Employees->Summ->GetValue () +
    $Employees->SpentHours->GetValue () );
}

```

```
}
```

Perl

```
sub Employees_BeforeShowRow() {  
  
    $Employees->{Summ}->SetValue($Employees->{Summ}->GetValue() +  
    $Employees->{SpentHours}->GetValue() );  
  
}
```

ColdFusion

```
<!--Employees_BeforeShowRow -->  
  
<CFSET fldSumm=Val(fldSumm) + fldSpentHours>
```

Java

```
//Employees_BeforeShowRow  
  
    long summ = e.getGrid().getControl("Summ").getValue() == null ? 0 :  
    Utils.convertToLong(e.getGrid().getControl("Summ").getValue()).longVal  
    ue();  
    long spent = e.getGrid().getControl("SpentHours").getValue() == null  
    ? 0 :  
    Utils.convertToLong(e.getGrid().getControl("SpentHours").getValue()).l  
    ongValue();  
    e.getGrid().getControl("Summ").setValue( summ + spent);
```

VB.Net

1. Add a variable called *Summ* below the *'End Forms Definition* section.

```
'End Forms Definition  
    Dim Summ As Integer ' or Double
```

2. Add the [Before Show Row](#) event for the Grid.

```
'Employees_Summ_BeforeShow  
  
    EmployeesSumm.Text = Summ.ToString()
```

3. Add the [Before Show](#) event for the *Summ* Label.

```
'Employees_BeforeShowRow  
  
    If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =  
    ListItemType.AlternatingItem Then  
        Summ += DataItem.SpentHours.Value  
    End If
```

C#

1. Add a variable called *Summ* below the *'End Forms Definition* section.

```
//End Forms Definition

protected Int64 Summ;
```

2. Add the [Before Show Row](#) event for the Grid.

```
//Employees_Summ_BeforeShow

EmployeesSumm.Text = Summ.ToString();
```

3. Add the [Before Show](#) event for the *Summ* Label.

```
//Employees_BeforeShowRow

if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem) {
    Summ += (Int64) (DataItem.SpentHours.Value);
}
```

See also:

[Before Show Row](#) event, [Grid](#) form, [Label](#) control

Hiding Components

Hiding a Grid If No Records are Found

This example shows how to hide a Grid if no records are found to be displayed within the grid.

Assume that a *Responses* Grid is located below a *Topics* grid on a page which implements a forum. You can use the code below to hide the *Responses* Grid if there are no posted responses.

1. In the [Before Show](#) event, add the code shown below:

ASP

```
Function Responses_BeforeShow()

If Responses.Recordset.EOF Then
    Responses.Visible = False
End If

End Function
```

PHP

```
function Responses_BeforeShow() {
global $Responses;

if ($Responses->DataSource->RecordsCount == 0) {
    $Responses->Visible = False;
}
```

```
}  
  
}
```

Perl

```
sub Responses_BeforeShow() {  
  
    if ($Responses->{DataSource}->{RecordsCount} == 0) {  
        $Responses->{Visible} = 0;  
    }  
  
}
```

ColdFusion

```
<!---Responses_BeforeShow --->  
  
<CFOUTPUT>#recordCountResponses#</CFOUTPUT>  
<CFIF recordCountResponses EQ 0>  
    <CFSET hideResponses=True>  
</CFIF>
```

VB.Net

```
'Responses_BeforeShow  
  
If PagesCount = 0 Then  
    ResponsesRepeater.Visible = False  
End if
```

C#

```
//Responses_BeforeShow  
  
if (PagesCount == 0) {  
    ResponsesRepeater.Visible = false;  
}
```

Java

```
//Responses_BeforeShow  
  
if ( e.getGrid().isEmpty() ) {  
    e.getGrid().setVisible(false);  
}
```

See also:

[Before Show](#) event

Hiding a Record Form Control

This example shows how to hide a control on a [Record](#) form. In this case, a [Record](#) form called *Responses* has a [Text box](#) control called *Author* that should not be displayed.

1. Switch to HTML mode and locate the code for the *Author* Textbox control. Just before the code for the *Author* Textbox control, add the code `<!-- BEGIN TextBox Author -->` then after the code for the *Author* Textbox control, add the code `<!-- END TextBox Author -->`:

```
2.
3.         <tr>
4.             <td>Author</td>
5.             <td>
6.                 <!-- BEGIN TextBox Author -->
7.                 <input name="Author" value="{Author}">
8.                 <!-- END TextBox Author -->
9.             </td>
```

```
</tr>
```

Or you can also add the code at the beginning and at the end of the table row to hide the control together with its caption:

```
<!-- BEGIN TextBox Author -->
<tr>
  <td>Author</td>
  <td><input name="Author" value="{Author}"></td>
</tr>
<!-- END TextBox Author -->
```

Control Level

One way to hide a control is to place code in the [Before Show](#) event of the control itself. Another method is to add code in the [Before Show](#) event of the form in which the control is contained. We shall begin with the first method where the code is added to the [Before Show](#) event of the control itself.

1. Add the [Before Show](#) event to the control.
2. Within the event, add the code below:

ASP

```
Function Responses_Author_BeforeShow()

    Responses.Author.Visible = False

End Function
```

PHP

```
function Responses_Author_BeforeShow() {
    global $Responses;
```

```
$Responses->Author->Visible = False;

}
```

Perl

```
sub Responses_Author_BeforeShow() {

    $Responses->{Author}->{Visible} = 0;

}
```

ColdFusion

```
<!--Responses_Author_BeforeShow -->

<CFSET hideAuthor = True>
```

VB.Net

```
'Responses_Author_BeforeShow

ResponsesAuthor.Visible = False
```

C#

```
//Responses_Author_BeforeShow

ResponsesAuthor.Visible = false;
```

Java

```
//Responses_Author_BeforeShow

e.getControl().setVisible(false);
```

Form Level

The second method of hiding controls is to place code in the [Before Show](#) event of the form in which the control is contained. Using this method, you can hide multiple controls within the form since they are all accessible at the form level. This is not the case when the code is added to the [Before Show](#) event of one of the controls.

1. Add the [Before Show](#) event to the form.
2. Within the event, add the code below:

ASP

```
Function Responses_BeforeShow()

    Responses.Author.Visible = False
```

```
Responses.AuthorEmail.Visible = False

End Function
```

PHP

```
function Responses_BeforeShow() {
global $Responses;

$Responses->Author->Visible = False;
$Responses->AuthorEmail->Visible = False;

}
```

Perl

```
sub Responses_BeforeShow() {

$Responses->{Author}->{Visible} = 0;
$Responses->{AuthorEmail}->{Visible} = 0;

}
```

ColdFusion

```
<!--Responses_BeforeShow --->

<CFSET hideAuthor = True>
```

VB.Net

```
'Responses_BeforeShow

ResponsesAuthor.Visible = False
```

C#

```
//Responses_BeforeShow

ResponsesAuthor.Visible = false;
```

Java

```
//Responses_BeforeShow

e.getRecord().getControl("Author").setVisible(false);
```

See Also:

[Before Show](#) event, How to [create a dynamic Login Control](#)

Customizing the Data Source

Dynamically Modify the "List Of Values" of a ListBox

The values in a List Box control can be specified by using a list of values which appear in pairs. Each entry in a List Box is composed of a value which is displayed and another which is submitted, hence the pairs. Using event code, you can specify the pairs that will make up the entries of the ListBox.

In this example, we have a *Status* [List Box](#) which is located in a *Tasks* Record form.

1. Select *ListOfValues* in the [Source Type](#) property of the List Box.
2. In the [Before Show](#) event to the Record form, add the custom code shown below:

ASP

```
Function Tasks_BeforeShow(Sender)

    Set Tasks.Status.DataSource = CCCreateDataSource(dsListOfValues,
Empty, _
    Array(Array("1","2","3"),Array("High","Normal","Low")))

End Function
```

PHP

```
function Tasks_BeforeShow(& $sender) {
    global $Tasks;

    $Tasks->Status->Values = array(array("1", "High"),array("2",
"Normal"), array("3", "Low"));
}
```

Perl

```
sub Tasks_BeforeShow() {

    $Tasks->{Status}->{Values} = [{"1", "High"}, {"2", "Normal"}, {"3",
"Low"}];

}
```

ColdFusion

```
<!--Tasks_BeforeShow --->

<CFSET arr_tasksStatus=ListToArray("1;High;2;Normal;3;Low",";")>
```

VB.Net

```
'Tasks_BeforeShow

item.statusItems.Clear()
tasksstatus.Items.Clear()
```

```

item.statusItems.Add("", "Select Value")
item.statusItems.Add("1", "High")
item.statusItems.Add("2", "Normal")
item.statusItems.Add("3", "Low")
item.statusItems.CopyTo(tasksstatus.Items)

```

C#

```

//Tasks_BeforeShow

item.statusItems.Clear();
tasksstatus.Items.Clear();
item.statusItems.Add("", "Select Value");
item.statusItems.Add("1", "High");
item.statusItems.Add("2", "Normal");
item.statusItems.Add("3", "Low");
item.statusItems.CopyTo(tasksstatus.Items);

```

Java

```

//Tasks_BeforeShow

e.getRecord().getList("status").setListOfValues("1;High;2;Normal;3;Low");

```

See Also:

[Before Show](#) event

Dynamically Modify the ORDER BY Clause

This example shows how to modify the Order By clause for a Grid form. Within an SQL statement, the Order By clause determines the order in which the selected records will be sorted. In this case, we have a *Tasks* Grid that is based on the *tasks* database table which has a text field called *task_name*. The code below will cause the records in the Grid to be sorted by *task_name* in descending order.

1. Add the [Before Build Select](#) event to the grid.
2. Within the event, add the code below:

ASP

```

Function Tasks_DataSource_BeforeBuildSelect(Sender)

If Tasks.DataSource.Order = "" Then
    Tasks.DataSource.Order = "task_name DESC"
End if

End Function

```

PHP

```
function Tasks_DataSource_BeforeBuildSelect(& $sender) {
    global $Tasks;

    If ($Tasks->DataSource->Order == "") {
        $Tasks->DataSource->Order = "task_name DESC";
    }
}

```

Perl

```
sub Tasks_DataSource_BeforeBuildSelect() {

    if ($Tasks->{DataSource}->{Order} eq "") {
        $Tasks->{DataSource}->{Order} = "task_name DESC";
    }

}

```

ColdFusion

```
<!--Tasks_BeforeBuildSelect --->

<CFIF strOrder eq "">
    <CFSET strOrder= "task_name DESC">
</CFIF>

```

Java

```
// Tasks_BeforeBuildSelect

if (StringUtils.isEmpty(e.getCommand().getOrder())) {
    e.getCommand().setOrder("task_name DESC");
}

```

VB.Net

1. Add the [Before Execute Select](#) event to the grid.
2. Within the event, add the code below:

```
' Tasks_BeforeExecuteSelect

If Select_.OrderBy = "" Then
    DirectCast(Select_,TableCommand).OrderBy = "task_name DESC"
End if

```

C#

1. Add the [Before Execute Select](#) event to the grid.
2. Add the code below:

```
// Tasks_BeforeExecuteSelect

```

```
if (Select.OrderBy == "") {
    Select.OrderBy = "task_name DESC";
}
```

See Also:

[Before Build Select](#) event

Dynamically Modify the WHERE and ORDER BY Clauses of a ListBox

When the the [List Source Type](#) property of a list box is set to *Table/View*, the values that are used to populate the Listbox come from a database table or view. In this case, it is possible to use event code to specify the Where clause as well as the Order By clause which will be used in the SQL statement that retrieves records to be shown in the Listbox.

In our example, we have a *Status* [List Box](#) that is located in a *Tasks* Record form.

1. In the **Properties** Window for the Listbox, set the [Source Type](#) property to *Table/View*.
2. Add the [Before Build Select](#) event to the Listbox.
3. Within the event, add the code below:

ASP

```
Function Tasks_Status_DataSource_BeforeBuildSelect(Sender)

    Tasks.Status.DataSource.Where = " status_id >2"
    Tasks.Status.DataSource.Order = " status_id DESC"

End Function
```

PHP

```
function Tasks_Status_DataSource_BeforeBuildSelect(& $sender) {
    global $Tasks;

    $Tasks->Status->DataSource->Where = " status_id >2";
    $Tasks->Status->DataSource->Order = " status_id DESC";

}
```

Perl

```
sub Tasks_Status_DataSource_BeforeBuildSelect() {

    $Tasks->{Status}->{DataSource}->{Where} = " status_id >2";
    $Tasks->{Status}->{DataSource}->{Order} = " status_id DESC";

}
```

ColdFusion

```
<!--Tasks_status_BeforeBuildSelect --->

<CFSET strWhereStatus=" status_id > 2">
<CFSET strOrderStatus=" status_id DESC">
```

Java

```
//Tasks_Status_BeforeBuildSelect

e.getCommand().setWhere(" status_id > 2");
e.getCommand().setOrder(" status_id DESC");
```

VB.Net

```
'Tasks_Status_BeforeBuildSelect

CType(statusDataCommand,TableCommand).Where="age_id > 5"
statusDataCommand.OrderBy="age_name"
```

C#

```
//Tasks_Status_BeforeBuildSelect

((TableCommand)statusDataCommand).Where="age_id > 5";
statusDataCommand.OrderBy="age_name";
```

See Also:

[Before Build Select](#) event

Dynamically Modify the WHERE Clause

This example shows how to modify the WHERE clause of a Grid form. Within an SQL statement, the Where clause is used to specify the criteria which will be used to select database records.

In this case, we have a *Tasks* grid that is based on the *tasks* database table which has a text field called *task_name*. The Grid will show only those task records whose names start with *"Improve" phrase*.

1. Add the [Before Build Select](#) event to the grid.
2. Within the event, add the code below:

ASP

```
Function Tasks_DataSource_BeforeBuildSelect(Sender)

If Tasks.DataSource.Where <> Empty Then
    Tasks.DataSource.Where = Tasks.DataSource.Where & " AND "
End if
```

```
Tasks.DataSource.Where = Tasks.DataSource.Where & "task_name  
like 'Improve%'"
```

```
End Function
```

Note: If the OR operator is used in the WHERE clause the conditions must be in brackets "()". For example:

```
Tasks.DataSource.Where = Tasks.DataSource.Where & "(task_name  
like 'Improve%' OR task_name like 'Add%')"
```

PHP

```
function Tasks_DataSource_BeforeBuildSelect(& $sender) {  
    global $Tasks;  
  
    if ($Tasks->DataSource->Where <> "") {  
        $Tasks->DataSource->Where .= " AND ";  
    }  
  
    $Tasks->DataSource->Where .= "task_name like 'Improve%'";  
  
}
```

Note: If the OR operator is used in the WHERE clause the conditions must be in brackets "()". For example:

```
$Tasks->DataSource->Where .= "(task_name like 'Improve%' OR  
task_name like 'Add%')";
```

Perl

```
sub Tasks_DataSource_BeforeBuildSelect() {  
  
    if ($Tasks->{DataSource}->{Where} ne "") {  
        $Tasks->{DataSource}->{Where} .= " AND ";  
    }  
  
    $Tasks->{DataSource}->{Where} .= "task_name like 'Improve%'";  
  
}
```

Note: If the OR operator is used in the WHERE clause the conditions must be in brackets "()". For example:

```
$Tasks->{ds}->{Where} .= "(task_name like 'Improve%' OR task_name  
like 'Add%')";
```

ColdFusion

```
<!--Tasks_BeforeBuildSelect -->  
<CFIF strWhere NEQ "">  
    <CFSET strWhere="#strWhere# AND ">  
</CFIF>
```

```
<CFSET strWhere="#strWhere#task_name like 'Improve%'">
```

Note: If the OR operator is used in the WHERE clause the conditions must be in brackets "()". For example:

```
<CFSET strWhere="#strWhere#(task_name like 'Improve%' OR task_name like 'Add%')">
```

Java

```
//Tasks_BeforeBuildSelect

if (! StringUtils.isEmpty(e.getCommand().getWhere())) {
    e.getCommand().setWhere(e.getCommand().getWhere() + " AND
task_name like '%Improve%'");
} else {
    e.getCommand().setWhere("task_name like '%Improve%'");
}
```

VB.Net

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
'Tasks_BeforeExecuteSelect

DirectCast(Select_,TableCommand).Where = "task_name like 'Improve%"
DirectCast(Select_,TableCommand).Operation = "AND"
```

C#

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
//Tasks_BeforeExecuteSelect

((TableCommand)Select).Where = "task_name like 'Improve%";
((TableCommand)Select).Operation = "AND";
```

See Also:

[Before Build Select](#) event, [Before Execute Select](#) event

Dynamically Modify the SQL Statement

This example shows how to modify the SQL statement that is used to retrieve records for a Grid form. The example shows a Grid called *Tasks* that is based on the *tasks* database table.

ASP

1. Add the [Before Build Select](#) event to the Grid.
2. Within the event, add the code below:

```
Function Tasks_DataSource_BeforeBuildSelect(Sender)

    Tasks.DataSource.SQL = "SELECT * FROM Tasks"

End Function
```

PHP

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
function Tasks_DataSource_BeforeExecuteSelect(& $sender) {
    global $Tasks;

    $Tasks->DataSource->SQL = "SELECT * FROM Tasks";
}
```

Perl

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
sub Tasks_DataSource_BeforeExecuteSelect() {

    $Tasks->{DataSource}->{SQL} = "SELECT * FROM Tasks";

}
```

Java

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
//Tasks_BeforeExecuteSelect

e.getCommand().setSql("SELECT * FROM Tasks");
```

VB.Net

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
'Tasks_BeforeExecuteSelect

CType(Select_,TableCommand).SqlQuery = new StringBuilder("select top 3
* from ages")
```

C#

1. Add the [Before Execute Select](#) event to the Grid.
2. Within the event, add the code below:

```
//Tasks_BeforeExecuteSelect
```

```
((TableCommand)Select).SqlQuery = new StringBuilder("select top 3 *  
from ages");
```

See Also:

[Before Build Select](#) event, [Before Execute Select](#) event

Prevent Operation for Individual Records via Editable Grid.

This example shows how to prevent an operation (Insert, Update, Delete) for individual records via Editable Grid. In this case, we have an Editable Grid called 'Tasks' which is based on the 'tasks' database table.

When the Editable Grid is submitted, corresponding events are called for each record. Insert will not be performed if the user assigns the task to himself or herself. If the current user is not the one who the task is assigned to, the Update operation for this record is not performed. If the user is not the one who assigned the task, the Delete operation is not performed.

In this example, we have a *Tasks* [Editable Grid](#).

1. Add the [Before Execute Insert](#) event to the Editable Grid
2. Add the [Before Execute Update](#) event to the Editable Grid
3. Add the [Before Execute Delete](#) event to the Editable Grid
4. Within the events, add the code below:

ASP

```
Function Tasks_DataSource_BeforeExecuteInsert(Sender)  
    If CCGetUserID\(\) = Tasks.user_id_assign_to.Value Then  
        Tasks.DataSource.CmdExecution = false  
    End if  
End Function  
  
Function Tasks_DataSource_BeforeExecuteUpdate(Sender)  
    If CCGetUserID\(\) <> Tasks.user_id_assign_to.Value Then  
        Tasks.DataSource.CmdExecution = false  
    End if  
End Function  
  
Function Tasks_DataSource_BeforeExecuteDelete(Sender)  
    If CCGetUserID\(\) <> Tasks.user_id_assign_by.Value Then  
        Tasks.DataSource.CmdExecution = false  
    End if  
End Function
```

PHP

```
function Tasks_DataSource_BeforeExecuteInsert(& $sender) {  
    global $Tasks;  
    if (CCGetUserID\(\) == $Tasks->user_id_assign_to->GetValue\(\)) {  
        $Tasks->DataSource->CmdExecution = false;  
    }  
}
```

```

}

function Tasks_DataSource_BeforeExecuteUpdate(& $sender) {
global $Tasks;
    if (CCGetUserID() != $Tasks->user_id_assign_to->GetValue()) {
        $Tasks->DataSource->CmdExecution = false;
    }
}

function Tasks_DataSource_BeforeExecuteDelete(& $sender) {
global $Tasks;
    if (CCGetUserID() != $Tasks->user_id_assign_by->GetValue()) {
        $Tasks->DataSource->CmdExecution = false;
    }
}
}

```

Perl

```

sub Tasks_DataSource_BeforeExecuteInsert() {
    if (CCGetUserID() == $Tasks->{user_id_assign_to}->GetValue()) {
        $Tasks->{DataSource}->{CmdExecution} = 0;
    }
}

sub Tasks_DataSource_BeforeExecuteUpdate() {
    if (CCGetUserID() <> $Tasks->{user_id_assign_to}->GetValue()) {
        $Tasks->{DataSource}->{CmdExecution} = 0;
    }
}

sub Tasks_DataSource_BeforeExecuteDelete() {
    if (CCGetUserID() != $Tasks->{user_id_assign_by}->GetValue()) {
        $Tasks->{DataSource}->{CmdExecution} = 0;
    }
}
}

```

ColdFusion

```

<!--- Tasks_BeforeExecuteInsert --->
<CFIF Session.UserID EQ flduser_id_assign_to>
    <CFSET endOperationTasks=True>
</CFIF>

<!--- Tasks_BeforeExecuteUpdate --->
<CFIF Session.UserID NEQ flduser_id_assign_to>
    <CFSET endOperationTasks=True>
</CFIF>

<!--- Tasks_BeforeExecuteDelete --->

```

```
<CFIF Session.UserID NEQ flduser_id_assign_to>
  <CFSET endOperationTasks=True>
</CFIF>
```

Java

```
//Tasks_BeforeExecuteUpdate
Object ctrlValue =
e.getEditableGrid().getControl("user_id_assign_to").getFormattedValue(
);
String user_id = Utils.getUserId(e.getPage());
if (user_id == null || !user_id.equals(ctrlValue)) {
  e.getCommand().setCmdExecution(false);
}

//Tasks_BeforeExecuteDelete
Object ctrlValue =
e.getEditableGrid().getControl("user_id_assign_to").getFormattedValue(
);
String user_id = Utils.getUserId(e.getPage());
if (user_id == null || !user_id.equals(ctrlValue)) {
  e.getCommand().setCmdExecution(false);
}
```

VB.Net

```
'Tasks_BeforeExecuteUpdate
If DBUtility.UserID = item.user_id_assign_to.Value Then
  CmdExecution = False
End If

'Tasks_BeforeExecuteDelete
If DBUtility.UserID = item.user_id_assign_to.Value Then
  CmdExecution = False
End If

'Tasks_BeforeExecuteInsert
If DBUtility.UserID = item.user_id_assign_to.Value Then
  CmdExecution = False
End If
```

C#

```
//Tasks_BeforeExecuteUpdate
if(DBUtility.UserID == item.user_id_assign_to.Value)
  CmdExecution = false;

//Tasks_BeforeExecuteDelete
if(DBUtility.UserID == item.user_id_assign_to.Value)
  CmdExecution = false;
```

```
//Tasks_BeforeExecuteInsert
if(DBUtility.UserId == item.user_id_assign_to.Value)
    CmdExecution = false;
```

See Also:

[Before Execute Insert](#) event,
[Before Execute Update](#) event,
[Before Execute Delete](#) event

Retrieve Values From a Multi-Select ListBox

This example shows how to retrieve values from a Listbox which allows for the selection of multiple values. In this case, we have a Grid called *Tasks* which receives an input parameter called *s_Project*. The value of *s_Project* originates from a multiselect Listbox control in a Search form.

1. Set the *Multiple* property in the **Format** tab of the Listbox or modify the tag in HTML mode by entering *multiple* in the tag.

```
<select name="{s_Project_Name}" multiple>
```

2. Add the [Before Build Select](#) event to the grid.
3. Within the event, add the code below:

ASP

```
Function Tasks_DataSource_BeforeBuildSelect(Sender)
' "s_Project" contains the comma delimited string with the selected
values

if CCGetParam("s_Project", Empty) <> Empty then
    If Tasks.DataSource.Where <> Empty Then _
        Tasks.DataSource.Where = Tasks.DataSource.Where & " AND "

    Tasks.DataSource.Where = Tasks.DataSource.Where & _
        "task_project_id IN (" & CCGetParam("s_Project", Empty) & ")"
End if

End Function
```

PHP

```
function Tasks_DataSource_BeforeBuildSelect(& $sender) {
global $Tasks;

$s_Project = CCGetParam("s_Project", "");
if (count($s_Project) > 0 AND is_array($s_Project)) {
    foreach ($s_Project as $key => $value) {
        if ($Projects != "") $Projects = $Projects.", ";
        $Projects = $Projects.$value;
    }
}

if ($Tasks->DataSource->Where != "")
```

```

        $Tasks->DataSource->Where .= " AND ";
    if ($Projects != "")
        $Tasks->DataSource->Where .= " tasks.project_id IN
        (\".$Projects .)\"";
    }
}

```

Perl

```

sub Tasks_DataSource_BeforeBuildSelect() {

    $s_Project = CCGetParam("s_Project", "");

    if (ref($s_Project) eq "ARRAY") {
        for ($i=0; $i <= $#{$s_Project}; $i++ ) {
            if ($Projects ne "") {
                $Projects .= ",";
            }
            $Projects .= $s_Project->[$i];
        }
    } elsif (length($s_Project)) {
        $Projects = $s_Project;
    }

    if ($Tasks->{DataSource}->{Where} ne "") {
        $Tasks->{DataSource}->{Where} .= " AND ";
    }

    if ($Projects ne "" ) {
        $Tasks->{DataSource}->{Where} .= " tasks.project_id IN
        (\".$Projects .)\"";
    }
}

```

ColdFusion

```

<!---Tasks_BeforeBuildSelect --->
<CFMODULE Template="CCGetParam.cfm" strName="s_Project"
outputVar="Projects">
    <!-- "Projects" contains the comma delimited string with the
    selected values -->
    <CFIF Projects NEQ "">
        <CFIF strWhere NEQ "">
            <CFSET strWhere=strWhere & " AND ">
        </CFIF>
        <CFSET strWhere=strWhere & " tasks.project_id IN (#Projects#)">
    </CFIF>

```

Java

```

//Tasks_BeforeBuildSelect

```

```

String[] s_Project =
e.getPage().getHttpGetParams().getParameterValues("s_Project");

if (s_Project != null && s_Project.length > 0 ) {
    StringBuffer projects = new StringBuffer();
    for ( int i = 0; i < s_Project.length; i++ ) {
        if ( projects.length() > 0 ) {
            projects.append(", ");
        }
        projects.append(s_Project[i]);
    }
    StringBuffer where = null;
    if ( ! StringUtils.isEmpty(e.getCommand().getWhere()) ) {
        where = new StringBuffer(e.getCommand().getWhere()).append(" AND
");
    } else {
        where = new StringBuffer();
    }

    if (projects.length() > 0) {
        where.append("tasks.project_id IN (" + projects.toString() + ")");
        e.getCommand().setWhere(where.toString());
    }
}
}

```

VB.Net

```

'Tasks_BeforeBuildSelect

If IsNothing(System.Web.HttpContext.Current.Request("s_Project")) Then
    CType(Select, TableCommand).Where = "tasks.project_id IN (" +
System.Web.HttpContext.Current.Request("s_Project") + ")"
    CType (Select, TableCommand).Operation = "AND"
    CType (Count, TableCommand).Where = "tasks.project_id IN (" +
System.Web.HttpContext.Current.Request("s_Project") + ")"
    CType (Count, TableCommand).Operation = "AND"
End If

```

C#

```

//Tasks_BeforeBuildSelect

if(System.Web.HttpContext.Current.Request["s_Project"]!=null)
{
    ((TableCommand)Select).Where = "tasks.project_id IN (" +
System.Web.HttpContext.Current.Request["s_Project"] + ")";
    ((TableCommand)Select).Operation = "AND";
    ((TableCommand)Count).Where = "tasks.project_id IN (" +
System.Web.HttpContext.Current.Request["s_Project"] + ")";
}

```

```
((TableCommand)Count).Operation = "AND";  
}
```

See Also:

- [Before Build Select](#) event
- [List Box](#) component

Working With Databases

Execute Custom SQL

This example shows how to execute a custom SQL statement. In this case, we have a Record form called *Tasks* which is based on the *tasks* database table. When a record is updated using the Record form, an SQL statement is executed to insert a new entry into the *reports* database table.

1. Add the [After Execute Update](#) event to the record.
2. Within the event, add the code below:

ASP

```
Function Tasks_DataSource_AfterExecuteUpdate(Sender)  
Dim SQL  
Dim Connection  
Dim ErrorMessage  
  
Set Connection = New clsDBConnection1  
Connection.Open  
SQL = "INSERT INTO report (report_task_id,report_creator) "&  
      "VALUES ("&  
Connection.ToSQL(CCGetFromGet("task_id",0),ccsInteger) &","&  
Connection.ToSQL(CCGetUserID(),ccsInteger) &")"  
Connection.Execute(SQL)  
ErrorMessage = CCProcessError(Connection)  
Connection.Close  
Set Connection = Nothing  
On Error Goto 0  
  
End Function
```

PHP

```
function Tasks_DataSource_AfterExecuteUpdate(& $sender) {  
  
    $db = new clsDBConnection1();  
    $SQL = "INSERT INTO report (report_task_id,report_creator) ".  
          "VALUES (" . $db->  
>ToSQL(CCGetFromGet("task_id",0),ccsInteger) ."," . $db->  
>ToSQL(CCGetUserID(),ccsInteger) .")";  
  
    $db->query($SQL);  
    $db->close();  
}
```

```
}
```

Perl

```
sub Tasks_DataSource_AfterExecuteUpdate() {

    $db = clsDBConnection1->new();
    my $SQL = "INSERT INTO report (report_task_id,report_creator) ".
        "VALUES (" . $db-
>ToSQL(CCGetFromGet("task_id",0),$ccsInteger) ."," . $db-
>ToSQL(CCGetUserID(),$ccsInteger) .)";
    $db->query($SQL);
    $db->{sth} = undef;
    $db->{dbh} = undef;

}
```

ColdFusion

```
<!--Tasks_AfterExecuteUpdate --->

<CFIF strOperation EQ "Update" AND NOT endOperationTasks>
    <CFSET strSQL="INSERT INTO report (report_task_id,report_creator)
VALUES (">
    <CFMODULE Template="CCGetParam.cfm" strName="task_id" def="0">
    <CFMODULE Template="CCToSQL.cfm" Value="#CCGetParam#"
Type="#ccsInteger#">
    <CFSET strSQL=strSQL & CCToSQL & ",">
    <CFMODULE Template="CCToSQL.cfm" Value="#Session.UserID#"
Type="#ccsInteger#">
    <CFSET strSQL=strSQL & CCToSQL & ")">
    <CFModule Template="CCExecuteSQL.cfm" sql="#strSQL#"
Connection="Connection1">
</CFIF>
```

Java

```
//Tasks_AfterExecuteUpdate

JDBCConnection conn =
JDBCConnectionFactory.getJDBCConnection("Connection1");
String sql = "INSERT INTO report (report_task_id,report_creator)
VALUES (" +

conn.toSql(e.getPage().getHttpGetParameter("task_id","0"),JDBCConnecti
on.INTEGER)+
    "," +
conn.toSql(Utils.getUserId(e.getPage()),JDBCConnection.INTEGER) + ")";
conn.executeUpdate(sql);
conn.closeConnection();
```

VB.Net

```
'Tasks_AfterExecuteUpdate

Dim Request As HttpRequest = HttpContext.Current.Request
Dim NewDao As DataAccessObject =
Settings.Connection1DataAccessObject
Dim Sql As String = "INSERT INTO report
(report_task_id,report_creator) " & _
    "VALUES ("&
NewDao.ToSql(Request.QueryString("task_id"),FieldType._Integer) &","&
_
NewDao.ToSql(DBUtility.UserId.ToString(),FieldType._Integer) &")"
NewDao.RunSql(Sql)
```

C#

```
//Tasks_AfterExecuteUpdate

System.Web.HttpContext context=System.Web.HttpContext.Current;
DataAccessObject NewDao=Settings.Connection1DataAccessObject;
string Sql = "INSERT INTO report (report_task_id,report_creator) " +
    "VALUES ("+
NewDao.ToSql(context.Request.QueryString["task_id"],FieldType.Integer)
+", "+
NewDao.ToSql(DBUtility.UserId.ToString(),FieldType.Integer) +")";
NewDao.RunSql(Sql);
```

See Also:

[After Execute Update](#) event

Lookup Single Field Value in a Database

This example shows how to read a single value from a database. In this case, we have a *Header* page which contains a *UserName* Label that is used to display the user's name.

Method 1

Use the DLookup action.

1. Add the [DLookup](#) action to the [Before Show](#) event of the page.
2. Set the properties of the action.
Note: The values of the **Expression**, **Domain** and **Criteria** properties should not be entered in double quotes.

Method 2

1. Add the [Before Show](#) event to the page.
2. Within the event, add the code below:

ASP

```
Function Header_BeforeShow(Sender)
```

```

Header.UserName.Value = CCDLookup("user_name","users","user_id="
& DBConnection1.ToSQL(CCGetUserID(),ccsInteger), DBConnection1)

End Function

```

PHP

```

function Header_BeforeShow(& $sender) {
global $Header;
global $DBConnection1;

$Header->UserName-
>SetValue(CCDLookup("user_name","users","user_id=". $DBConnection1-
>ToSQL(CCGetUserID(), ccsInteger), $DBConnection1) );
}

```

Perl

```

sub Header_BeforeShow() {

$DBConnection1 = clsDBConnection1->new();
$result = CCDLookup("first_name", "users", "user_id="
$DBConnection1->ToSQL(CCGetUserID(), $ccsInteger), $DBConnection1);
$Header->{UserName}->SetValue($result);

}

```

ColdFusion

```

<!--Header_BeforeShow --->

<CFModule Template="CCDLookup.cfm" table="Users" field="first_name"
where="User_ID=1" Connection="Connection1">
<CFSET fldUserName=CCDLookup>

```

VB.Net

```

'Header_BeforeShow

UserName.Text = (New
TextField("",Settings.Connection1DataAccessObject.ExecuteScalar("SELEC
T user_name FROM users WHERE
user_id=" & DBUtility.UserId)).GetFormattedValue("")

```

C#

```

//Header_BeforeShow

UserName.Text = (new
TextField("",Settings.Connection1DataAccessObject.ExecuteScalar("SELEC

```

```
T user_name FROM users WHERE user_id=" +
DBUtility.UserId)).GetFormattedValue("");
```

Java

```
//Header_BeforeShow

e.getPage().getControl("UserName").setValue(
DBTools.dLookUp("User_Name", "Users", "User_ID="+Utils.getUserId(e.getPa
ge()), "Connection1"));
```

See Also:

[Before Show](#) event, How to [read multiple values from a database](#)

Retrieve Multiple Field Values from a Database

This example shows how to read values from a database using event code. In this case, the page contains a *UserInfo* label. If the *user_id* URL parameter is not empty, the name and work phone number corresponding to the *user_id* value will be displayed in the *UserInfo* label.

1. Add the [Before Show](#) event to the Label.
2. Within the event, add the code below:

ASP

```
Function UserInfo_BeforeShow(Sender)
    Dim SQL
    Dim RecordSet
    Dim UserId
    Dim UserName
    Dim WorkPhone

    ' Read user_id from URL
    UserId = CCGetFromGet("user_id", 0)

    If UserID > 0 then
        SQL = "SELECT emp_name, phone_work FROM employees WHERE
emp_id=" & UserId
        ' Open the recordset
        Set RecordSet = DBConnection1.Execute(SQL)
        If DBConnection1.Errors.Count = 0 Then
            If NOT RecordSet.EOF then
                UserName = CCGetValue(RecordSet, "emp_name")
                WorkPhone = CCGetValue(RecordSet, "phone_work")
            End if
        ' Close the recordset
        RecordSet.Close
        Set RecordSet = Nothing
    Else
        Print "SQL Execution Failed."
```

```

        DBConnection1.Errors.Clear
    End If

    ' Show a label value
    UserInfo.Value = UserName & ", phone: "&WorkPhone
Else
    UserInfo.Visible = False
End if
End Function

```

PHP

```

function UserInfo_BeforeShow(& $sender) {
global $UserInfo;

// Read user_id from URL
$UserId = CCGetFromGet("user_id", 0);

if ($UserId > 0) {
// Open connection
$db = new clsDBConnection1();
$sql = "SELECT emp_name, phone_work FROM employees WHERE
emp_id=".$UserId;
$db->query($sql);
$result = $db->next\_record();
if ($result) {
    $UserName = $db->f("emp_name");
    $WorkPhone = $db->f("phone_work");
}
$db->close();

// Show a label value
$UserInfo->SetValue($UserName .", phone: ". $WorkPhone);
} else {
    $UserInfo->Visible = False;
}
}

```

Perl

```

sub UserInfo_BeforeShow() {

# Read user_id from URL
$UserId = CCGetFromGet("user_id", 0);

if ($UserId > 0) {
# Open connection
$db = clsDBConnection1->new();

```

```

    $SQL = "SELECT emp_name, phone_work FROM employees WHERE
emp_id=".$UserId;
    $db->query($SQL);
    $Result = $db->next_record();
    if ($Result) {
        $UserName = $db->f("emp_name");
        $WorkPhone = $db->f("phone_work");
    }
    $db->{sth} = undef;
    $db->{dbh} = undef;

    # Show a label value
    $UserInfo->SetValue($UserName .", phone: ". $WorkPhone);
} else {
    $UserInfo->{Visible} = 0;
}
}
}

```

ColdFusion

```

<!---UserInfo_BeforeShow --->
    <CFMODULE Template="CCGetParam.cfm" strName="user_id" def="0"
outputVar="userId">
    <CFIF userId GT 0>
        <CFSET tempSQL = "SELECT emp_name, phone_work FROM employees WHERE
emp_id=#userId#">
        <CFMODULE Template="CCOPENRS.cfm" strName="Temp"
Connection="IntranetDB" sql="#tempSQL#">
        <CFIF queryTemp.RecordCount GT 0>
            <CFSET fldUserInfo = "#queryTemp.emp_name#, phone:
#queryTemp.phone_work#">
        </CFIF>
    <CFELSE>
        <CFSET hideUserInfo = True>
    </CFIF>

```

Java

```

//UserInfo_BeforeShow
// Read user_id from URL
long userId =
Utils.convertToLong(e.getPage().getHttpGetParams().getParameter("user_
id", "0")).longValue();
if (userId > 0) {
    // Open connection
    JDBCConnection conn =
JDBCConnectionFactory.getJDBCConnection("IntranetDB");
    String sql = "SELECT emp_name, phone_work FROM employees WHERE
emp_id=" +
                conn.toSql(String.valueOf(userId),
JDBCConnection.INTEGER);
    DbRow row = conn.getOneRow(sql);

```

```

conn.closeConnection();
String userName = "";
String workPhone = "";
if(row!=null) {
    userName = Utils.convertToString(row.get("emp_name"));
    workPhone = Utils.convertToString(row.get("phone_work"));
}

// Show a label value
e.getControl().setValue(userName+", phone: "+workPhone);
} else {
    e.getControl().setVisible(false);
}
}

```

C#

```

//UserInfo_BeforeShow
DataAccessObject NewDao = Settings.IntranetDBDataAccessObject;
string UserName = "";
string WorkPhone = "";

if (Request.QueryString["user_id"] != null &&
Request.QueryString["user_id"].Length != 0) {
    SqlCommand Select = new SqlCommand("SELECT emp_name, phone_work
FROM employees WHERE emp_id="+

NewDao.ToSql(Request.QueryString["user_id"].ToString(),FieldType.Integ
er),NewDao);

    DataRowCollection newDr = Select.Execute().Tables[0].Rows;
    for(int i = 0; i < newDr.Count; i++){
        UserName = newDr[i]["emp_name"].ToString();
        WorkPhone = newDr[i]["phone_work"].ToString();
    }

    // Show a label value
    UserInfo.Text = UserName + ", phone: "+WorkPhone;
}
}

```

VB.NET

```

' UserInfo_BeforeShow
Dim NewDao As DataAccessObject = Settings.IntranetDBDataAccessObject
Dim UserName As String = ""
Dim WorkPhone As String = ""

If Not IsNothing(Request.QueryString("user_id")) Then
    Dim [Select] As New SqlCommand("SELECT emp_name, phone_work FROM
employees WHERE emp_id=" +
        NewDao.ToSql(Request.QueryString("user_id").ToString(),
FieldType._Integer), NewDao)

```

```

    Dim newDr As DataRowCollection =
[Select].Execute().Tables(0).Rows
    Dim i As Integer
    For i = 0 To newDr.Count - 1
        UserName = newDr(i)("emp_name").ToString()
        WorkPhone = newDr(i)("phone_work").ToString()
    Next i

    ' Show a label value
    UserInfo.Text = UserName & ", phone: " & WorkPhone
End If

```

See Also:

[Before Show](#) event, How to [read a single value from a database](#)

Utilize MySQL Enum Field Type

This example shows how to use the MySQL enum type field to provide the list of values for a listbox. In this case, we have a Record form called *Tasks* which has a *status* List Box. We use event code to set the values for the List Box to correspond to the values of the *statuses* field (*enum("High","Normal","Low")*) in the *Tasks* database table.

1. In the Properties window for the Listbox, set the [List Source Type](#) property to *ListOfValues*.
2. Type any value (e.g. "0;0") into the [List Data Source](#) property of the Listbox.
3. Add the [Before Show](#) event to the [Record](#) form.
4. Within the event, add the code below:

ASP

```

Function Tasks_BeforeShow(Sender)
    Dim SQL
    Dim RecordSet
    Dim Enumfield
    Dim ArrayEnum

    SQL = "SHOW COLUMNS FROM tasks LIKE 'statuses'"

    ' Open the recordser
    Set RecordSet = DBConnection1.Execute(SQL)
    If DBConnection1.Errors.Count = 0 Then
        If NOT RecordSet.EOF then
            Enumfield = RecordSet.Fields(1)
        End If
    ' Close the recordser
    RecordSet.Close
    Set RecordSet = Nothing
Else
    Print "SQL Execution Failed."
    DBConnection1.Errors.Clear
End If

```

```

' Parse string from DB
if Enumfield <> Empty Then
    Enumfield = Replace(Mid(Enumfield,6,len(Enumfield)-6),"","")
    ArrayEnum = split(Enumfield ,",")

    Set Tasks.Status.DataSource = CCCreateDataSource(dsListOfValues,
Empty, _
    Array(ArrayEnum,ArrayEnum)
End if

End Function

```

PHP

```

function Tasks_BeforeShow(& $sender) {
global $Tasks;

    $SQL = "SHOW COLUMNS FROM tasks LIKE 'statuses'";

// Open the connection
$db = new clsDBConnection1();
$db->query($SQL);
$result = $db->next\_record();
if($result) {
    $Enumfield = $db->f(1);
}
$db->close();

// Parse string from DB
if ($Enumfield != "") {
    $Enumfield =
str_replace("'", "", substr($Enumfield,6,strlen($Enumfield)-7));
    $ArrayEnum = split(",", $Enumfield);
    foreach ($ArrayEnum as $value) {
        $values[] = array($value, $value);
    }
    $Tasks->Status->Values = $values;
}
}

```

Perl

```

sub Tasks_BeforeShow() {

    $SQL = "SHOW COLUMNS FROM tasks LIKE 'statuses'";

# Open the connection
$db = clsDBConnection2->new();

```

```

$db->query($SQL);
$Result = $db->next_record();
if ($Result) {
    $Enumfield = $db->f(1);
}
$db->{sth} = undef;
$db->{dbh} = undef;

# Parse string from DB
if ($Enumfield ne "") {
    $Enumfield =~ s/enum\\(|\\'|\\)//ig;
    my @ArrayEnum = split /\,/,$Enumfield ;

    my @values;
    for (my $i= 0; $i <= $#{@ArrayEnum}; $i++ ) {
        $values[$i] = [$ArrayEnum[$i],$ArrayEnum[$i]];
    }
    $Tasks->{Status}->{Values} = \@values;
}
}

```

Java

```

//Tasks_BeforeShow

String sql = "SHOW COLUMNS FROM tasks LIKE 'statuses'";

// Open the connection
JDBCConnection conn =
JDBCConnectionFactory.getJDBCConnection("Connection2");
DbRow row = conn.getOneRow(sql);
conn.closeConnection();
String enumField = null;
if ( row != null ) {
    enumField = (String) row.get(new Integer(1));
}

if ( enumField != null ) {
    enumField = StringUtils.replace(enumField.substring(5,
enumField.length()-1),"'", "");
    Enumeration opts = StringUtils.split(enumField, ',');
    Vector options = new Vector();
    int i = 1;
    com.codecharge.components.List list =
e.getRecord().getList("Status");
    while ( opts.hasMoreElements() ) {
        DbRow optRow = new DbRow();
        optRow.put(list.getBoundColumn(),String.valueOf(i++));
        optRow.put(list.getTextColumn(),(String) opts.nextElement());
    }
}

```

```

options.add(optRow);
}
e.getRecord().getControl("Status").setValue(enumField);
list.setOptions(options.elements());
}

```

See Also:

[BeforeShow](#) event, How to [modify the 'ListOfValues' of a list box](#)

Validating Data Input

Using the Control Validation Event

This example shows how to validate a field value using the control's **On Validate** Event. In this case, we have a form called *Form1* which contains a Text Box called *TextBox1*. The value of *Textbox1* should be greater than 10 and smaller than 50.

1. Add an [On Validate](#) event to the control. Note that you can just as well add the [On Validate](#) event to the form instead of the control. Applying the [On Validate](#) event to the form enables you to validate several controls belonging to the form.
2. Within the event, add the code below:

ASP

```

Function Form1_TextBox1_OnValidate(Sender)

    If Form1.TextBox1.Value < 10 or Form1.TextBox1.Value > 50 Then
        Form1.Errors.addError("The value must be greater than 10 and
smaller than 50")
    End if

End Function

```

PHP

```

function Form1_TextBox1_OnValidate(& $sender) {
    global $Form1;

    if ($Form1->TextBox1->GetValue() < 10 or $Form1->TextBox1->
GetValue() > 50) {
        $Form1->Errors->addError("The value must be greater than 10 and
smaller than 50");
    }
}

```

Perl

```

sub Form1_TextBox1_OnValidate() {

    if ($Form1->{TextBox1}->{Value} < 10 or $Form1->{TextBox1}->{Value}
> 50) {

```

```

    $Form1->{Errors}->addError("The value must be greater than 10 and
smaller than 50");
}

}

```

ColdFusion

```

<!--Form1_TextBox1_OnValidate-->
<CFIF fldTextBox1 LT 10 OR fldTextBox1 GT 50>
    <CFSET strErrRegistration=strErrRegistration & "The value
must be greater than 10 and smaller than 50<br>">
</CFIF>

```

VB.Net

```

'Form1_TextBox1_OnValidate

If TextBox1.Value < 10 or TextBox1.Value > 50 Then
    errors.Add("TextBox1","The value must be greater than 10 and
smaller than 50")
End if

```

C#

Note: Select *Integer* in the [Data Type](#) property of the *TextBox1* Text Box.

```

//Form1_TextBox1_OnValidate

if (TextBox1.Value != null && ((Int64)TextBox1.Value < 10 ||
(Int64)TextBox1.Value > 50)) {
    errors.Add("TextBox1","The value must be greater than 10 and
smaller than 50");
}

```

Java

```

//Form1_TextBox1_OnValidate

if ( e.getControl().getValue() != null &&
    (Utils.convertToLong(e.getControl().getValue()).longValue() <
10 ||
    Utils.convertToLong(e.getControl().getValue()).longValue() >
50) ) {
    e.getControl().addError("The value must be greater than 10 and
smaller than 50");
}

```

See Also:

[On Validate](#) event, How to [use the Form Validation Event](#)

Using the Form Validation Event

This example shows how to validate a field value using the form's **On Validate** event. In this case we have a *Registration* Record form in which the value of the *Password* field should be identical to the value of the *Confirm* field.

1. Add the [On Validate](#) event to the form.
2. Within the event, add the code below:

ASP

```
Function Registration_OnValidate(Sender)

    If Registration.Password.Value <> Registration.Confirm.Value Then
        Registration.Errors.addError("The values of Password and
Confirm fields do not match.")
    End if

End Function
```

PHP

```
function Registration_OnValidate(& $sender) {
    global $Registration;

    if ($Registration->Password->GetValue() != $Registration->Confirm-
>GetValue()) {
        $Registration->Errors->addError("The values of Password and
Confirm fields do not match.");
    }
}
```

Perl

```
sub Registration_OnValidate() {

    if ($Registration->{Password}->{Value} ne $Registration->{Confirm}-
>{Value}) {
        $Registration->{Errors}->addError("The values of Password and
Confirm fields do not match.");
    }

}
```

ColdFusion

```
<!--Registration_OnValidate-->
<CFIF fldPassword NEQ fldConfirm>
    <CFSET strErrRegistration=strErrRegistration & "<br>" &
"The values of Password and Confirm fields do not match.">
</CFIF>
```

VB.Net

```
'Registration_OnValidate

If Password.Value <> Confirm.Value Then
    errors.Add("Password","The values of Password and Confirm fields
do not match.")
End if
```

C#

```
//Registration_OnValidate

if ((Password.Value != null && Confirm.Value != null) &&
(Password.Value.ToString() != Confirm.Value.ToString() )) {
    errors.Add("Password","The values of Password and Confirm fields
do not match.");
}
}
```

Java

```
//Registration_OnValidate

if ( e.getRecord().getControl("Password").getValue() != null &&
! e.getRecord().getControl("Password").getValue().equals(
e.getRecord().getControl("Confirm").getValue())) {
    e.getRecord().addError("The values in fields Password and Confirm
do not match.");
}
}
```

See Also:

[On Validate](#) event, How to [Use Control Validation](#)

Processing Errors

Display Custom Error Messages

This example shows how to process critical errors. In this case, the error condition is based on the availability of an input parameter called *task_id* which is received over the URL.

Method 1

In the first instance, we want to prevent the page from displaying if the input parameter is not received. We place code in the [After Initialize](#) event of the [page](#) to check for the input parameter and display an error message if no value is received for it.

1. Add the [After Initialize](#) event to the [page](#).
2. Within the event, add the code below:

ASP

```
Function Page_AfterInitialize(Sender)

If IsEmpty(CCGetParam("task_id", Empty)) Then
```

```

UnloadPage
response.write("Error: ""task_id"" parameter is not specified.")
response.end
End If

End Function

```

PHP

```

function Page_AfterInitialize(& $sender) {

    if (CCGetFromGet("task_id","") == "") {
        print ("Error: \"task_id\" parameter is not specified.");
        exit;
    }

}

```

Perl

```

sub Page_AfterInitialize() {

    if (CCGetFromGet("task_id","") eq "") {
        if (!$isHTTPHeader) {
            print $page_header;
            print "Content-type: text/html\n\n";
        }
        print "Error: \"task_id\" parameter is not specified.\n";
        exit 0;
    }

}

```

ColdFusion

```

<!--Page_AfterInitialize --->
<CFMODULE Template="CCGetParam.cfm" strName="task_id">
<CFIF CCGetParam EQ "">
    <CFOUTPUT>Error: "task_id" parameter is not specified.</CFOUTPUT>
    <CFEXIT>
</CFIF>

```

VB.Net

```

'Page_AfterInitialize

If IsNothing(Request.QueryString("task_id")) OR
Request.QueryString("task_id") = "" Then
    Response.Write("Error: ""task_id"" parameter is not specified.")
    Response.[End]
End If

```

C#

```
//Page_AfterInitialize

    if (Request.QueryString["task_id"] == null ||
Request.QueryString["task_id"].Length == 0) {
        Response.Write("Error: 'task_id' parameter is not specified.");
        Response.End();
    }
}
```

Java

```
//Page_AfterInitialize

    if
(StringUtils.isEmpty(e.getPage().getHttpGetParams().getParameter("task
_id"))) {
        Utils.printFatalError("Error: \"task_id\" parameter is not
specified.", e.getPage().getResponse());
    }
}
```

Method 2

In the second instance, we want to prevent a form called *Tasks* from displaying if the input parameter *task_id* is not received.

1. Add a [Label](#) control on the page next to the *Tasks* record form.
2. Rename the Label to *Tasks_Error*.
3. Add the [After Initialize](#) event to the [page](#).
4. Within the event, add the code below:

ASP

```
Function Page_AfterInitialize()

    If IsEmpty(CCGetParam("task_id", Empty)) Then
        Tasks.Visible=False
        Tasks_Error.Value="Error: ""task_id"" parameter is not specified."
    End If

End Function
```

PHP

```
function Page_AfterInitialize() {
global $Tasks;
global $Tasks_Error;

    if (CCGetFromGet("task_id","") == "") {
        $Tasks->Visible = False;
    }
}
```

```

    $Tasks_Error->SetValue("Error: \"task_id\" parameter is not
specified.");
}
}

```

Perl

```

sub Page_AfterInitialize() {

    if (CCGetFromGet("task_id","") eq "") {
        $Tasks->{Visible} = 0;
        $Tasks_Error->SetValue("Error: \"task_id\" parameter is not
specified.");
    }

}

```

ColdFusion

Set the [Default Value](#) property of the 'Task_Error' label to *"Error: 'task_id' parameter is not specified."*

```

<!--Page_AfterInitialize --->
<CFMODULE Template="CCGetParam.cfm" strName="task_id">
<CFIF CCGetParam EQ "">
    <CFSET hideTasks=True>
<CFELSE>
    <CFSET hideTasks_Error=True>
</CFIF>

```

ColdFusion with Templates

1. Set the [Default Value](#) property of the *Task_Error* label to *"Error: 'task_id' parameter is not specified."*
2. Add "`<!-- BEGIN Label Tasks_Error -->`" at the beginning and "`<!-- END Label Tasks_Error -->`" at the end of the [Label](#) control:

```

<!-- BEGIN Label Tasks_Error -->
{Tasks_Error}
<!-- END Label Tasks_Error -->

```

3. Add the code below:

```

<!--Page_AfterInitialize --->
<CFMODULE Template="CCGetParam.cfm" strName="task_id">
<CFIF CCGetParam EQ "">
    <CFSET hideTasks=True>
<CFELSE>
    <CFSET hideTasks_Error=True>
</CFIF>

```

Java

```
//Page_AfterInitialize

if
(StringUtils.isEmpty(e.getPage().getHttpGetParams().getParameter("task_id"))) {
    e.getPage().getRecord("Tasks").setVisible(false);
    e.getPage().getControl("Tasks_Error").setValue("Error: \"task_id\" parameter is not specified.");
}
```

VB.Net

1. Set the [Default Value](#) property of the *Task_Error* Label to "Error: 'task_id' parameter is not specified."
2. Add the code below:

```
'Page_AfterInitialize

If IsNothing(Request.QueryString("task_id")) OR
Request.QueryString("task_id") = "" Then
    TasksHolder.Visible=False
Else
    Tasks_Error.Visible=False
End If
```

C#

1. Set the [Default Value](#) property of the *Task_Error* Label to "Error: 'task_id' parameter is not specified."
2. Add the code below:

```
//Page_AfterInitialize

if (Request.QueryString["task_id"] == null ||
Request.QueryString["task_id"].Length == 0) {
    TasksHolder.Visible = false;
} else {
    Tasks_Error.Visible = false;
}
```

Dynamically Modifying HTML Output

Add Custom HTML to a Page

This example shows how to display custom HTML dynamically using a Label control.

1. Add a Label control called *CustomHTML*.
2. Set the [Content](#) property of the Label to *HTML*.
3. In the [Before Show](#) event of the Label, add the code below:

ASP

```
Function Page_BeforeShow(Sender)

    CustomHTML.Value="MyData <table><tr><td>Data 1</td><td>Data
2</td></tr></table>"

End Function
```

PHP

```
function Page_BeforeShow(& $sender) {
    global $CustomHTML;

    $CustomHTML->SetValue("MyData <table><tr><td>Data 1</td><td>Data
2</td></tr></table>" );
}

}
```

Perl

```
sub Page_BeforeShow() {

    $CustomHTML->SetValue("MyData <table border=1><tr><td>Data
1</td><td>Data 2</td></tr></table>" );

}

}
```

ColdFusion

```
<!--Page_BeforeShow -->

<CFSET fldCustomHTML="MyData <table border=1><tr><td>Data
1</td><td>Data 2</td></tr></table>">
```

VB.Net

```
'Page_BeforeShow

CustomHTML.Text = "MyData <table><tr><td>Data 1</td><td>Data
2</td></tr></table>"
```

C#

```
//Page_BeforeShow

CustomHTML.Text = "MyData <table><tr><td>Data 1</td><td>Data
2</td></tr></table>";
```

Java

```
//Page_BeforeShow
```

```
e.getPage().getControl("CustomHTML").setValue("MyData <table border=1><tr><td>Data 1</td><td>Data 2</td></tr></table>" );
```

See also:

[Before Show](#) event

Change Control Value

This example shows how to change the value of a control in the **Before Show** event.

Method 1: Control Level

In this instance, we will change the value of a control called *FirstName* which is located in a form called *User*.

1. Add the [Before Show](#) event to the control
2. Within the event, add the code below:

ASP

```
Function User_FirstName_BeforeShow(Sender)

    User.FirstName.Value = "Bob"

End Function
```

PHP

```
function User_FirstName_BeforeShow(& $sender) {
    global $User;

    $User->FirstName->SetValue ("Bob");
}
```

Perl

```
sub User_FirstName_BeforeShow() {

    $User->{FirstName}->SetValue ("Bob");

}
```

ColdFusion

```
<!-- Users_FirstName_BeforeShow --->

<CFSET fldFirstName = "Bob">
```

VB.Net

```
'Users_FirstName_BeforeShow

UserFirstName.Text= "Bob"
```

C#

```
//Users_FirstName_BeforeShow  
  
UserFirstName.Text= "Bob";
```

Java

```
//Users_FirstName_BeforeShow  
  
e.getControl().setValue("Bob");
```

Method 2: Form Level

In this instance, we will change the value of a Label field called *Total* which is located within a Grid called *Orders*. Unlike the first example, we will use the **Before Show** event of the form instead of the **Before Show** event of the control.

1. Add the [Before Show Row](#) event to the *Orders* form
2. Within the event, add the code below:

ASP

```
Function Orders_BeforeShowRow()  
  
Orders.Total.Value=Orders.Quantity.Value * Orders.Price.Value  
  
End Function
```

PHP

```
function Orders_BeforeShowRow() {  
global $Orders;  
  
$Orders->Total->SetValue( $Orders->Quantity->GetValue() *  
$Orders->Price->GetValue() );  
  
}
```

Perl

```
sub Orders_BeforeShowRow() {  
  
$Orders->{Total}->SetValue( $Orders->{Quantity}->GetValue() *  
$Orders->{Price}->GetValue() );  
  
}
```

ColdFusion

```
<!-- Orders_BeforeShow --->  
  
<CFSET fldTotal=fldQuantity * fldPrice>;
```

VB.Net

Note: Set the [Data Type](#) property to *Integer* or *Float* for the *Quantity* and *Price* Labels..

```
'Orders_BeforeShow

OrdersTotal.Text = (item.Quantity.Value *
item.Price.Value).ToString()
```

C#

Note: Set the [Data Type](#) property to *Integer* or *Float* for the *Quantity* and *Price* Labels.

```
//Orders_BeforeShow

OrdersTotal.Text = (((Int64)item.Quantity.Value) *
((Int64)item.Price.Value)).ToString();
```

Java

Note: Set the [Data Type](#) property to *Integer* or *Float* for the *Quantity* and *Price* Labels.

```
//Orders_BeforeShow

e.getRecord().getControl("Total").setValue(e.getRecord().getControl("Q
uantity").getValue().longValue() *
e.getRecord().getControl("Price").getValue().doubleValue() );
```

Method 3: Page Level

In the last instance, we will change the value of a Label field called *CurrentUser* which is owned by the page.

1. Add the [Before Show](#) event to the page
2. Within the event, add the code below:

ASP

```
Function Page_BeforeShow()

CurrentUser.Value="Bob"

End Function
```

PHP

```
function Page_BeforeShow() {
global $CurrentUser;

$CurrentUser->SetValue ("Bob");
```

```
}
```

Perl

```
sub Page_BeforeShow() {  
  
    $CurrentUser->SetValue("Bob");  
  
}
```

ColdFusion

```
<!-- Page_BeforeShow --->  
<CFSET fldCurrentUser="Bob">
```

VB.Net

```
'Page_BeforeShow  
  
CurrentUser.Text="Bob"
```

C#

```
//Page_BeforeShow  
  
CurrentUser.Text="Bob";
```

Java

```
//Page_BeforeShow  
  
e.getPage().getControl("CurrentUser").setValue("Bob");
```

See also:

[BeforeShow](#) event

Changing Table Cell Color

This example demonstrates how to change cell color in a table. We have a Grid called *Tasks* in which every second row should have a background color different from the preceding row.

For ASP, PHP, Perl, ColdFusion, Java

1. Switch to **HTML** mode
2. Select Forms Tab in the Toolbox
3. Add a [Label](#) called *RowStyle*:

```
4.         <!-- BEGIN Row -->  
5.         <tr class="{RowStyle}">  
6.             <td>{TaskName}</td>  
7.             <td>{Status}</td>  
8.         </tr>
```

```
9.      <!-- END Row -->
```

10. Add the [Before Show Row](#) event to the *Tasks* grid.

11. Within the event, add the code below:

ASP

```
'Global variable
Dim Counter

Function Tasks_BeforeShowRow(Sender)

    If Counter = 0 Then
        Tasks.RowStyle.Value = "Row"
        Counter = 1
    Else
        Tasks.RowStyle.Value = "AltRow"
        Counter = 0
    End if

End Function
```

PHP

```
function Tasks_BeforeShowRow(& $sender) {
    global $Counter;
    global $Tasks;

    if ($Counter == 0) {
        $Tasks->RowStyle->SetValue("Row");
        $Counter = 1;
    } else {
        $Tasks->RowStyle->SetValue("AltRow");
        $Counter = 0;
    }
}
```

Perl

```
sub Tasks_BeforeShowRow() {

    if ($Counter == 0) {
        $Tasks->{RowStyle}->SetValue("Row");
        $Counter = 1;
    } else {
        $Tasks->{RowStyle}->SetValue("AltRow");
        $Counter = 0;
    }
}
```

```
}
```

ColdFusion

```
<!--Tasks_BeforeShowRow -->

<CFPARAM Name="Counter" Default="0">
<CFIF Counter EQ 0>
    <CFSET fldRowStyle="Row">
    <CFSET Counter = 1>
<CFELSE>
    <CFSET fldRowStyle="AltRow">
    <CFSET Counter = 0>
</CFIF>
```

Java

```
//Tasks_BeforeShowRow

if ((e.getGrid().getCurrentRowNumber()%2) == 0) {
    e.getGrid().getControl("RowStyle").setValue("Row");
} else {
    e.getGrid().getControl("RowStyle").setValue("AltRow");
}
```

For .NET

1. Switch to **HTML** mode
2. Modify the *TR* tag as follows

```
<!-- BEGIN Row -->
<tr id="task_row" runat="server">
    <td>{TaskName}</td>
    <td>{Status}</td>
</tr>
<!-- END Row -->
```

3. Add the [Before Show Row](#) event to the *Tasks* grid.
4. Add a variable called *Counter* below the *'End Forms Definition* section
5. Add the code below:

VB.Net

```
Dim Counter As Integer

'Tasks_BeforeShowRow
If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =
ListItemType.AlternatingItem Then
```

```

Dim task_row As System.Web.UI.HtmlControls.HtmlTableCell =
DirectCast(e.Item.FindControl("task_row"), System.Web.UI.HtmlControls.H
tmlTableCell)
If Counter = 0 Then
    task_row.Attributes("Class") = "Row"
    Counter = 1
Else
    task_row.Attributes("Class") = "AltRow"
    Counter = 0
End IF
End If

```

C#

```

protected int Counter;

//Tasks_BeforeShowRow
if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem) {
    System.Web.UI.HtmlControls.HtmlTableCell task_row =
(System.Web.UI.HtmlControls.HtmlTableCell)e.Item.FindControl("task_row
");
    if (Counter == 0) {
        task_name_td.Attributes["Class"] = "Row";
        Counter = 1;
    } else {
        task_name_td.Attributes["Class"] = "AltRow";
        Counter = 0;
    }
}
}

```

See also:

[Before Show Row](#) event

Changing Text Color

This example shows how to change the value of a Label control before it is shown. In particular, we have a Grid called *Tasks* which has a Label field called *Priority*. We shall add event code so that the color of the Label text changes depending on the field value (*red* for high, *black* for normal, and *green* for low priority).

Method 1

Use custom HTML

1. Set the [Content](#) property of the Label field to *HTML*.
2. Add the [Before Show](#) event to the *Priority* Label.
3. Within the event, add the code below:

ASP

```
Function Tasks_Priority_BeforeShow(Sender)
```

```

    If Tasks.Priority.Value = "High" Then
        Tasks.Priority.Value = "<font
color=RED>"&Tasks.Priority.Value&"</font>"
    ElseIf Tasks.Priority.Value = "Low" Then
        Tasks.Priority.Value = "<font
color=GREEN>"&Tasks.Priority.Value&"</font>"
    End if

End Function

```

PHP

```

function Tasks_Priority_BeforeShow(& $sender) {
    global $Tasks;

    if ($Tasks->Priority->GetValue() == "High") {
        $Tasks->Priority->SetValue("<font color=RED>".$Tasks->Priority->
GetValue()."</font>");
    } else if ($Tasks->Priority->GetValue() == "Lowest") {
        $Tasks->Priority->SetValue("<font color=GREEN>".$Tasks->
Priority->GetValue()."</font>");
    }
}

```

Perl

```

sub Tasks_Priority_BeforeShow() {

    if ($Tasks->{Priority}->GetValue() eq "High") {
        $Tasks->{Priority}->SetValue("<font color=RED>".$Tasks->
{Priority}->GetValue()."</font>");
    } elsif ($Tasks->{Priority}->GetValue() eq "Lowest") {
        $Tasks->{Priority}->SetValue("<font color=GREEN>".$Tasks->
{Priority}->GetValue()."</font>");
    }
}

```

ColdFusion

```

<!--Tasks_Priority_BeforeShow -->

<CFIF fldPriority EQ "High">
    <CFSET fldPriority="<font color=RED>#fldPriority#</font>">
<CFELSEIF fldPriority EQ "Lowest">
    <CFSET fldPriority="<font color=GREENRED>#fldPriority#</font>">
</CFIF>

```

VB.Net

```
'Tasks_Priority_BeforeShow

If TasksPriority.Text = "High" Then
    TasksPriority.Text = ("" & (TasksPriority.Text).ToString() & "</font>").ToString()
ElseIf TasksPriority.Text = "Lowest" Then
    TasksPriority.Text = ("" & (TasksPriority.Text).ToString() & "</font>").ToString()
End if
```

C#

```
//Tasks_Priority_BeforeShow

if (TasksPriority.Text == "High") {
    TasksPriority.Text = "<font
color=RED>"+TasksPriority.Text+"</font>";
} else if (TasksPriority.Text == "Lowest") {
    TasksPriority.Text = "<font
color=GREEN>"+TasksPriority.Text+"</font>";
}
```

Java

```
//Tasks_Priority_BeforeShow

if ("High".equals(e.getControl().getValue())) {
    e.getControl().setValue("<font
color=RED>"+e.getControl().getValue()+"</font>");
} else if ("Low".equals(e.getControl().getValue())) {
    e.getControl().setValue("<font
color=GREEN>"+e.getControl().getValue()+"</font>");
}
```

Method 2

Use A Label control

1. Switch to the **HTML** mode
2. Add a [Label](#) control called *Priority_Color* into the tag.

```
3.         <!-- BEGIN Row -->
4.         <tr>
5.             <td class="ClearDataTD">{task_name} </td>
6.             <td class="ClearDataTD">{project_name} </td>
7.             <td class="ClearDataTD">{status_name} </td>
8.             <td class="ClearDataTD"><font
color="{Priority_Color}">{Priority}</font> </td>
9.             <td class="ClearDataTD">{date_resolve} </td>
10.        </tr>
```

```
<!-- END Row -->
```

11. Add the [Before Show Row](#) event to the *Tasks* Grid.

12. Within the event, add the code below:

ASP

```
Function Tasks_BeforeShowRow()  
  
    If Tasks.Priority.Value = "High" Then  
        Tasks.Priority_Color.Value = "RED"  
    ElseIf Tasks.Priority.Value = "Lowest" Then  
        Tasks.Priority_Color.Value = "GREEN"  
    Else  
        Tasks.Priority_Color.Value = "BLACK"  
    End if  
  
End Function
```

PHP

```
function Tasks_BeforeShowRow() {  
    global $Tasks;  
  
    if ($Tasks->Priority->GetValue() == "High") {  
        $Tasks->Priority_Color->SetValue("RED");  
    } else if ($Tasks->Priority->GetValue() == "Lowest") {  
        $Tasks->Priority_Color->SetValue("GREEN");  
    } else {  
        $Tasks->Priority_Color->SetValue("BLACK");  
    }  
  
}
```

Perl

```
sub Tasks_BeforeShowRow() {  
  
    if ($Tasks1->{Priority}->GetValue() eq "High") {  
        $Tasks1->{Priority_Color}->SetValue("RED");  
    } elsif ($Tasks1->{Priority}->GetValue() eq "Lowest") {  
        $Tasks1->{Priority_Color}->SetValue("GREEN");  
    } else {  
        $Tasks1->{Priority_Color}->SetValue("BLACK");  
    }  
  
}
```

Java

```
//Tasks BeforeShowRow
```

```

if ("High".equals(e.getGrid().getControl("Priority").getValue())) {
    e.getGrid().getControl("Priority_Color").setValue("RED");
} else if
("Lowest".equals(e.getGrid().getControl("Priority").getValue())) {
    e.getGrid().getControl("Priority_Color").setValue("GREEN");
} else {
    e.getGrid().getControl("Priority_Color").setValue("BLACK");
}

```

See also:

[Before Show](#) event, [Before Show Row](#) event, [Content](#) property

Working With Pages

Add URL Parameters to a Page

This example shows how to reload a page if the URL lacks some parameters. In this case, the URL for the page must contain a *status=1* URL parameter otherwise the page will be reloaded and the parameter added to the URL.

1. Add the [After Initialize](#) event to the page.
2. Within the event, add the code below:

ASP

```

Function Page_AfterInitialize(Sender)

    If IsEmpty(CCGetParam("status", Empty)) Then
        Response.Redirect FileName & "?" &
CCAddParam(Request.ServerVariables("QUERY_STRING"), "status", 1)
    end if

End Function

```

PHP

```

function Page_AfterInitialize(& $sender) {
    global $FileName;

    if (CCGetFromGet("status","") == "") {
        header("Location: " . $FileName."?").
CCAddParam(CCGetQueryString("QueryString", ""), "status", 1) );
    }
}

```

Perl

```

sub Page_AfterInitialize() {

    if (CCGetFromGet("status","") eq "") {
        print $page_header . "Status: 302 Found\nLocation: ".

```

```

    $FileName."?". CCAddParam(CCGetQueryString("status", ""),
"status", 1) ."\n\n";
    exit 0;
}
}

```

ColdFusion

```

<!--Page_AfterInitialize --->
<CFPARAM Name="URL.status" Default="">
<CFIF URL.status EQ "">
    <CFMODULE Template="CCGetQueryString.cfm"
strCollection="QueryString"
arrRemoveParameters="#ListToArray('CFID;CFTOKEN',';')#">
    <CFMODULE Template="CCAddParam.cfm"
strQueryString="#CCGetQueryString#" strName="status" strValue="1">
    <CFLOCATION URL="#strFileName##CCAddParam#">
</CFIF>

```

VB.Net

```

'Page_AfterInitialize

If IsNothing(Request.QueryString("status")) Or
Request.QueryString("status") = "" Then
    Dim params As New LinkParameterCollection()
    params.Add("status","1")
    Response.Redirect(Request.Url.AbsolutePath +
params.ToString("GET","status"))
End If

```

C#

```

//Page_AfterInitialize

if (Request.QueryString["status"] == null ||
Request.QueryString["status"].Length == 0) {
    LinkParameterCollection HrefParams = new
LinkParameterCollection();
    HrefParams.Add("status","1");
    Response.Redirect(Request.Url.AbsolutePath +
HrefParams.ToString("GET","status"));
}

```

Java

```

//Page_AfterInitialize

if (
StringUtils.isEmpty(e.getPage().getHttpGetParams().getParameter("statu
s"))) ) {

```

```

String servletPath = e.getPage().getRequest().getRequestURI();
String queryString = e.getPage().getRequest().getQueryString();
queryString = StringUtils.isEmpty(queryString) ? "status=1" :
queryString + "&" + "status=1";
e.getPage().setRedirectString(servletPath + "?" + queryString);
}

```

See Also:

[After Initialize](#) event

Working with Custom Template Blocks

This example shows how to work with a custom template block and custom template variables.

1. Add a custom block in the HTML mode (shown in **blue**)
2. Add custom template variables. (shown in **red**)

```

3.      <table>
4.      <tr>
5.          <td>Server Variable</td>
6.          <td>Value</td>
7.      </tr>
8.      <!-- BEGIN MyTable -->
9.      <tr>
10.         <td>{VariableName}</td>
11.         <td>{VariableValue}</td>
12.     </tr>
13.     <!-- END MyTable -->

```

```
</table>
```

14. To set values for the template variables, enter the code below:

ASP

Version 1

For a top level block which is not nested within another block:

```

Function Page_BeforeShow(Sender)
Dim CustomBlock
Dim strKey

Set CustomBlock = Tpl.Block("MyTable")
For Each strKey In Request.ServerVariables
    CustomBlock.Variable("VariableName") = strKey
    CustomBlock.Variable("VariableValue") =
Request.ServerVariables(strKey)
    CustomBlock.Parse ccsParseAccumulate
Next

End Function

```

Version 2

For a custom block located inside another block (e.g. *Mytable* block inside *Row* block inside *Tasks* block):

```
Function Tasks_BeforeShowRow(Sender)
Dim CustomBlock
Dim strKey

    Set CustomBlock = Tasks.TemplateBlock.Block("Row").Block("MyTable")
    CustomBlock.Clear

    For Each strKey In Request.ServerVariables
        CustomBlock.Variable("VariableName") = strKey
        CustomBlock.Variable("VariableValue") =
Request.ServerVariables(strKey)
        CustomBlock.Parse ccsParseAccumulate
    Next

End Function
```

Note: If you need to parse the block only once you can use 'ccsParseOverwrite' instead of 'ccsParseAccumulate'.

PHP

Version 1

For a top level block which is not nested within another block:

```
function Page_BeforeShow(& $sender) {
global $HTTP_SERVER_VARS;
global $Tpl;

    foreach($HTTP_SERVER_VARS as $strKey => $value) {
        $Tpl->SetVar("VariableName",$strKey);
        $Tpl->SetVar("VariableValue",$value);
        $Tpl->Parse("MyTable",True);
    }

}
```

Version 2

For a custom block located inside another block (e.g. *Mytable* block inside *Row* block inside *Tasks* block):

```
function Tasks_BeforeShowRow(& $sender) {
global $HTTP_SERVER_VARS;
global $Tpl;

    $Tpl->SetBlockVar("MyTable","");
    foreach($HTTP_SERVER_VARS as $strKey => $value) {
        $Tpl->SetVar("VariableName",$strKey);
        $Tpl->SetVar("VariableValue",$value);
        $Tpl->Parse("MyTable",True);
    }

}
```

```
}  
}
```

Perl

Version 1

For a top level block which is not nested within another block:

```
sub Page_BeforeShow() {  
  
    while (($key,$value) = each %ENV) {  
        $Tpl->setvar("VariableName",$key);  
        $Tpl->setvar("VariableValue",$value);  
        $Tpl->parse("MyTable","true");  
    }  
  
}
```

Version 2

For a custom block located inside another block (e.g. *Mytable* block inside *Row* block inside *Tasks* block):

```
sub Tasks_BeforeShowRow() {  
  
    $Tpl->setblockvar("MyTable","");  
    while (($key,$value) = each %ENV) {  
        $Tpl->setvar("VariableName",$key);  
        $Tpl->setvar("VariableValue",$value);  
        $Tpl->parse("MyTable","true");  
    }  
  
}
```

ColdFusion with Templates

Version 1

For a top level block which is not nested within another block:

```
<!--Page_BeforeShow --->  
  
    <CFLOOP LIST="#StructKeyList(CGI)#" INDEX="item">  
        <CFMODULE Template="CCSetVar.cfm" Var="@VariableName"  
Value="#item#">  
        <CFMODULE Template="CCSetVar.cfm" Var="@VariableValue"  
Value="#StructFind(CGI, item)#">  
        <CFMODULE Template="CCParse.cfm" mode="Parse" blockName="MyTable"  
bAccumulate="True">  
    </CFLOOP>
```

Version 2

For a custom block located inside another block (e.g. *Mytable* block inside *Row* block inside *Tasks* block):

```

<!--Tasks_BeforeShowRow --->

    <CFMODULE Template="CCSetVar.cfm" Var="#strBlockToParse#/Row/MyRow"
Value="">
    <CFLOOP LIST="#StructKeyList(CGI)#" INDEX="item">
        <CFMODULE Template="CCSetVar.cfm"
Var="#strBlockToParse#/Row/MyRow/@VariableName1" Value="#item#">
            <CFMODULE Template="CCSetVar.cfm"
Var="#strBlockToParse#/Row/MyRow/@VariableValue1"
Value="#StructFind(CGI, item)#">
                <CFMODULE Template="CCParse.cfm" mode="Parse"
blockName="#strBlockToParse#/Row/MyRow" bAccumulate="True">
                    </CFLOOP>

```

Servlet with Templates

Version 1

For a top level block which is not nested within another block:

```

//Page_BeforeShow

//The data used for output into the template variables.
Enumeration enum = StringUtils.split("1;111;2;222;3;333;4;444");

Template tmpl = e.getPage().getTemplate();
String currentPath = e.getPage().getCurrentPath();
while (enum.hasMoreElements()) {
    tmpl.setVar(currentPath+ "/MyTable/@VariableName", (String)
enum.nextElement());
    if (enum.hasMoreElements()) {
        tmpl.setVar(currentPath+ "/MyTable/@VariableValue", (String)
enum.nextElement());
    }
    tmpl.parse(currentPath+ "/MyTable");
}

```

Version 2

For a custom block located inside another block (e.g. *Mytable* block inside *Row* block inside *Tasks* block):

```

// Tasks_BeforeShowRow

//The data used for output into the template variables.
Enumeration enum = StringUtils.split("5;555;6;666;7;777;8;888");

Template tmpl = e.getPage().getTemplate();
String currentPath = e.getPage().getCurrentPath();
tmpl.setVar(currentPath+ "/Row/MyTable");
while (enum.hasMoreElements()) {

```

```

    tmpl.setVar(currentPath+ "/Row/MyTable/@VariableName", (String)
enum.nextElement());
    if (enum.hasMoreElements()) {
        tmpl.setVar(currentPath+ "/Row/MyTable/@VariableValue", (String)
enum.nextElement());
    }
    tmpl.parse(currentPath+ "/Row/MyTable");
}

```

See Also:

[Before Show](#) event, [Before Show Row](#) event

Dynamically Redirect Users to Another Page

This example shows how to redirect the user to different pages after the Insert or Update operation is performed. In this case, we have a Record form which has an Insert button named *Insert1* and an update button named *Update1*. We add code in the [On Click](#) event of each of the buttons to perform the redirection:

Note: The Redirect variable can be modified in the onClick event of the Button only if the Return Page property of this Button is empty, otherwise Redirect will be overwritten in page's code.

1. Add the [On Click](#) event for each of the buttons.
2. Within the events , add the code below:

ASP

```

Function Form1_Insert1_OnClick(Sender) {

    Redirect = "InsertMessagePage.asp"

}

Function Form1_Update1_OnClick(Sender) {

    Redirect = "UpdateMessagePage.asp"

}

```

PHP

```

function Form1_Insert1_OnClick(& $sender) {
global Redirect;

    Redirect = "InsertMessagePage.php";

}

function Form1_Update1_OnClick(& $sender) {
global Redirect;

```

```
$Redirect = "UpdateMessagePage.php";  
  
}
```

Perl

```
sub Form1_Insert1_OnClick() {  
  
    $Redirect = "InsertMessagePage.php";  
  
}  
  
sub Form1_Update1_OnClick() {  
  
    $Redirect = "UpdateMessagePage.php";  
  
}
```

ColdFusion

```
<!--Form1_Insert1_OnClick --->  
  
    <CFSET strRedirect = "InsertMessagePage.cfm">  
  
<!--Form1_Update1_OnClick --->  
  
    <CFSET strRedirect = "UpdateMessagePage.cfm">
```

VB.Net

```
'Form1_Insert1_OnClick  
  
RedirectUrl = "InsertMessagePage.aspx"  
  
'Form1_Update1_OnClick  
  
RedirectUrl = "UpdateMessagePage.aspx"
```

C#

```
//Form1_Insert1_OnClick  
  
RedirectUrl = "InsertMessagePage.aspx";  
  
//Form1_Update1_OnClick  
  
RedirectUrl = "UpdateMessagePage.aspx";
```

Java

```
//Form1_Insert1_OnClick

    e.getPage().setRedirectString("InsertMessagePage.do");

//Form1_Update1_OnClick

    e.getPage().setRedirectString("UpdateMessagePage.do");
```

See also:

[OnClick](#) event, [BeforeUnload](#) event

Dynamic Login Field in a Registration Form

This example shows how to use a Login control on a registration form.

Assume that you have a *Registration* [Record](#) form with a *Logi* [nText Box](#) control where users can enter a Login name when creating a record

Login:

but existing users should not be able to change the Login name

Login: rob_mcdonald

1. Create a registration form.
2. Click on the Text Box icon in the **Forms** tab of the Toolbox to add the *Login Text Box* control. In the same tab click on the Label icon to add a [Label](#) control directly after the *Login* Textbox.
3. Set the name of the Label control to *Login_View*.
4. Add `<!-- BEGIN TextBox Login -->` at the beginning and `<!-- END TextBox Login -->` at the end of the Login Textbox.
5. Add `<!-- BEGIN Label Login_View -->` at the beginning and `<!-- END Label Login_View -->` at the end of the Login Label.
6. The illustration below shows how the Login will look:

- In the Design mode

Login: [A](#) {Login_View} [A](#)

- In the **HTML** mode

```
•
•
• <tr>
• <td>Login:</td>
• <td>
• <!-- BEGIN TextBox Login -->
• <input type="text" name="{Login_Name}"
value="{Login}">
• <!-- END TextBox Login -->
```

```

•      <!-- BEGIN Label Login_View -->
•      {Login_View}
•      <!-- END Label Login_View -->
•      </td>
•      </tr>

```

7. Add a [Before Show](#) event for the [Record](#) form.

8. Within the event, add the code below:

PS: Don't forget to retrieve the login value in the edit mode

ASP

```

Function Registration_BeforeShow(Sender)

If (Registration.EditMode) Then
    'Edit Mode
    Registration.Login.Visible=False
Else
    'Add Mode
    Registration.Login_View.Visible=False
End If

End Function

```

PHP

```

function Registration_BeforeShow(& $sender) {
    global $Registration;

    if ($Registration->EditMode) {
        //Edit Mode
        $Registration->Login->Visible = False;
    } else {
        //Add Mode
        $Registration->Login_View->Visible = False;
    }
}

```

Perl

```

sub Registration_BeforeShow() {

    if ($Registration->{EditMode} == 1) {
        #Edit Mode
        $Registration->{Login}->{Visible} = 0;
    } else {
        #Add Mode
        $Registration->{Login_View}->{Visible} = 0;
    }
}

```

```
}
```

ColdFusion

```
<!--Registration_BeforeShow --->  
<CFIF blnEditModeRegistration>  
  <CFSET hideLogin=True>  
<CFELSE>  
  <CFSET hideLogin View=True>  
</CFIF>
```

VB.Net

```
'Registration_BeforeShow  
  
If (IsInsertMode) Then  
  RegistrationLogin_View.Visible = False  
Else  
  RegistrationLogin.Visible = False  
End if
```

C#

```
//Registration_BeforeShow  
  
if (IsInsertMode) {  
  RegistrationLogin_View.Visible = false;  
} else {  
  RegistrationLogin.Visible = false;  
}
```

Java

```
//Registration_BeforeShow  
  
if (e.getRecord().isEditMode()) {  
  //Edit Mode  
  e.getRecord().getControl("Login").setVisible(false);  
} else {  
  //Add Mode  
  e.getRecord().getControl("Login_View").setVisible(false);  
}
```

See also:

[BeforeShow](#) event, How to [hide a record control](#)

Working With Panels

The examples below demonstrate the code used when working with controls and HTML snippets contained within [Panels](#), including assigning control values and hiding the panels.

Note that the panel component is transparent when accessing embedded controls' properties and methods, therefore you may set control values using the same code whether components are within a panel or not.

Single Control within a Panel

Here is how to assign a value to, or hide a single control that is embedded within a panel. In this example we assume that a label named "UserLogin" is embedded within a panel named "Panel1" that is placed directly on the page (not inside a form). If a user is logged into the system we want to populate the label with the user's name, otherwise we want to hide the panel containing an HTML snippet and the label.

The following HTML code represents the panel structure:

```
<!-- BEGIN Panel Panel1 -->
<table>
  <tr>
    <td><Welcome {UserLogin}"></td>
  </tr>
</table>
<!-- END Panel Panel1 -->
```

The following [Before Show](#) event code is assigned to the Panel:

ASP

```
Function Panel1_BeforeShow(Sender)
  If CCGetUserID() Then
    UserLogin.Value = CCGetUserLogin()
  Else
    Panel1.Visible = false
  End if
End Function
```

PHP

```
function Panel1_BeforeShow(& $sender) {
  global $Panel1;
  global $UserLogin;

  if (CCGetUserID()) {
    $UserLogin->SetValue(CCGetUserLogin());
  } else {
    $Panel1->Visible = false;
  }
}
```

Perl

```
sub Panel1_BeforeShow() {
```

```

if (CCGetUserID()) {
    $UserLogin->SetValue(CCGetUserLogin());
} else {
    $Panell->{Visible} = 0;
}
}

```

ColdFusion

```

<!---Event: BeforeShow. Control: Panell --->
<CFIF IsDefined("Session.UserID")>
    <CFSET fldUserLogin = Session.UserLogin>
<CFELSE>
    <CFSET hidePanell = True>
</CFIF>

```

VB.Net

```

'Panel Panell Event BeforeShow.
If Not IsNothing(DBUtility.UserID) Then
    UserLogin.SetValue(DBUtility.UserLogin)
Else
    Panell.Visible = False
End if

```

C#

```

//Panel Panell Event BeforeShow.
if (DBUtility.UserID != null) {
    UserLogin.SetValue(DBUtility.UserLogin);
} else {
    Panell.Visible = false;
}

```

Java

```

//Panell_BeforeShow
String login = Utils.getUserLogin(e.getPage());
if (login != null) {
    e.getPage().getControl("UserLogin").setValue(login);
} else {
    e.getPanel().setVisible(false);
}

```

Multiple Controls within a Panel

Working with multiple controls embedded within a panel is similar to working with a single control.

In this example we assume that two record rows, one with a label named "*User_id_assigned_by*" and another with a label *tasks_start_date*" are embedded within a panel named "Panel1". In the insert mode we want to hide both record rows and control, while in the Edit mode we want to assign a custom value to the label *User_id_assigned_by*. The record form is named "tasks".

The following HTML code represents the panel structure:

```
<!-- BEGIN Panel Panell1 -->
<tr>
  <td>Assigned By</td>
  <td><{User_id_assigned_by}></td>
</tr>
<tr>
  <td>Start Date</td>
  <td><{task_start_date}></td>
</tr>
<!-- END Panel Panell1 -->
```

The following [Before Show](#) event code is assigned to the "tasks" form:

ASP

```
Function tasks_BeforeShow()
  If tasks.EditMode Then
    tasks.user_id_assign_by.Value = CCDLookUp("emp_name",
"employees", "emp_id=" &_
      DBIntranetDB.ToSQL(tasks.user_id_assign_by.Value,
ccsInteger), DBIntranetDB)
  Else
    tasks.Panell1.Visible = false
  End if
End Function
```

PHP

```
function tasks_BeforeShow() {
global $tasks;
global $DBIntranetDB;
  if ($tasks->EditMode)
  {
    $emp_name = CCDLookUp("emp_name", "employees", "emp_id=" .
      $DBIntranetDB->ToSQL($tasks->user_id_assign_by->GetValue(),
ccsInteger), $DBIntranetDB);
    $tasks->user_id_assign_by->SetValue($emp_name);
  }
  else
  {
    $tasks->Panell1->Visible = false;
  }
}
```

Perl

```
sub tasks_BeforeShow() {
  $DBIntranetDB = clsDBIntranetDB->new();
  if ( $tasks->{EditMode} )
  {
    $tasks->{user_id_assign_by}->SetValue(CCDlookUp("emp_name",
"employees", " emp_id=" .
```

```

        $DBIntranetDB->ToSQL($tasks->{user_id_assign_by}-
>GetValue(), $ccsInteger), $DBIntranetDB);
    }
    else
    {
        $tasks->{Panell}->{Visible} = 0;
    }
}

```

ColdFusion

```

<!---Event: BeforeShow. Control: tasks --->
<CFIF blnEditModeTasks>
    <CF_CCToSQL Value= "#flduser_id_assign_by#" Type= "#ccsInteger#" >
    <CF_CCDLookUp Field="emp_name" Table="employees"
Where="emp_id=#CCToSQL#" Connection="IntranetDB">
    <CFSET flduser_id_assign_by=CCDLookUp>
<CFELSE>
    <CFSET hidePanell = True>
</CFIF>

```

VB.Net

```

'Record tasks Event BeforeShow.
Dim NewDao As DataAccessObject = Settings.IntranetDBDataAccessObject
If IsInsertMode = True Then
    TasksPanell.Visible = False
Else
    tasksuser_id_assign_by.Text =
Convert.ToString(NewDao.ExecuteScalar( _
                                "SELECT emp_name FROM employees WHERE
emp_id=" & _
                                Convert.ToString(
tasksuser_id_assign_by.Text)))
End if

```

C#

```

//Record tasks Event BeforeShow.
DataAccessObject NewDao = Settings.IntranetDBDataAccessObject;
if (IsInsertMode)
{
    TasksPanell.Visible = false;
}
else
{
    SqlCommand userNameCmd = new SqlCommand( "SELECT emp_name FROM
employees " +
                                "WHERE emp_id=" +
NewDao.ToSql(item.user_id_assign_by.GetFormattedValue,
FieldType.Integer), NewDao);
    tasksuser_id_assign_by.Text =
userNameCmd.ExecuteScalar().ToString();
}

```

```
}
```

Java

```
//tasks_BeforeShow

if (e.getRecord().isEditMode())
{
    e.getControl().setValue(DBTools.dLookUp("emp_name", "employees",
"emp_id=" +

DBTools.toSql(e.getControl().getFormattedValue(), JDBCConnection.INTEGE
R,"IntranetDB"), "IntranetDB"));
}
else
{
    e.getRecord().getControl("Panel1").setVisible(false);
}
}
```

HTML Snippet within a Panel

In this example we hide sample panels that contain only HTML code and no controls. The following HTML code represents the panel structure:

```
<!-- BEGIN Panel Panel1 -->
<table>
<tr> <td>Welcome to our site</td> </tr>
<tr>
<td>
<!-- BEGIN Panel Panel2 -->
<table>
<tr> <td>Administration</td> </tr>
</table>
<!-- END Panel Panel2 -->
</td>
</tr>
</table>
<!-- END Panel Panel1 -->
```

Note: to embed an HTML section within a panel, select an HTML code, then click on the panel icon on the Forms tab of the Toolbox.

The following [Before Show](#) event code is assigned to the panel:

ASP

1. Page level, where panels are placed on the page, outside of forms.

```
Function Panel1_BeforeShow()
    Panel1.Visible = false
End Function
```

2. Includable page level, where panels are placed on an includable page called "header".

```
Function header_Panel1_BeforeShow()
    header.Panel1.Visible = false
```

```
End Function
```

3. Form level, where panels are placed inside a record form called '*messages*',

```
Function messages_Panell_BeforeShow()  
    messages.Panell.Visible = false  
End Function
```

4. Nested panels, where '*Panel2*' is placed within '*Panel1*'. The event code can be assigned to either '*Panel1*' or '*Panel2*'.

```
Function Panel2_BeforeShow()  
    Panel2.Visible = false  
End Function
```

PHP

1. Page level, where panels are placed on the page, outside of forms.

```
function Panell_BeforeShow() {  
    global $Panell;  
    $Panell->Visible = false;  
}
```

2. Includable page level, where panels are placed on an includable page called "header".

```
function header_Panell_BeforeShow() {  
    global $header;  
    $header->Panell->Visible = false;  
}
```

3. Form level, where panels are placed inside a record form called '*messages*',

```
function messages_Panell_BeforeShow() {  
    global $messages;  
    $messages->Panell->Visible = false;  
}
```

4. Nested panels, where '*Panel2*' is placed within '*Panel1*'. The event code can be assigned to either '*Panel1*' or '*Panel2*'.

```
function Panel2_BeforeShow() {  
    global $Panel2;  
    $Panel2->Visible = false;  
}
```

Perl

1. Page level, where panels are placed on the page, outside of forms.

```
sub Panell_BeforeShow() {  
    $Panell->{Visible} = 0;  
}
```

2. Includable page level, where panels are placed on an includable page called "header".

```
sub header_Panell_BeforeShow() {  
    $header->{Panell}->{Visible} = 0;  
}
```

```
}
```

3. Form level, where panels are placed inside a record form called '*messages*',

```
sub messages_Panel1_BeforeShow() {  
    $messages->{Panel1}->{Visible} = 0;  
}
```

4. Nested panels, where '*Panel2*' is placed within '*Panel1*'. The event code can be assigned to either '*Panel1*' or '*Panel2*'.

```
sub Panel2_BeforeShow() {  
    $Panel2->{Visible} = 0;  
}
```

ColdFusion

1. Page level, where panels are placed on the page, outside of forms.

```
<!--Event: BeforeShow. Control: NewPage1 -->  
<CFSET hidePanel1 = True>
```

2. Includable page level, where panels are placed on an includable page called "header".

```
<!--Event: BeforeShow. Control: header -->  
<CFSET hidePanel1 = True>
```

3. Form level, where panels are placed inside a record form called '*messages*',

```
<!--Event: BeforeShow. Control: messages -->  
<CFSET hidePanel1 = True>
```

4. Nested panels, where '*Panel2*' is placed within '*Panel1*'. The event code can be assigned to either '*Panel1*' or '*Panel2*'.

```
<!--Event: BeforeShow. Control: Panel2 -->  
<CFSET hidePanel2 = True>
```

VB.Net

1. Page level, where panels are placed on the page, outside of forms.

```
'Page NewPage1 Event BeforeShow.  
Panel1.Visible = False
```

2. Includable page level, where panels are placed on an includable page called "header".

```
'Page header Event BeforeShow.  
Panel1.Visible = False
```

3. Form level, where panels are placed inside a record form called '*messages*',

```
'Record messages Event BeforeShow.  
messagesPanel1.Visible = False
```

4. Nested panels, where '*Panel2*' is placed within '*Panel1*'. The event code can be assigned to either '*Panel1*' or '*Panel2*'.

```
'Panel Panel2 Event BeforeShow.  
Panel2.Visible = False
```

C#

1. Page level, where panels are placed on the page, outside of forms.

```
//Page NewPage1 Event BeforeShow.  
    Panel1.Visible = false;
```

2. Includable page level, where panels are placed on an includable page called "header".

```
//Page header Event BeforeShow.  
    Panel1.Visible = false;
```

3. Form level, where panels are placed inside a record form called 'messages',

```
//Record messages Event BeforeShow.  
    messagesPanel1.Visible = false;
```

4. Nested panels, where 'Panel2' is placed within 'Panel1'. The event code can be assigned to either 'Panel1' or 'Panel2'.

```
//Panel Panel2 Event BeforeShow.  
    Panel2.Visible = false;
```

Java

1. Page level, where panels are placed on the page, outside of forms.

```
//page_BeforeShow  
    e.getPage().getPanel("panel1").setVisible(false);
```

2. Includable page level, where panels are placed on an includable page called "header".

```
//header_BeforeShow  
    e.getPage().getPanel("Panel1").setVisible(false);
```

3. Form level, where panels are placed inside a record form called 'messages',

```
//messages_BeforeShow  
    e.getRecord().getPanel("Panel1").setVisible(false);
```

4. Nested panels, where 'Panel2' is placed within 'Panel1'. The event code can be assigned to either 'Panel1' or 'Panel2'.

```
//Panel2_BeforeShow  
    e.getPage().getPanel("Panel2").setVisible(false);
```

See also

- [Panel Control](#)
- [Panel Reference](#)
- [Placing an existing component within a new panel](#)
- [Restricting SQL Execution for Components Hidden Using a Panel](#)
- [Iterating Panel Controls](#)

Iterating Panel Controls

ASP

Regular Page:

```
Dim Ctrl, Key
For Each Key in Panell.Components
    Set Ctrl = Panell.Components(Key)
    Ctrl.Visible = False
Next
```

PHP

Regular page:

```
global $Panell;

foreach ($Panell->Components as $Name => $Component) {
    $Component->Visible = false;
}
```

Includable page:

```
global $Page1;

foreach ($Page1->Panell->Components as $Name => $Component) {
    $Component->Visible = false;
}
```

Perl

Regular page:

```
foreach my $Component (values %{$Panell->{Components}}){
    $Component->{Visible} = 0;
}
```

Includable page:

```
foreach my $Component (values %{$Page1->{Panell}->{Components}}){
    $Component->{Visible} = 0;
}
```

VB.NET

```
Dim c As Control

For Each c In Panell.Controls
    c.Visible = False
```

C#.NET

```
foreach(Control c in Panell.Controls)
    c.Visible = false;
```

Java

```
for (Iterator it =
e.getPage().getPanel("Panel1").getComponents().iterator();
it.hasNext(); ) {
    ((Model) components.next()).setVisible(false);
}
```

See also

[Panel Control](#),

[Panel Reference](#),

[Placing an existing component within a new panel](#),

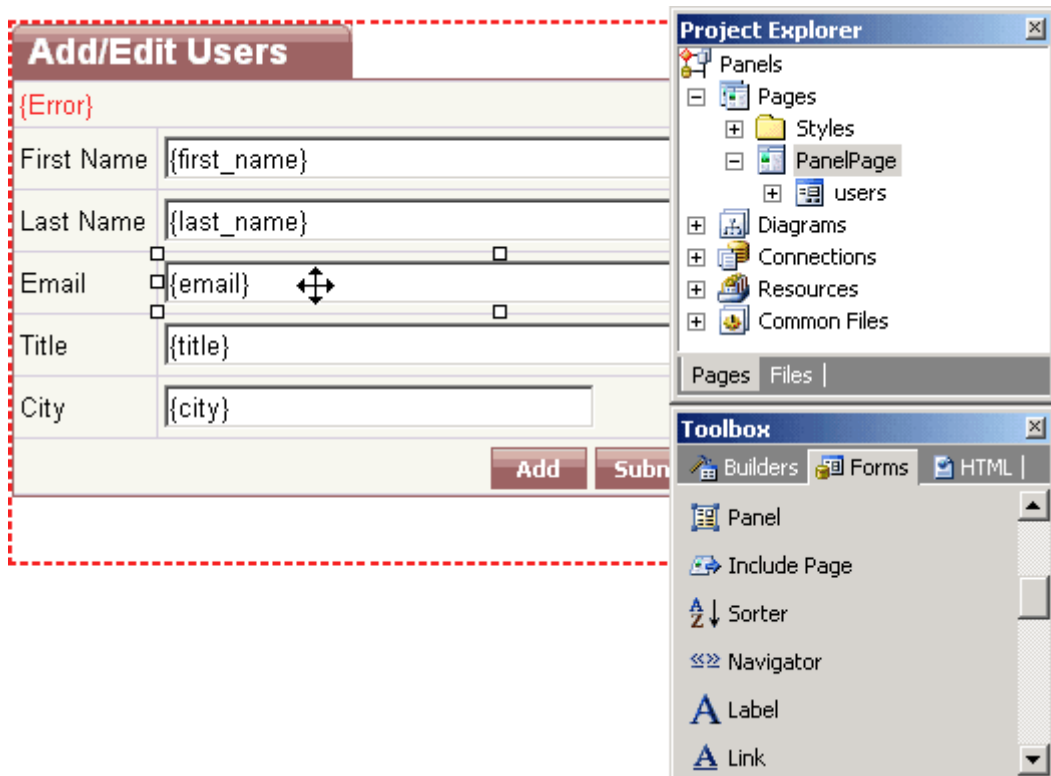
[Restricting SQL Execution for Components Hidden Using a Panel](#),

[Working with Panels](#)

Placing an Existing Component Within a New Panel

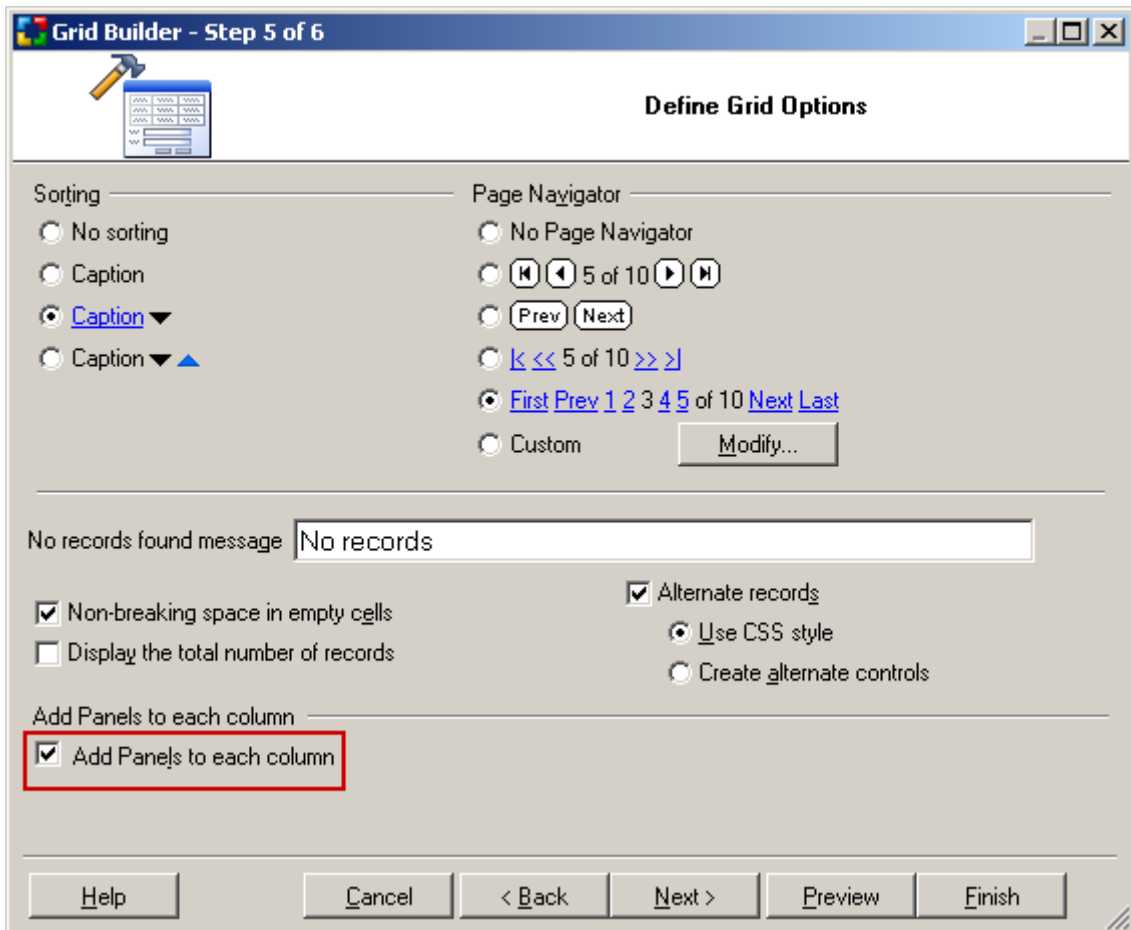
To place an existing component within a new panel:

- Select the part of HTML that should be enclosed within a panel.
- Select panel component on the Toolbar

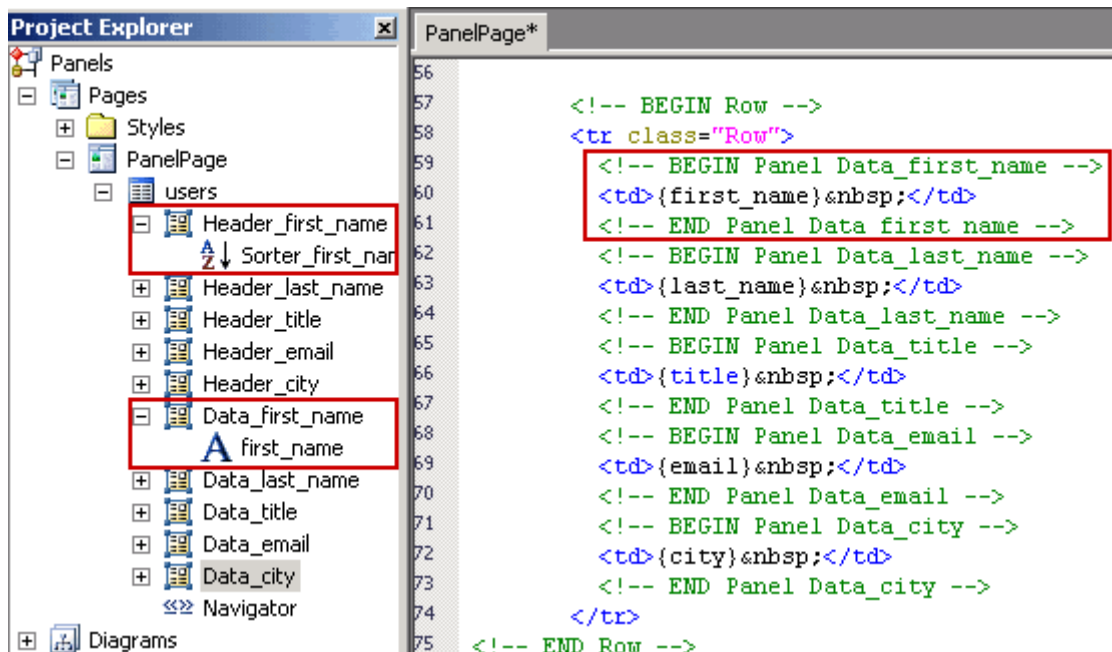


Creating grids with columns that can be hidden programmatically

The Grid Builder contains an option that automatically places all grid columns within panels for programmatic visibility control.



Note that two or three panels will be added for each column: for the column caption, data row and alternative data row (if alternate rows are used).



To hide a column, it is necessary to hide all panels corresponding to that column. Use the grid's *Before Show* event to hide the "Header_columnname" panel, and the *Before Show Row* event to hide the "Data_columnname" and "Data_Alt_columnname" panels.

Moving HTML code into/from a Panel

To move parts of HTML code from or into a panel, highlight an HTML snippet and drag it inside or outside the panel.

See also

[Panel Control](#),

[Panel Reference](#),

[Restricting SQL Execution for Components Hidden Using a Panel](#),

[Iterating Panel Controls](#),

[Working with Panels](#)

Restricting SQL Execution for Components Hidden Using a Panel

When using PHP, ASP or Perl; hiding a panel means that SQL queries will not be performed for any form included into the Panel. When using Java or .NET the data reading is performed prior to it being displayed, therefore it is necessary to use custom code that sets the component's *AllowRead* property to *False* if panel's *Visibility* property is *False*. The following custom code can be added in the page's *After Initialize* and component's *Before Select* events:

Java - After Initialize event

```
for (Iterator it =
e.getPage().getPanel("Panel1").getComponents().iterator();
it.hasNext(); ) {
    Object obj = it.next();
    if (obj instanceof Component) {
        ((Component) obj).setAllowRead(false);
    }
}
```

C# - After Initialize event

For each container component with the specified datasource add the code below:

```
<componentName>Operations.AllowRead =
<componentName>Operations.AllowRead && <panelName>.Visible;
```

For example, if Grid 'MyGrid' is inserted into Panel 'MyPanel' the code line should look like the following:

```
MyGridOperations.AllowRead = MyGridOperations.AllowRead &&
MyPanel.Visible;
```

VB - After Initialize event

For each container component with the specified datasource add the code below:

```
<componentName>Operations.AllowRead =
<componentName>Operations.AllowRead AndAlso <panelName>.Visible
```

For example, if Grid 'MyGrid' is inserted into Panel 'MyPanel' the code line should look like the following:

```
MyGridOperations.AllowRead = MyGridOperations.AllowRead AndAlso
MyPanel.Visible
```

See also

- [Panel Control](#)
- [Panel Reference](#)
- [Placing an existing component within a new panel](#)
- [Iterating Panel Controls](#)
- [Working with Panels](#)

Working with Reports

Inserting Row Number

This example shows how to insert row number ReportLabel in Report Component. We will change the value of control *Label1* located in a Report *Users* using several methods.

Method 1: Using Total Functions

This method allows you to show the Row Number and reset it for the whole report, separate page or needed group depending on the Reset At property value.

1. Specify [Control Source Type](#) property of Label1 as Database Column.
2. Leave [Control Source](#) property of Label1 empty.
3. Specify [Function](#) property of Label1 as Count.
4. Specify [Reset At](#) property of Label1 as you need.

Method 2: Using Special Value

This method allows you to show the Row Number and reset it for report.

1. Specify [Control Source Type](#) property of Label1 as Special Value.
2. Specify [Control Source](#) property of Label1 as Row Number.

Method 3: Using the Custom Code

1. Add the [Before Show](#) event to the Label1 control
2. Within the event, add the code below:

ASP

```
Users.Label1.Value = "Row Number: " & Users.RowNumber
```

Java

```
e.getReport().getControl("Label1").setValue("Row Number: " +  
e.getReport().getRowNumber());
```

PHP

```
global $Users;  
$Users->Label1->SetValue ("Row Number: " . $Users->RowNumber);
```

Perl

```
$Users->{Label1}->SetValue ("Row Number: " . $Users->  
>{RowNumber});
```

ColdFusion

```
<CFSET fldLabel1=RowNumber>
```

ASP.NET C#

```
UsersLabel1.Text = "Row Number: " +  
Users.RowNumber.ToString();
```

ASP.NET VB

```
UsersLabel1.Text = "Row Number: " + Users.RowNumber.ToString()
```

See also

[BeforeShow](#) event,
[Control Source](#) property,
[Control Source Type](#) property

Translating Control Values

Using Action to Dynamically Translate Control Value

This example explains how to retrieve a translation string by its key and assign it to a label control using the [Retrieve Value for Control](#) Action. The event code is placed in the [Before Show](#) event.

We will use the "CCS_Today" translation key from CCS project resources.

1. Add the [Retrieve Value for a Control](#) action to the [Before Show](#) event of the control.
2. Set the **Source Type** property of the action to *Translation* and the **Source Name** property to *CCS_Today*.

See also

- [Translating Control Values via Custom Code](#)
- [BeforeShow](#) event
- [Retrieve Value for a Control](#) action

Translating Control Values via Custom Code

This example demonstrates how to retrieve a translation string by its key and assign it to a label using the [Before Show](#) event.

We will use the "CCS_Today" translation key from CCS project resources.

1. Add the [Before Show](#) event to the Label control
2. Enter the code below into the event:

ASP

```
Function Label1_BeforeShow(Sender)  
    Sender.Value = CCSLocales.GetText("CCS_Today", Empty)  
End Function
```

PHP

```
function Label1_BeforeShow(& $sender) {  
    global $CCSLocales;  
    $Component->SetValue($CCSLocales->GetText("CCS_Today"));  
}
```

Perl

```
sub Label1_BeforeShow() {
    $Label1->SetValue($CCSLocales->GetText("CCS_Today"));
}
```

ColdFusion

```
<!--- Event: BeforeShow. Action: Custom Code. Control: Label1
--->
<CFSET fldLabel1 = Request.CCSLocales.GetText("CCS_Today")>
```

VB.Net

- o Net Framework 1.0, Net Framework 1.1:

```
o 'Label Label1 Event BeforeShow
```

```
Label1.Text = rm.GetString("CCS_Today")
```

- o Net Framework 2.0:

```
o 'Label Label1 Event BeforeShow
```

```
Label1.Text = Resources.strings.CCS_Today
```

C#

- o Net Framework 1.0, Net Framework 1.1:

```
o //Label Label1 Event BeforeShow
```

```
Label1.Text = rm.GetString("CCS_Today");
```

- o Net Framework 2.0:

```
o //Label Label1 Event BeforeShow
```

```
Label1.Text = Resources.strings.CCS_Today;
```

Java

```
//Label1_BeforeShow

e.getControl().setValue(e.getPage().getResourceString("CCS_Today"));
```

See also

- [Using Action to Dynamically Translate Control Value](#)
- [BeforeShow](#) event

Other

Create a Grid Column with a Calculated Value

This example shows how to create a column with a calculated value in a Grid. In this case, we have a Grid called *Tasks* which has the fields *task_name*, *estimated_time* and *expended_time*. We shall add a new field called *task_percent* which will contain the calculated value.

1. Add an additional column to the grid. There are two ways to do this:

- Switch to **HTML mode** and manually add HTML code for a new column `<td></td>` or
 - Switch to **Design mode**, place the cursor in the last column, invoke the context menu (right-click the mouse) and select the **Insert Column** option
2. Click the **Label** icon on the **Forms tab** (of the Toolbox) to insert a new [Label](#) control into the Grid column.
 3. Type `task_percent` into the [Name](#) property.
 4. Select `float` in the [Data Type](#) property.
 5. Enter `0.00` or any other value into the [Format](#) property.
 6. Add the [Before Show Row](#) event to the Grid.
 7. Inside the event, add the code below:

ASP

```
Function Tasks_BeforeShowRow(Sender)

    If Tasks.Estimated_Time.Value > 0 Then
        Tasks.Task_Percent.Value =
        (Tasks.Expended_Time.Value*100)/Tasks.Estimated_Time.Value
    Else
        Tasks.Task_Percent.Value = 0
    End if

End Function
```

PHP

```
function Tasks_BeforeShowRow(& $sender) {
    global $Tasks;

    if($Tasks->Estimated_Time->GetValue() > 0){
        $Tasks->Task_Percent->SetValue( ($Tasks->Expended_Time->
GetValue()*100)/$Tasks->Estimated_Time->GetValue());
    } else{
        $Tasks->Task_Percent->SetValue(0);
    }

}
```

Perl

```
sub Tasks_BeforeShowRow() {

    if ($Tasks->{Estimated_Time}->GetValue() > 0) {
        $Tasks->{Task_Percent}->SetValue( ($Tasks->{Expended_Time}-
        >GetValue()*100)/$Tasks->{Estimated_Time}->GetValue());
    } else {
        $Tasks->{Task_Percent}->SetValue(0);
    }

}
```

```
}
```

ColdFusion

```
<!--Tasks_BeforeShowRow -->  
  <CFIF fldEstimated Time GT 0 >  
    <CFSET  
fldTask Percent=(fldExpended Time*100)/fldEstimated Time>  
  <CFELSE>  
    <CFSET fldTask Percent=0>  
  </CFIF>
```

VB.Net

```
'Tasks_BeforeShowRow  
  If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =  
  ListItemType.AlternatingItem Then  
    Dim TasksTask_Percent As System.Web.UI.WebControls.Literal =  
    DirectCast(e.Item.FindControl("TasksTask_Percent"), System.Web.UI.WebCo  
    ntrols.Literal)  
    If DataItem.Estimated_Time.Value > 0 Then  
      TasksTask_Percent.Text =  
      ((DataItem.Expended_Time.Value*100)/DataItem.Estimated_Time.Value).ToS  
      tring()  
    Else  
      TasksTask_Percent.Text = "0"  
    End if  
  End If
```

C#

```
//Tasks_BeforeShowRow  
  
  if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==  
  ListItemType.AlternatingItem) {  
    System.Web.UI.WebControls.Literal TasksTask_Percent =  
    (System.Web.UI.WebControls.Literal)e.Item.FindControl("TasksTask_Perce  
    nt");  
    if ((Double)DataItem.Estimated_Time.Value > 0) {  
      TasksTask_Percent.Text =  
      (((Double)DataItem.Expended_Time.Value*100)/(Double)DataItem.Estimated  
      _Time.Value).ToString();  
    } else {  
      TasksTask_Percent.Text = "0";  
    }  
  }  
}
```

Java

```
//Tasks_BeforeShowRow  
  
  if(e.getGrid().getControl("Estimated_Time").getValue() != null &&
```

```

Utils.convertToDouble(e.getGrid().getControl("Estimated_Time").getValue())
.doubleValue()>0 ){
    e.getGrid().getControl("Task_Percent").setValue(

(Utils.convertToDouble(e.getGrid().getControl("Expended_Time").getValue())
.doubleValue()*100)/Utils.convertToDouble(e.getGrid().getControl("
Estimated_Time").getValue()).doubleValue());
} else {
    e.getGrid().getControl("Task_Percent").setValue(0);
}

```

See also:

[Before Show Row](#) event

Create Custom Session Variables

This example shows how to add a custom session variable when a user is logged in. In this case, we will store the user's name in a session variable.

1. Add an [On Click](#) event to the 'Login' [button](#) of a Login form.
2. Within the event, add the code below:

Note: In addition to creating the session variable, you should also clear it when the user logs out.

ASP

```

Function Login_DoLogin_OnClick(Sender)
Dim Connection1
    If Login_DoLogin_OnClick = True Then
        Set Connection1 = New clsDBConnection1
        Connection1.Open
        Session("User_Name") =
CCDLookUp("User_Name","Users","User_ID=" &
Connection1.ToSQL(CCGetUserID(),ccsInteger), Connection1)
        Connection1.Close
        Set Connection1 = Nothing
    End if
End Function

```

PHP

```

function Login_DoLogin_OnClick(& $sender) {

    if ($Login_DoLogin_OnClick == true) {
        $db = new clsDBConnection1();
        CCSetSession("User_Name",
CCDLookUp("User_Name","Users","User_ID=".$db-
>ToSQL(CCGetUserID(),ccsInteger), $db) );
        $db->close();
    }
}

```

```
}
```

Perl

```
sub Login_DoLogin_OnClick() {  
  
    if ($Login_DoLogin_OnClick == 1) {  
        $DBConnection1 = clsDBConnection1->new();  
        $User_Name = CCDLookUp("First_Name","Users","User_ID=".$db->  
>ToSQL(CCGetUserID(),$ccsInteger), $DBConnection1);  
        CCSetsession("User_Name", $User_Name);  
        $DBConnection1->{sth} = undef;  
        $DBConnection1->{dbh} = undef;  
    }  
  
}
```

ColdFusion

```
<!--DoLogin_OnClick --->  
  
    <CFIF CCLoginUser>  
        <CFModule Template="CCDLookUp.cfm" table="employees"  
field="emp_name" where="emp_id=#Session.UserID#"  
Connection="IntranetDB">  
            <CFLOCK NAME="Session" TIMEOUT="30" TYPE="Exclusive">  
                <CFSET Session.User_Name=CCDLookUp>  
            </CFLOCK>  
        </CFIF>
```

VB.NET

```
'Button Button_DoLogin Event OnClick. Action Custom Code  
  
Session("User_Name") =  
Settings.Connection1DataAccessObject.ExecuteScalar("SELECT User_Name  
FROM Users WHERE User_ID=" &  
Settings.Connection1DataAccessObject.ToSql(DBUtility.UserId.ToString()  
, FieldType._Integer))
```

C#

```
//Button Button_DoLogin Event OnClick. Action Custom Code  
  
Session["User_Name"] =  
Settings.Connection1DataAccessObject.ExecuteScalar("SELECT  
User_Name FROM Users WHERE User_ID=" +  
Settings.Connection1DataAccessObject.ToSql(DBUtility.UserId.To  
String(), FieldType.Integer));
```

Java

```
//DoLogin_OnClick
```

```

    if (Utils.getUserId(e.getPage()) != null ) {
        String userId =
DBTools.toSql(Utils.getUserId(e.getPage()),
JDBCConnection.TEXT, "Connection1");

        SessionStorage.getInstance(e.getPage().getRequest()).set
Attribute("User_Name",

        DBTools.dLookUp("User_Name","Users","User_ID="+userId,"
Connection1"));
    }

```

See also:

[On Click](#) event

Dynamically Modify a Hyperlink

Description

This example shows how to dynamically modify a hyperlink for the [Link](#) and [Image Link](#) controls.

Method 1


Use this method if you have all required data in database fields and want to lay-out controls in HTML.

1. Use the Toolbox to create one or more [Label](#) controls
2. In the [Control Source](#) property set up the database field that will be used as the link name
3. Modify the HTML for the link as shown below:

```
<a href="{label1}?parameter={label2}">{label3}</a>
```

Method 2

Use this method if you have all required data in database fields and want to modify hyperlink parameters only.

1. Use the Toolbox to add a [Link](#) or [Image Link](#) control to the form
2. Specify the text to be shown for the link by selecting a database column in [Control Source](#) property
3. Specify the hyperlink by selecting a database column in [Href Source](#) property
4. Click the  icon on the right side of the [Href Source](#) property in the Properties window
5. Add the required parameters

Method 3

Use this method if you want to have full control over the hyperlink modification.

1. Use the Toolbox to add a [Link](#) or [Image Link](#) control to the form
2. Add an *event* that will allow you to modify the control's value:
3. Use the [Before Show Row](#) event for the Grid or Editable Grid form or use the [Before Show](#) event for a control or the Record form.
4. Inside the event, add the code below:

ASP

```
Function Users_Link1_BeforeShowRow(Sender)

    Link1.Page = "UserPage.asp"

    Link1.Parameters = CCAddParam(Link1.Parameters,
    "user_id", "1")

    Link1.Value = Users.First_Name & " " & Users.Last_Name

End Function
```

PHP

```
function Users_Link1_BeforeShowRow(& $sender) {
    global $Users;

    $Users->Link1->Page = "UserPage.php";
    $Users->Link1->Parameters =
    CCGetQueryString("QueryString", "");
    $Users->Link1->Parameters = CCAddParam($Users-
    >Link1->Parameters, "user_id", "1");
    $Users->Link1->SetValue($Users->First_Name-
    >GetValue() ." ". $Users->Last_Name->GetValue() ) ;
}
```

Perl

```
sub Users_Link1_BeforeShowRow() {

    $Users->{Link1}->{Page} = "UserPage.cgi";
    $Users->{Link1}->{Parameters} =
    CCGetQueryString("QueryString", "");
    $Users->{Link1}->{Parameters} = CCAddParam($Users-
    >{Link1}->{Parameters}, "user_id", "1");
    $Users->{Link1}->{SetValue}($Users->First_Name-
    >{GetValue}() ." ". $Users->Last_Name->{GetValue}()) ;
}
```

```
}
```

ColdFusion

```
<!--Users_BeforeShowRow --->

<CFSET fldLink1 =fldFirst Name & " " &
fldLast Name>

<CFMODULE Template="CCGetQueryString.cfm"
strCollection="QueryString"
arrRemoveParameters="#ListToArray('CFID;CFTOKEN',';')#"
>

<CFMODULE Template="CCAddParam.cfm"
strQueryString="#CCGetQueryString#" strName="user_id"
strValue="#flduser id#" outputVar="CCGetQueryString">

<CFSET fldLink1Link="UserPage.cfm" &
IIF(CCGetQueryString NEQ
"", "'?#CCGetQueryString#'", "'')>
```

Java

```
//Users BeforeShowRow

Link link1 = e.getGrid().getLink("Link1");
link1.setHrefSourceValue("UserPage.do");
link1.setPreserveType(PreserveParameterType.GET);
link1.clearParameters();
LinkParameter userId = new LinkParameter("user_id",
"", ParameterSource.EXPRESSION);
userId.setValue("1");
link1.addParameter(userId);

link1.setValue(e.getGrid().getControl("First_Name").get
Value() + " " +
e.getGrid().getControl("Last_Name").getValue() );
```

For .NET

Use this method if you want to have full control over the hyperlink modification.

1. Use the Toolbox to add a [Link](#) or [Image Link](#) control to the form
2. In the [Before Show](#) event of the Link, add the code below

VB.Net

```
'Tasks_Link1_BeforeShow

item.Link1HrefParameters.Add("user_id",1)
Link1.HRef = "UserPage.aspx" &
item.Link1HrefParameters.ToString("GET", "")
```

```
Link1.InnerText = UsersFirst_Name &" "&
UsersLast_Name
```

C#

```
//Task_Link1_BeforeShow

item.Link1HrefParameters.Add("user_id",1);
Link1.HRef = "UserPage.aspx" +
item.Link1HrefParameters.ToString("GET","");
Link1.InnerText = UsersFirst_Name +" "+
UsersLast_Name;
```

See Also:

How to [create a MailTo hyperlink](#)

Dynamically Set the Default Value of a ListBox

This example shows how to dynamically set a default value for a control. In this case, we have a Listbox called *MonthBox* which is located in a Record form called *Registration*. The event code checks for an input parameter called *MonthBox* and if the input parameter is not available, the Listbox value is set to the current month. Add the [Before Show](#) event for the [list box](#). Within the event, add the code below:

ASP

```
Function Registration_MonthBox_BeforeShow(Sender)

If IsEmpty(CCGetParam("MonthBox",Empty)) Then
    Registration.MonthBox.Value = Month(NOW())
End if

End Function
```

PHP

```
function Registration_MonthBox_BeforeShow(& $sender) {
global $Registration;

if (CCGetParam("MonthBox","") == "") {
    $Registration->MonthBox->SetValue(date("n"));
}

}
```

Perl

```
sub Registration_MonthBox_BeforeShow() {

    $mon = (localtime())[4];
    if (CCGetParam("MonthBox","") eq "") {
        $Registration->{MonthBox}->{Value} = $mon+1;
    }

}
```

ColdFusion

```
<!--Registration_MonthBox_BeforeShow --->
    <CFMODULE Template="CCGetParam.cfm"
    strName="MonthBox">
    <CFIF CCGetParam EQ "">
        <CFSET fldMonthBox=DateFormat(Now(),"m")>
    </CFIF>
```

VB.Net

```
'Registration_MonthBox_BeforeShow

    If IsNothing(Request.QueryString("MonthBox")) Or
Request.QueryString("MonthBox") = "" Then
        RegistrationMonthBox.Value = Month(NOW())
    End if
```

C#

```
//Registration_MonthBox_BeforeShow
    if (Request.QueryString["MonthBox"] == null ||
Request.QueryString["MonthBox"].Length == 0) {
        RegistrationMonthBox.Value =
DateTime.Now.Month.ToString();
    }
```

Java

```
//Registration_MonthBox_BeforeShow

    if
(StringUtils.isEmpty(e.getPage().getParameter("MonthBox
")))) {
        e.getControl().setValue(new
java.text.SimpleDateFormat("M").format(new Date()));
    }
```

See Also:

[Before Show](#) event

Examine Form's Mode

This example shows how to identify if a record form is in **Add** or **Edit** mode. **Edit** mode occurs when a record has been retrieved and can be altered or deleted. **Add** mode occurs when no record has been retrieved in which case the only possible operation would be to add a new record. Add the [Before Show](#) Event. Within the event, add the code below:

ASP

```
Function Form1_BeforeShow(Sender)

    If (EventCaller.EditMode) Then
        'Edit Mode
    Else
        'Add Mode
    End If

End Function
```

PHP

```
function Form1_BeforeShow(& $sender) {
    global $Form1;

    If ($Form1->EditMode)
        // Edit Mode
    Else
        // Add Mode
    }
}
```

Perl

```
sub Form1_BeforeShow() {

    if ($Form1->{EditMode} == 1) {
        #Edit Mode
    } else {
        #Add Mode
    }
}
```

```
}
```

ColdFusion

```
<--Form1_BeforeShow -->  
<CFIF blnEditModeForm1>  
  <!-- Edit mode --->  
<CFELSE>  
  <!-- Add mode --->  
</CFIF>
```

VB.Net

```
'Form1_BeforeShow  
  
If (IsInsertMode) Then  
  'Add Mode  
Else  
  'Edit Mode  
End If
```

C#

```
//Form1_BeforeShow  
  
if (IsInsertMode) {  
  //Add Mode  
} else {  
  //Edit Mode  
}
```

Java

```
//Form1_BeforeShow  
  
if (e.getRecord().isEditMode()) {  
  //Edit Mode  
} else {  
  //Add Mode  
}
```

See Also:

[Before Show](#) Event

Replace Control Value Before Inserting

This example shows how to change the value of a control before the Insert and Update operation. In this case, we have a Record form called *Tasks* which has two hidden fields called *LastUpdateDate* and *LastUpdateUserid*. This fields will store the current date and ID of the user who performs the operation.

1. Add the custom *LastUpdateInfo()* function.
2. Add the [Before Insert](#) and [Before Update](#) events for the *Tasks* Grid.
3. Call the *LastUpdateInfo()* function in both of the events.

ASP

```
Function LastUpdateInfo()  
  
    Tasks.LastUpdateDate.Value = Now()  
    Tasks.LastUpdateUserid.Value = CCGetUserID\(\)  
  
End function  
  
Function Tasks_BeforeInsert()  
    LastUpdateInfo()  
End Function  
  
Function Tasks_BeforeUpdate()  
    LastUpdateInfo()  
End Function
```

PHP

```
function LastUpdateInfo(){  
    global $Tasks;  
  
    $Tasks->LastUpdateDate->SetValue(date("m.d.y"));  
    $Tasks->LastUpdateUserid->  
>SetValue(CCGetUserID\(\));  
  
}  
  
function Tasks_BeforeInsert() {  
    LastUpdateInfo();  
}  
  
function Tasks_BeforeUpdate() {  
    LastUpdateInfo();  
}
```

Perl

```
sub LastUpdateInfo() {  
  
    $Tasks->{LastUpdateDate}->SetValue(time());  
    $Tasks->{LastUpdateUserid}-  
>SetValue(CCGetUserID\(\));  
  
}  
sub Tasks_BeforeInsert() {  
    LastUpdateInfo();  
}  
  
sub Tasks_BeforeUpdate() {  
    LastUpdateInfo();  
}
```

ColdFusion

```
<!--Tasks_BeforeInsert -->  
  
    <CFSET fldLastUpdateDate=Now()>  
    <CFSET fldLastUpdateUserid=Session.UserID>  
<!--Tasks_BeforeUpdate -->  
  
    <CFSET fldLastUpdateDate=Now()>  
    <CFSET fldLastUpdateUserid=Session.UserID>
```

Java

```
class LUI {  
    public void lastUpdateInfo(Page page) {  
  
page.getRecord("Tasks").getControl("LastUpdateDate").set  
tValue(new Date());  
  
page.getRecord("Tasks").getControl("LastUpdateUserid").  
setValue(Utils.getUserId(page));  
    }  
}  
  
//Tasks_BeforeInsert  
  
    new LUI().lastUpdateInfo(e.getPage());  
  
//Tasks_BeforeUpdate  
  
    new LUI().lastUpdateInfo(e.getPage());
```

VB.Net

```
Function LastUpdateInfo()  
    TasksLastUpdateDate.Text = DateTime.Now.ToString()  
    TasksLastUpdateUserid.Value =  
    DBUtility.UserId.ToString()  
End function  
'Tasks_BeforeInsert code  
    LastUpdateInfo()  
'Tasks_BeforeUpdate code  
    LastUpdateInfo()
```

C#

```
void LastUpdateInfo()  
{  
    TasksLastUpdateDate.Text = DateTime.Now.ToString();  
    TasksLastUpdateUserid.Value =  
    DBUtility.UserId.ToString();  
}  
\\Tasks_BeforeInsert code  
    LastUpdateInfo();  
\\Tasks_BeforeUpdate code  
    LastUpdateInfo();
```

See also:

[Before Insert](#) event,
[Before Update](#) event

Working With Component Attributes

This example demonstrates how to dynamically set value of a component attribute, in this case disabling textbox when users view their profile.

Let's assume that we have a TextBox named *user_login*, located in *Registration* form. The event code disables the textbox if the URL contains *mode=profile*.

1. While in the HTML mode use the context menu to insert `{user_login:disabled}` [attribute](#) into TextBox template. The HTML should look as follows:

```
<input name="{user_login_Name}"  
value="{user_login}" {user_login:disabled}>
```

2. 'Custom Code' action to the [Before Show](#) event of the TextBox.
3. Use the code shown below within the action body:

ASP

```
If CCGetFromGet("mode", "") = "profile" Then _
    Sender.Attributes("disabled") = "disabled"
```

PHP

```
if (CCGetParam('mode', '') == 'profile') {
    $Component->Attributes->SetValue('disabled',
    'disabled');
}
```

Perl

```
if (CCGetFromGet('mode', '') eq 'profile') {
    $Component->{Attributes}->SetValue('disabled',
    'disabled');
}
```

ColdFusion

```
<CFPARAM Name="URL.mode" Default="">
<CFIF URL.mode EQ "profile">
    <CFSET StructInsert(attrUser_login, "disabled",
    "disabled", True)>
</CFIF>
```

VB.Net

```
If Request.QueryString("mode") = "profile" Then
    ControlAttributes.Add(usersuser_login, New
    CCSCControlAttribute("disabled", FieldType._Text,
    "true"))
End If
```

C#

```
if (Request.QueryString["mode"] == "profile") {
    ControlAttributes.Add(usersuser_login, new
    CCSCControlAttribute("disabled", FieldType.Text,
    "true"));
}
```

Java

```
if
("profile".equals(e.getPage().getHttpGetParams().getPar
ameter("mode"))) {
    ((com.codecharge.util.ModelAttribute)e.getModel().g
etAttribute("disabled")).setValue("disabled");
}
```

See Also:

- [Before Show](#) event

- [Component Attributes](#)

Debugging

Displaying Output

Although CodeCharge Studio doesn't include a debugger, you can perform various debugging tasks using a small amount of manual coding within the events. One of the common debugging tasks is to examine the value of a variable within the application. You can often do this by adding a line of code to your program that will display the value of any variable.

ASP

When using ASP, you can use the following code to print out variables within most events:

```
response.write variable_name
```

In a similar fashion you can also display the value of a control, such as a Text Box, within its **Before Show** event:

```
response.write TextBox1.value
```

Additionally, you can use the **Before Execute Select** event of a Grid to examine parts of the SQL statement of the form by using the following code:

```
response.write "SQL:" & GridName.datasource.SQL & "<br>"  
  
response.write "ORDER BY:" & GridName.datasource.Order & "<br>"  
  
response.write "WHERE:" & GridName.datasource.Where & "<br>"
```

(similar code can also be used to dynamically alter the SQL)

Keep in mind that sometimes the VBScript command `response.write` may appear not to work, especially when the page is being redirected to another page or refreshed. In this case you can disable the page buffering option by adding the following code to the beginning of the Common.asp file (after the 'End Include Files comment):

```
response.buffer = False
```

The following MSDN articles also provide information about debugging when using ASP:

- [Debugging ASP Applications in IIS](#)
- [Debugging Server Script](#)

PHP

When using PHP, you can use the following code to print out variables within most events:

```
global $variable_name;

echo $variable_name;
```

In a similar fashion you can also display the value of a control, such as a Text Box, within its **Before Show** event:

```
global $TextBox1;

echo $TextBox1->GetValue\(\);
```

Additionally, you can use the **Before Execute Select** event of a Grid to examine parts of the SQL statement of the form by using the following code:

```
global $GridName;

echo "SQL:" . $GridName->DataSource->SQL . "<br>";
echo "ORDER BY:" . $GridName->DataSource->Order . "<br>";
echo "WHERE:" . $GridName->DataSource->Where . "<br>";
```

(similar code can also be used to dynamically alter the SQL)

Keep in mind that sometimes the PHP command `echo` may appear not to work, especially when the page is being redirected to another page or refreshed. In this case you can disable the page buffering option by adding the following code to the beginning of the Common.php file (after the `//End Include Files` comment):

```
ini_set("output_buffering", "Off");
```

Perl

As a prerequisite to printing any debug information in Perl, make sure that you set common header information as follows:

```
if (!$isHTTPHeader) {
    $isHTTPHeader = 1;
    print $page_header;
    print "Content-type: text/html\n\n";
}
```

After setting the header information, you can use the following code to print out variables within most events:

```
print $variable_name;
```

In a similar fashion you can also display the value of a control, such as a Text Box, within its **Before Show** event:

```
print $TextBox1->GetValue();
```

Additionally, you can use the **Before Execute Select** event of a Grid to examine parts of the SQL statement of the form by using the following code:

```
print "SQL:" . $GridName->{DataSource}->{SQL} . "<br>";  
  
print "ORDER BY:" . $GridName->{DataSource}->{Order} . "<br>";  
  
print "WHERE:" . $GridName->{DataSource}->{Where} . "<br>";
```

(similar code can also be used to dynamically alter the SQL)

ColdFusion

When using ColdFusion, you can use the following code to print out variables within most events:

```
<cfoutput>#variable name#</cfoutput>
```

In a similar fashion you can also display the value of a control, such as a Text Box, within its **Before Show** event:

```
<cfoutput>#fldTextBox1#</cfoutput>
```

Additionally, you can use the **Before Execute Select** event of a Grid to examine parts of the SQL statement of the form by using the following code:

```
<cfoutput><br>SQL: #strSQL#</cfoutput>  
  
<cfoutput><br>ORDER BY: #strOrder#</cfoutput>  
  
<cfoutput><br>WHERE: #strWhere#</cfoutput>
```

(similar code can also be used to dynamically alter the SQL)

Keep in mind that sometimes the ColdFusion command `<cfoutput>` may appear not to work, especially when the page is being redirected to another page or refreshed. You can disable redirection by using the following command:

```
<CFSET strRedirect="">
```

Java

The code generated by CodeCharge Studio Servlets and JSP patterns includes logging capabilities that can be configured to trace your application's execution. The logging

properties are located in the **Properties** window when the project name is selected in the **Project Explorer** window. These properties are as follows:

Path to logfile	The absolute path to a file where the log messages should be written. This property also supports predefined values such as <i>out</i> and <i>err</i> to write to the standard output and standard error streams respectively. The value <i>none</i> will switch off the logging functionality.
Level of Logging	There are four levels that determine the verbosity of logging: <i>debug</i> , <i>info</i> , <i>warn</i> , <i>error</i> . <i>Debug</i> is the most verbose level and <i>error</i> is the least. The <i>Debug</i> level will produce messages about page processing and output SQL statements being executed.

To add your messages to the logging stream from events or custom code, you can use the following code:

```
CCLogger.GetInstance().debug("In BeforeShow event, changing label color");
```

.NET

The .NET framework provides a various methods for debugging and profiling applications. Similar to ASP you can use `Response.Write` statements in events to examine variables and controls values

```
System.Web.HttpContext.Current.Response.Write(variable_name);
```

```
System.Web.HttpContext.Current.Response.Write(variable_name)
```

Keep in mind that the variable whose value should be displayed must be of the `System.String` data type. If the variable is of a different data type, you should call the `ToString()` method to produce a value of the type `System.String`.

In a similar fashion you can also display the value of a control, such as a Text Box, within its **Before Show** event:

```
System.Web.HttpContext.Current.Response.Write(TextBox1.Text);
```

```
System.Web.HttpContext.Current.Response.Write(TextBox1.Text)
```

Additionally, you can use the **Before Execute Select** event of a Grid to examine the Grid's SQL statement with the following code:

```
System.Web.HttpContext.Current.Response.Write(Select.ToString());
```

```
System.Web.HttpContext.Current.Response.Write(Select_.ToString())
```

You can also use **tracing** to produce debugging information. This method includes functionality that allows you to write debug statements directly in your code, without having to remove the statements from your application when it is deployed to production servers. For detailed information about this, please refer to the section on [ASP.NET Trace](#) in the MSDN documentation.

The following MSDN articles also provide information about debugging in the .Net environment:

- Using the [Common Language Runtime \(CLR\) Debugger](#)
- Using the [Visual Studio. Net Debugger](#)
- [Debugging ASP.Net Web Applications](#)

CodeCharge Studio Example Pack

The CodeCharge Studio example pack comes in the form of a CodeCharge Studio project within which there are different sub-projects which implement various programming techniques. To create an instance of the CodeCharge Studio example pack, proceed as follows:

1. In the **File** menu, select the **New --> Project** option.
2. In the **Add New Project** dialog, click on the **Solutions** tab and select **CCS Example Pack** from the available options.
3. Specify a Name and Path where the project files will be created.
4. Click **OK** to create the Project.

The following table contains descriptions of the different examples found in the example pack.

Example	Description
Index/Directory Usage	
Directory & Path Implementation	This example shows how to implement a simple Yellow Pages application that consists of a Path, Directory and a Grid.
Directory Maintenance System	This example shows a sample implementation of a system that can be used to manage hierarchical data. In this case, it is the list of categories that are later used in the directory.
Editable Grid Usage	
Record Maintenance with Navigation	This example shows how to use the editable grid component to maintain individual records while at the same time letting users navigate between the records without having to go back to a grid page to select another record.
Editable Grid with 'Add Row' Button	This example shows how to create an Editable Grid in which you can dynamically add empty rows for adding new records. Usually, an editable grid has a set number of empty rows for adding new records. This example shows how the empty rows can be dynamically displayed as needed by clicking on a 'Add Row' button.
Editable Grid with Auto New Rows	This example shows how to create an Editable Grid in which new empty rows are automatically added as you progressively enter information now new records. Usually, an editable grid has a set number of empty rows for adding new records. This example

	shows how the empty rows can be dynamically displayed as the user fills in information. JavaScript code is used to detect whenever new information is added to a field so that a new row can be displayed automatically.
Creating Master/Detail Forms	
Simple Master-Detail for Employees/Projects	This example shows how to implement a master-detail page consisting of grid, record and editable grid forms. The grid form lists all the 'master' records while the record form and editable grid display detail information for a selected master record.
Master-Detail Form for Order Entry	This example shows how to create a master-detail order entry page. A grid form is used to display a list of all the orders while a record and editable grid form display detailed information about a selected order.
Working with Multiple Selections	
Updating Many-to-Many via Multi-select ListBox	This example shows how to implement a sample record form that allows users to assign multiple employees to a project using a multi-select ListBox. This is an example of a many-to-many relationship whereby one project can have many employees and one employee can have many projects.
Updating Many-to-Many via CheckBox List	This example shows how to implement a sample record form that allows users to assign multiple employees to a project using a CheckBox List control. This is an example of a many-to-many relationship whereby one project can have many employees and one employee can have many projects.
Updating Many-to-Many via Two ListBoxes	This example shows how to implement a sample record form that allows users to assign multiple projects to an employee using two Listboxes. This is an example of a many-to-many relationship whereby one employee can have many projects and one project can have many employees.
Search Form with Multi-select ListBox	This example shows how use a multi-select ListBox in a search form.
Data Presentation Techniques	

Grid with Navigable Detail View	This example shows how to use Grid forms to traverse multiple records and view their details at the same time. In this case, there is no need to switch between a Grid and a Record form to show record details since a grid with a Navigator controls shows all the records with their details.
Data Input Techniques	
Pop-up List & Smart Lookup	This example shows an implementation of a pop-up list for selecting values and a smart lookup for quick data entry. The lookup further enhances the process by allowing the user to specify partial information when searching for a manager i.e. the managers initials, first or last name or a partial name.
Multi-Step User Registration	This example shows how to create a multi-page user registration and editing form.
Data Entry page with Input Mask	This example shows how to create a form with input mask fields.
Other	
Using a Multi-Column Primary Key	This example shows how to implement a sample record form based on a table that uses a multi-column primary key.

CodeCharge Studio Example Solutions

CodeCharge Studio comes with a number of example solutions that you can use:

- To learn how to build projects of your own.
- As a basis for building your own projects by making use of the pre-built functionality in the solutions.

To create a project based on one of the example solutions:

1. In the **File** menu, select the **New --> Project** option.
2. In the **Add New Project** dialog, click on the **Solutions** tab and select the solution you want to base your project on.
3. You will then need to specify a name for the project as well as the location where the new project files will be saved. You also have to select the language in which the solution will be based.
4. Once you click on the **OK** button, CodeCharge Studio will create the solution pages and at the end of it, prompt you to choose a style to be used for the project. You can either select a style from the list or click on the 'Cancel' button to use the default style.

After selecting the style, the new project and all its pages appear within the CodeCharge Studio IDE. In its current state, you can configure the project settings then deploy the project immediately. If desired, you can also customize the project pages or add pages of your own.

For each of the solutions, you can create the project in any one of the supported languages. Please note that for databases like MSSQL Server, MySQL Server and other please use the SQL scripts located in the following folders to create the databases: `⌘:\Program Files\CodeChargeStudio\Examples\Internet\` and `⌘:\Program Files\CodeChargeStudio\Examples\Intranet\`.

The following table contains descriptions of the different example solutions.

Example Solution	Description
Intranet	
Employee Directory	Implements a solution to manage employees who belong to different departments.
Task Manager	Implements a task management solution. The tasks are categorized by project, priority, status as well as type.
Internet	
Portal	Implements a club portal with different sections for news, user registration, events, links, club officers and administration.
Forum	Implements a simple forum where users can post and reply to messages.
Registration Form	Implements a simple registration system where users can be registered as well as view their existing information.
Store	Implements a basic online store application.

Common Errors and Warnings

Common Errors

Could not load type error (.NET)

Parser Error Message: Could not load type 'TaskManager.tasks_list.tasks_list'.

Source Error

```
Line 1: <!--ASPX page @1-3BD59FD0-->
Line 2: <%@ Page language="vb" Codebehind="tasks_list.aspx.vb"
AutoEventWireup="false" Inherits="TaskManager.tasks_list.tasks_list"
%>
```

```
Line 3: <%@ Import namespace="TaskManager.tasks_list" %>
Line 4: <%@ Import namespace="TaskManager.Configuration" %>
```

Solution

The directory you publish to must be a virtual directory.

See also

[ASP.Net\(C#\) Deployment Guide.](#)

HTTP 500 Internal Server Error

This message is a "friendly" representation of a non-standard error that occurred on the web server. To see the full error message, turn off "friendly error messages in your web browser".

For more information, refer to: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q294807>

Incorrect File Paths

While working with CodeCharge Studio projects in the Microsofté FrontPageé Add-In the project paths containing a URL (<http://>) like <http://hostname/projectname/> are not supported.

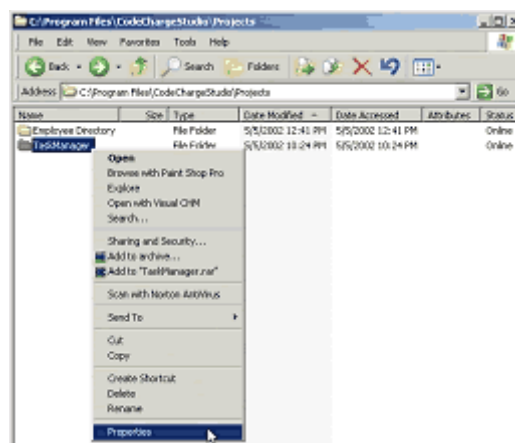
It is recommended that you create projects in the file system, for example `c:\projectfolder\projectname\`, taking into consideration that:

- The project should not be created in the IIS folders (`c:\inetpub\wwwroot\projectname`).
- The project name should contain only alphanumeric characters.

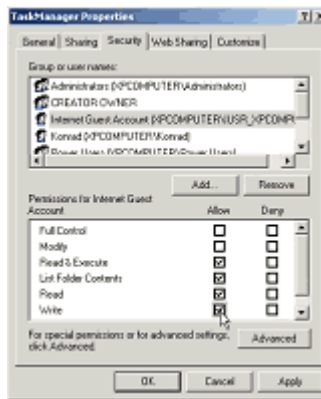
Microsoft JET Database Engine (0x80004005) Could not use "; file already in use.

This error usually happens on Windows NT/2000/XP when your database file or database folder doesn't have write permissions. You can often solve it by following these steps:

1. Right-click on the project folder on your disk and select **Properties**.



2. In the **Security tab** add Internet Guest Account and set the Write permissions checkbox.



For more information, refer to: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q174943>

Operation must use an updateable query. (Microsoft JET Database Engine)

This is one of the most common ASP errors that occurs when the data is being updated in a Microsoft Access database that doesn't have sufficient access privileges.

Solution

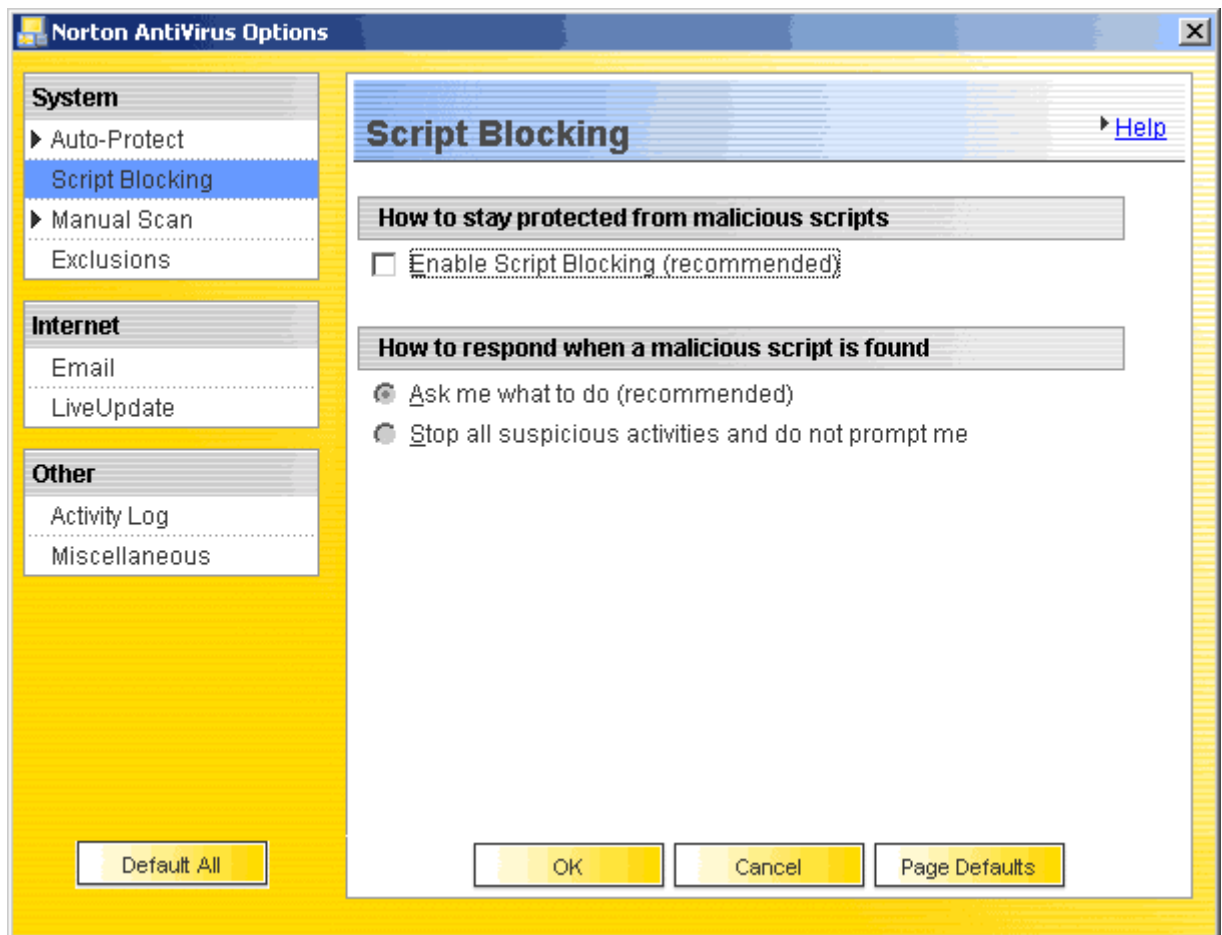
- In Windows 98 or ME: Right click on the MS Access file (.mdb) and uncheck the "Read-only" property.
- In Windows NT, 2000 or XP: Ask your system administrator to setup full access permissions for the anonymous user account called IUSR_<MachineName> to access the database.

If you are hosting your website externally, your web hosting company most likely provides a special read-write folder for databases or will create one for you. Please check their FAQ and other information on your hosting company's website or contact them directly.

For more information, refer to: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q175168>

Page takes forever to load or the IIS web server appears to hang

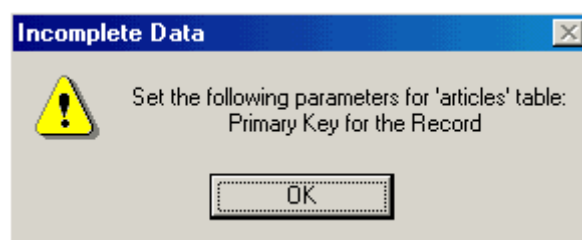
This issue is usually caused by Norton AntiVirus being configured to block script execution. If you have Norton AntiVirus installed on your machine, disable Script Blocking as shown below.



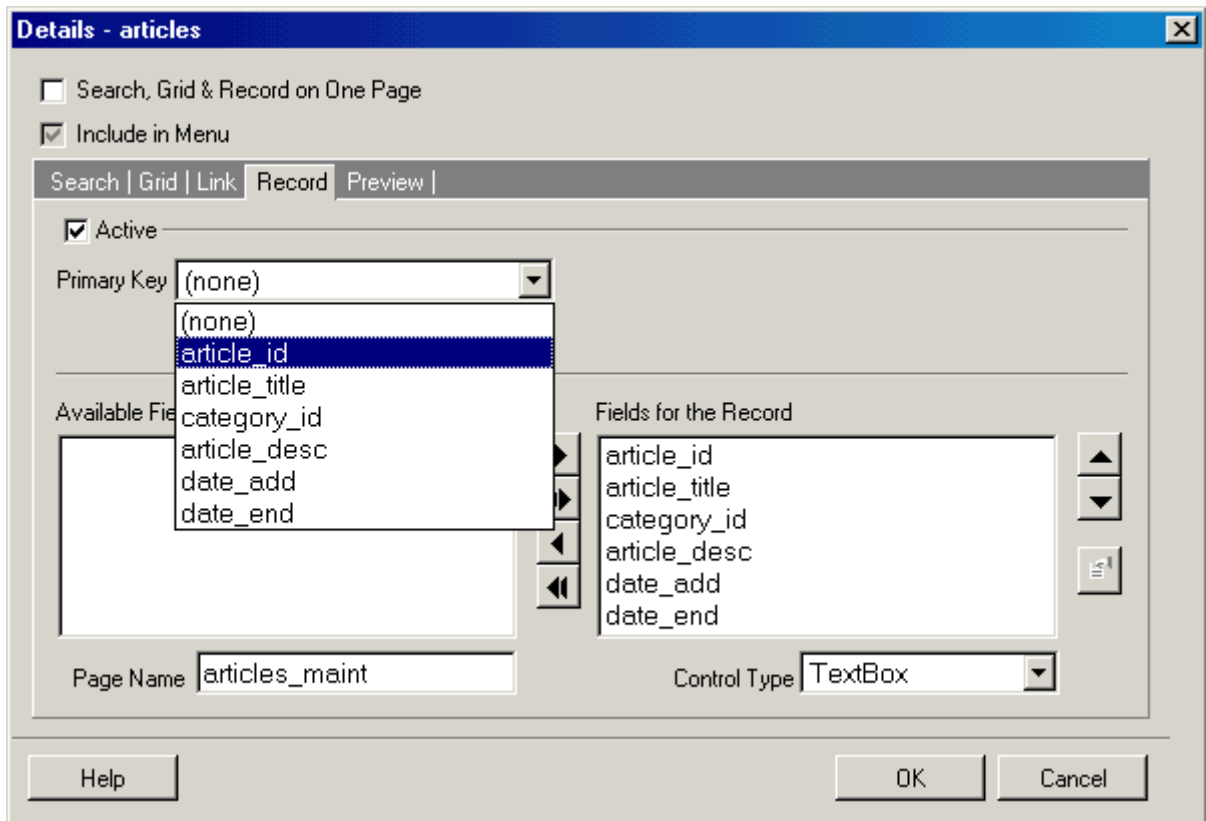
Warnings

Application Builder: Incomplete Data Message.

The following message window appears on Step six (6) of the Application Builder if the database table does not have the Primary Key or ODBC connection is used as a Data Source.



If ODBC is used, it is unable to retrieve the Primary Key for some databases. In this case please select a particular table name under the Table column, select the Primary Key field or select the field to be used as a Primary Key for the Record tab.



See also
[Application Builder](#)