

The Crescent Master/Detail Control and Objects Help Contents

The Crescent Master/Detail control consists of the following controls:

- CMasterDetail Control
- CParameter Object
- CParameters Collection



Crescent Master/Detail Control

The Crescent Master/Detail control is a custom control provided by Crescent DBPak. It is an invisible at run-time control that allows you to more easily create master/detail presentations of information from a database. The Master/Detail control can be used in either a single-form or a multiform scenario.

Typically, the Master/Detail control will interact with two data controls; one, a master data control, and the other a detail data control. The master data control is used to navigate through a specified recordset. The detail data control navigates through a recordset based on a field in the current record of the master data control. Both data controls can be used to display information in bound controls that you place on the form.

This help file describes the properties, methods, and events for the Master/Detail control.

Properties

<u>About</u> *	<u>CParameters</u> *	<u>Custom</u> *
<u>DataField</u> *	<u>DataSource</u> *	<u>DetailDataSource</u> *
<u>DetailRefreshMode</u> *	<u>Enabled</u> *	Index
Left	Name	<u>SQL</u> *
<u>Substituted SQL</u> *	Tag	Top

Methods

Object	Parent	<u>RefreshDetail</u> *
--------	--------	------------------------

Events

<u>BeforeRefreshDetail</u> *	<u>Error</u> *
------------------------------	----------------

Indicates a custom or modified property, method, or event.

(About) Property

Applies To

Master/Detail

Purpose

Provides access to version and copyright information for the Master/Detail control at design time.

Comments

Double-click this property in the Visual Basic property sheet for the Master/Detail control to display the About dialog box for the control.

CParameters Property

Applies To

Master/Detail

Purpose

The CParameters property is a parameters collection.

Syntax

```
[Form1.]CMasterDetail1.CParameters
```

Data Type

CParameter Collection

Usage

Read/write at design-time and at run time. This property can be manipulated at design-time using the custom property page. It can be manipulated at run time using the standard collection manipulation syntax.

Comments

The Master/Detail Control supports a property page that allows you to define the set of CParameter Objects in the CParameters Collection at design-time. The CParameters property page lists the current CParameters. You can add and delete CParameter Objects. The Value of any CParameter added in this page will be of type String, and will be saved with the project when the form is saved.

The following example shows accessing a CParameter Object:

```
MasterDetail1.CParameters(recent).Value = #5/5/96#
```

You can access a CParameter Collection either by using name of a CParameter or by the number of the CParameters index. The numbered index begins with zero. The next CParameter has an index of one. For example, you could access a CParameter in the following ways:

```
CMasterDetail1.CParameterCollection(A1).Value  
CMasterDetail1.CParameterCollection.Item(0).Value
```

The first line in this example accesses the CParameter Object named A1. The second line accesses the first CParameter Object in the CParameters Collection.

See the documentation on the Property page for more information.

(Custom) Property

Applies To

Master/Detail

Purpose

Provides access to custom property sheets for the Master/Detail control at design-time.

Comments

Double-click this property in the Visual Basic property sheet for the Master/Detail control to display the Properties page for the Master/Detail control. The Properties page contains both the General and CParameters custom property sheets for the control. You can set the properties for the Master/Detail control either programmatically or on the Properties page.

The General custom property sheet provides an interface to easily set the following information for the Master/Detail control:

- [SQL](#)
- [DetailDataSource](#)
- [DetailRefreshMode](#)
- [Enabled](#)

The CParameters custom property sheet provides an interface to easily set the following information for the Master/Detail control:

- CParameters
- Add CParameter - You can add CParameter Objects by clicking in the Name and Value boxes and entering the names and values. See the [CParameter Object](#) topic for more information on setting the properties for a CParameter Object.
- Delete selected CParameters - You can delete CParameter Objects by selecting them in the CParameters box. Once you select one or more CParameter Objects, the Delete selected CParameters box is enabled. Click on it to delete the selected CParameter Objects.

DataField Property

Applies To

Master/Detail

Purpose

The DataField property specifies the field in the Master data controls RecordSet to be the key field used for selecting in the Detail data controls RecordSource (or SQL) property.

Syntax

```
[Form1.]CMasterDetail1.DataField [= value]
```

Data Type

String

Usage

Read/Write at design-time and run time. Normally set at design-time but can be set at run time.

Comments

The valid values for this property are the names of the fields available to the designated master data control. The default value is the NULL string. The master data control is specified in the DataSource property of the Master/Detail data control. The fields available to the master data control are the fields in the master data controls RecordSet. These fields are listed in the DataField property of the master data control. If no master data control has been specified or no master data controls RecordSource has been selected for access, there are no values listed for selection in this property.

When set at run time, this property causes an implicit RefreshDetail if the property specifies a valid field in the master data controls RecordSource.

If your master data controls RecordSource contains expressions, then it is recommended that you code your SQL queries to alias expression columns as shown in the following example:

```
Data1.RecordSource = "Select AVG(Sales) " _  
    & " AS AverageSales From SalesTable"  
CMasterDetail1.DataField = "AverageSales"  
CMasterDetail1.DataSource = Data1  
Data1.Refresh
```

DataSource Property

Applies To

Master/Detail

Purpose

The DataSource property specifies a data control to be the master data control.

Syntax

```
[Form1.]CMasterDetail1.DataSource [ = DataControlObjectName ]
```

Data Type

String

Usage

Read/Write at design-time.

Comments

Valid values include any data control on the form. There is no default value.

DetailDataSource Property

Applies To

Master/Detail

Purpose

The DetailDataSource property specifies the data control to act as the detail data control. This is the data control that accesses the database to retrieve the data specified in the SubstitutedSQL property.

Syntax

```
[Form1.]CMasterDetail1.DetailDataSource [ = DataControlName ]
```

Data Type

String

Usage

Read/Write at design-time and run time.

Comments

Valid values include any data control on the form. There is no default detail data control. You have to specify a detail data control in this property.

The master and detail data controls cannot be the same control.

DetailRefreshMode Property

Applies To

Master/Detail

Purpose

The DetailRefreshMode property specifies the mode for data to be refreshed, either automatically or manually. If the mode is Automatic, the data is refreshed whenever the data changes without requiring a request from the user or the programmer. If the mode is Manual, the data is only refreshed when the RefreshDetail method is invoked.

Syntax

```
[Form1.]CMasterDetail1.DetailRefreshMode [ = <mode value > ]
```

Data Type

Integer

Usage

Read/Write at design-time and run time.

Comments

This property has the following values:

Use this value... to specify this...

0 - Automatic automatic updating of the data when it changes

1 - Manual updating of the data only occurs when you specify an update action to take place.

The default value is 0 - Automatic.

Enabled Property

Applies To

Master/Detail

Purpose

The Enabled property allows the Master/Detail control to work. If it is not enabled, the control cannot perform any actions.

Syntax

```
[Form1.]CMasterDetail1.Enabled
```

Data Type

Boolean

Usage

Read/Write at design-time and run time.

Comments

Valid values are True and False. The default value is True.

Use this value... to specify this...

True to allow the Master/Detail control to access a database and display the data.

False to disable the Master/Detail control from accessing a database or displaying the data.

SQL Property

Applies To

Master/Detail

Purpose

The SQL property specifies an SQL statement that can have parameters that are replaced with values in the SubstitutedSQL property. See the SubstitutedSQL property for more information.

Syntax

```
[Form1.]CMasterDetail1.SQL [= string]
```

Data Type

String

Usage

Read/Write at design-time and run time.

Comments

Any valid SQL statement for the Detail data controls database can be entered.

Whenever the master data control repositions to a new record or the RefreshDetail method is invoked, the Master/Detail control performs the following actions:

1. Scans the string in the SQL property and substitutes any parameters specified with their current value. The SQL statement, with the parameter substitutions made, can be read using the SubstitutedSQL property.
2. The SQL statement, with the parameter substitutions made, is set into the RecordSource (or SQL) property of the data control specified by the DetailDataSource property.

If the DetailRefreshMode property is set to MANUAL, these steps are only performed when the RefreshDetail method is invoked. If the DetailDataSource property is set to an empty string, the second step is not performed.

Typically, the key parameter in the SQL statement is the field value in the master data control specified by the DataField property. This parameter is named **:key**. Parameter names are case-insensitive. It is up to the application programmer to compose the string in the SQL property, such that, when parameter substitutions are made, the result is a valid SQL statement for the detail controls RecordSource (or SQL) property. Do not have a space between the colon and the CParameter name.

For example, if you want to display master/detail information for the Customers and Orders tables in an SQL database and that the key field common to both tables is the Cust-Num field. The master data control, data control Data1 in this example, could have a RecordSource of SELECT * FROM Customers. The detail data control is Data2. The Master/Detail properties are set as follows:

```
DataField: Cust-Num  
DataSource: Data1  
DetailDataSource: Data2  
SQL: SELECT * FROM Orders WHERE Cust-Num = :key
```

Each time Data1, the master control, is positioned to a new record, the value of the Cust-Num field is read and substituted for **:key** in the string specified in the SQL property. In turn, this parameter substituted string is used to set the RecordSource (or SQL) property of the detail data control. For example, if the master data control was positioned on a new record and the Cust-Num field contained a 5, this SQL statement would read as follows after substitution:

```
SELECT * FROM Orders WHERE Cust-Num = 5
```

SubstitutedSQL Property

Applies To

Master/Detail

Purpose

The SubstitutedSQL property contains an SQL statement with substitutions for the current CParameter values as specified in the SQL property. The SubstitutedSQL statement is assigned to the RecordSource property (or SQL property) of the Detail data control when the Detail data control is refreshed or when the master data control repositions to a new record.

Syntax

```
[Form1.]CMasterDetail1.SubstitutedSQL
```

Data Type

String

Usage

Read-only at run time.

Comments

The parameter value(s) are substituted wherever the CParameter names appear in the SQL property. The CParameter names appear following a colon (:) in the SQL property.

For example, if the CParameters Collection contained the following object:

```
CParameter.name = recent  
CParameter.value = #1/7/96#
```

And you have the following SQL statement in the SQL property:

```
SELECT * FROM Orders WHERE Cust-Num = :key AND Date >= :recent
```

Each time the master control is positioned to a new record, the value of the Cust-Num field is read and substituted for **:key** in the string specified in the SQL property. In turn, this parameter substituted string is used to set the RecordSource (or SQL) property of the detail data control. For example, if the master data control was positioned on a new record and the Cust-Num field contained a 5, the SubstitutedSQL property will read as:

```
SELECT * FROM Orders WHERE Cust-Num = 5 AND Date >= 1/7/96
```

The master data control substitutes the CParameter value of 1/7/96 for the CParameter named recent.

Any CParameter Objects that are in the CParameters Collection but are not used in the SQL string are ignored. If a CParameter Object that does not exist in the CParameters Collection is used in the SQL string, either the Error event is fired or a trappable error occurs. If the substitution is being performed because the DetailRefreshMode property is set to Automatic and the master data control is selecting a new record, then the Error event is fired. However, if the substitution is being performed because the RefreshDetail method is being invoked, then a trappable error occurs.

RefreshDetail Method

Applies To

Master/Detail

Purpose

Substitutes values for the parameters in the SQL statement. Resets the SQL statement in the DetailDataSource control. Refreshes the DetailDataSource control. This method is usually used in conjunction with setting the DetailRefreshMode to Manual.

Syntax

```
[Form1.]CMasterDetail1.RefreshDetail
```

Comments

When the method is called, the SubstitutedSQL property changes and the BeforeRefreshDetail event is fired.

Whenever the master data control repositions to a new record, or the RefreshDetail method is invoked, the Master/Detail control does three things:

1. Scans the string in the SQL property and substitutes any parameters specified with their current value. The SQL statement, with the parameter substitutions made, can be read using the SubstitutedSQL property.
2. The SQL statement, with the parameter substitutions made, is set into the RecordSource (or SQL) property of the data control specified by the DetailDataSource property.
3. The Refresh method of the data control specified by the DetailDataSource property is invoked.

If the DetailRefreshMode property is set to Manual, these steps are only performed when the RefreshDetail method is invoked. If the DetailDataSource property is set to an empty string, steps 2 and 3 above are not performed.

BeforeRefreshDetail Event

Applies To

Master/Detail

Purpose

This event is fired whenever the detail data control is about to be refreshed by the Master data control.

Syntax

```
CMasterDetail1.BeforeRefreshDetail(Cancel As Boolean)
```

Comments

Setting the Cancel parameter to True cancels the refresh operation; not setting the Cancel parameter to False allows the refresh operation to continue. The default value is False which means the refresh operation is performed.

Just before this event is fired, a new version of the SubstitutedSQL property is generated so the Visual Basic programmer will have an up-to-date version of the SQL statement while processing the event. After the event has fired, a new version of the SubstitutedSQL property is generated in case the SQL property or any of the parameters were changed during processing of this event.

Error Event

Applies To

Master/Detail

Purpose

This event is fired whenever an error occurs in the Master/Detail control when the master data control pointing to a new record and the DetailRefreshMode property is set to Automatic.

Syntax

```
CMasterDetaill_Error(Number As Integer, Description As String, Scode As Long, Source As String, HelpFile As String, HelpContext As Long, CancelDisplay As Boolean)
```

Constants

This event has the following parameters:

- Number As Integer:

Constant	Value	Meaning
mderrCantCoerce	1002	The data control cannot convert the CParameter's Value to a string.
mderrCantInvokeRefresh	1009	The detail data control has a Refresh method, but it cannot be invoked.
mderrCantSetSQL	1008	The detail data control has a RecordSource or SQL property but the property cannot be set.
mderrDataCopy	1003	The master/detail control cannot make a copy of the CParameter's Value.
mderrInvalidEscape	1001	The SQL property contains a single colon that is not followed by a key.
mderrNoDetailDispatch	1007	The master/detail control cannot find the IDispatch corresponding to the Detail Dispatch name.
mderrNoDetailRefresh	1004	The Refresh method does not exist in the DetailDataSource method list.
mderrParameterNotFound	1000	The SQL property contains a parameter name that does not exist.
mderrSetDetailSQL	1006	Neither the RecordSource nor SQL properties exist in the DetailDataSource property list.
<other>		An error generated by the DetailDataSource object when an attempt was made to set its RecordSource or SQL property, or to invoke its Refresh method.



Description As String - A short description of the error.



Scode As Long - The actual numeric value for the error.



Source as String - The name of the application in which the error occurred.



HelpFile As String - The filename of the help file if known. If the filename is not known or if a help file does not exist, the value is NULL.



HelpContext As Long - The context ID passed to get the help file.



CancelDisplay As Boolean - Setting the CancelDisplay parameter to True tells the control not to report the error. Use this value if you will handle the error in your program. Setting the CancelDisplay parameter to False tells the control to report the error. This is the default behavior.

CParameters Collection

This section describes the properties, methods, and events for the CParameters Collection.

Properties

Count*

Methods

Append*

Delete*

Item*

An asterisk in superscript font indicates a custom or modified property, method, or event.

Count Property

Applies To

CParameters Collection

Purpose

The Count property shows the number of objects in the CParameters Collection.

Syntax

```
[short] = [Form1.]CMasterDetail1.CParameters.Count
```

Data Type

Integer

Usage

Read-only at run time.

Append Method

Applies To

CParameters Collection

Purpose

Adds a new CParameter Object to the CParameters Collection.

Syntax

```
[Form1.]CMasterDetail1.CParameters.Append CParameter
```

Comments

This method expects a CParameter Object to be appended to the CParameters Collection.

You cannot add a CParameter Object with the same name as one already in the collection. You cannot add a CParameter Object named key.

Delete Method

Applies To

CParameters Collection

Purpose

Removes a CParameter Object from the collection.

Syntax

```
[Form1.]CMasterDetail1.CParameters.Delete CParameterName
```

Comments

This method expects a string representing the name of the CParameter Object to be deleted from the CParameters Collection.

Item Method

Applies To

CParameter Collection

Purpose

Returns the CParameter Object at the specified index from the collection.

Syntax

```
CMasterDetail1.CParameters.Item(n)  
CMasterDetail1.CParameters.Item(Pn)  
CMasterDetail1.CParameters(n)  
CMasterDetail1.CParameters(Pn)
```

Data Type

CParameter Object

Comments



Returns a CParameter Object



You can index an item either by offset , that is, 0 ... (n-1), or by name.



If you specify an invalid index, you will receive the following error message:

```
Run-time error 9:Subscript out of range
```

Specify an index that is in the valid range of indexes. The valid range is from 0 to the integer specified in the Count property.



The following example shows the code used to access the first item in a CParameters Collection:

```
Dim ParameterX as CParameter  
ParameterX = CMasterDetail1.CParameters.Item(0)  
Msgbox Parameters Name & ParameterX.Name -  
    & Parameters Value & ParameterX.Value
```

CParameter Object

This section describes the properties for the CParameter Object. It has no methods or events.

Properties

Name*

Value*

An asterisk in superscript font indicates a custom or modified property, method, or event.

Name Property

Applies To

CParameter Object

Purpose

The Name property is assigned the name of the CParameter Object.

Syntax

```
[Form1.]CParameter.Name [ = CParameterName ]
```

Data Type

String

Usage

Read/Write at design-time using the Custom Property Page. Read/Write at run time using standard object manipulation syntax.

Comments

Valid names must begin with a letter followed by any combination of alphanumeric characters. Names cannot begin with a number. Names are not case-sensitive but the case is preserved. No two parameter objects in the collection can have the same name.

The Name cannot be changed after the object has been appended to a collection.

Value Property

Applies To

CParameter Object

Purpose

The Value property of the CParameter Object is assigned the value of the CParameter Object.

Syntax

```
[Form1.]CParameter.Value [ = value ]
```

Data Type

Variant

Usage

Read/Write at design-time using the Custom Property Page. Read/Write at run time using standard object manipulation syntax.

Comments

The value of the CParameter Object is substituted where the CParameter Object's name appears in an SQL property in the Master/Detail control code. The CParameter names appear following a colon (:) in the SQL property. Do not have a space between the colon and the CParameter Object's name.

For example, if the CParameters Collection contained the following object:

```
CParameter.name = recent  
CParameter.value = #1/7/96#
```

You have the following SQL statement in your code:

```
SELECT * FROM Orders WHERE Cust-Num = 5 AND Date >= :recent
```

If the master data control was positioned on customer number 5, the SQL statement will look like the following:

```
SELECT * FROM Orders WHERE Cust-Num = 5 AND Date >= 1/7/96
```

The master data control substitutes the CParameter value of 1/7/96 for the CParameter name of recent.

You cannot have more than one level of a CParameters Object in the Value property. That is, you cannot nest aliases more than one level. The Master/Detail control will only expand the first level of an alias. For example, you already defined a CParameter called recent to be 1/7/96. The following code will not work:

```
Dim CParam as new CParameter  
CParam.name = where_clause  
CParam.value = Cust-Num = 5 AND Date >= :recent  
CMasterDetail1.SQL = SELECT * FROM Orders WHERE :where_clause
```

In this example, the Master/Detail control cannot substitute the value of 1/7/96 for the CParameter called recent. .

Any CParameter Objects that are in the CParameters Collection but are not used in the SQL string are ignored. If a CParameter Object that does not exist in the CParameters Collection is used in the SQL string, either the Error event is fired or a trappable error occurs. If the substitution is being performed because the DetailRefreshMode property is set to Automatic and the master data control is selecting a new record, then the Error event is fired. However, if the substitution is being performed because the RefreshDetail method is being invoked, then a trappable error occurs.

General Tab - Crescent Master/Detail Control

Use this tab to define the following properties for the Master/Detail control:

SQL	An SQL statement that can have parameters that are replaced with values in the SubstitutedSQL property. See the <u>SQL</u> property of the Master/Detail control for more information.
DetailDataSource	The name of the data control to act as the detail data control which accesses the database to retrieve the data specified in the SubstitutedSQL property. See the <u>DetailDataSource</u> property of the Master/Detail control for more information.
DetailRefreshMode	The mode for data to be refreshed, either automatically or manually. See the <u>DetailRefreshMode</u> property of the Master/Detail control for more information.
Enabled	Allows the Master/Detail control to work. If it is not enabled, the control cannot perform any actions. See the <u>Enabled</u> property of the Master/Detail control for more information.

OK

Apply changes to the CParameters Collection for the Master/Detail control and exit the property sheet.

Cancel

Exit the property sheet without saving unapplied changes to the CParameters Collection of the Master/Detail control.

Apply

Apply changes to the CParameters Collection.

CParameters Tab - Crescent Master/Detail Control

The text boxes in this frame allow you to add, select, and/or delete a CParameter Object from the CParameters Collection associated with the current Master/Detail control.

CParameters

Lists the CParameter Objects that are currently defined in the CParameters Collection associated with the current Master/Detail control. See the Name property of the Master/Detail control for more information on setting the name. See the Value property of the Master/Detail control for more information on setting the value.

Add CParameter

The text boxes in this frame allow you to set the name and value of a CParameter Object. Click the Add button to add the new CParameter Object to the CParameters Collection.

Delete selected CParameters

Deletes the currently highlighted CParameter Objects listed in the **CParameters** list box.

OK

Apply changes to the CParameters Collection for the Custom Data control and exit the property sheet.

Cancel

Exit the property sheet without saving unapplied changes to the CParameters Collection of the Custom Data control.

Apply

Apply changes to the CParameters Collection.

