

# Defining Extended Rules

Extended rules apply to GUI controls such as push buttons, radio buttons or check boxes. These GUI controls are defined using a resource editor.

This chapter covers the following topics:

- Using a Resource Editor
- Opening a DLL
- Opening a Dialog
- Closing a Dialog
- Defining the Dialog Properties
- Defining the Control Properties
- Defining the Font
- Using Styles
- Saving the DDT File
- Building the BDD File
- Detecting the Screen on which the Extended Rules are to be Applied (Single Screen)

See the *Extended Rules Reference* for a detailed description of each rule.

See the *First Steps* for a step-by-step description of how to define extended rules.

See also: *Using Basic Rules Together with Extended Rules*.

---

## Using a Resource Editor

You can use any resource editor that can create standard Windows resource DLLs (dynamic link libraries), for example Microsoft Visual Studio. Other editors, such as Visual Basic 6.0, use a different format for resource DLLs and can therefore not be used with Entire Screen Builder.

The following topics are covered below:

- Creating a DLL
- Reserved Control IDs
- Reserved Dialog Names

## Creating a DLL

In the resource editor, a DLL has to be created containing a dialog for each screen that is to be presented in an extended dialog. You can only use the controls that are supported by Entire Screen Builder. See the *Extended Rules Reference* for further information.

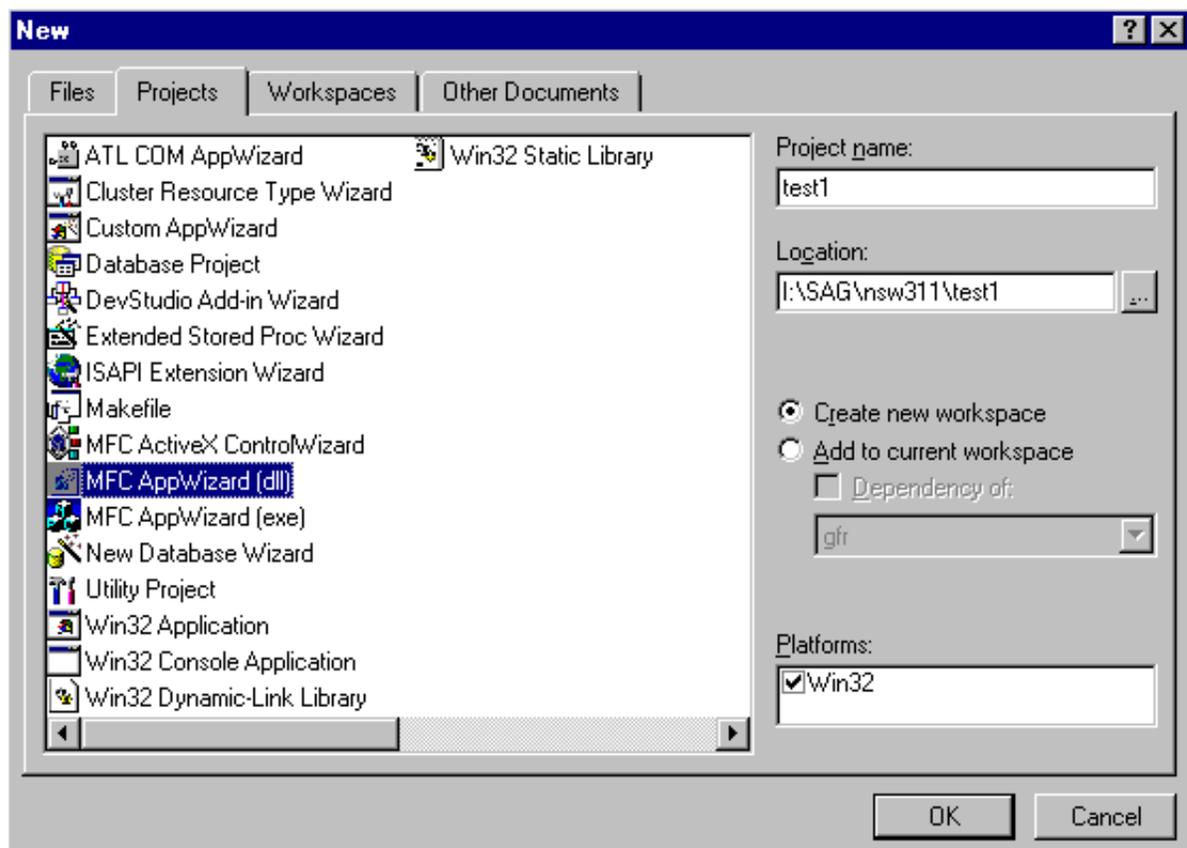
The following description applies when you use Microsoft Visual C++ Version 6, which is part of Microsoft Visual Studio, to create the DLL.

### ▶ To create a DLL containing a dialog using Microsoft Visual C++

1. From the **File** menu, choose **New**.

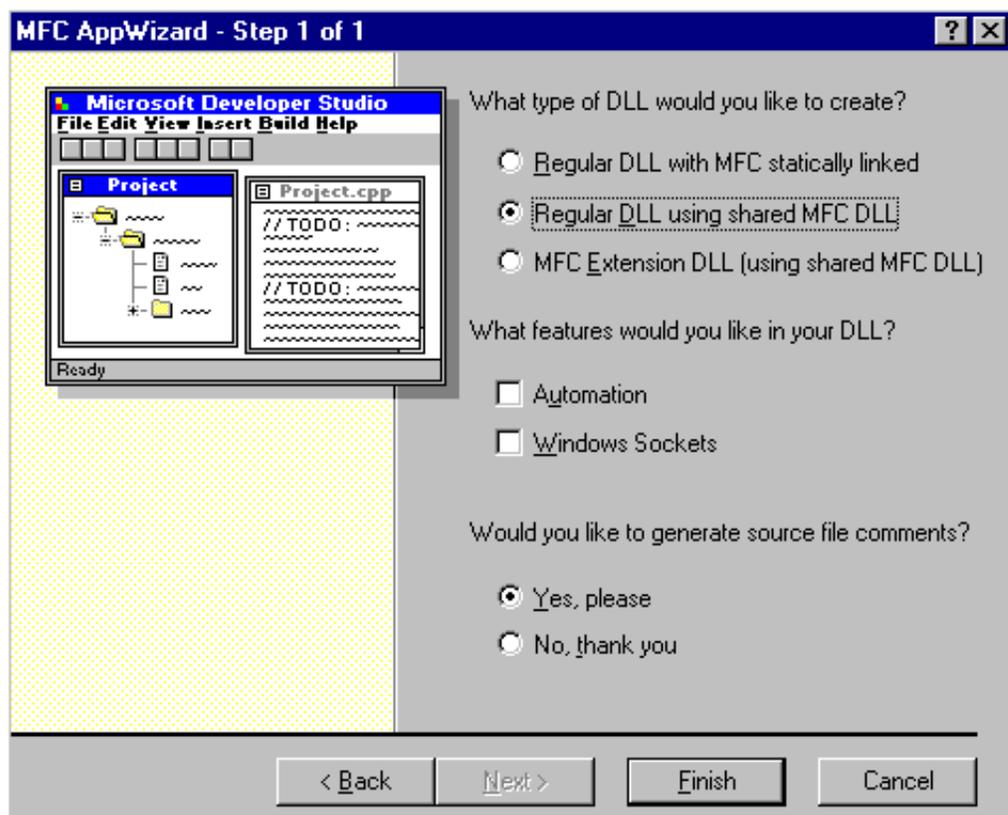
The New dialog box appears.

2. Make sure that the Projects page is shown.
3. Select **MFC AppWizard (dll)**.
4. In the **Project name** text box, specify the desired project name.



5. Choose the **OK** button.

The following dialog box appears.



Leave the default options as shown in the above dialog box.

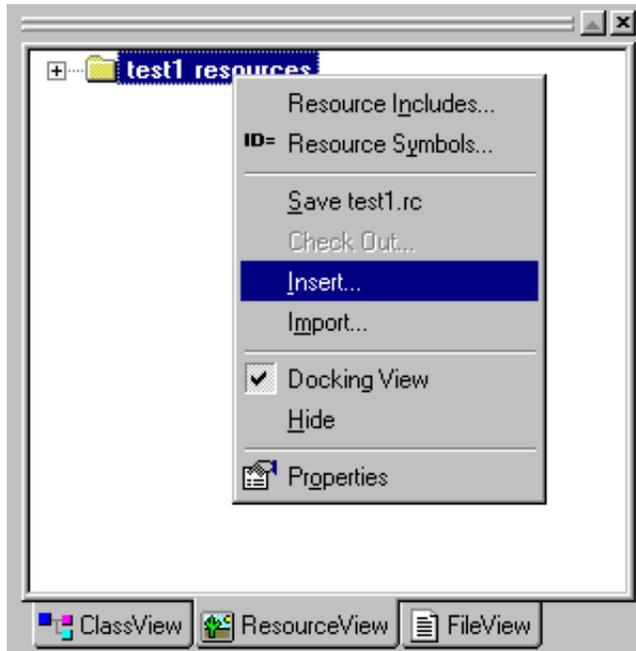
6. Choose the **Finish** button.

The New Project Information dialog box appears.

7. Choose the **OK** button.

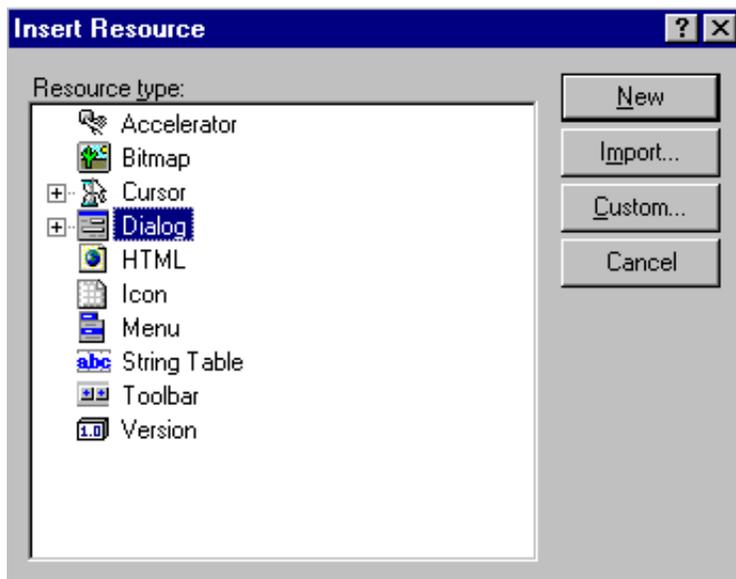
The DLL is now created. The name you have defined for the project is shown in the workspace window.

8. Make sure that the ResourceView page is shown.
9. Select the resource name and invoke the context menu.
10. From the context menu, choose **Insert**.



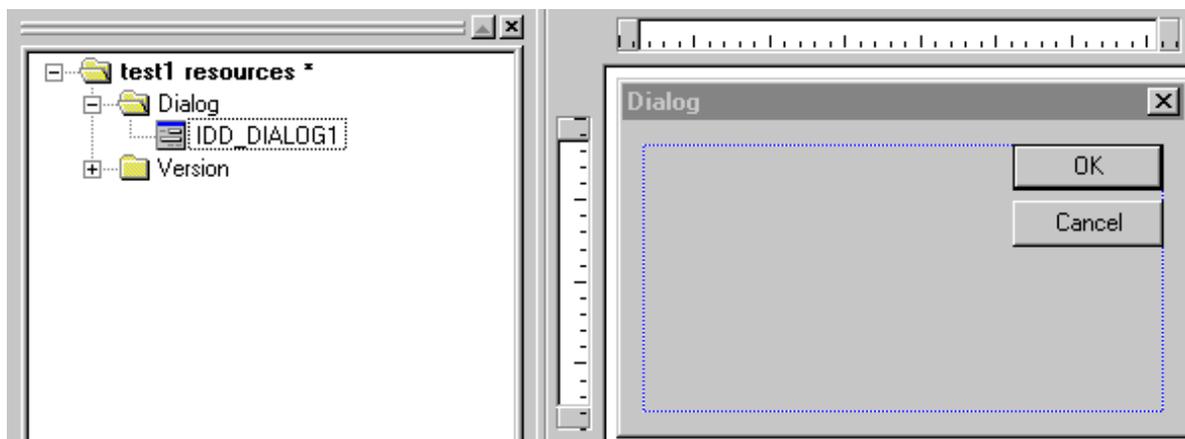
The Insert Resource dialog box appears.

11. Select the resource type **Dialog**.



12. Choose the **New** button.

A new dialog is now shown. Its name is shown in the tree.

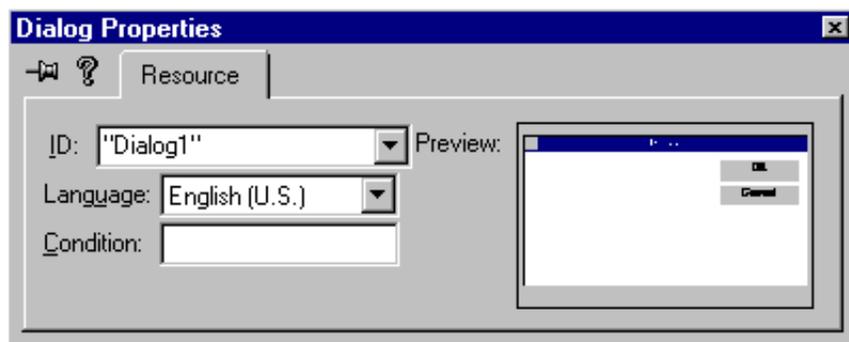
**Important:**

Change the ID of the new dialog to a string value as described below.

13. In the tree, select the dialog name and invoke a context menu.
14. From the context menu, choose **Properties**.

The Dialog Properties dialog box appears.

15. In the **ID** text box, specify a new ID enclosed in quotation marks.



16. Press ENTER to close the dialog box.

You can now design the dialog as follows:

- either design a new dialog, or
- load an existing screen file that has been saved using the Entire Screen Builder SDK into the dialog (see the description below).

When the dialog design has been completed, you have to build the DLL.

17. From the **Build** menu, choose **Build name.dll**.

Or:

Press F7.

When the DLL has been built successfully, you can open it using the Entire Screen Builder SDK. See *Opening a DLL* for further information.

▶ **To load an Entire Screen Builder screen file into the dialog**

1. From the **File** menu, choose **Open**.

The Open dialog box appears.

2. Select the desired screen file.
3. Choose the **Open** button.

Information similar to the following is now shown:



4. Open the dialog (for example, by double-clicking its name).

The content of the screen file is now shown in a dialog.

5. Make sure that this dialog is selected.
6. From the **Edit** menu, choose **Select All**.

Or:

Press CTRL+A.

7. From the **Edit** menu, choose **Copy**.

Or:

Press CTRL+C.

8. In the tree view, open the dialog you have created in the previous exercise.
9. If required, resize the dialog.
10. From the **Edit** menu, choose **Paste** to copy the contents of the screen file to the dialog.

Or:

Press CTRL+V.

## Reserved Control IDs

The following control IDs are reserved for Entire Screen Builder. Therefore, all controls that you create in the resource editor (for example, push buttons, group boxes or list boxes) must to have a control ID bigger than 0x5000 (20480).

Control ID	Description
0x0000 - 0x0009	Reserved for Windows
0x000A - 0x0100	Undefined
0x0101 - 0x2480	Screen ID (see <i>Field Identification</i> for further information)
0x2481 - 0x24FD	Undefined
0x24FE	Message line (basic rules)
0x24FF	ENTER key
0x2500 - 0x2563	Host keys
0x2564 - 0x25C7	Images
0x25C8	Group box
0x25C9 - 0x25CF	3270 special keys
0x25D0 - 0x26D0	Popup menu
0x26D1 - 0x2AF7	Undefined
0x2AF8	Message line (extended rules)
0x2AF9 - 0x4999	Undefined

## Reserved Dialog Names

The following dialog names are reserved for Entire Screen Builder. They are used by the Frames rule.

- "TOP"
- "LEFT"
- "RIGHT"
- "BOTTOM"

### Important:

The quotation marks are part of the frame name.

## Opening a DLL

A DLL contains the dialogs that have been created using a resource editor.

### **Important:**

You can only open a DLL when you are working in application or map scope.

If you want to open another DLL, you must first close an already open dialog.

### **To open a DLL**

1. From the **File** menu, choose **Open DLL**.

Or:

Choose the following toolbar button:



The Open DLL dialog box appears.

2. Select the desired DLL.
3. Choose the **Open** button.

A dialog box appears providing for selection all dialogs that have been defined in the DLL (see below).

## Opening a Dialog

After the DLL has been opened, you can open a dialog contained in it. To do so, you can either use a menu command or the drop-down list box in the toolbar. Only one dialog can be shown at one time. If you want to display another dialog, you must first close an already open dialog.

### **To open a dialog using a menu command**

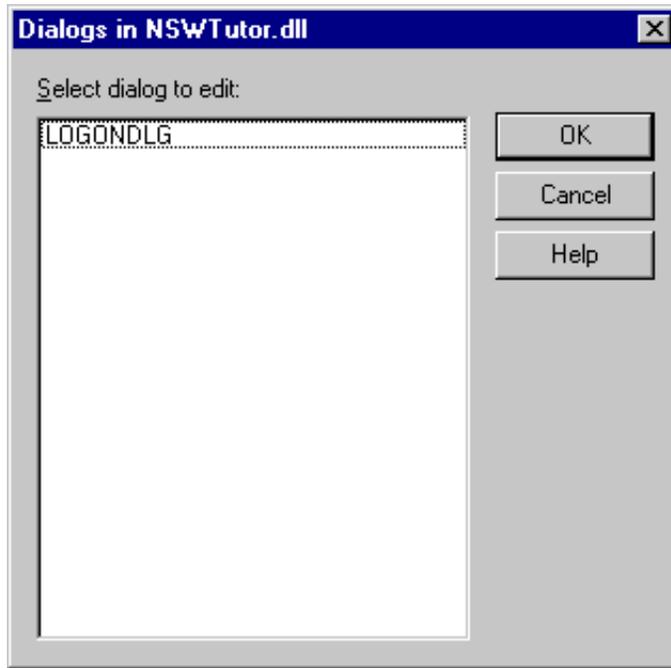
1. From the **Extended** menu, choose **Open Dialog**.

Or:

Choose the following toolbar button:



A dialog box appears listing all dialogs that have been defined in the DLL.



2. Select the desired dialog.
3. Choose the **OK** button.

▶ **To open a dialog using the drop-down list box in the toolbar**

- From the drop-down list box in the toolbar, select the desired dialog.



## Closing a Dialog

You can either close a dialog with the standard button in the upper right corner of the dialog or with a command.

▶ **To close a dialog using a command**

- From the **Extended** menu, choose **Close**.

Or:

Choose **Close** from the context menu that is invoked by pressing the right mouse button.

## Defining the Dialog Properties

You can define a background color or image for the current dialog. When a dialog style has been defined for the background color, you also define this style.

### Note:

You can also use the font toolbar to define a background color or style.

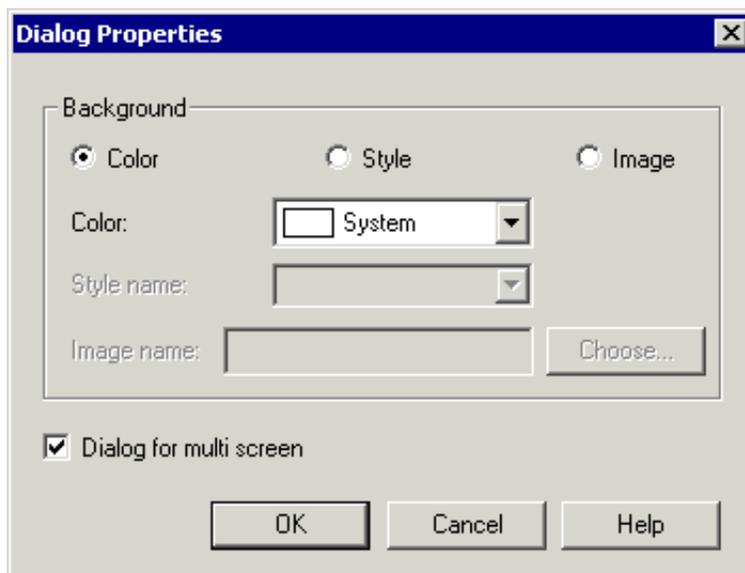
### ► To define the dialog properties

1. Open the dialog for which you want to define the dialog properties.
2. From the **Extended** menu, choose **Dialog Properties**.

Or:

Click the right mouse button in an area of the dialog that does not contain a control and choose **Dialog Properties** from the resulting context menu.

The Dialog Properties dialog box appears.



3. In the Background group box, select one of the following option buttons:

### Color

When this option button is selected, the **Color** drop-down list box is available and you can select the desired color from this drop-down list box.

When you select **System** from this drop-down list box, the background color as defined under Windows will be used.

When you select **Custom...** from this drop-down list box, the Color dialog box appears and you can select additional colors or define custom colors. The currently defined color is shown next to **Custom...**

### Style

When this option button is selected, the **Style name** drop-down list box is available. When a style has been defined, you can select it from this drop-down list box. See *Defining a Dialog Style*.

### Image

When this option button is selected, the **Image name** text box is available and you can specify the path to an image in the rules repository (relative to the root folder of the rules repository). You can also choose the **Choose** button to select the file from the "Open" dialog box.

The background image can be any GIF, JPG or BMP file. See *General Information on Image Files*.

4. If you want to use the multi screen feature, select the following check box:

#### **Dialog for multi screen**

When this check box is selected, an additional page, *Screen Number*, is available in the control properties. See also: *Using the Multi Screen Feature*.

**Note:**

This check box is disabled when a tab control has been defined in the extended dialog.

5. Choose the **OK** button.

## Defining the Control Properties

The context menu that appears when you click the right mouse button shows all commands that apply to the selected control. In order to apply extended rules to the selected control, you have to choose the command **Control Properties**. This command is available in the context menu and in the **Extended** menu.

### To define the control properties

1. Select the control to which you want to apply extended rules.
2. From the **Extended** menu, choose **Control Properties**.

Or:

Invoke the context menu and choose **Control Properties**.

Or:

Double-click the control.

Depending on the control that is currently selected, different pages are available in the resulting dialog box. The pages that are used by more than one control are explained in the following topics below:

- Field Identification
- Display Styles
- Action
- Mouse
- Screen Number

For information on the rule-specific pages not covered below, see the descriptions of the corresponding rules in the *Extended Rules Reference*.

3. Specify all required information on the different pages.
4. Choose the **OK** button.

## Field Identification

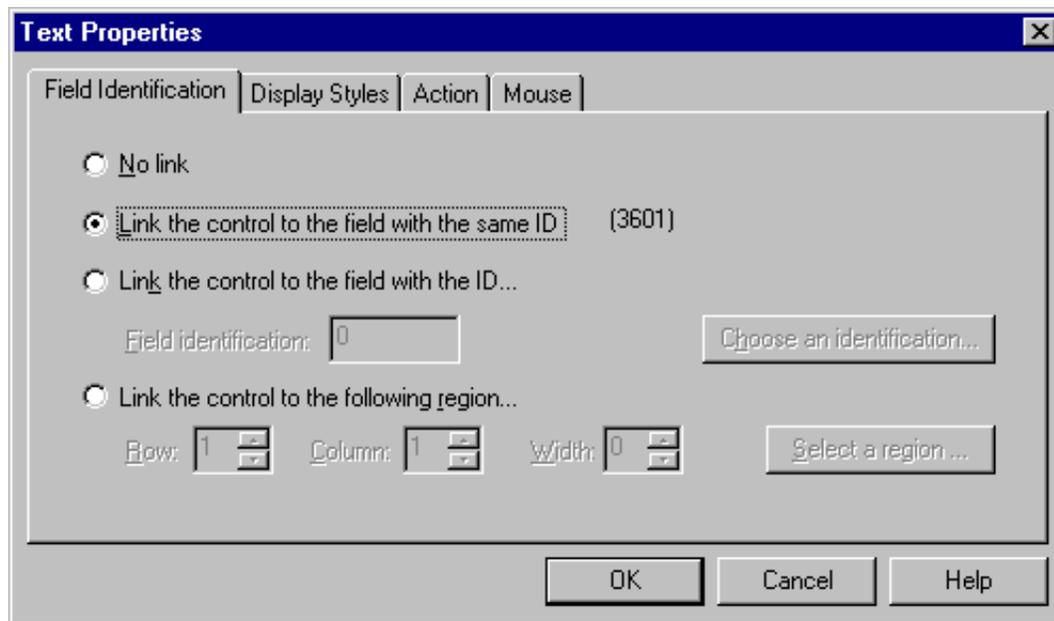
In order to transfer the values that a user enters in an extended dialog to the character screen and vice versa, you have to define the corresponding field for a number of dialog controls. Each field has a unique ID based on its position in the screen. The ID is calculated as follows:

$$\text{ID} = (\text{row} * 256) + \text{column}$$

There are two different ways of creating a dialog:

- You save a character screen as a screen file and then modify it using the resource editor. In this case, it is not required that you define the corresponding fields. By saving the screen, all screen IDs (0x0101-0x2480) are automatically defined. However, if you create new dialog controls, you have to define all corresponding fields manually. By default, the option button **Link the control to the field with the same ID** (see below) is enabled for controls such as edit boxes. For a static text control, the option button **No link** is enabled by default.
- You start from scratch, i.e. you create a new dialog for a screen using the resource editor. In this case, you have to define all corresponding fields manually.

When the definition of the corresponding field is required, the Field Identification page is available.



### No link

The content of the control is not changed.

For example, if you have defined a static text control with the label "None:", this label is always shown.

### Link the control to the field with the same ID

The content of the control is substituted with the content of a field that has the same ID.

For example, you have defined a static text control with ID 257 and the label "None:". When the following has been defined for the field, the label "Same:" is then shown on the control:

```
ID = 257, row=1, column=1, text="Same:"
```

### Link the control to the field with the ID

The content of the control is substituted with the content of a field that has another ID. In this case, you can either type the field ID in the text box or choose the **Choose an identification** button to select the desired field in the scope window.

For example, you have defined a static text control with ID 257 and the label "None:". When the following has been defined for the field, the label "Other:" is then shown on the control:

```
ID = 2580, row=10, column=20, text="Other:"
```

### Link the control to the following region

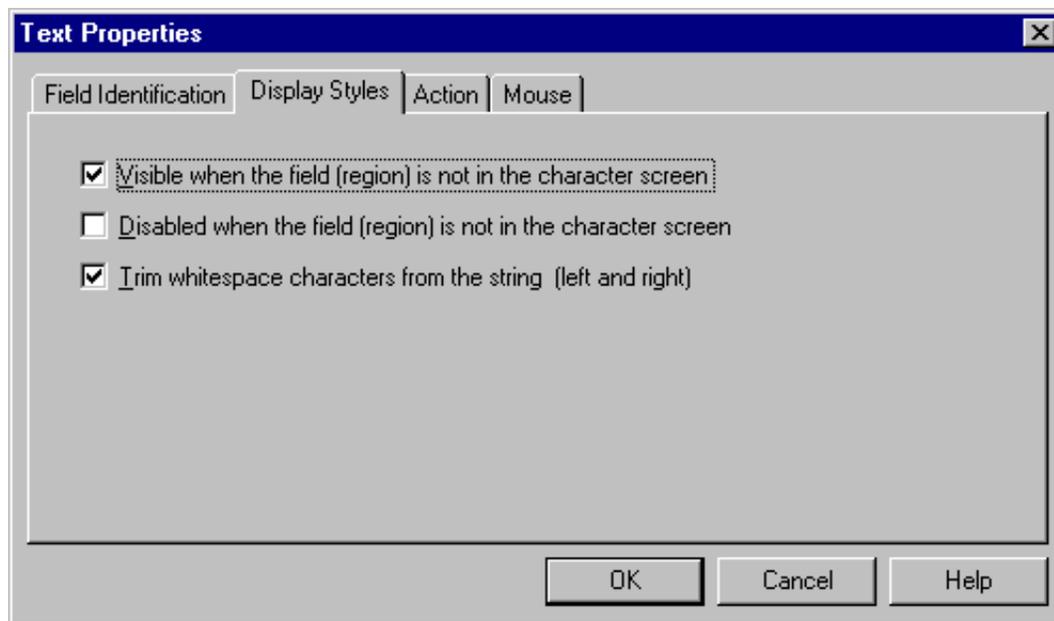
The content of the control is substituted with the content of the defined region. In this case, you can either specify the coordinates of the region using the spin boxes (row, column and width) or choose the **Select a region** button to define the desired region in the scope window.

For example, you have defined a static text control with ID 257 and the label "None:". When the following has been defined for the region, the label "Region" is then shown on the control:

```
row=20, column=5, width=10, text="Region"
```

## Display Styles

When the definition of the corresponding field is required, the Display Styles page is also available (in addition to the Field Identification page). The options on the Display Styles page are only available when an option other than **No link** is selected on the Field Identification page.



### **Visible when the field (region) is not in the character screen**

When this check box is selected, the control will be shown even if the corresponding field or region is not available in the character screen.

### **Disabled when the field (region) is not in the character screen**

This check box is only available when the check box **Visible when the field (region) is not in the character screen** has also been selected.

When this check box is selected, the control will be disabled if the corresponding field or region is not available in the character screen. Otherwise, it is not shown at all.

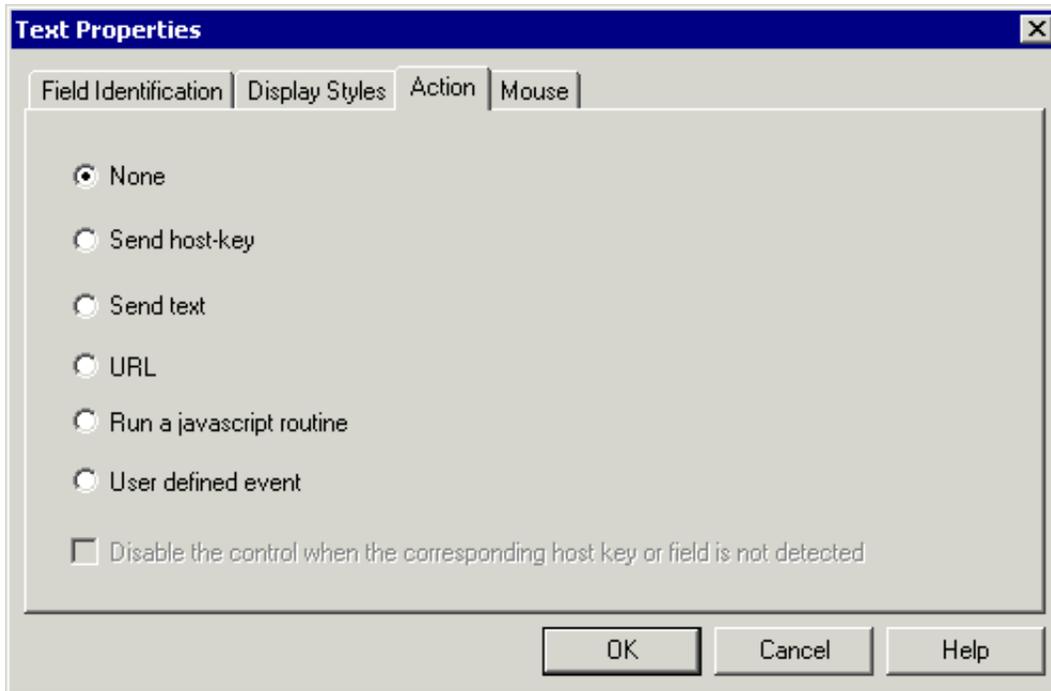
When the corresponding field for a control is a protected field (this may happen when the application changes an unprotected field to a protected field), the control is automatically disabled. The setting of this option is ignored in this case.

### **Trim whitespace characters from the string (left and right)**

This check box is only available for static text. When this check box is selected, any blank characters before and after the string which makes up the static text are removed.

## Action

The Action page is available for push button and static text controls. When you define an action, it is recommended that you enable the **Enable the single click** check box on the Mouse page.



### None

When this option button is selected, no action will be taken when the user chooses the control. Use this option for a push button control, if you want to display an image only.

### Send host key

When this option button is selected, additional options are shown:



You can send the defined function key to the application.

- **Host key**

From the drop-down list box, select the host key you want to associate with the control.

This drop-down list box also contains BS2000 keys.

- **Field identification**

Specify the identification of the input field which receives the focus when the user chooses the control. To do so, make sure that the corresponding screen file is shown in the scope window and choose the **Choose** button. You can now click the desired field in the scope window.

## Send text

When this option button is selected, additional options are shown:

You can display text in an input field and then send the defined function key to the application.

- **Host key**

From the drop-down list box, select the host key you want to associate with the control.

This drop-down list box also contains BS2000 keys.

- **Field identification**

Specify the identification of the input field which receives the focus when the user chooses the control. To do so, make sure that the corresponding screen file is shown in the scope window, choose the **Choose** button and click the desired field in the scope window.

- **Text**

In the text box, specify the text that is to be shown in the input field.

- **Don't clear field**

When this check box is not selected, the content of the input field is cleared before new information is written to it. However, when this check box is selected, the new information just overwrites (part of) the information in this input field.

For example: the input field contains the text "ABCDEF" and the new text "?" is sent to this input field. When this check box is not selected, "?" is then shown in the input field. When this check box is selected, "?BCDEF" is shown in the input field.

## URL

When this option button is selected, an additional text box is shown:

You can open a new browser window.

- **Address**

Specify the URL of the web site that is to be opened in a new browser window when the user chooses the control. For example:

*<http://www.softwareag.com/corporat/default.htm>*

### Run a JavaScript routine

When this option button is selected, an additional text box is shown:



The drop-down list box provides for selection the script files stored in the *Production* folder. Select the script file that is to be executed when the user chooses the control.

See also: *General Information on Script Files* in the *Script Files* documentation.

### User defined event

When this option button is selected, an additional text box is shown:



Specify the event that is to be fired when the user chooses the control:

- **Text is not specified**

When only the option button has been selected and a text has not been defined in the text box, the event `UserDefinedEvent` will be fired into the HTML page. This enables you to have local processing in the HTML page on the client workstation using a scripting language supported by the browser (for example, VBScript).

- **Text is specified**

Optional. You can specify a text for the user-defined event. In this case, the event `UserDefinedEvent(szText)`, including the text that you have specified, will be fired into the HTML page. This is helpful when you have specified a user-defined event for more than one control in the dialog. The event will then be identified by its name.

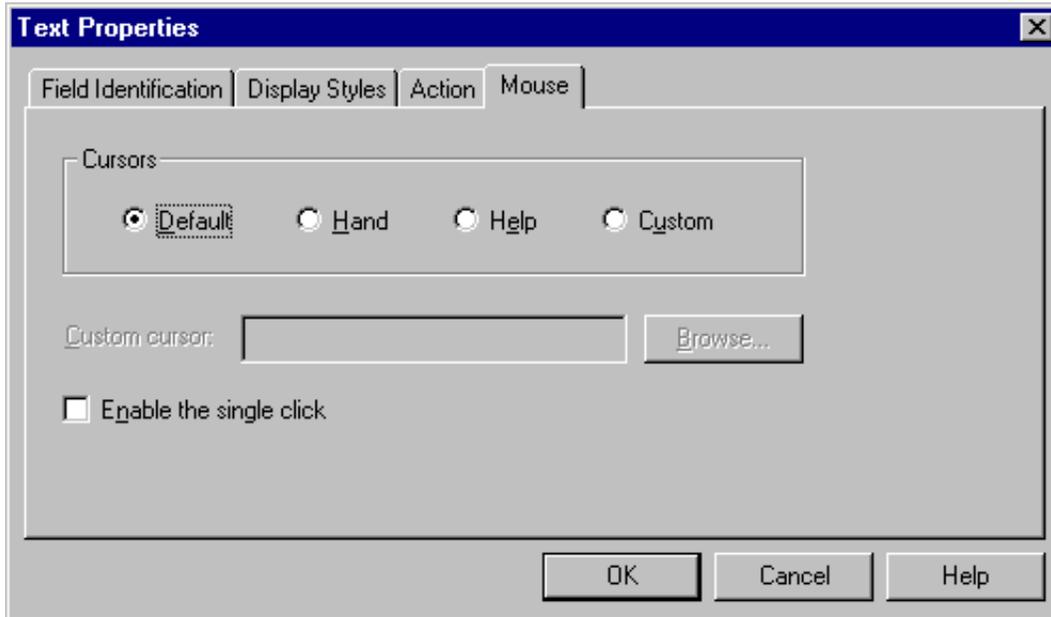
See also: *Events* in the *Web Viewer Client Functions (API)* section of the *User Exits* documentation.

### Disable the control when the corresponding host key or field is not detected

This check box is only available for push buttons. It is available when an option other than **None** has been selected. When this check box is selected, the control is disabled if the corresponding host key or field is not available in the character screen.

## Mouse

The Mouse page is available for most resource editor controls that are supported by Entire Screen Builder.



### Cursors

You can define the type of cursor that is to be used. Select one of the following option buttons:

- **Default**  
The cursor as defined under Windows.
- **Hand**  
Hand-shape cursor.
- **Help**  
Cursor in the shape of a question mark.
- **Custom**  
You can define your own cursor (see below).

### Custom cursor

When the **Custom** option button has been selected, you must specify a path and file name in this text box. This can be a file with the extension *cur* or *ani*.

Using the **Browse** button, you can also choose the file from a dialog box.

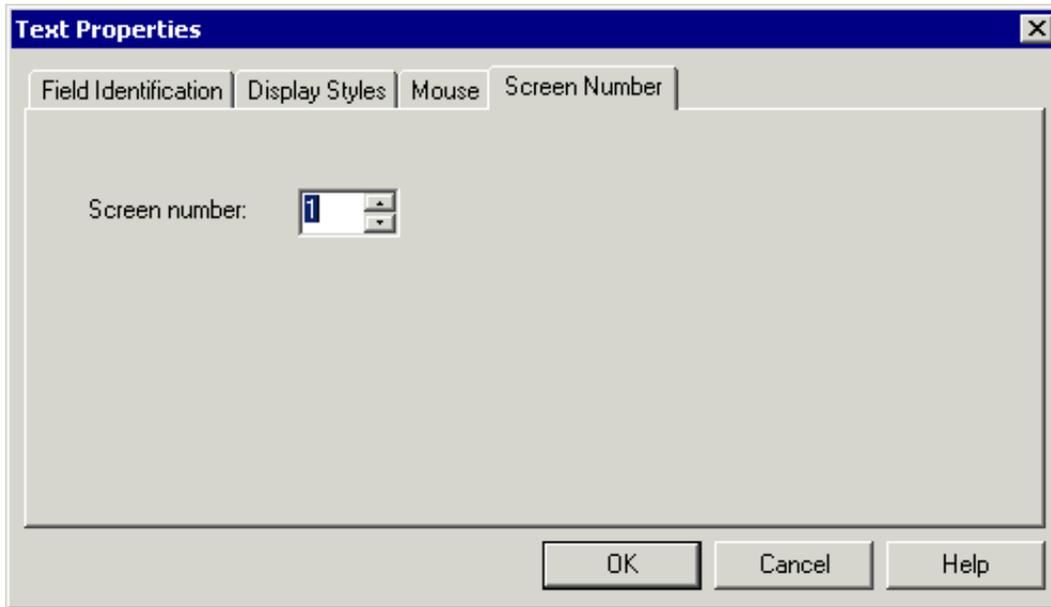
### Enable the single click

When this check box is selected, a single click on a control has the same effect as a double-click. If an action has been defined for the control (**Send host key**, **Send text**, **URL** or **Run a JavaScript routine**), this action is executed after a single click. If no action has been defined, the corresponding host key that has been defined with the basic rule Control Attributes is sent to the host.

## Screen Number

The Screen Number page is only available when the option **Dialog for multi screen** has been activated in the dialog properties (see *Defining the Dialog Properties*).

See also: *Using the Multi Screen Feature*.



### Screen number

With the multi screen feature, fields from different terminal screens can be shown in one dialog.

You have to specify the number of the terminal screen which contains the corresponding field for the current control.

## Defining the Font

You can define another font or character set for a control.

**Note:**

You can also use the font toolbar to define a font.

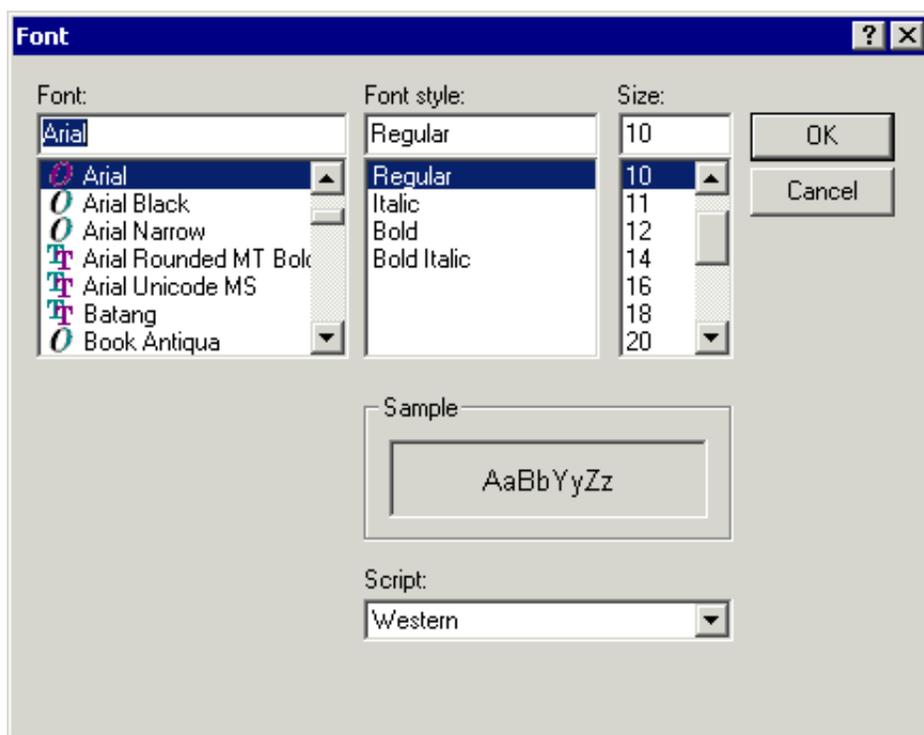
▶ **To define another font**

1. Select the control for which you want to define another font.
2. From the **Extended** menu, choose **Font**.

Or:

Invoke the context menu and choose **Font**.

The Font dialog box appears.



3. Select the desired font, font style and size.
4. If you want to use another character set, select it from the **Script** drop-down list box.
5. Choose the **OK** button.

## Using Styles

You can define styles for frequently-used font and color settings. You can then apply these styles to your controls and dialogs.

The following topics are covered below:

- Defining a Control Style
- Defining a Dialog Style
- Applying a Style to a Control or Dialog

### Defining a Control Style

Control styles can be applied to the following controls:

Check boxes

Edit boxes (one input field and several input fields)

Group boxes

List box controls (list boxes and combo boxes)

List view controls

Radio buttons

Static text

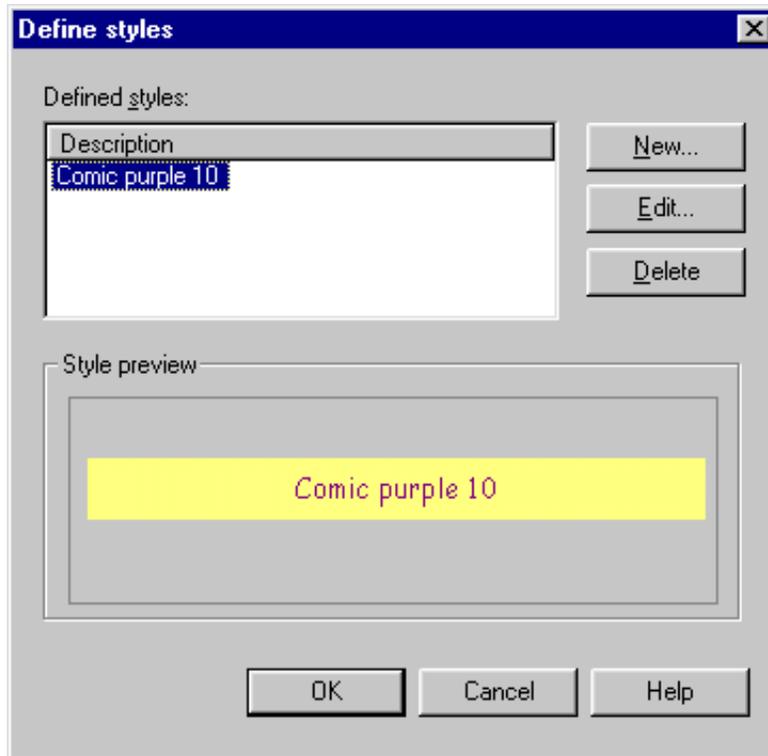
You can only define control styles when you are working in application or map scope. A dialog must not be open.

Using the Control Attributes rule, you can define that the size of a control is automatically increased or reduced according to the font that has been defined for the text in this control.

► To define a new control style

1. From the **Extended** menu, choose **Define Styles > Control Styles**.

The Define Styles dialog box appears.



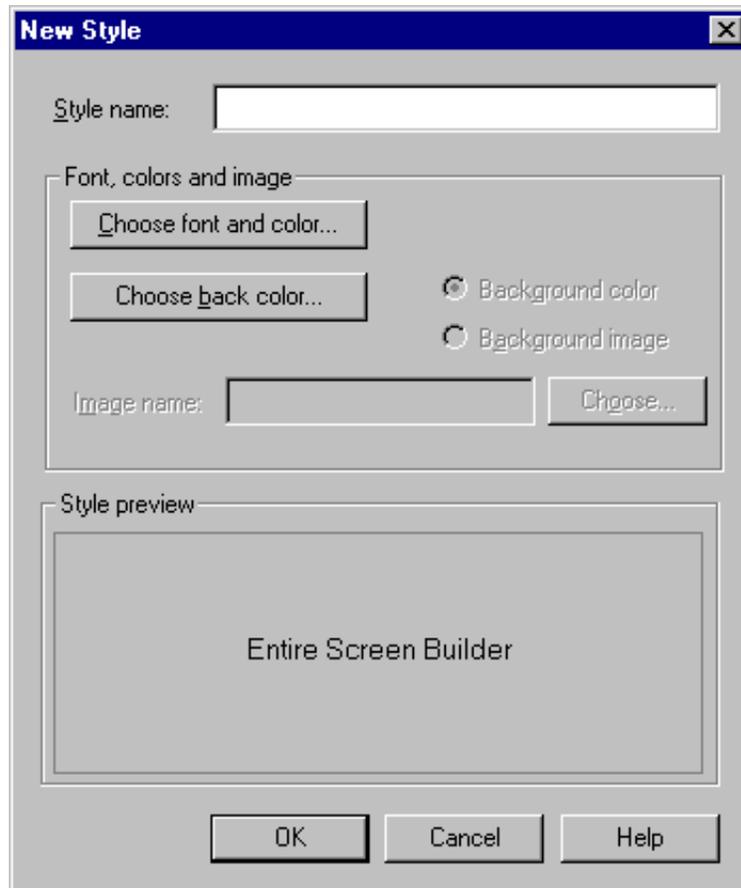
When control styles have already been defined, the names you defined for them are shown in this dialog box. When you select a control style, the preview section shows how this style looks like.

The following command buttons are available:

<b>New</b>	Define a new style (see below).
<b>Edit</b>	Modify the selected style.
<b>Delete</b>	Delete the selected style.

2. To define a new control style, choose the **New** button.

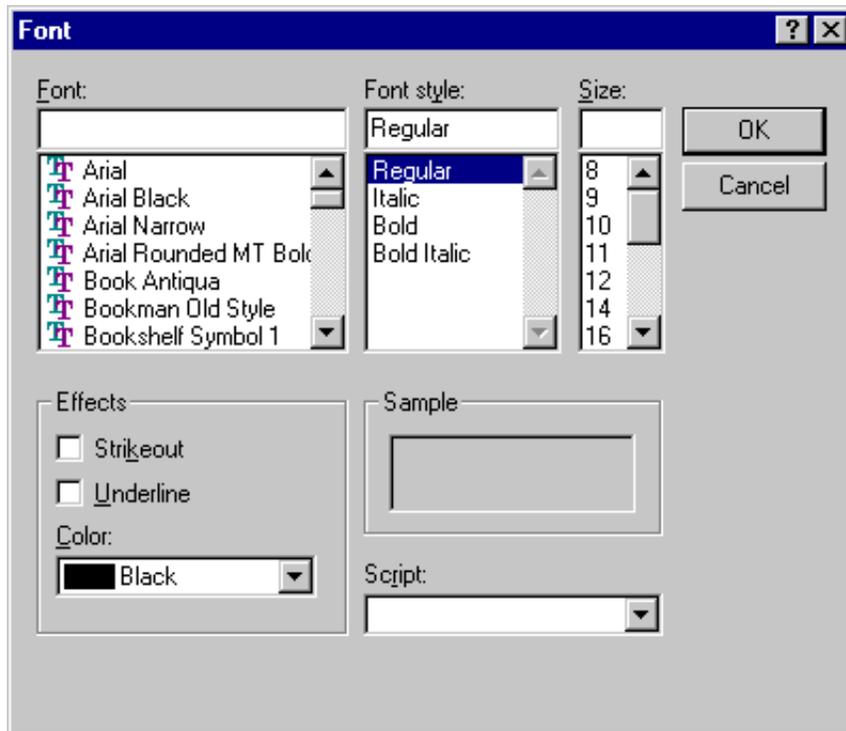
The New Style dialog box appears.

**Note:**

The option buttons for background color and background image are not available for control styles.

3. In the **Style name** text box, enter a name for the style.
4. To define the font and/or color, choose the corresponding command button.

The Font dialog box appears.



- Specify all required options and choose the **OK** button.

The new font and color are now shown in the preview section of the New Style dialog box.

- To define the background color, choose the corresponding command button.

The Color dialog box appears.



7. Define the desired color and choose the **OK** button.

The new background color is now shown in the preview section of the New Style dialog box.

8. Choose the **OK** button to close the New Style dialog box.

## Defining a Dialog Style

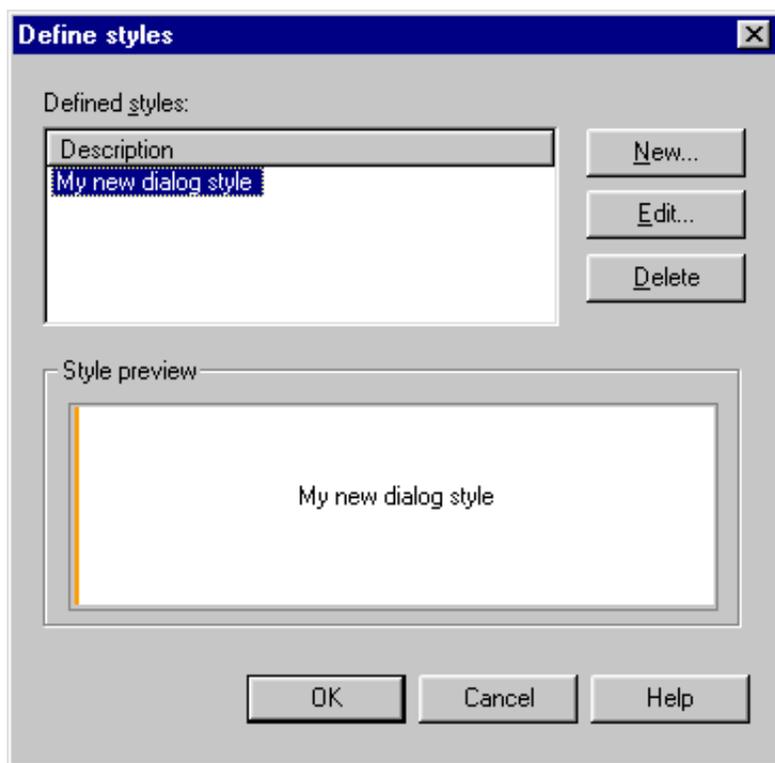
Dialog styles can only be applied to dialogs. You can define a background color or a background image.

You can only define dialog styles when you are working in application or map scope. A dialog must not be open.

### ▶ To define a new dialog style

1. From the **Extended** menu, choose **Define Styles > Dialog Styles**.

The Define Styles dialog box appears.



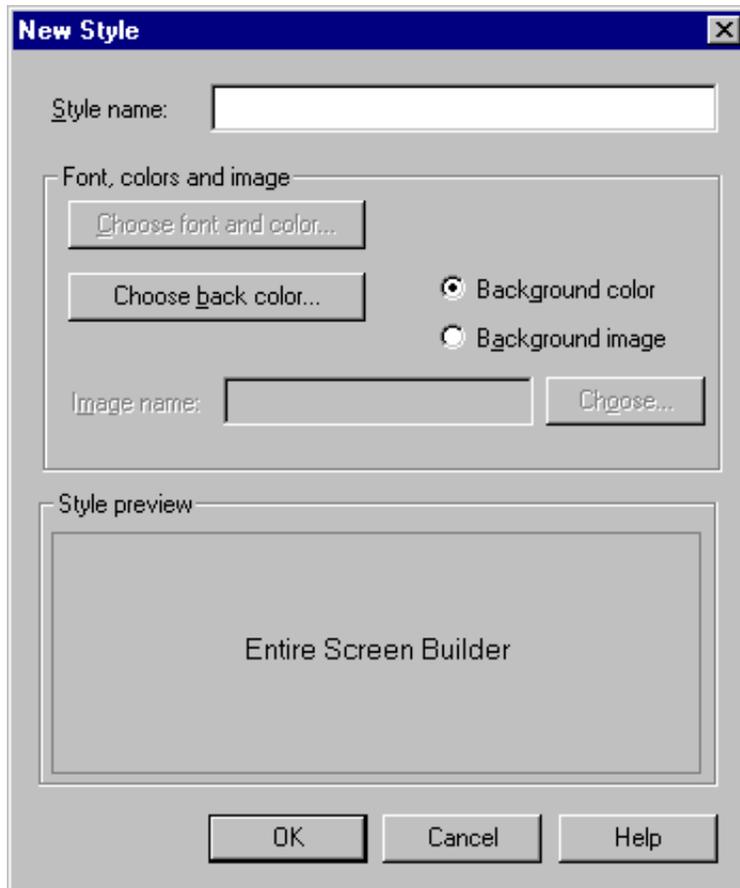
When dialog styles have already been defined, the names you defined for them are shown in this dialog box. When you select a dialog style, the preview section shows how this style looks like.

The following command buttons are available:

<b>New</b>	Define a new style (see below).
<b>Edit</b>	Modify the selected style.
<b>Delete</b>	Delete the selected style.

2. To define a new dialog style, choose the **New** button.

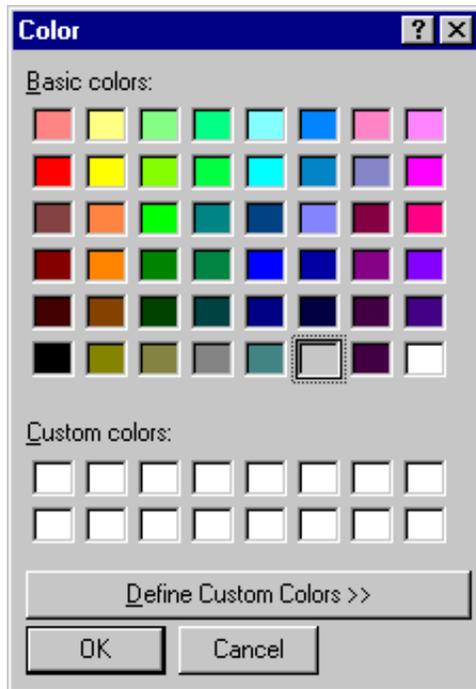
The New Style dialog box appears.

**Note:**

The command button for choosing a font and color is not available for dialog styles.

3. In the **Style name** text box, enter a name for the style.
4. To define a background color, make sure that the **Background color** option button is selected and choose the **Choose back color** command button.

The Color dialog box appears and you can define the desired color.



Or:

To define a background image, make sure that the **Background image** option button is selected. In the **Image name** text box, specify the path to the image in the rules repository (relative to the root folder of the rules repository). You can also choose the **Choose** button to select the file from the Open dialog box. The background image can be any GIF, JPG or BMP file. See *General Information on Image Files*.

When you have defined a new background color or background image, it is shown in the preview section of the New Style dialog box.

5. Choose the **OK** button to close the New Style dialog box.

## Applying a Style to a Control or Dialog

When you have defined a control or dialog style, you can apply it to a control or dialog.

### Note:

A dialog style can also be defined in the dialog properties. See *Defining the Dialog Properties*.

### ▶ To apply a style to a control or dialog

1. In the dialog, select the control(s) to which you want to apply a style.

If you do not select a control, a dialog style can be applied.

2. Invoke the context menu and choose **Apply Style > *stylename***.

You can also apply a style named "Default". In this case, the control or dialog is reset to the original settings as defined in the resource editor.

Or:

Choose the desired style from the font toolbar.

## Saving the DDT File

DDT stands for Dialog Definition Table.

The DDT contains all information that has been defined using the SDK (such as font and color, list box properties, correspondence between character screen fields and dialog controls).

This file is always created in the same folder as the DLL containing the dialogs. It receives the same name as the DLL. If you copy the DLL to another location, you must also copy the DDT to this location.

### ▶ To save the DDT file

- From the **File** menu, choose **Save DDT File**.

Or:

Choose **Save DDT** from the context menu that is invoked by pressing the right mouse button.

## Building the BDD File

BDD stands for Binary Dialog Definition.

The DDT is the basis for building the BDD file. The BDD contains the DLL and DDT in binary format. When you define the extended detection rules for a character screen (see below), you have to specify the name of the BDD file.

### To build the BDD file

- From the **Extended** menu, choose **Build BDD**.

Or:

Choose **Build BDD** from the context menu that is invoked by pressing the right mouse button.

Information about the build process is now shown in the output window.

When two or more IDs with the same numbers are detected, an error message such as the following is shown in the output window:

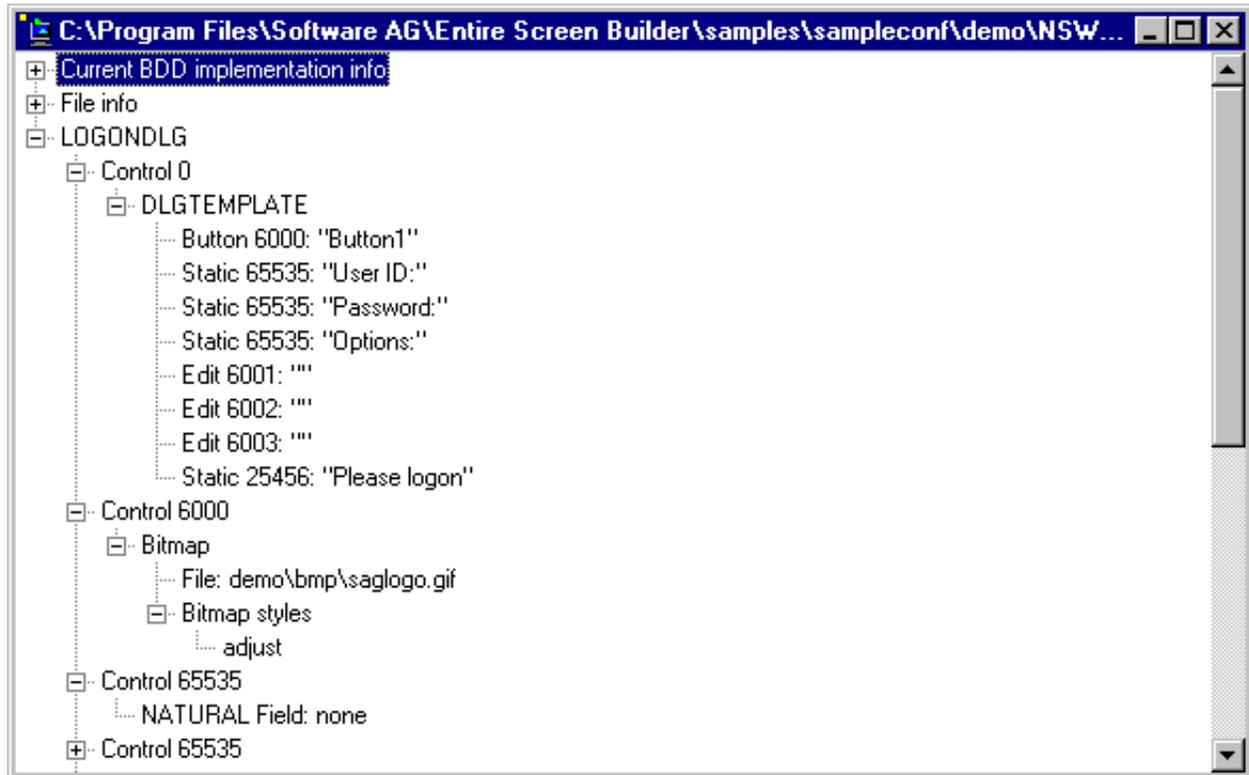
```
Error: duplicate ID 512 found in dialog "MYDIALOG", controls tab order "10" and "20".
```

The numbers given for the tab order refer to your specifications in the resource editor. This is the order in which the **TAB** key moves the input focus from one control to the next within a dialog box. Microsoft Visual Studio, for example, provides the menu command **Tab Order**. With this command, you can easily identify the controls that caused the error: just check the number in the upper-left corner of a control and compare it with the numbers in the error message.

#### **Note:**

If you use another resource editor, a **Tab Order** command may not be available or it may have another name.

Using *BDDView.exe* in Entire Screen Builder's *Bin* folder, you can view the contents of a BDD file. For example:



## Detecting the Screen on which the Extended Rules are to be Applied (Single Screen)

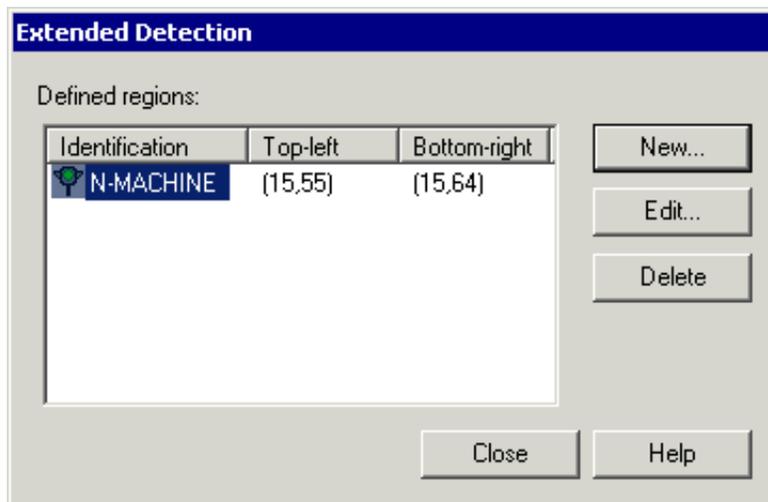
When the BDD has been built, you have to associate each extended dialog with the character screen for which it is to be displayed.

A screen is detected when a defined region is found on a host screen, or when a defined region *and* all defined subregions are found on a host screen. For each host screen, you can define more than one region.

### ▶ To display all defined detection rules for a single screen

- From the **Extended** menu, choose **Detection > Single Screen**.

The Extended Detection dialog box appears. When this dialog box is shown, a check mark is shown next to the **Single Screen** command.



When detection rules have already been defined, they are shown in this dialog box. The scope window then shows the outlines of all defined detection rules.

The color red in the symbol to the left of a defined region indicates that this rule has been disabled. A symbol with the color green indicates that the rule is enabled.

The following command buttons are available:

<b>New</b>	Add a new detection rule (see below).
<b>Edit</b>	Modify the selected detection rule. Alternative: double-click a detection rule.
<b>Delete</b>	Delete the selected detection rule. Alternative: choose <b>Delete</b> from the <b>Edit</b> menu or press DEL to delete the selected detection rule.

▶ **To associate a dialog with a character screen**

1. Make sure that the resource file that is to be used as a template is shown in the scope window.
2. Display the Extended Detection dialog box as described above.
3. Choose the **New** button.
4. In the scope window, use the mouse to select the region which contains the string that is to be used to detect the screen.

The Extended Detection Configuration dialog box appears.

5. Specify all required information as described below.
6. Choose the **OK** button.

## **Overview of Extended Detection Configuration Options**

Different pages are available in the Extended Detection Configuration dialog box:

- General
- Region

## General



### Disable the Dialog Detection rule

When this check box is selected, this rule is disabled.

### Get the content of the region as the dialog name

When this check box is selected, the server will use the content of the first region that is defined in the BDD file as the dialog name (see the description of the Regions page).

#### **Important:**

When subregions have been defined, the content of the region is only used as the dialog name if all defined subregions are also found. The subregions itself are not used as dialog names.

This feature is helpful, if the character screens always display their different names in the same region. In this case, you only have to define one rule.

### BDD file

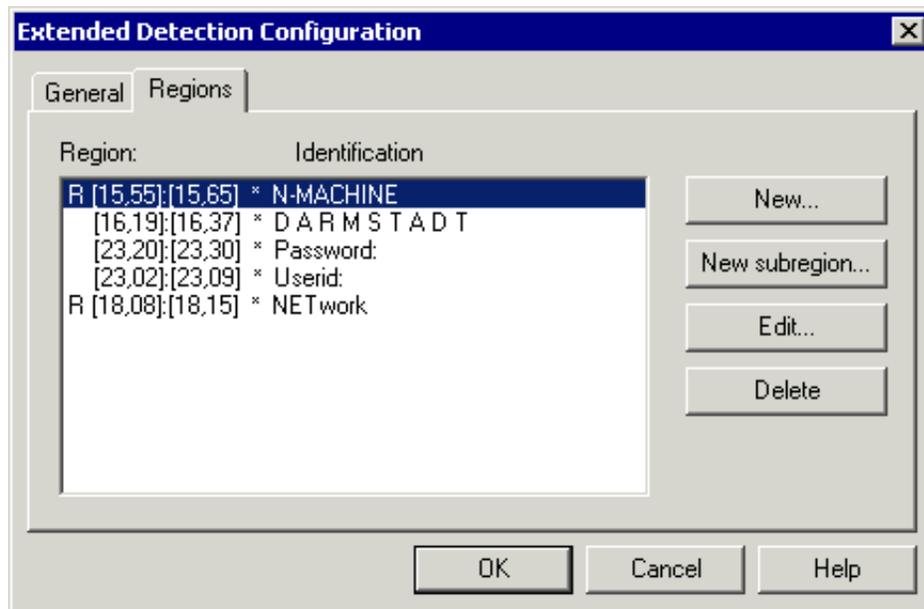
Specify the path to the BDD file containing the extended dialog for the character screen. You can also choose the **Browse** button to select the BDD file from the Open dialog box.

### Name

When a BDD file has been specified, you can select one of the extended dialogs contained in the BDD file from this drop-down list box. This is the dialog that is to be shown instead of the character screen. This text box is dimmed if the **Get the content of the region as the dialog name** check box is selected.

## Region

The Regions page is used to define regions and subregions, i.e. unique strings which appear on the host screen that is to be detected.



On this page, a defined region is indicated by the letter "R" at the beginning of a line. Lines which do not begin with the letter "R" apply to subregions.

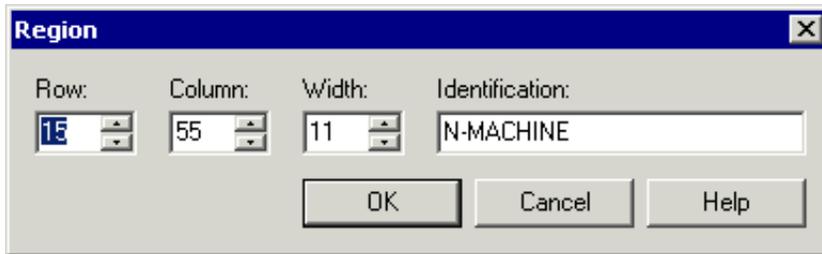
Example: with the above definitions, the screen is detected when region 1 *and* all defined subregions are found, *or* when region 2 is found.

The following command buttons are provided:

<b>New</b>	Define a new region for the current extended dialog.
<b>New subregion</b>	Define a new subregion for the selected region. For one region several subregions can be defined.
<b>Edit</b>	Modify the selected region or subregion. Alternative: double-click a region or subregion to modify it. See <i>Modifying a Region or Subregion</i> below.
<b>Delete</b>	Delete the selected region or subregion. When you delete a region, all of its subregions are automatically deleted.

## Modifying a Region or Subregion

The Region dialog box appears when you select a region or subregion on the Regions page and choose the **Edit** button (or when you double-click it).



### Row / Column / Width

The position of the string that is to be used to detect the screen. This is the region that has been selected using the mouse. Using the spin buttons, you can manually adjust the values.

### Identification

This is the string that has been selected using the mouse. This string is used to detect the screen on which the rules for the extended dialog are to be applied.