

Visual Modeler

Tutorial

Release 9.1



Copyright

This edition applies to the 9.1 Version of Visual Modeler and to all subsequent releases and modifications until otherwise indicated in new editions.

Before using this information and the product it supports, read the information in *Notices* on page 29.

Licensed Materials - Property of IBM

Visual Modeler

© Copyright IBM Corp. 1999, 2011. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP

Schedule Contract with IBM Corp.

Contents

3	
Introduction	4
Creating Product Models	5
Create the Model.	5
Properties	8
Defining Properties for the MXWS-7700 Model	8
Attaching Properties	9
Rules	11
Creating a Rule	13
Using Rules to Control Display of Option Items	14
Creating a Constraint Table	17
UI Controls.	19
Display Properties	19
Pre-Pick Guiding Text.	19
Ignore in Quote.	19
Tabular Displays	20
Calculated Property Values.	20
User-Entered Values.	23
Restricting User Entered Values	24
Images	25
Layout Management	26
Notices	29
Trademarks.	31
Index	33

Introduction

The topics in this guide provide step-by-step lessons on creating product models.

Creating Product Models

The Visual Modeler application enables you to create configurable product models. Customers can customize a product by selecting choices that the product offers to ensure that the product precisely meets their needs.

Before creating a configurable product model, you must ensure that all the products to be used in the model are already created using the IBM® Sterling Business Center application. For more information about creating a configurable product, refer to the *Sterling Business Center: Item Administration Guide*.

In this lesson, we create a configurable product model. This comprises the following steps:

1. Create the product model
2. Test the product configuration experience

After a configurable product model is created, it needs to be associated to a product. For more information about associating a model to a bundle product, refer to the *Sterling Business Center: Item Administration Guide*.

Once we have created and tested the basic model, the lesson continues by demonstrating the use of more advanced configuration options and UI controls that help you manage the customer's experience as they configure the product.

Create the Model

1. In the Product and Catalog Administration panel, click **Configuration Models**.

The Visual Modeler administration page displays.

2. Click **New Model Group**.
3. Enter Computers in the Name and Description fields and click **Save**.
4. In the model group hierarchy tree, select the Computers node.
5. Click **New Model Group**.
6. Enter Workstations in the Name and Description fields and click **Save**.

Note: We are creating a hierarchy of model groups that mirror the hierarchy of product categories. This is not necessary, but can often help to maintain the organization of models and the products to which they correspond.

7. In the model group hierarchy tree, select the Workstations node.
8. Click **New Model**.
9. Click **Browse...**
10. In the product picker window that opens up, navigate to the Computers -> Workstations product category and select the MXWS-7700 product.
11. Click **Done**.

12. Notice that the Assigned Product ID field now reads MXWS-7700 and the Name and Description fields are populated with the name and description of the MXWS-7700 product.
13. Click **Save**.
14. The new model, MXWS-7700, is displayed in the Models and Groups panel.
15. Select the MXWS-7700 model and click the **Edit** icon on the Visual Modeler toolbar.
16. The Model Detail page is now displayed.

To begin our modeling, we will create an option class and two option items in the option class: this will enable customers to choose between two different monitors to go with their workstation.

17. Click **New Option Class**.

The following diagram displays the Visual Modeler Model Detail Page: New Option Class Tab

18. Enter Monitors in the Name field and enter “Please select a monitor” in the Description field.
19. Click **Save**.

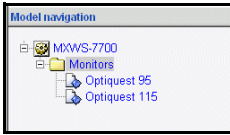
The option class is created and added to the model tree in the Model navigation panel. Now we will create two option items in this option class: these will represent the selectable items that user may select.

20. Select the Monitors option class in the model tree.
21. Click **New Option Item**.

The following diagram displays the Visual Modeler Model Detail Page: New Option Item Tab:

22. Enter Optquest Q95 in the Name and Description fields.
23. Click **Save**.
24. The New Option Item tab is re-displayed. This time, enter Optquest Q115 in the Name and Description field.
25. Click **Save**.

26. If you examine the Model navigation tree, you see that the model is now displayed with one option class and two option items.

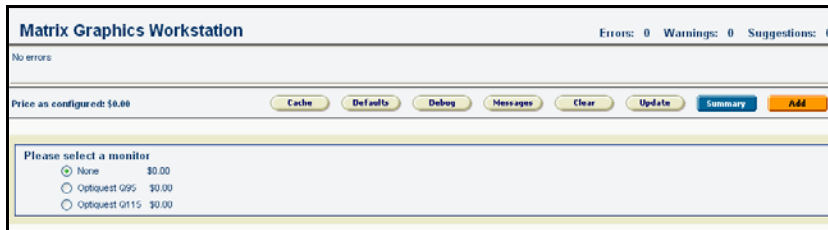


Now we are ready to compile and test this simple model.

27. Click the **Compile and Test** icon on the Visual Modeler toolbar.

Behind the scenes, an XML file is generated that holds the structure of this model. Each time we make changes to the model, we must re-compile the model so that our changes will become part of the customer's configuration experience.

A second browser window is displayed: this is the configuration experience that customers will have when they configure the MXWS-7700 product.



As you can see, this page provides a UI in which customers can select either of the two monitors (or neither), and the text displayed on the page is the descriptions that we provided for the option class and option items.

28. Close this window.

This completes the creation of the basic model. Next we must hook it up to the product.

29. Click **My Home** to return to the Enterprise Home page.

30. In the Product manager, navigate to the Product Detail page for the MXWS-7700 product.

31. In the Component Type drop-down list, select Configurable.

32. Notice that in the Model drop-down list, "Computers/Workstations/MXWS-7700" is selected automatically.

33. Click **Save Changes**.

You have completed the basic setup of this configurable product. If a customer selects the MXWS-7700 product and adds it to their shopping cart, then they will be able to configure it by choosing between the two monitor models provided.

Properties

Most of the customer's experience of configuring a product is determined by properties: these are attributes of the model, its option classes, and option items, and they are used to determine how rules fire, the behavior of the UI, and information that can be displayed to the user. The Visual Modeler provides a set of built-in properties that control the behavior of the IBM® Sterling Configurator and how the model is presented to the end-user. You can also define properties that the Configurator will use to ensure that the user's product configuration choices make sense.

This section describes how to define and attach properties to a model. You define a property at the model group or model level, and you attach a property to a model, option class, or option item. The basic steps are:

1. Determine at what level to define and attach a property. Where you attach the property determines where you can use it: properties are available for use in the model group and model hierarchy beneath the point at which they are defined. For example, you may want to define a property, `MonitorSize`, at the `Monitors` option class level so that it is available for use with each of the option items (types of monitors) in that option class.
2. Once you define the property, then you attach it at the appropriate level of the model hierarchy and, if appropriate, assign a value to it. For example, once you have defined the `MonitorSize` property, then you attach it to the option item defining a specific monitor and set the value representing the size of the monitor.

We will define properties at the model level for the monitors available with the `MXWS-7700 Workstation` product. Once the properties are defined, we will assign property values and attach them at the option item level.

Defining Properties for the `MXWS-7700 Model`

1. In the Visual Modeler, navigate to the `Workstations` level of the `Model Group` hierarchy. `MXWS-7700` displays in the `Models and Groups` panel.
2. Select `MXWS-7700`, then click the `Edit Model` icon in the Visual Modeler toolbar.
3. `MXWS-7700` displays at the top of the `Model Navigation` panel, and the **General Info** tab for the model displays.
4. Click the **Properties** tab. The `Properties` page displays as shown in the following figure.
5. Click the **Define** tab.
The `Define Properties` page displays, as shown in the following figure.
6. Define the `MXWS-7700` properties:
 - a. Enter the property name: `MonitorSize`.
 - b. Choose `Number` from the `Type` drop-down list.
 - c. Click `Add`.

The `MonitorSize` property displays in the `Defined Properties` list.

7. Add the rest of the MXWS-7700 model properties using the property names and types shown in the following table.

Name	Type
MaximumResolution	String
MonitorWeight	Number
MaximumWeight	Number
RequiresXVGA	Number

When you have finished defining and adding the properties, the Define Properties page displays as shown in the following figure.

8. Click Save All Changes.

The properties are now part of the MXWS-7700 model.

Attaching Properties

This section assumes that you are still editing the MXWS-7700 model in the Visual Modeler.

First, we will attach a property that applies to the entire model.

1. In the Model Navigation panel, click the MXWS-7700 link.

The General Info page displays.

2. Click the **Properties** tab.

The Properties page displays as shown in the following figure.

Notice the two lists: Unattached Properties and Attached Properties.

3. Choose MaximumWeight from the Unattached Properties drop-down list.
4. Enter 0 for the property value.
5. Click Attach.

The MaximumWeight property displays in the Attached Properties list.

6. Click Save All Changes.

Next, we will attach the properties that apply to the option items for the Monitors option class.

7. In the Model Navigation panel, click the Optquest Q95 link.

The **Properties** tab displays as shown in the following figure:

Notice the two lists: Unattached Properties and Attached Properties.

- a. Choose MonitorSize from the **Unattached Properties** drop-down list.
- b. Enter 19 in the Value field, then click **Attach**.

The MonitorSize property displays in the Attached Properties list.

Continue attaching properties to the option items as follows

Option Item	Property	Value
Optiquest Q95	MonitorSize	19
	MaximumResolution	1024x768
	MonitorWeight	7.2
Optiquest Q115	MonitorSize	21
	MaximumResolution	1280x1024
	MonitorWeight	8.3

When you have finished attaching properties to each option item, click **Save All Changes**.

Rules

The Visual Modeler provides you with the ability to precisely control the customer's selections so that the selections that they make are compatible with each other and that as users make selections, they see selections that are related.

For example, suppose that you want to ensure that customers pick a good graphics card to go with their monitor. You can create a rule based on their monitor selection that displays a graphics card option class that displays only compatible graphics cards. Alternatively, you can create a constraint table that specifies which monitor can be selected with which graphics card.

Before continuing this section, create a new option class and three option items. For example, you may create Graphic Cards as the Option Class with the following items:

Property Definition	Option Item	Property Value
Resolution	GC-1000	Resolution: VGA
	GC-2000	Resolution: XVGA
	GC-3000	Resolution: XVGA

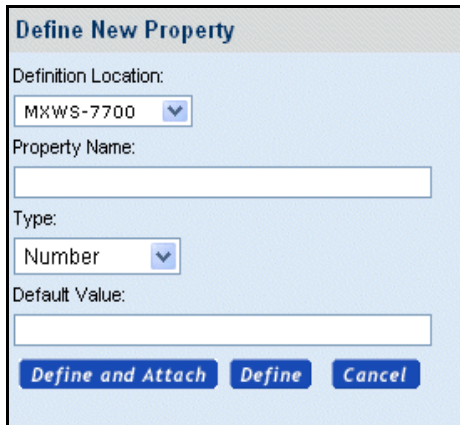
Define the Resolution property at the Graphics Cards level of the model hierarchy as follows:

1. In the Model navigation panel, click the MXWS-7700 link.
The **General Info** tab displays.
2. Click New Option Class.
The New Option Class page displays.
3. Enter Graphics Cards in the Name field and in the Description field.
Click Save and Edit.
4. The **General Info** tab for the new Graphics Cards option class displays.
Click the **Properties** tab.

5. The **Properties** tab displays.

Click New Property.

The Define New Property pop-up window appears, as shown in the following figure.



The image shows a dialog box titled "Define New Property". It contains the following fields and controls:

- Definition Location:** A drop-down menu with "MXWS-7700" selected.
- Property Name:** An empty text input field.
- Type:** A drop-down menu with "Number" selected.
- Default Value:** An empty text input field.
- At the bottom, there are three buttons: "Define and Attach", "Define", and "Cancel".

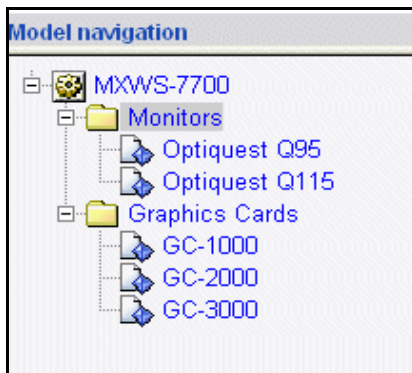
6. Choose Model Groups from the Definition Location drop-down list.

7. Enter Resolution for the Property Name.

8. Choose String from the Type drop-down list.

9. Click Define.

Define the three new option items and attach the Resolution property to them using the values in the table mentioned above. Define the option items as described in "[Create the Model](#)" and attach the Resolution property as described in "[Attaching Properties](#)". Be sure to click Save All Changes as you complete each step. When you are finished, the Model navigation panel displays as shown in the following figure.



In the Model navigation panel, click the Graphics Cards link, then click the **Display** tab. Choose Invisible from the Option Class Display drop-down list, then click Save All Changes.

On the Monitors option class, create a new property as follows

Property Definition	Option Item	Property Value
IsPicked	Optiquest Q95	1
	Optiquest Q115	1

Creating a Rule

The first rule we make is to show the Graphics Card option class if a user selects one of the monitor option items.

1. Navigate to the model MXWS-7700.
2. Click the Rules tab.
3. Click the Define sub-tab.
4. Click **New....**

5. Enter the following information:
 - a. Name: Display Graphics Card
 - b. Description: Display the Graphics Card option class if a monitor is selected.
6. Select Rule Triggered on: Success.
7. Click **Save and Edit**.
8. Click the New Fragment icon.

The New Fragment page displays.
9. Define the fragment:
 - a. Choose value from the Function1 drop-down list.
 - b. Choose IsPicked from the Property1 drop-down list.
 - c. Choose = from the Operator drop-down list.
 - d. Choose literal from the Function2 drop-down list.

- e. Enter 1 in the Property2 drop-down list.
- f. Choose Rule is false from the If not specified drop-down list.

10. Click **Save and Return**.

We want the rule to assign the `_isVisibleable` property to the Graphics Cards option class if the rule is true, so define the assignment action like this:

1. In the Assignment Actions panel, select `_isVisibleable` property from the drop-down list.
2. Enter 1 for the Value.
3. Click the **...** button next to the Assign To field, and in the pop-up window, navigate to the Graphics Card option class. Select it and click **Done**.

The Assign To field is populated with the value `*.Graphics Cards`.

4. Click **Add Item**.
5. Click **Save**.

Having defined the rule, now we must attach it to the model.

1. Select the Model MXWS-7700 from the navigation panel, and click the **Rules** tab.
2. On the **Attach** sub-tab, select the Display Graphics Card rule from the drop-down list, and click **Attach**.
3. Click **Save All Changes**.

Now compile and test the model. You see that the Graphics Card option class is hidden until you select a monitor, and then it is displayed so that a graphics card can be selected.

Using Rules to Control Display of Option Items

You can use assignment actions to control the display of option items: this gives you the ability to ensure that customers only make valid selections. In "[Creating a Constraint Table](#)" we demonstrate an alternate approach using constraint tables. In this section, we use a rule with an assignment action to ensure that a customer will only pick valid combinations of graphics cards and monitors.

For the purposes of this example, our rule will say that if you select a monitor whose maximum resolution is 1280 x 1024, then you must select a graphics card that supports XVGA.

1. Navigate to the model MXWS-7700.
2. Click the Rules tab.
3. Click the Define sub-tab.
4. Click **New...**
5. Enter the following information:
 - a. Name: Display Compatible Graphics Cards
 - b. Description: This rule ensures that only graphics cards that support each monitor are displayed.
6. Select Rule Triggered on: Success.
7. Click **Save and Edit**.
8. Click the **New Fragment** icon.

9. Specify the fragment as follows:
 - a. Choose value from the Function1 drop-down list.
 - b. Choose MaximumResolution from the Property1 drop-down list.
 - c. Choose = from the Operator drop-down list.
 - d. Choose literal from the Function2 drop-down list.
 - e. Enter 1280x1024 in the Property2 field.

Specifying the fragment in this way is equivalent to the following formula:

```
value(Maximum Resolution) = literal(1280 x 1024)
```

10. Choose Rule is false from the If not specified drop-down list.

11. Click **Save and Return**.

If the rule is true, then we want the rule to assign the RequiresXVGA property to the model, and then have a rule that ensures that only compatible graphics cards option items are displayed, so define the assignment action like this:

12. In the Assignment Actions panel, select the RequiresXVGA property from the drop-down list.
13. Enter 1 for the Value.
14. Click the ... button next to the Assign To field, and in the pop-up window, navigate to the MXWS-7700 model. Select it and click **Done**.

The Assign To field is populated with the value “MXWS-7700”.

15. Click Add Item.
16. Click **Save**.

17. Now attach this rule to the model.

So far, we have created a rule that tells the model that if certain monitors are selected, then the graphics card that is selected must support XVGA. Now we create a rule that can be attached to each graphics card option item that determines whether it can be displayed.

1. Navigate to the model MXWS-7700.
 - a. Click the Rules tab.
 - b. Click the Define sub-tab.
 - c. Click **New...**
 - d. Enter the following information:
 - e. Name: Display if Support XVGA
 - f. Description: Display this card if XVGA support is required and the card supports XVGA.
2. Select Rule Triggered on: Failure.
3. Click **Save and Edit**.
4. Click the New Fragment icon.

5. Specify the fragment as follows:
 - a. Choose value from the Function1 drop-down list.
 - b. Choose RequiresXVGA from the Property1 drop-down list.
 - c. Choose = from the Operator drop-down list.
 - d. Choose literal from the Function2 drop-down list.
 - e. Enter 1 in the Property2 field.

Specifying the fragment in this way is equivalent to the following formula:

```
value(RequiresXVGA) = literal(1) in any location
```

6. Set **If not specified** to Rule is true.

In effect, this says that if the RequiresXVGA property is not set, then assume that all graphics cards are valid selections.

7. Click **Save and Return**.

8. Click the New Fragment icon.

9. Specify the fragment as follows:

- a. Choose propval from the Function1 drop-down list.
- b. Choose Resolution from the Property1 drop-down list.
- c. Choose = from the Operator drop-down list.
- d. Choose literal from the Function2 drop-down list.
- e. Enter XVGA in the Property2 field.

Specifying the fragment in this way is equivalent to the following formula:

```
propval(Resolution) = literal(XVGA) in any location
```

Note that we have to use the propval function here rather than value: this is because the option item will not have been picked at the time the rule fires.

10. Set **If not specified** to Rule is true.

11. Click **Save and Return**.

12. In the Assignment Actions panel, select `_isViewable` property from the drop-down list.

13. Enter 0 for the Value.

14. Leave the Assign To field value blank. This is to indicate that the property is at the node at which the rule is attached.

15. Click Add Item.

16. Click **Save**.

17. Now attach this rule to the each graphics card.

Before compiling our model, the last thing we have to do is to manage the order in which the rules fire. We want to ensure that the rule that tests to see if XVGA is required fires before the rules that determine if each graphics card option item is compatible.

18. Navigate to the Model node.

19. Click the Rules tab.
20. Click the Firing Sequence sub-tab.
21. Change the Priority value of the Display Compatible Graphics Cards rule to 10.

This ensures that this rule will fire first.

22. Click Save All Changes.

Now compile and test the model. You see that the Graphics Card option class is hidden until you select a monitor, and then it is displayed so that a graphics card can be selected. If you select the Optquest Q115 monitor, then the GC-1000 graphics card is not displayed.

In the next section, "[Creating a Constraint Table](#)", you will create a table which constrains which option items are compatible with each other. Before proceeding to the next section, remove the rule Display if Support XVGA from each of the graphics cards.

Creating a Constraint Table

Not all of the graphics cards may be compatible with all of the monitors, and so you want to specify what combinations of monitors and graphics cards are acceptable. In this section, we create a constraint table to express this.

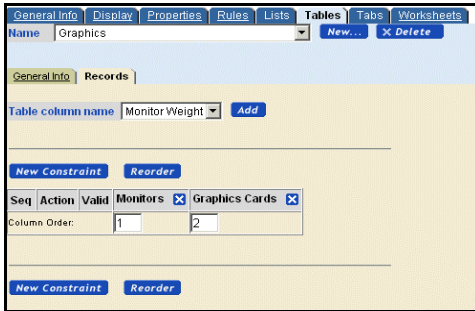
Assume that the following combinations of Graphics Card and Monitor are compatible:

Compatible?	Optquest Q95	Optquest Q115
GC-1000	Yes	No
GC-2000	Yes	Yes
GC-3000	Yes	Yes

You can express constraint tables either by specifying what option items can be selected together or which cannot. In this example, it is easier to specify that GC-1000 and Optquest Q115 cannot be selected together: the other selections are assumed to be compatible.

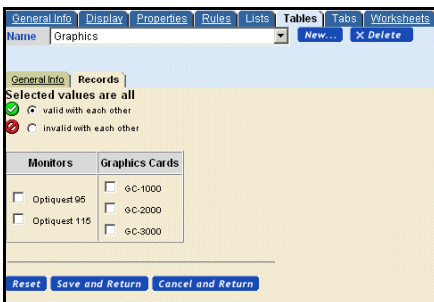
1. Navigate to the model MXWS-7700.
2. Click the Tables tab.
3. Click **New...**
4. Enter the following information:
 - a. Name: Graphics
 - b. Description: Constrains the selection of graphics cards and monitors.
 - c. Message: You cannot select this combination of graphics card and monitor.
5. Click **Save Changes**.
6. Click the Records tab.
7. Select Monitors from the Table Column Name drop-down list and click **Add**.

8. Select Graphics Cards from the Table Column Name drop-down list and click **Add**.



9. Click **New Constraint**.

10. Click **Edit**.



11. Select the **invalid with each other** radio button, and check the Optiquet Q115 and GC-1000 check boxes.

12. Click **Save and Return**.

Now compile and test the model. You see that the Graphics Card option class is hidden until you select the Optiquet Q115 monitor, and then it is displayed so that a graphics card can be selected. You see that the GC-1000 option item is displayed with a clickable icon that indicates that it should not be selected. If you do select it, then an error message is displayed. You can click the icon to ask the Visual Modeler to help resolve the conflict: in this case, it will suggest an alternate selection of monitor.

UI Controls

The Visual Modeler provides a rich set of controls that can provide a flexible and attractive UI to help customers make their selections. This topic describes how to use them.

Display Properties

Each model, option class, and option item has a set of properties that determine how the configurable product will be displayed to the customer: these are known as display properties. Every display property corresponds to a UI property as indicated below.

Pre-Pick Guiding Text

Suppose that you want to provide some text to help customers make a selection, but want to remove the text once the customer has done so.

1. Navigate to the Monitors option class.
2. Click the **Display** tab.
3. Enter the following in the Pre-Pick Guiding Text field: “The larger the monitor, the easier it is to manage multiple displays on it.”
4. Click **Save All Changes**.

If you now compile and test the model, then you will see that this text is displayed when you first display the model. However, if you select one of the monitors, then when the page is re-displayed, you can see that the text is now removed.

This display property corresponds to the UI: PRE_PICK GUIDING TEXT property.

Ignore in Quote

By default, option classes and option items are displayed in the customer’s cart when they completed their product configuration and put their configured product in their cart. If you do not want an option class to be displayed in the customer’s cart, then do this:

1. Navigate to the Monitors option class.
2. Check the Ignore In Quote check box.
3. Click **Save**.

If you now compile and test the model, then you will see that the monitor class is displayed. However, if you click the Summary button, then the Monitors option class is not displayed on the Summary page (though any option items picked are).

This display property corresponds to the UI: IGNORE IN QUOTE property.

Tabular Displays

Suppose that you would like to display the monitor option items with some of their properties to help users choose among them. You can do this in the form of a table as follows.

1. Navigate to the Monitors option class.
2. Click the **Display** tab.
3. Select Tabular Display from the UI control drop-down list.
4. Click **Save All Changes**.
5. In the Tabular Display Control Settings section of this page, enter:
 - a. Column Headings: Size;Resolution
 - b. Column Properties: MonitorSize;MaximumResolution
 - c. Column Alignment: Left;Left
6. Click **Save All Changes**.
7. Click **Compile and Test**.

The screenshot shows the Matrix Graphics Workstation interface. At the top, it displays 'Matrix Graphics Workstation' and 'Errors: 0 Warnings: 0 Suggestions: 0'. Below this, there is a 'No errors' message. A price summary shows 'Price as configured: \$0.00' with buttons for 'Cache', 'Defaults', 'Debug', 'Messages', 'Clear', 'Update', 'Summary', and 'Add'. The main content area is titled 'Please select a monitor.' and contains a table with the following data:

Name	Size	Resolution
Optquest Q95	19.0	1024x768
Optquest Q115	21.0	1280x1024

Below the table, there is a 'Graphics Cards' section with radio button options and prices:

- None \$0.00
- GC-1000 \$0.00
- GC-2000 \$0.00
- GC-3000 \$0.00

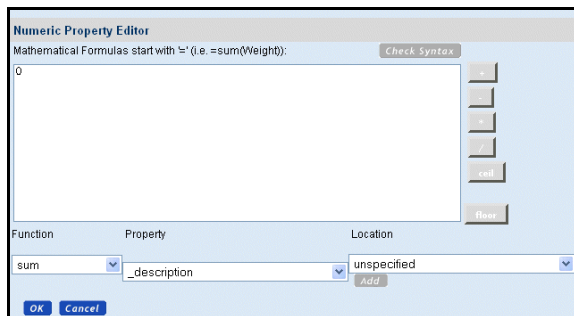
Note that the option items are not selectable in this display. You probably want to add a second option class that makes them selectable.

Calculated Property Values

The Visual Modeler provides a simple, yet powerful, means to use property values to calculate other property values. In this section, we show how use this mechanism to display extra information to customers. Suppose that you know that for each monitor, you know the diagonal linear dimension (d) of the monitor, but you want to present to the customer the total area (A) of the monitor screen. This can be approximately calculated as $A = d*d/2$. You can do this along these lines:

1. Navigate to the Monitors option class.
2. Define a numeric property called Monitor Area.
3. Attach this property to the two monitors.
4. Navigate to the Monitors option class.
5. Click the **Display** tab.

6. Change the table column properties as follows:
 - a. Column Headings: Size;Area;Resolution
 - b. Column Properties: MonitorSize;MonitorArea;MaximumResolution
 - c. Column Alignment: Left;Left;Left
7. Click **Save All Changes**.
8. Navigate to the model node.
9. Click the **Worksheets** tab, and then create a worksheet as follows:
10. Click **New...**
11. Enter the following information:
 - a. Name: Calculate Area
 - b. Click **Create**.
12. Click **Add Column** and add first Monitor Size and then Monitor Area.
13. Click Add Row and, using the entity picker, add the monitor option items.
14. In the first row and Monitor Area column, click the Edit Property Value button.



15. Enter “=” as the first character in the text area.
16. In the Numeric Property Editor window, select the Function value, MonitorSize Property, and the unspecified Location and click **Add**.
17. Click * from the mathematical symbols along the side.
18. In the Numeric Property Editor window, select the Function value, MonitorSize Property, and the unspecified Location and click **Add**.
19. Click / from the mathematical symbols along the side.
20. Enter 2.
- You should see the following in the text area:


```
value("Monitor Size")*value("Monitor Size")/2
```
21. Click **OK**.
22. In the second row and Monitor Area column, click the Edit Property Value button and repeat the steps above for this monitor.
23. Click **Save All Changes**.

If you now compile and test the model, then you will see that the Monitors option class is displayed as a three-column table, and the Area column is calculated from the Size column.

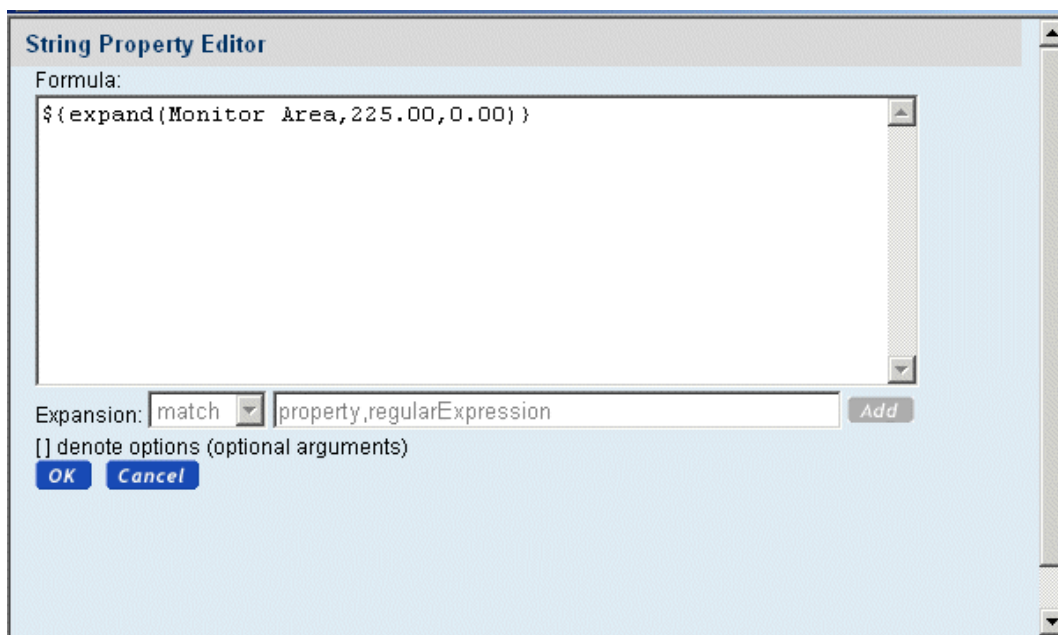
You can also use Java classes in the Numeric and String Editor windows. For example, suppose that the monitors are circular, and the Monitor Size (d) property is the radius of the monitor. Then the area (A) should be calculated as $\pi*d*d$.

You can modify the Monitor Area property formula to read:

```
=java.lang.Math.PI*value("Monitor Size")*value("Monitor Size")
```

To make this more readable, you can use a string-formatting expression to define a more readable property as follows.

1. Navigate to the Monitors option class.
2. Define a new String property called MonitorAreaString.
3. Navigate to the MXWS-7700 model and click the **Worksheet** tab.
4. Add a new column to the worksheet by adding the MonitorAreaString property.
5. In the first row and MonitorAreaString column, click the Edit Property Value button.



6. Enter “`${expand(Monitor Area,225.00,0.00 inches)}`” in the text area.
7. Click **OK**.
8. Now navigate to the Column Properties field on the Monitors class **Display** tab, and change it to: `MonitorSize;MonitorAreaString;MaximumResolution`

If you now compile and test the model, then you will see that the Monitors option class is displayed as a three-column table, and the Area column is calculated from the Size column and displayed in the form 1134.11 inches.

User-Entered Values

You may want to permit customers to enter values for properties: these can be used to check user requirements against rules that determine whether option items match the requirements. To do so, you must specify that the relevant option class UI Control is a User Entered Value, and then at the option item level, specify how the user-entered value is bound to a property. Typically, you want to bind the value to a property that can be used in a model rule.

For example, suppose that you want to enable users to specify a maximum weight for their computer monitor. You can do this as follows:

1. Create an option class called MonitorWeight.
2. On its **Display** tab, select User Entered Values from the UI Control drop-down list.
3. Set the Number of Columns display property to 2.
4. Click **Save All Changes**.
5. Create a single option item below this option class: call it Weight.
6. On the **Display** tab for Weight, in the User Entered Value Settings section, enter:
 - a. For the User Entered Value Type, select Numeric
 - b. Assign Value to Property: *.MaximumWeight
 - c. This references the MaximumWeight property which is attached at the model level.
 - d. Text before Entry Field: Enter the maximum weight for the monitor in kg.
7. Click **Save All Changes**.

If you compile and test this model, you will see the following option class section:



Enter the maximum weight for the monitor in kg.: 0.0

To show how the user-entered value can be used:

8. Navigate to the Model level, and then create the following rule:
 - Name: Maximum Weight at Model
 - Triggered on: Failure
 - Fragment: In Function1, select value, MonitorWeight, any from the drop-down lists, select ">" from the Operator drop-down list, and in Function2, select value, MaximumWeight, relative, and select Rule is true from the If not specified drop-down list, so that the rule reads:

```
value(MonitorWeight) > value(.MaximumWeight)
```
 - Error message: The selected monitor exceeds your specified maximum weight.
9. Save this rule.
10. Attach this rule at the MXWS-7700 node.

When you compile and test this model, you will see that depending on your choice of monitor and the value you enter for the maximum weight of the monitor, an error message is displayed when the monitor weight exceeds the maximum weight.

What happens here is that when a customer specifies a value in the Weight text field and updates the model, the value they enter is assigned to the MXWS-7700.MaximumWeight property, and then the rule compares this value to the value of the MonitorWeight property defined anywhere in the model. In this case, the only place where MonitorWeight has been attached to the nodes is at the two monitor option items, and so the value of the MonitorWeight property at the selected monitor node is used.

An alternative approach to writing this rule is to specify that the MonitorWeight property should be retrieved from the point where the rule is attached and then attach this version of the rule to each of the monitor option items. Try this as follows:

1. Detach the rule Maximum Weight at Model rule from the model, so that it does not participate in the rule firing process.
2. Create the following rule:
 - Name: Maximum Weight at Monitor
 - Triggered on: Failure
 - Fragment: In Function1, select value, MonitorWeight, relative from the Location1 drop-down list, select ">" from the Operator drop-down list, and in Function2, select value, MaximumWeight, relative, and select Rule is true from the If not specified drop-down list, so that the rule reads:

```
value(.MonitorWeight) > value(.MaximumWeight)
```
 - Error message: The selected monitor exceeds your specified maximum weight.
3. Save this rule.
4. Attach this rule at the monitor option item nodes.

When you compile and test this model, you will see that depending on your choice of monitor and the value you enter for the maximum weight of the monitor, an error message is displayed when the monitor weight exceeds the maximum weight.

What happens in this case is that when a customer specifies a value in the Weight text field and updates the model, is that the value they enter is assigned to the MXWS-7700.Maximum Weight property, and then the rule compares this value to the value of the Monitor Weight property defined at the node where this rule is attached. In this case, the only place where Monitor Weight has been attached to the nodes is at the two monitor option items, and so the value of the Monitor Weight property at the selected monitor node is used.

Before proceeding with this lesson, detach the Maximum Weight at Monitor rule from the Monitor option items.

Restricting User Entered Values

Some of the time you may want to restrict the possible values that a customer can enter in a user-entered value field. You can do this using the Allowed Values display property.

1. Navigate to the Weight option item of the Monitor Weight option class and click the **Display** tab.
2. In the Allowed Values display property field, enter: 0-20.

When you compile and test this model, you see that the Weight text field is now a drop-down list which is populated by integer values from 0.0. to 20.0. A customer can only select one of these values, and when they do, their selection is automatically submitted to the server.

As another example, suppose that you want another user-entered value field for a customer's color preference, and again you want to indicate that the customer must select only from a choice of colors. Do the following:

1. Create an option class called Color.
2. On its **Display** tab, select User Entered Values from the UI Control drop-down list.
3. Click **Save All Changes**.
4. Create a single option item below this option class: call it Color Choice.
5. On the **Display** tab, in the User Entered Value Settings section, enter:
 - a. For the User Entered Value Type, select String
 - b. Enter Black,Blue,Green,Red,White in the Allowed Values field.
 - c. Assign Value to Property: *.Color
 - d. This references the Color property which you can define and attach at the model level.
 - e. Text before Entry Field: Select your preferred color.
6. On the **Properties** tab, select the UI: SUPPRESS UEV NONE VALUE property and enter yes for its value, and click **Attach**.
7. Click **Save All Changes**.

When you compile and test this model, you see the new Color option class, and a drop-down list of values from which the customer can make their selection. Note that None is not a selectable item.

Images

You can associate images with a model, option class, or option item simply by specifying the Image display property: this takes as values relative URLs or absolute URLs:

- ☐ If you begin the URL with "http://", then the URL is assumed to be absolute;
- ☐ If you begin the URL with "/", then the URL is interpreted relative to the servlet container;
- ☐ If you begin the URL without either, then the URL is interpreted relative to the current URL:.

Image Value	URL
http://webserver:port/images/4Stars.gif	http://webserver:port/images/4Stars.gif
/images/4Stars.gif	http://server:port/images/4Stars.gif
4Stars.gif	http://server:port/en/US/images/4Stars.gif

At the model or option class level, the Image display property corresponds to the UI: ICON GRAPHIC property. At the option item level, the Image display property corresponds to the UI: ITEM IMAGE NAME property. Note that if you define an Image display property at the option item level, then you must also set the UI: SHOW ITEM IMAGES property to be true at the option class level.

Layout Management

You can use UI properties to manage the basic layout of a configurable product. By specifying the numbers of rows and columns each option class occupies, and by specifying the number of columns on the page, you can fine-tune the look-and-feel of your page without touching the underlying JSP page.

In this section, we will add another option class, and then manage the page layout so that the Monitors option class occupies all of the first row and the other two option classes occupy the second row.

First, create new option classes as follows

Option Item Name	Description
Pentium 4 2 GHz	Pentium 4 2.8 GHz
Pentium 4 2A GHz	Pentium 4 2.8A GHz
Pentium 4 2C GHz	Pentium 4 2.8C GHz

Option Item Name	Description
SDRAM 256MB	SDRAM 256MB
DDR 256MB	DDR 256MB
RDRAM 256MB	RDRAM 256MB

Option Item Name	Description
Logitech 967300	Logitech 967300
Gyration GP170	Gyration GP170
Adesso 595	Adesso 595

Now manage the layout of the configurable product as follows:

1. Navigate to the MXWS-7700 model.
2. On the **Display** tab, set the Number of Columns property to 3. This is equivalent to setting the UI: NUMBER OF COLUMNS property to 3.
3. Navigate to the Monitors option class.
4. On the **Display** tab, set the Number of Columns property to 3. This is equivalent to setting the UI: NUMBER OF COLUMNS property to 3.
5. Navigate to the Graphics Cards option class.
6. On the **Display** tab, set the Number of Columns property to 1.
7. Navigate to the MonitorWeight option class.

8. On the **Display** tab, set the Option Class Display to Invisible.
9. Navigate to the Processors option class.
10. On the **Display** tab, set the Number of Columns property to 1.
11. On the **Properties** tab, enter Processors in the value field of the _description property.
12. Navigate to the RAM option class.
13. On the **Display** tab, set the Number of Columns property to 1.
14. On the **Properties** tab, enter RAM in the value field of the _description property.
15. Navigate to the Keyboards option class.
16. On the **Display** tab, set the Number of Columns property to 1.
17. On the **Properties** tab, enter Keyboards in the value field of the _description property.

When you compile and test this model, and then select a monitor, then the page is laid out as illustrated below.

Name	Size	Area	Resolution
Optquest G95	19.0	180.5	1024x768
Optquest G115	21.0	220.5	1280x1024

Suppose instead that you want the Graphics Cards option class to be displayed in two rows and one column, and then have the Processors class take up two columns. Then do this:

18. Navigate to the Graphics Cards option class.
19. On the **Display** tab, set the Number of Rows property to 2. This is equivalent to setting the UI: ROW SPAN property to 2.
20. Navigate to the Processors option class.
21. On the **Display** tab, set the Number of Columns property to 2.

When you compile and test this model, and then select a monitor, then the page is laid out as illustrated below.

Finally, add one more option class. This is a user-entered value class called Case with one option item called Color. Set up the option item as a String-valued property that can take a value from this list: Black,Blue,Green,Red,White.

When you compile and test this model you see that the layout is skewed by the Case option class sticking out on the third row. To correct this, you need to specify that the RAM option class must skip a column: this accounts for the fact that the Graphics Cards option class takes up two rows.

22. Navigate to the RAM option class.

23. On the **Display** tab, set the Number of Columns to Skip property to 1. This is equivalent to setting the UI: SKIP COLUMNS property to 1.

When you compile and test the model, you now see that the rows and columns are again what you expect.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual

Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS

FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA__95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are

fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2011. Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2011.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise, Gentran®, Gentran:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

A

Allowed Values display property 24

D

define and attach properties 8

display properties 19

E

expand function 22

I

Introduction 4

P

propval function 16

U

UI: ICON GRAPHIC property 25

UI: IGNORE IN QUOTE property 19

UI: ITEM IMAGE NAME property 25

UI: NUMBER OF COLUMNS property 26

UI: PRE_PICK GUIDING TEXT property 19

UI: ROW SPAN property 27

UI: SHOW ITEM IMAGES property 25

UI: SKIP COLUMNS property 28

UI: SUPPRESS UEV NONE VALUE property 25

user-entered values 23