RulesManager for WinForms

2020.08.25 更新

グレープシティ株式会社

RulesManager for WinForms

目次

RulesManagerの概要

RulesManager is a UI library that allows you to manage conditional formatting rules at runtime. It has easy to use API that allows you to add conditional formatting easily through code. With Excel like conditional formatting UI, RulesManager can be attached to any data aware control such as FlexGrid, DataGrid or FlexChart to provide complete runtime conditional formatting management. It has an expression parser and formula editor that can evaluate any formula that returns true or false. In addition, it supports colors, data bars, gradients for formatting cells based on rules.

Documentation
クイックスタート
主な特長
条件付き書式ルールの管理
書式ルールの適用

API References
C1.Win.RulesManager.4.5.2 アセンブリ

Note: ComponentOne Rules Manager is compatible with both .NET 4.5.2 and .NET 5 Frameworks.

主な特長

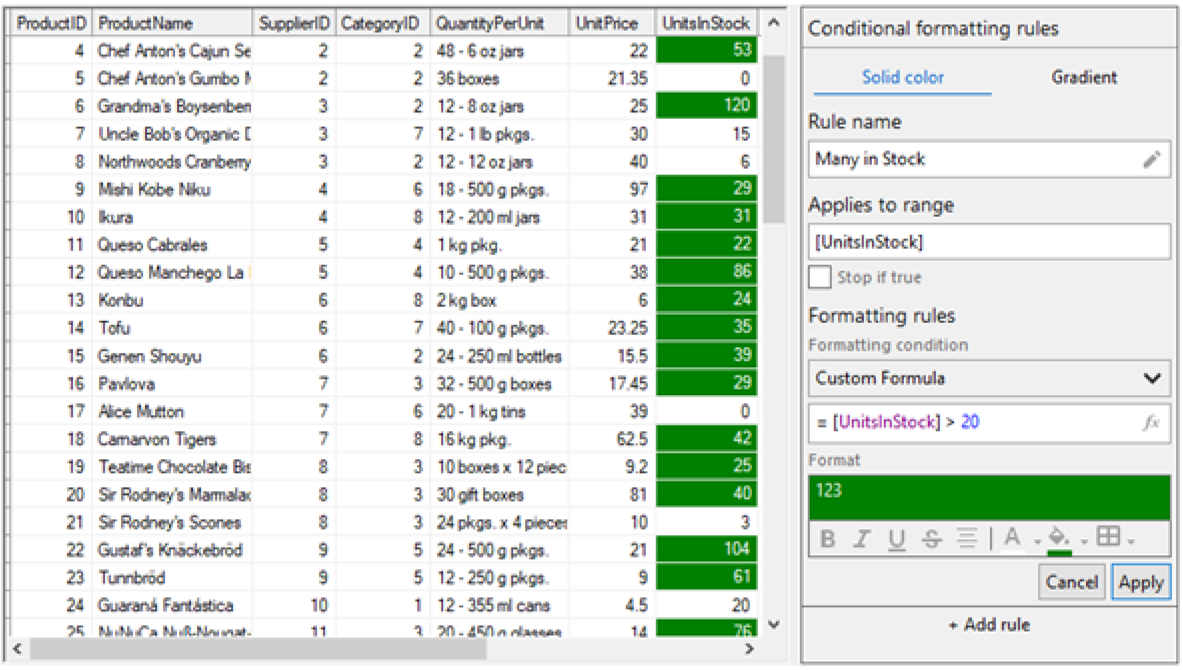
Some of the key features of Rules Manager that you may find useful include the following:

- **Complete Rule Management UI**
  Rules Manager UI library displays a list of formatting rules for any data which gives the power of conditional formatting in your hands and lets you manage the rules at runtime.
- **Familiar Excel-like Conditional Formatting**
  Rules Manager allows you to deliver Microsoft Excel like features, such as adding data bars, color scales, gradients, icons and indicators to highlight cells with key values.
- **Runtime Rule Management**
  Rules Manager allows you to perform many actions to manage the rules at runtime. This includes adding, deleting, reordering rules and changing the precedence in which rules are applied.
- **Manage Grid Cell Styling**
  Rules Manager lets you edit the cell styles and define the logic by using built-in conditions or creating an expression.
- **Manage Indicators**
  Rules Manager provides you the ability to manage indicators on a chart or any other similar control.

クイックスタート

This quick start will guide you through the steps of adding **C1FlexGrid** and **C1RulesManager** controls, binding C1FlexGrid to data source, integrate C1RulesManager to the C1FlexGrid control, and applying conditional formatting on the grid data.

You can achieve the following output through the design view or completely through the code.



ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
4	Chef Anton's Cajun Se	2	2	48 - 6 oz jars	22	53
5	Chef Anton's Gumbo I	2	2	36 boxes	21.35	0
6	Grandma's Boysenben	3	2	12 - 8 oz jars	25	120
7	Uncle Bob's Organic [	3	7	12 - 1 lb pkgs.	30	15
8	Northwoods Cranberry	3	2	12 - 12 oz jars	40	6
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	29
10	Ikura	4	8	12 - 200 ml jars	31	31
11	Queso Cabrales	5	4	1 kg pkg.	21	22
12	Queso Manchego La	5	4	10 - 500 g pkgs.	38	86
13	Konbu	6	8	2 kg box	6	24
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39
16	Pavlova	7	3	32 - 500 g boxes	17.45	29
17	Alice Mutton	7	6	20 - 1 kg tins	39	0
18	Carnarvon Tigers	7	8	16 kg pkg.	62.5	42
19	Teatime Chocolate Bis	8	3	10 boxes x 12 piec	9.2	25
20	Sir Rodney's Marmalac	8	3	30 gift boxes	81	40
21	Sir Rodney's Scones	8	3	24 pkgs. x 4 pieces	10	3
22	Gustaf's Knäckebröd	9	5	24 - 500 g pkgs.	21	104
23	Tunnbröd	9	5	12 - 250 g pkgs.	9	61
24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.5	20
25	NuNuCa Nuß-Nougat-	11	3	20 - 450 g glasses	14	76

Conditional formatting rules

Solid color | Gradient

Rule name: Many in Stock

Applies to range: [UnitsInStock]

□ Stop if true

Formatting rules

Formatting condition: Custom Formula

= [UnitsInStock] > 20

Format: 123

Through Design View

Note: WinForms .NET 5 Edition has only runtime assemblies. Due to the new design-time model in VS2019 Preview, which is not complete yet from the Microsoft side, we do not supply any special design-time features as of yet. So, a RulesManager application for .NET 5 can only be created programmatically for now. For the programmatic steps for this quick start sample, see the "Through Code View" implementation tab.

Set up the application

1. Create a new **Windows Forms App**.
2. Drag and drop the **C1FlexGrid** and **C1RulesManager** controls from the **Toolbox** onto the **Form**.
3. From the **Properties** window, set the **Name** property of the C1RulesManager control to rulesManager and C1FlexGrid to flexGrid.

Bind FlexGrid to a data source

1. Click on the C1FlexGrid's smart tag (▶) to open the C1FlexGrid Tasks menu.
2. In the **C1FlexGrid Tasks** menu, click the **Choose Data Source** drop-down arrow and select the **Add Project Data Source** link from the drop-down box.
3. The **Data Source Configuration Wizard** appears. Leave the default setting, **Database**, selected on the **Choose a Data Source Type** page, and click **Next**.
4. On the **Choose a Database Model** page, leave **Dataset** selected and click **Next**.
5. Click the **New Connection** button to create a new connection or choose one from the drop-down list. When you click **New Connection**, the **Add Connection** dialog box appears.
6. Leave **Microsoft Access Database File** as the **Data source**.
7. Click the **Browse** button under **Database file name**. In the **Select Microsoft Access Database File** dialog box,

browse to the C1NWind.mdb database in the **Documents\ComponentOne Samples\Common** directory. Select the C1NWind.mdb file and click **Open**.

8. In the **Add Connection** dialog box, click the **Test Connection** button to make sure that you have successfully connected to the database or server and click **OK**.
9. Click **OK** again to close the **Add Connection** dialog box.
10. Click the **Next** button to continue. A dialog box will appear asking if you would like to add the data file to your project and modify the connection string. Since it is not necessary to copy the database to your project, click **No**.
11. Save the connection string in the application configuration file by checking the **Yes, save the connection as** box and entering a name. Click the **Next** button to continue.
12. On the **Choose Your Database Objects** page, expand the **Tables** node, and select the **Products** table. Click **Finish** to exit the wizard.
13. A DataSet and connection string are added to your project. Additionally, Visual Studio automatically creates the following code to fill the DataSet.

    C#

    ```
    this.productsTableAdapter.Fill(this.c1NWindDataSet.Products);
    ```

Integrate RulesManager with FlexGrid

To integrate RulesManager with FlexGrid, use the following code:

C#

```
rulesManager.SetC1RulesManager(flexGrid, rulesManager);
```

Apply conditional formatting

1. Click on the C1RuleManager's smart tag (▶) to open the C1RuleManager Tasks menu.
2. In the **C1RuleManager Tasks** menu, select **Edit Rule**. This opens the **IRule Collection Editor**.
3. In the **IRule Collection Editor**, click **Add** to add a rule and set the following properties:

Property	**Value**
Name	Many In Stock
Expression	= [UnitsInStock] > 20
Style > BackColor	Green
Style > ForeColor	White

4. In the **IRule Collection Editor,** click the **AppliesTo** ellipsis button. This opens the **IItemRange Collection Editor.**
5. In the **IItemRange Collection Editor**, click **Add,** set the **Fields** property to "UnitsInStock" and click **OK**. This closes the editor and applies the rule you set to the "UnitsInStock" column of the grid.
6. Click **OK** to close the **IRule Collection Editor.** Observe that the rule you just created gets added in the RulesManager control in the design view.

Through Code View

Set up the application

1. Create a new **Windows Forms App**.
2. Switch to the code view and add the following references:
    - using C1.Win.FlexGrid;
    - using C1.Win.RulesManager;
3. Initialize an instance of the **C1FlexGrid** and C1RulesManager class using the following code:

C#

```
C1FlexGrid c1FlexGrid = new C1FlexGrid();
C1RulesManager rulesManager = new C1RulesManager();
```

Bind FlexGrid to a data source

1. Set the data source for FlexGrid.

   C#

   ```
   //Set datasource for FlexGrid
           private DataTable GetDataSource()
           {
               var rs = "select * from Products;";
               var cn = GetConnectionString();
               var da = new OleDbDataAdapter(rs, cn);
               var dt = new DataTable();
               da.Fill(dt);
               return dt;
           }

           static string GetConnectionString()
           {
               var path =
   Environment.GetFolderPath(Environment.SpecialFolder.Personal) + @"\ComponentOne
   Samples\Common";
               var conn = @"provider=microsoft.jet.oledb.4.0;data source=
   {0}\c1nwind.mdb;";
               return string.Format(conn, path);
           }
   ```

2. Assign data source to the FlexGrid control.

   C#

   ```
   flexGrid.DataSource = GetDataSource();
   ```

Integrate RulesManager with FlexGrid

To integrate RulesManager with FlexGrid, use SetC1RulesManager method of the RulesManager class as shown in the following code:

C#

```
rulesManager.SetC1RulesManager(flexGrid, rulesManager);
```

Apply conditional formatting

1. Create a method, say ApplyPredefinedRules, to add rules to RulesManager which are to applied to the FlexGrid control and define the rule in RulesManager, using the Name, Expression and Style properties of the Rule class. In the following example, rule applies to "UnitsInStock" column where the fore color of the cell is set to White and the back color is set to Green if the value in the cells of the "UnitsInStock" column is greater than 20.

   C#

   ```
   private void ApplyPredefinedRules()
           {
               //Define the rule in RulesManager
   ```

```
            var rule1 = new C1.Win.RulesManager.Rule()
            {
                Name = "Many In Stock",
                Expression = "= [UnitsInStock] > 20",
                Style = new ItemStyle()
                {
                    ForeColor = Color.White,
                    BackColor = Color.Green
                }
            };
```

2. Add the "UnitsInStock" column to which the rule applies using the FieldRange class.

C#

```
rule1.AppliesTo.Add(new FieldRange(new string[] { "UnitsInStock" }));
```

3. Add the rule to the RuleManager.

C#

```
rulesManager.Rules.Add(rule1);
```

4. Invoke the method used to add rules to RulesMangager and apply to FlexGrid in the Form's Load method.

C#

```
ApplyPredefinedRules();
```

条件付き書式ルールの管理

RulesManager allows you to perform many actions to manage the rules at runtime as well as through code. These actions include creating, editing and deleting rules as discussed below.

Create Rule

You can create and add a rule to the RulesManager programmatically and at runtime.

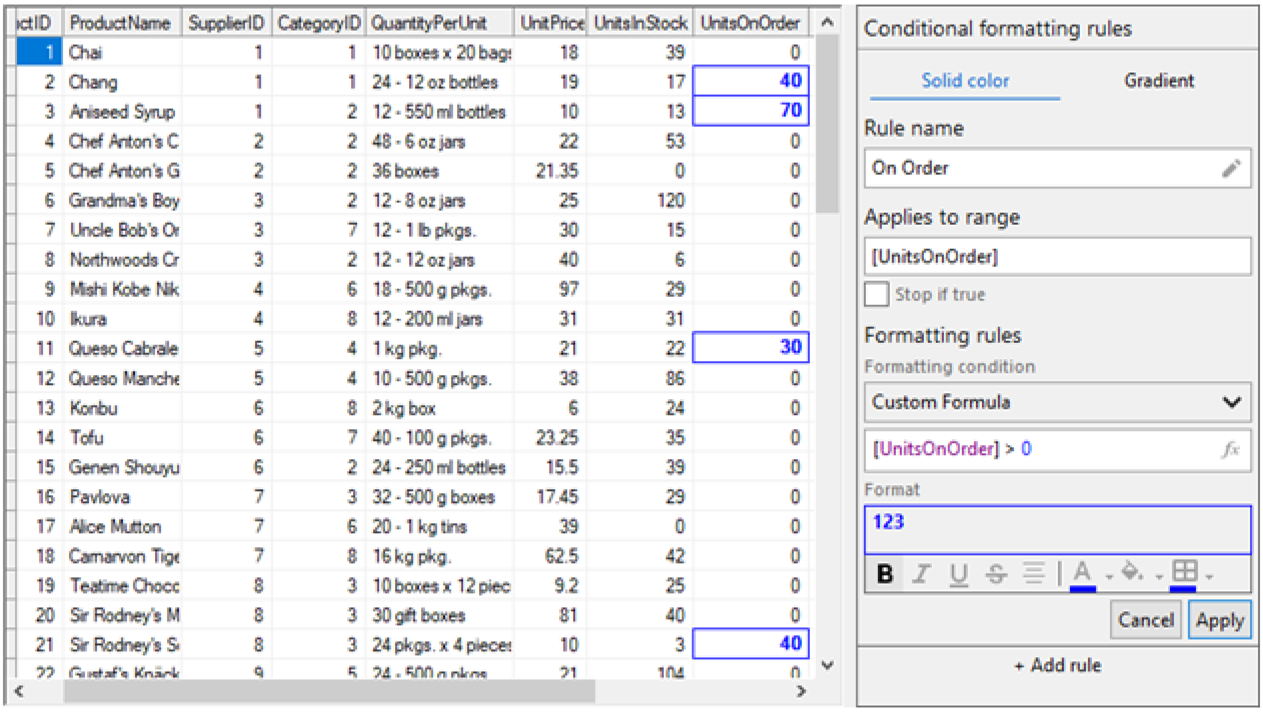
To create a rule programmatically, use the following code. This example uses the same data source which is configured in Quick Start.

C#

```
var rule1 = new C1.Win.RulesManager.Rule()
{
      Name = "On Order",
      Expression = "= [UnitsOnOrder] > 0",
      Style = new ItemStyle()
      {
          ForeColor = Color.DarkBlue,
          BorderColor = Color.DarkBlue,
          FontStyle = FontStyle.Bold
      }
};
```

The above code defines a rule in RulesManager using the Name, Expression and Style property of the Rule class. This rule applies to "UnitsOnOrder" column where the condition is that the fore color and border color of the cells the cells with value greater than 0 will be set to DarkBlue and Font style will be set to Bold.

Similarly, you can create a rule at runtime. The following image shows the conditional formatting rules set in the RulesManager control. Here, rule applies to the "UnitsOnOrder" column where the condition is that the fore color and border color of the cells the cells with value greater than 0 will be set to DarkBlue and Font style will be set to Bold.



ctID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder
1	Chai	1	1	10 boxes x 20 bag:	18	39	0
2	Chang	1	1	24 - 12 oz bottles	19	17	**40**
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13	**70**
4	Chef Anton's C	2	2	48 - 6 oz jars	22	53	0
5	Chef Anton's G	2	2	36 boxes	21.35	0	0
6	Grandma's Boy	3	2	12 - 8 oz jars	25	120	0
7	Uncle Bob's Or	3	7	12 - 1 lb pkgs.	30	15	0
8	Northwoods Cr	3	2	12 - 12 oz jars	40	6	0
9	Mishi Kobe Nik	4	6	18 - 500 g pkgs.	97	29	0
10	Ikura	4	8	12 - 200 ml jars	31	31	0
11	Queso Cabrale	5	4	1 kg pkg.	21	22	**30**
12	Queso Manche	5	4	10 - 500 g pkgs.	38	86	0
13	Konbu	6	8	2 kg box	6	24	0
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35	0
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39	0
16	Pavlova	7	3	32 - 500 g boxes	17.45	29	0
17	Alice Mutton	7	6	20 - 1 kg tins	39	0	0
18	Carnarvon Tige	7	8	16 kg pkg.	62.5	42	0
19	Teatime Chocc	8	3	10 boxes x 12 piec	9.2	25	0
20	Sir Rodney's M	8	3	30 gift boxes	81	40	0
21	Sir Rodney's S	8	3	24 pkgs. x 4 piece:	10	3	**40**
22	Gustaf's Knäck	9	5	24 - 500 g pkgs	21	104	0

Conditional formatting rules

Rule name: On Order

Applies to range: [UnitsOnOrder]

□ Stop if true

Formatting rules

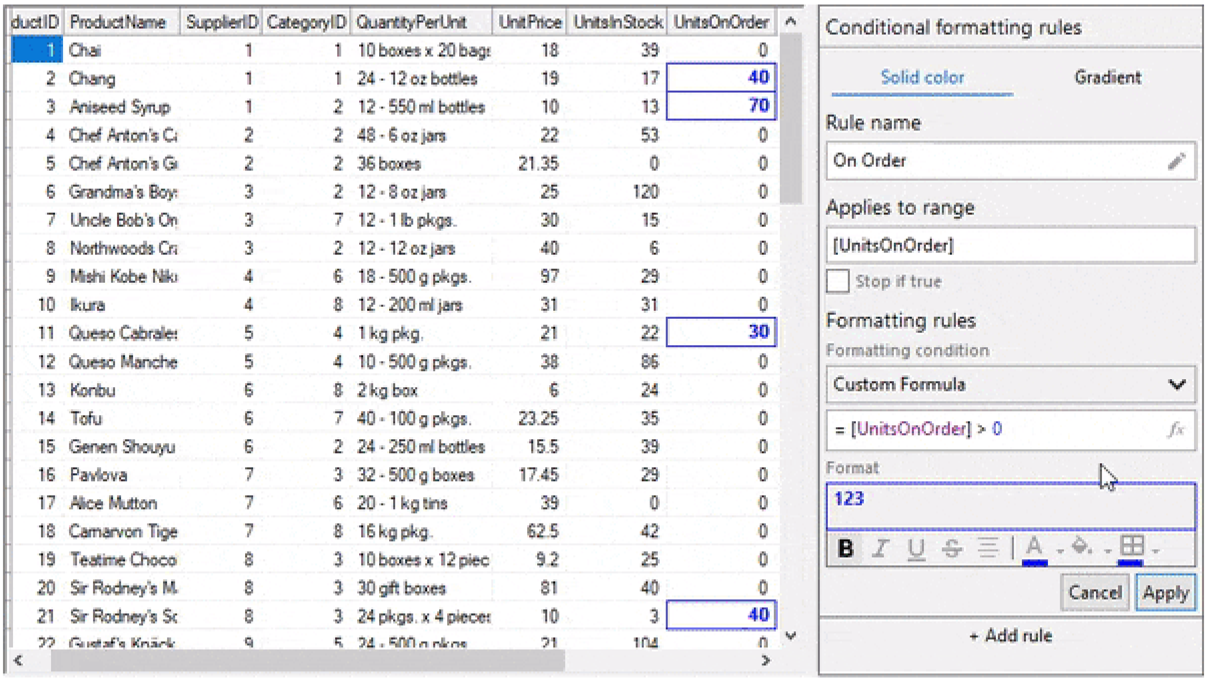
Formatting condition: Custom Formula

[UnitsOnOrder] > 0

Format: 123

Edit Rule

There can be a scenario where you might want to modify an existing rule. This can be done by changing the range, style or condition. The following GIF shows how you can modify a rule by changing the style of an existing rule at runtime.



ductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder
1	Chai	1	1	10 boxes x 20 bag:	18	39	0
2	Chang	1	1	24 - 12 oz bottles	19	17	40
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13	70
4	Chef Anton's Ca	2	2	48 - 6 oz jars	22	53	0
5	Chef Anton's Gi	2	2	36 boxes	21.35	0	0
6	Grandma's Boy:	3	2	12 - 8 oz jars	25	120	0
7	Uncle Bob's On	3	7	12 - 1 lb pkgs.	30	15	0
8	Northwoods Cra	3	2	12 - 12 oz jars	40	6	0
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	29	0
10	Ikura	4	8	12 - 200 ml jars	31	31	0
11	Queso Cabrale:	5	4	1 kg pkg.	21	22	30
12	Queso Manche	5	4	10 - 500 g pkgs.	38	86	0
13	Konbu	6	8	2 kg box	6	24	0
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35	0
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39	0
16	Pavlova	7	3	32 - 500 g boxes	17.45	29	0
17	Alice Mutton	7	6	20 - 1 kg tins	39	0	0
18	Carnarvon Tige	7	8	16 kg pkg.	62.5	42	0
19	Teatime Choco	8	3	10 boxes x 12 piec	9.2	25	0
20	Sir Rodney's M	8	3	30 gift boxes	81	40	0
21	Sir Rodney's Sc	8	3	24 pkgs. x 4 piece:	10	3	40
22	Gustaf's Knäck	9	5	24 - 500 g pkgs.	21	104	0

Conditional formatting rules

Rule name: On Order

Applies to range: [UnitsOnOrder]

☐ Stop if true

Formatting rules

Formatting condition: Custom Formula

= [UnitsOnOrder] > 0

Format: 123

Delete Rule

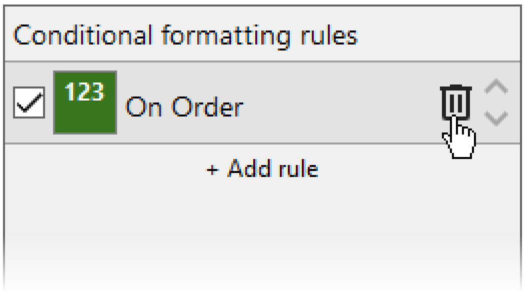
RulesManager also allows you to delete a rule programmatically and at runtime using the Remove method.

To delete a rule programmatically, use the following code.

C#

```
rulesManager.Rules.Remove(rule1);
```

Similarly, you can delete a rule at runtime by clicking the "bin" icon appearing next to the created rule.



Conditional formatting rules

☑ 123 On Order

書式ルールの適用

With Rules Manager, you can apply the conditional formatting rules to each element of a grid, be it a cell, row or column. You can also apply conditional formatting to a cell range or to the complete grid to change its appearance based on the conditions you specify.

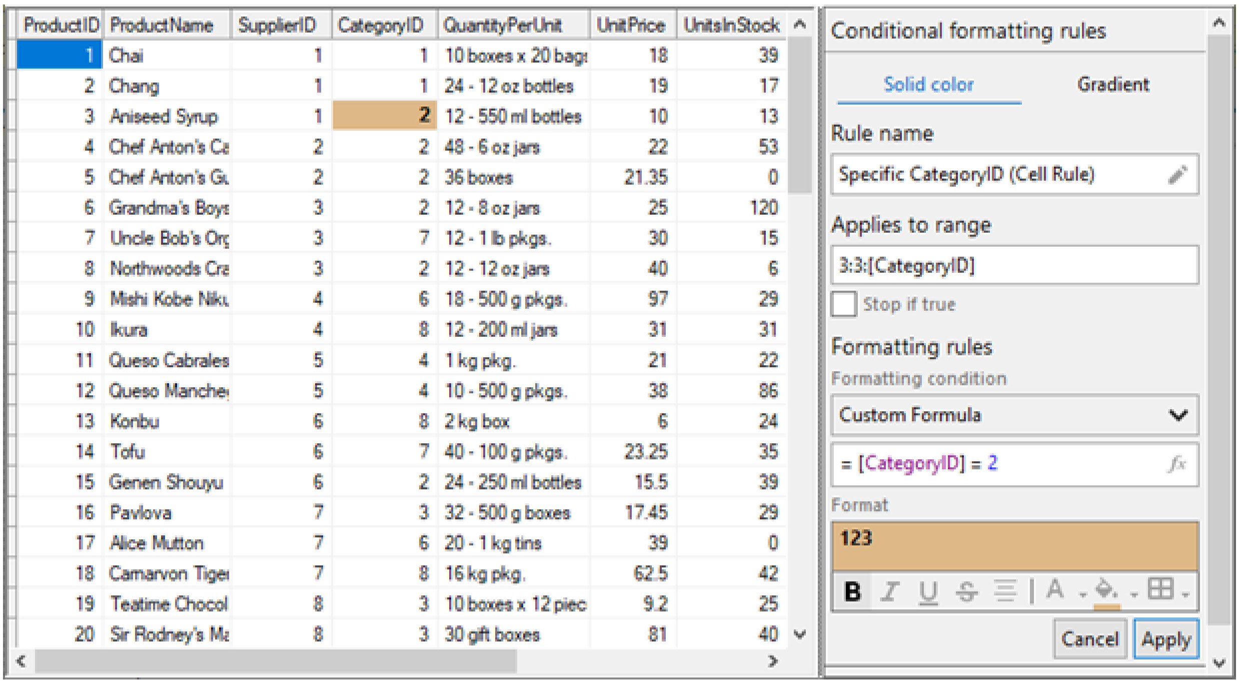
Apply Rules on a Cell

Rules Manager allows you to apply conditional formatting rules on a cell programmatically and at runtime. To apply the formatting rules programmatically, use the following code. This example uses the same data source which is configured in Quick Start. In this example, we create a rule wherein the conditional formatting is applied on a cell, D4, from the CategoryID column. Here, the specified cell from the categoryID column is added to the rule using the CustomItemRange class.

C# copyCode

```
public void ApplyCellRule()
{
    //単一のセル(D4)に書式ルールを適用します。
    var cellRule = new C1.Win.RulesManager.Rule()
    {
        Name = "Specific CategoryID (Cell Rule)",
        Expression = "= [CategoryID] = 2",
        Style = new ItemStyle()
        {
            BackColor = Color.BurlyWood,
            FontStyle = FontStyle.Bold
        }
    };
    cellRule.AppliesTo.Add(new CustomItemRange(3, 3, new string[] { "CategoryID" }));
    //RuleManagerにルールを追加します。
    c1RulesManager1.Rules.Add(cellRule);
}
```

The same formatting can be applied to the cell at runtime. The following image shows the above conditional formatting rule set in the RulesManager control at runtime.



ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
1	Chai	1	1	10 boxes x 20 bag	18	39
2	Chang	1	1	24 - 12 oz bottles	19	17
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13
4	Chef Anton's Ca	2	2	48 - 6 oz jars	22	53
5	Chef Anton's Gu	2	2	36 boxes	21.35	0
6	Grandma's Boys	3	2	12 - 8 oz jars	25	120
7	Uncle Bob's Org	3	7	12 - 1 lb pkgs.	30	15
8	Northwoods Cra	3	2	12 - 12 oz jars	40	6
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	29
10	Ikura	4	8	12 - 200 ml jars	31	31
11	Queso Cabrales	5	4	1 kg pkg.	21	22
12	Queso Manche	5	4	10 - 500 g pkgs.	38	86
13	Konbu	6	8	2 kg box	6	24
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39
16	Pavlova	7	3	32 - 500 g boxes	17.45	29
17	Alice Mutton	7	6	20 - 1 kg tins	39	0
18	Carnarvon Tiger	7	8	16 kg pkg.	62.5	42
19	Teatime Chocol	8	3	10 boxes x 12 piec	9.2	25
20	Sir Rodney's Ma	8	3	30 gift boxes	81	40

Conditional formatting rules

Rule name: Specific CategoryID (Cell Rule)

Applies to range: 3:3:[CategoryID]

☐ Stop if true

Formatting rules

Formatting condition: Custom Formula

= [CategoryID] = 2

Format: 123

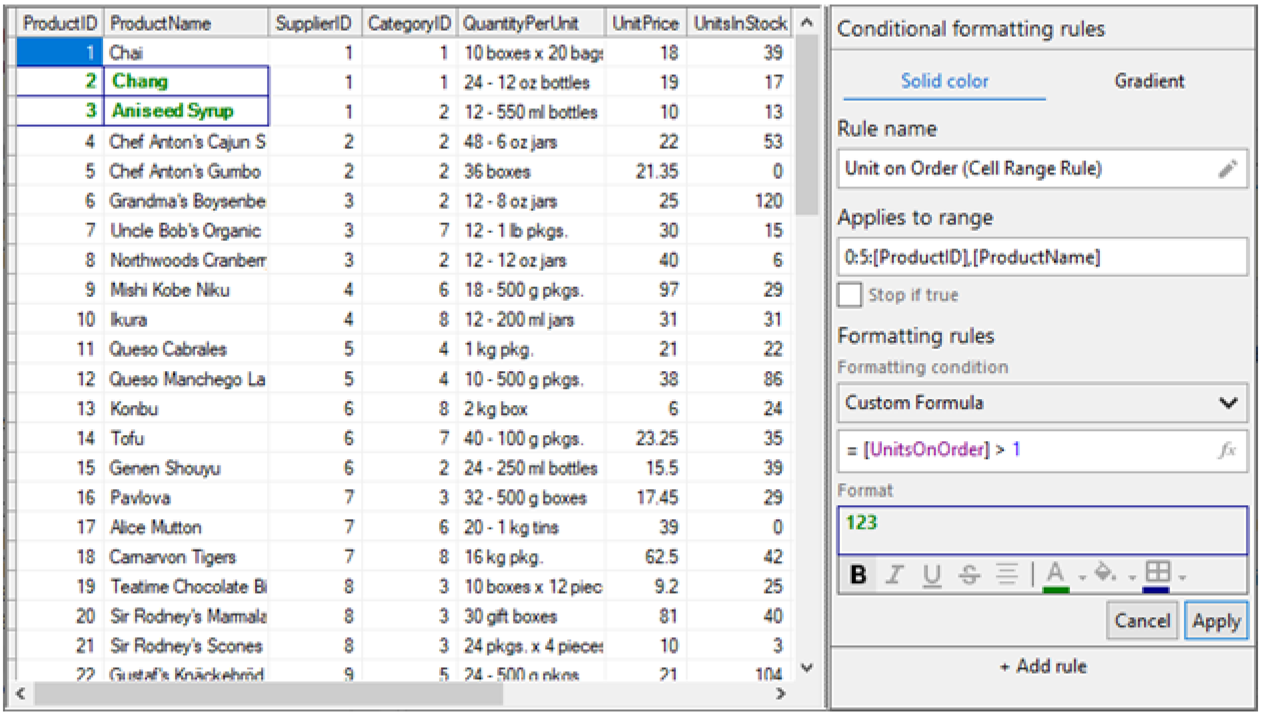
Apply Rules on a Cell Range

Using Rules Manager, you can apply conditional formatting rules on a cell range programmatically and at runtime. To apply the formatting rules programmatically, use the following code. This example uses the same data source which is configured in Quick Start. In this example, we create a rule wherein the conditional formatting is applied on a cell range that includes first two columns and first five rows where the value in UnitsOnOrder column is greater than 1. Here, the cell range is added to the rule using the **CustomItemRange** class.

C# copyCode

```
public void ApplyCellRangeRule()
{
    //最初の2列と最初の5行を含むセル範囲に書式設定のルールを適用します。
    var cellRangeRule = new C1.Win.RulesManager.Rule()
    {
        Name = "Unit on Order (Cell Range Rule)",
        Expression = "= [UnitsOnOrder] > 1",
        Style = new ItemStyle()
        {
            ForeColor = Color.Green,
            BorderColor = Color.DarkBlue,
            FontStyle = FontStyle.Bold
        }
    };
    cellRangeRule.AppliesTo.Add(new CustomItemRange(0, 5, new string[] { "ProductID",
"ProductName" }));
    //RuleManagerにルールを追加します。
    c1RulesManager1.Rules.Add(cellRangeRule);
}
```

The same formatting can be applied to the cell range at runtime as well. The following image shows the above conditional formatting rule set in the RulesManager control at runtime.



ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
1	Chai	1	1	10 boxes x 20 bag:	18	39
2	Chang	1	1	24 - 12 oz bottles	19	17
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13
4	Chef Anton's Cajun S	2	2	48 - 6 oz jars	22	53
5	Chef Anton's Gumbo	2	2	36 boxes	21.35	0
6	Grandma's Boysenbe	3	2	12 - 8 oz jars	25	120
7	Uncle Bob's Organic	3	7	12 - 1 lb pkgs.	30	15
8	Northwoods Cranberr	3	2	12 - 12 oz jars	40	6
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	29
10	Ikura	4	8	12 - 200 ml jars	31	31
11	Queso Cabrales	5	4	1 kg pkg.	21	22
12	Queso Manchego La	5	4	10 - 500 g pkgs.	38	86
13	Konbu	6	8	2 kg box	6	24
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39
16	Pavlova	7	3	32 - 500 g boxes	17.45	29
17	Alice Mutton	7	6	20 - 1 kg tins	39	0
18	Camarvon Tigers	7	8	16 kg pkg.	62.5	42
19	Teatime Chocolate Bi	8	3	10 boxes x 12 piec	9.2	25
20	Sir Rodney's Marmala	8	3	30 gift boxes	81	40
21	Sir Rodney's Scones	8	3	24 pkgs. x 4 pieces	10	3
22	Gustaf's Knäckebröd	9	5	24 - 500 g pkgs.	21	104

Conditional formatting rules

Rule name: Unit on Order (Cell Range Rule)

Applies to range: 0:5:[ProductID],[ProductName]

☐ Stop if true

Formatting rules

Formatting condition: Custom Formula

= [UnitsOnOrder] > 1

Format: 123

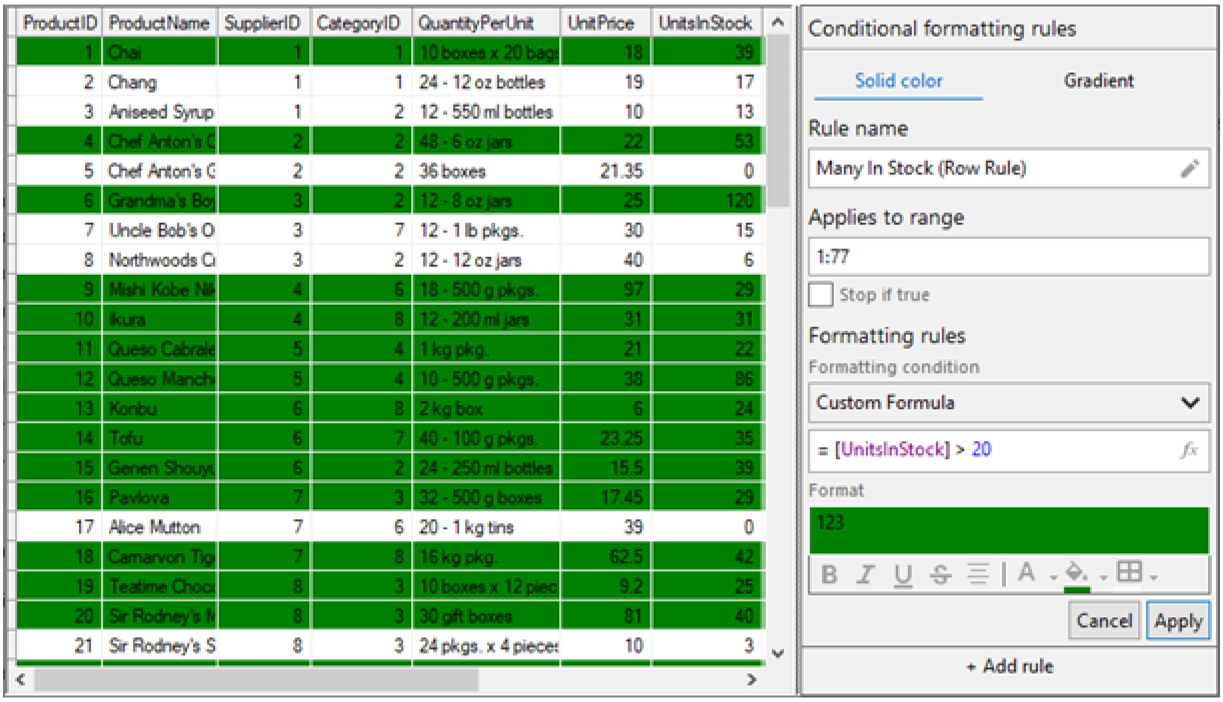
Apply Rules on a Row

You can apply conditional formatting rules on a row programmatically and at runtime. To programmatically apply the formatting rules on the rows, use the following code. This example uses the same data source which is configured in Quick Start. In this example, the conditional formatting is applied to all the rows in FlexGrid where the value in UnitsInStock column is greater than 20. Here, all the rows are added to the rule using the ItemRange class.

C# copyCode

```
public void ApplyRowRule()
{
    //FlexGridのすべての行に書式設定のルールを適用します。
    var rowRule = new C1.Win.RulesManager.Rule()
    {
        Name = "Many In Stock (Row Rule)",
        Expression = "= [UnitsInStock] > 20",
        Style = new ItemStyle()
        {
            BorderColor = Color.White,
            BackColor = Color.Green
        }
    };
    rowRule.AppliesTo.Add(new ItemRange(c1FlexGrid1.Rows.Fixed,
c1FlexGrid1.Rows.Count - 1));
    //RuleManagerに行のルールを追加します。
    c1RulesManager1.Rules.Add(rowRule);
}
```

The same formatting can be applied to the rows at runtime. The following image shows the above conditional formatting rule set in the RulesManager control at runtime.



ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
1	Chai	1	1	10 boxes x 20 bag	18	39
2	Chang	1	1	24 - 12 oz bottles	19	17
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13
4	Chef Anton's C	2	2	48 - 6 oz jars	22	53
5	Chef Anton's G	2	2	36 boxes	21.35	0
6	Grandma's Boy	3	2	12 - 8 oz jars	25	120
7	Uncle Bob's O	3	7	12 - 1 lb pkgs.	30	15
8	Northwoods C	3	2	12 - 12 oz jars	40	6
9	Mishi Kobe Ni	4	6	18 - 500 g pkgs.	97	29
10	Ikura	4	8	12 - 200 ml jars	31	31
11	Queso Cabrale	5	4	1 kg pkg.	21	22
12	Queso Manch	5	4	10 - 500 g pkgs.	38	86
13	Konbu	6	8	2 kg box	6	24
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39
16	Pavlova	7	3	32 - 500 g boxes	17.45	29
17	Alice Mutton	7	6	20 - 1 kg tins	39	0
18	Carnarvon Tig	7	8	16 kg pkg.	62.5	42
19	Teatime Choc	8	3	10 boxes x 12 piec	9.2	25
20	Sir Rodney's M	8	3	30 gift boxes	81	40
21	Sir Rodney's S	8	3	24 pkgs. x 4 pieces	10	3

Conditional formatting rules

Rule name: Many In Stock (Row Rule)

Applies to range: 1:77

□ Stop if true

Formatting rules

Formatting condition: Custom Formula

= [UnitsInStock] > 20

Format: 123

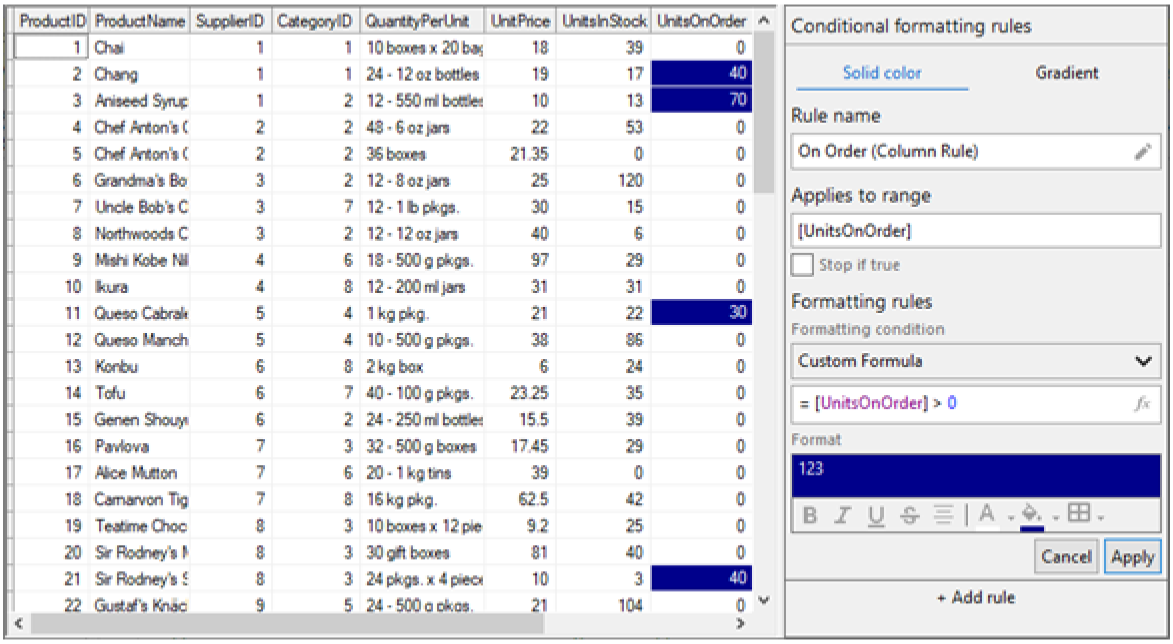
Apply Rules on a Column

With Rules Manager, you can apply conditional formatting rules on a column programmatically and at runtime. To programmatically apply the formatting rules on a column, say UnitsOnOrder, use the following code. This example uses the same data source which is configured in Quick Start. In this example, the conditional formatting is applied on the values in UnitsOnOrder column which are greater than 0. Here, this column is added to the rule using the FieldRange class.

C# copyCode

```
public void ApplyColumnRule()
{
    //「UnitsOnOrder」列に書式設定のルールを適用します。
    var columnRule = new C1.Win.RulesManager.Rule()
    {
        Name = "On Order (Column Rule)",
        Expression = "= [UnitsOnOrder] > 0",
        Style = new ItemStyle()
        {
            ForeColor = Color.White,
            BackColor = Color.DarkBlue
        }
    };
    columnRule.AppliesTo.Add(new FieldRange(new string[] { "UnitsOnOrder" }));
    //RuleManagerに列のルールを追加します。
    c1RulesManager1.Rules.Add(columnRule);
}
```

The same formatting can be applied to a column at runtime. The following image shows the above conditional formatting rule set in the RulesManager control at runtime.



ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder
1	Chai	1	1	10 boxes x 20 ba	18	39	0
2	Chang	1	1	24 - 12 oz bottles	19	17	40
3	Aniseed Syru	1	2	12 - 550 ml bottle	10	13	70
4	Chef Anton's (	2	2	48 - 6 oz jars	22	53	0
5	Chef Anton's (	2	2	36 boxes	21.35	0	0
6	Grandma's Bo	3	2	12 - 8 oz jars	25	120	0
7	Uncle Bob's C	3	7	12 - 1 lb pkgs.	30	15	0
8	Northwoods C	3	2	12 - 12 oz jars	40	6	0
9	Mishi Kobe Ni	4	6	18 - 500 g pkgs.	97	29	0
10	Ikura	4	8	12 - 200 ml jars	31	31	0
11	Queso Cabral	5	4	1 kg pkg.	21	22	30
12	Queso Manch	5	4	10 - 500 g pkgs.	38	86	0
13	Konbu	6	8	2 kg box	6	24	0
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35	0
15	Genen Shouy	6	2	24 - 250 ml bottle	15.5	39	0
16	Pavlova	7	3	32 - 500 g boxes	17.45	29	0
17	Alice Mutton	7	6	20 - 1 kg tins	39	0	0
18	Carnarvon Tig	7	8	16 kg pkg.	62.5	42	0
19	Teatime Choc	8	3	10 boxes x 12 pie	9.2	25	0
20	Sir Rodney's I	8	3	30 gift boxes	81	40	0
21	Sir Rodney's S	8	3	24 pkgs. x 4 piec	10	3	40
22	Gustaf's Knäc	9	5	24 - 500 g pkgs.	21	104	0

Conditional formatting rules

Solid color Gradient

Rule name

On Order (Column Rule)

Applies to range

[UnitsOnOrder]

□ Stop if true

Formatting rules

Formatting condition

Custom Formula

= [UnitsOnOrder] > 0

Format

123

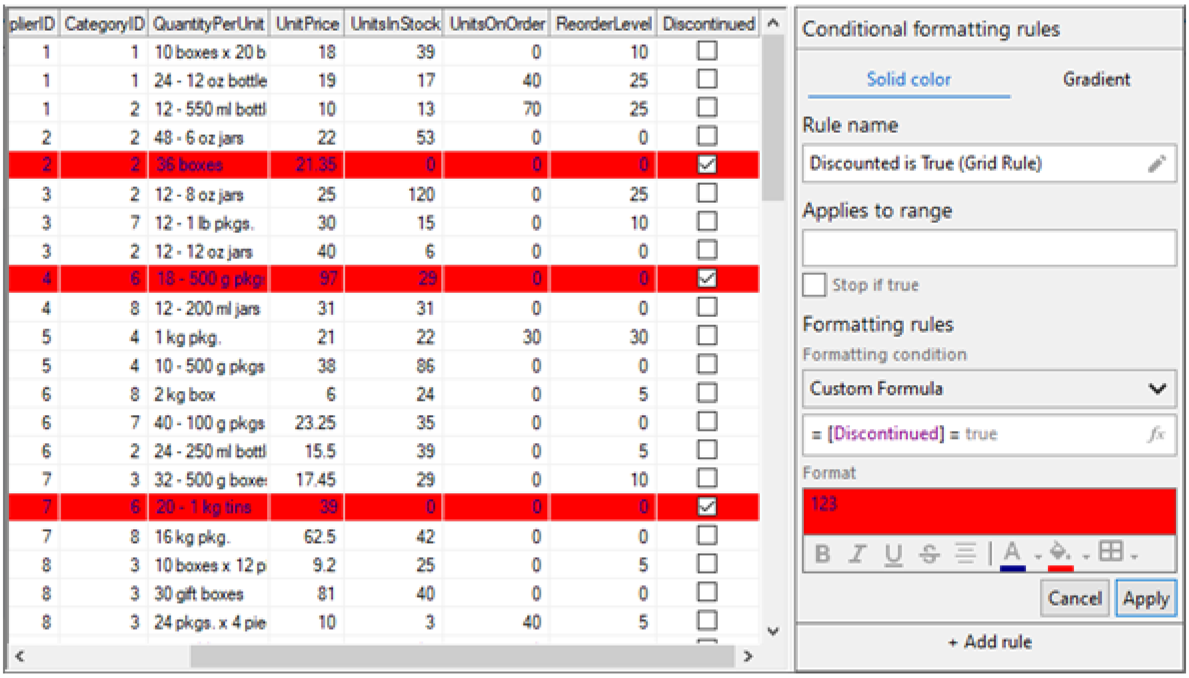
Apply Rules on the Grid

Using Rules Manager, you can apply conditional formatting rules on the complete grid programmatically and at runtime. To apply the formatting rules on the grid programmatically, use the following code. This example uses the same data source which is configured in Quick Start. In this example, we create a rule wherein the conditional formatting is applied to the complete grid where the value of Discontinued products is true.

C# copyCode

```
public void ApplyGridRule()
{
    //FlexGrid全体(完全な行とすべての列)に書式設定のルールを適用します。
    var gridRule = new C1.Win.RulesManager.Rule()
    {
        Name = "Discounted is True (Grid Rule)",
        Expression = "= [Discontinued] = true",
        Style = new ItemStyle()
        {
            BackColor = Color.Red,
            ForeColor = Color.DarkBlue
        }
    };
    //RuleManagerにルールを追加します。
    c1RulesManager1.Rules.Add(gridRule);
}
```

The same formatting can be applied to the complete grid at runtime. The following image shows the above conditional formatting rule set in the RulesManager control at runtime.



plierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	1	10 boxes x 20 b	18	39	0	10	□
1	1	24 - 12 oz bottle	19	17	40	25	□
1	2	12 - 550 ml bottl	10	13	70	25	□
2	2	48 - 6 oz jars	22	53	0	0	□
2	2	36 boxes	21.35	0	0	0	☑
3	2	12 - 8 oz jars	25	120	0	25	□
3	7	12 - 1 lb pkgs.	30	15	0	10	□
3	2	12 - 12 oz jars	40	6	0	0	□
4	6	18 - 500 g pkgs	97	29	0	0	☑
4	8	12 - 200 ml jars	31	31	0	0	□
5	4	1 kg pkg.	21	22	30	30	□
5	4	10 - 500 g pkgs	38	86	0	0	□
6	8	2 kg box	6	24	0	5	□
6	7	40 - 100 g pkgs	23.25	35	0	0	□
6	2	24 - 250 ml bottl	15.5	39	0	5	□
7	3	32 - 500 g boxe:	17.45	29	0	10	□
7	6	20 - 1 kg tins	39	0	0	0	☑
7	8	16 kg pkg.	62.5	42	0	0	□
8	3	10 boxes x 12 p	9.2	25	0	5	□
8	3	30 gift boxes	81	40	0	0	□
8	3	24 pkgs. x 4 pie	10	3	40	5	□

Conditional formatting rules

Rule name: Discounted is True (Grid Rule)

Applies to range:

□ Stop if true

Formatting rules

Formatting condition: Custom Formula

= [Discontinued] = true

Format: 123