Reports Dynamic Cell Colors Code

Color report cells dynamically.

This page presents information about how to use dynamic cell colors in reports.

- Dynamic Cell Colors
- How to use
  - Creating Callback Method
  - Callback Method Usage
  - Report with Colored DataGrid
  - Additional Information

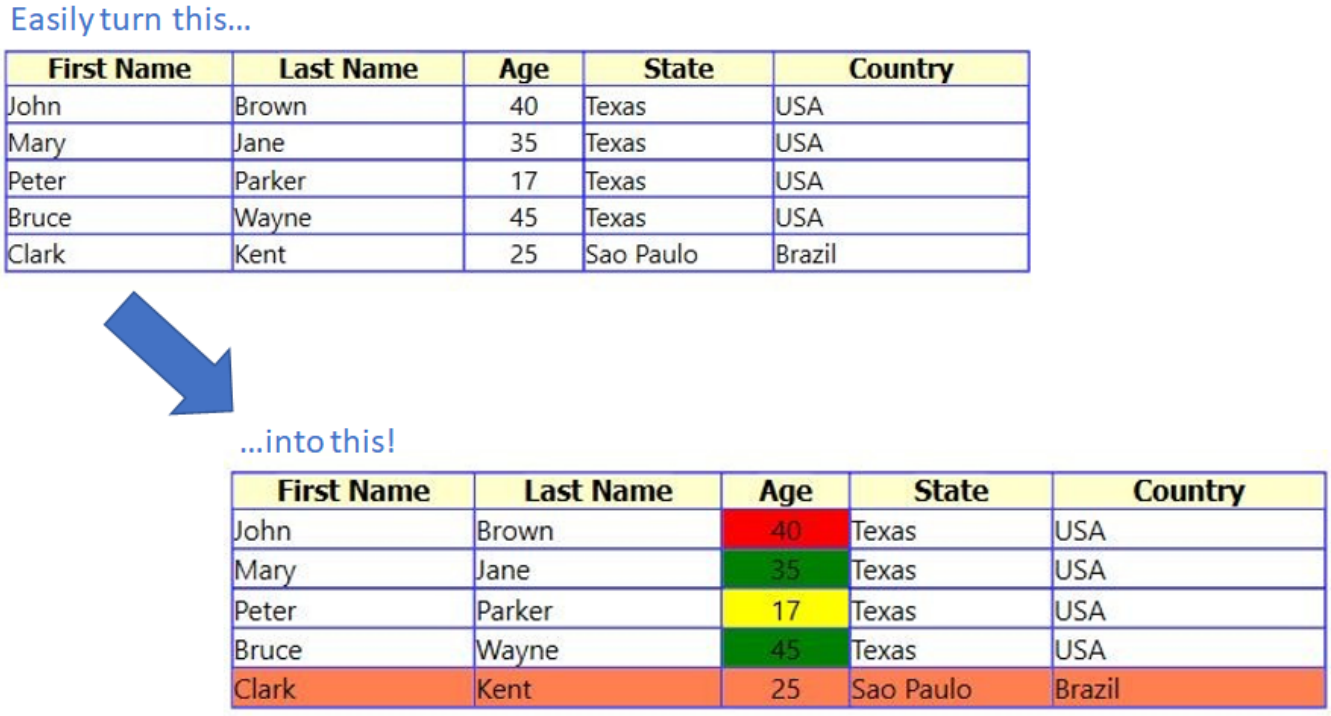
Dynamic Cell Colors

Improve your reports with Dynamic Cell Colors. Calculated at runtime, you can vary the cell color of your report tables, making them deliver even more value faster.

This is great for regulatory reporting as well as a quick visual history of when values were in or out of acceptable ranges.

Have you ever wished you could quickly spot areas that need attention in the shift report? Now you can!

Spend more time fixing issues and celebrating wins instead of staring at numbers trying to figure out what is great, or not-so-great, about that last run.



How to use

Creating Callback Method

Table cells can be dynamically colored by using the input parameters of the Callback Method. You can filter tables by **ColumnName** and **Rows**.

To use this feature, you will need to add a callback method in the **ClientMain** class that is called every time the DataGrid is modified.

The function has the following syntax:

```
public void OnReportCustomTableCell(string reportName, string columnName, System.Data.DataRow row, System.
Windows.Documents.TableCell tableCell)
{
// Insert Code Here
}
```

In this example, we will add the following table to our report:

First Name	Last Name	Age	State	Country
John	Brown	40	Texas	USA
Mary	Jane	35	Texas	USA
Peter	Parker	17	Texas	USA
Bruce	Wayne	45	Texas	USA
Clark	Kent	25	Sao Paulo	Brazil

Callback Method Usage

The code below shows an example of how this feature can be implemented.

```
public void OnReportCustomTableCell(string reportName, string columnName, System.Data.DataRow row, System.
Windows.Documents.TableCell tableCell)
{
if (columnName == "Age")
{
if (TConvert.To<int>(row["Age"]) == 40)
tableCell.Background = Brushes.Red;

else if (TConvert.To<int>(row["Age"]) < 30)
tableCell.Background = Brushes.Yellow;

else
tableCell.Background = Brushes.Green;
}

if (row["Country"].ToString() == "Brazil")
tableCell.Background = Brushes.Coral;
}
```

The result achieved by the example code is presented in the image below.

First Name	Last Name	Age	State	Country
John	Brown	40	Texas	USA
Mary	Jane	35	Texas	USA
Peter	Parker	17	Texas	USA
Bruce	Wayne	45	Texas	USA
Clark	Kent	25	Sao Paulo	Brazil

The new colored DataGrid is added into the report following the same logic as before.

Report with Colored DataGrid

Insert a table with the correct number of columns and only **TWO** rows. In the first column, write the text that will be the column name in the printed report.

In the second column, write the correct column name that is in the database.

First Name	Last Name	Age	Address	State	Country
FirstName	LastName	Age	Address	State	Country

Additional Information

Since the callback function is placed into the ClientMain class, the reports containing DataGrids will only be colored if the report is being saved by one of the following methods:

- Using the SaveCommandAsync method in **CodeBehind** of a Display
- Using the SaveCommandAsync method inside a script created at Scripts -> **Tasks** with Client Domain
- Using a tag in the report **SaveTrigger** column at **Reports Forms**

Legacy | 9.2 Video Tutorial

In this section...
