

ThemesLib

Generated by Doxygen 1.9.4

1 Namespace Index	1
1.1 Package List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 Feuster Namespace Reference	7
5 Class Documentation	9
5.1 Feuster.Themes.Theme Class Reference	9
5.1.1 Detailed Description	9
5.1.2 Member Function Documentation	9
5.1.2.1 GetColor()	9
5.1.3 Property Documentation	10
5.1.3.1 ColorCount	10
5.1.3.2 ColorList	10
5.1.3.3 DarkMode	10
5.1.3.4 Id	11
5.1.3.5 Name	11
5.2 Feuster.Themes Class Reference	11
5.2.1 Detailed Description	12
5.2.2 Member Function Documentation	12
5.2.2.1 GetThemeByGradient() [1/2]	12
5.2.2.2 GetThemeByGradient() [2/2]	12
5.2.2.3 GetThemeById()	13
5.2.2.4 GetThemeByName()	13
5.2.2.5 GetThemeColor() [1/4]	14
5.2.2.6 GetThemeColor() [2/4]	14
5.2.2.7 GetThemeColor() [3/4]	15
5.2.2.8 GetThemeColor() [4/4]	15
5.2.2.9 GetThemeColorCount() [1/2]	16
5.2.2.10 GetThemeColorCount() [2/2]	16
5.2.2.11 GetThemeNames()	17
5.2.2.12 GetThemeNamesByColorCount()	17
5.2.3 Member Data Documentation	17
5.2.3.1 ThemeList	18
6 File Documentation	19
6.1 C:/Git/ThemesLib/Themes/Themes.cs File Reference	19
6.2 Themes.cs	19

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

Feuster	7
-----------------------------------	-------------------

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Feuster.Themes.Theme	9
Feuster.Themes	11

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

C:/Git/ThemesLib/Themes/[Themes.cs](#) 19

Chapter 4

Namespace Documentation

4.1 Feuster Namespace Reference

Classes

- class [Themes](#)

Chapter 5

Class Documentation

5.1 Feuster.Themes.Theme Class Reference

Public Member Functions

- Color [GetColor](#) (int Index)
Get a single color by index depending on the actual DarkMode setting

Properties

- string [Name](#) [get, set]
- int [Id](#) [get, set]
[Theme](#) internal Id
- bool [DarkMode](#) [get, set]
- int [ColorCount](#) [get, set]
Available colors in the actual theme
- Color[] [ColorList](#) = null! [get, set]
List of colors for the regular color scheme

5.1.1 Detailed Description

Definition at line 10 of file [Themes.cs](#).

5.1.2 Member Function Documentation

5.1.2.1 GetColor()

```
Color Feuster.Themes.Theme.GetColor (
    int Index )
```

Get a single color by index depending on the actual DarkMode setting

Parameters

<i>Index</i>	
--------------	--

Returns

Color

Definition at line 52 of file [Themes.cs](#).

5.1.3 Property Documentation

5.1.3.1 ColorCount

```
int Feuster.Themes.Theme.ColorCount [get], [set]
```

Available colors in the actual theme

Definition at line 40 of file [Themes.cs](#).

5.1.3.2 ColorList

```
Color [ ] Feuster.Themes.Theme.ColorList = null! [get], [set]
```

List of colors for the regular color scheme

Definition at line 45 of file [Themes.cs](#).

5.1.3.3 DarkMode

```
bool Feuster.Themes.Theme.DarkMode [get], [set]
```

Definition at line 31 of file [Themes.cs](#).

5.1.3.4 Id

```
int Feuster.Themes.Theme.Id [get], [set]
```

[Theme](#) internal Id

Definition at line 25 of file [Themes.cs](#).

5.1.3.5 Name

```
string Feuster.Themes.Theme.Name [get], [set]
```

Definition at line 16 of file [Themes.cs](#).

The documentation for this class was generated from the following file:

- C:/Git/ThemesLib/Themes/[Themes.cs](#)

5.2 Feuster.Themes Class Reference

Classes

- class [Theme](#)

Static Public Member Functions

- static [Theme](#) [GetThemeById](#) (int Id)
Get a [Theme](#) by its Id
- static [Theme](#) [GetThemeByName](#) (string Name)
Get a [Theme](#) by its Name
- static Color [GetThemeColor](#) (string ThemeName, int ColorIndex, bool DarkMode)
Get a single Color from a [Theme](#) by its Name, DarkMode setting and the index in the color list
- static Color [GetThemeColor](#) (int Themeld, int ColorIndex, bool DarkMode)
Get a single Color from a [Theme](#) by its Name, DarkMode setting and the index in the color list
- static Color [GetThemeColor](#) (int Themeld, int ColorIndex)
Get a single Color from a [Theme](#) by its Id and the index in the color list using the active DarkMode setting
- static Color [GetThemeColor](#) (string ThemeName, int ColorIndex)
Get a single Color from a [Theme](#) by its Name and the index in the color list using the active DarkMode setting
- static int [GetThemeColorCount](#) (string ThemeName)
Get the Color count from a [Theme](#) by its Name
- static int [GetThemeColorCount](#) (int Themeld)
Get the Color count from a [Theme](#) by its Id
- static List< string > [GetThemeNames](#) (bool SortAtoZ=true, bool ShowColorCount=true)
List all available [Themes](#) by Name
- static List< string > [GetThemeNamesByColorCount](#) (int ColorCount, bool ColorCountAsMin=false, bool SortAtoZ=true, bool ShowColorCount=true)
List all available [Themes](#) by Name and by their Color Count
- static [Theme](#) [GetThemeByGradient](#) (Color Start, Color End, int ColorCount, string Name="GradientTheme 1")
Generate a theme by 2 gradient colors
- static [Theme](#) [GetThemeByGradient](#) (Color Start, Color Center, Color End, string Name="GradientTheme 1")
Generate a theme by 3 gradient colors

Static Public Attributes

- static [Theme\[\] ThemeList](#)
List of all themes

5.2.1 Detailed Description

Definition at line 5 of file [Themes.cs](#).

5.2.2 Member Function Documentation

5.2.2.1 GetThemeByGradient() [1/2]

```
static Theme Feuster.Themes.GetThemeByGradient (
    Color Start,
    Color Center,
    Color End,
    string Name = "GradientTheme 1" ) [static]
```

Generate a theme by 3 gradient colors

Parameters

<i>Start</i>	Start Color of Gradient
<i>Center</i>	Center Color of Gradient
<i>End</i>	End Color of Gradient
<i>Name</i>	optional Name for the generated theme

Returns

[Theme](#)

Definition at line 1968 of file [Themes.cs](#).

5.2.2.2 GetThemeByGradient() [2/2]

```
static Theme Feuster.Themes.GetThemeByGradient (
    Color Start,
    Color End,
    int ColorCount,
    string Name = "GradientTheme 1" ) [static]
```

Generate a theme by 2 gradient colors

Parameters

<i>Start</i>	Start Color of Gradient
<i>End</i>	End Color of Gradient
<i>ColorCount</i>	Color count between 2 to 8 colors
<i>Name</i>	optional Name for the generated theme

Returns

[Theme](#)

Definition at line 1894 of file [Themes.cs](#).

5.2.2.3 GetThemeById()

```
static Theme Feuster.Themes.GetThemeById (
    int Id ) [static]
```

Get a [Theme](#) by its Id

Parameters

<i>Id</i>	Theme Id Number
-----------	---------------------------------

Returns

[Theme](#) or on error first default [Theme](#)

Definition at line 1638 of file [Themes.cs](#).

5.2.2.4 GetThemeByName()

```
static Theme Feuster.Themes.GetThemeByName (
    string Name ) [static]
```

Get a [Theme](#) by its Name

Parameters

<i>Name</i>	Theme Name String
-------------	-----------------------------------

Returns

[Theme](#) or on error first default [Theme](#)

Definition at line 1669 of file [Themes.cs](#).

5.2.2.5 GetThemeColor() [1/4]

```
static Color Feuster.Themes.GetThemeColor (
    int ThemeId,
    int ColorIndex ) [static]
```

Get a single Color from a [Theme](#) by its Id and the index in the color list using the active DarkMode setting

Parameters

<i>Themeld</i>	
<i>ColorIndex</i>	Color Index starting at 0

Returns

Requested Color value or in error case black/white depending on DarkMode setting

Definition at line 1774 of file [Themes.cs](#).

5.2.2.6 GetThemeColor() [2/4]

```
static Color Feuster.Themes.GetThemeColor (
    int ThemeId,
    int ColorIndex,
    bool DarkMode ) [static]
```

Get a single Color from a [Theme](#) by its Name, DarkMode setting and the index in the color list

Parameters

<i>Themeld</i>	
<i>ColorIndex</i>	Color Index starting at 0
<i>DarkMode</i>	DarkMode setting

Returns

Requested Color value or in error case black/white depending on DarkMode setting

Exceptions

<i>Exception</i>	Theme is null
------------------	-------------------------------

Definition at line 1750 of file [Themes.cs](#).

5.2.2.7 GetThemeColor() [3/4]

```
static Color Feuster.Themes.GetThemeColor (
    string ThemeName,
    int ColorIndex ) [static]
```

Get a single Color from a [Theme](#) by its Name and the index in the color list using the active DarkMode setting

Parameters

<i>ThemeName</i>	Theme Name String
<i>ColorIndex</i>	Color Index starting at 0

Returns

Requested Color value or in error case black/white depending on DarkMode setting

Definition at line 1786 of file [Themes.cs](#).

5.2.2.8 GetThemeColor() [4/4]

```
static Color Feuster.Themes.GetThemeColor (
    string ThemeName,
    int ColorIndex,
    bool DarkMode ) [static]
```

Get a single Color from a [Theme](#) by its Name, DarkMode setting and the index in the color list

Parameters

<i>ThemeName</i>	Theme Name String
<i>ColorIndex</i>	Color Index starting at 0
<i>DarkMode</i>	DarkMode setting

Returns

Requested Color value or in error case black/white depending on DarkMode setting

Exceptions

<i>Exception</i>	Theme is null
------------------	-------------------------------

Definition at line 1725 of file [Themes.cs](#).

5.2.2.9 GetThemeColorCount() [1/2]

```
static int Feuster.Themes.GetThemeColorCount (
    int ThemeId ) [static]
```

Get the Color count from a [Theme](#) by its Id

Parameters

<i>ThemeId</i>	Theme Id
----------------	--------------------------

Returns

Number of available Colors

Exceptions

<i>Exception</i>	Theme is null
------------------	-------------------------------

Definition at line 1825 of file [Themes.cs](#).

5.2.2.10 GetThemeColorCount() [2/2]

```
static int Feuster.Themes.GetThemeColorCount (
    string ThemeName ) [static]
```

Get the Color count from a [Theme](#) by its Name

Parameters

<i>ThemeName</i>	Theme Name String
------------------	-----------------------------------

Returns

Number of available Colors

Exceptions

<i>Exception</i>	Theme is null
------------------	-------------------------------

Definition at line 1805 of file [Themes.cs](#).

5.2.2.11 GetThemeNames()

```
static List< string > Feuster.Themes.GetThemeNames (
    bool SortAtoZ = true,
    bool ShowColorCount = true ) [static]
```

List all available [Themes](#) by Name

Parameters

<i>SortAtoZ</i>	list alphabetically sorted
<i>ColorCount</i>	include color count of the theme

Returns

String list with [Theme](#) names

Definition at line [1845](#) of file [Themes.cs](#).

5.2.2.12 GetThemeNamesByColorCount()

```
static List< string > Feuster.Themes.GetThemeNamesByColorCount (
    int ColorCount,
    bool ColorCountAsMin = false,
    bool SortAtoZ = true,
    bool ShowColorCount = true ) [static]
```

List all available [Themes](#) by Name and by their Color Count

Parameters

<i>ColorCount</i>	Color count which is needed to show the theme in the list
<i>ColorCountAsMin</i>	List also themes which have more Colors than the given count
<i>SortAtoZ</i>	list alphabetically sorted
<i>ColorCount</i>	include color count of the theme

Returns

String list with [Theme](#) names

Definition at line [1868](#) of file [Themes.cs](#).

5.2.3 Member Data Documentation

5.2.3.1 ThemeList

```
Theme [ ] Feuster.Themes.ThemeList [static]
```

List of all themes

Definition at line 78 of file [Themes.cs](#).

The documentation for this class was generated from the following file:

- C:/Git/ThemesLib/Themes/[Themes.cs](#)

Chapter 6

File Documentation

6.1 C:/Git/ThemesLib/Themes/Themes.cs File Reference

Classes

- class [Feuster.Themes](#)
- class [Feuster.Themes.Theme](#)

Namespaces

- namespace [Feuster](#)

6.2 Themes.cs

[Go to the documentation of this file.](#)

```
00001 using System.Drawing;
00002
00003 namespace Feuster
00004 {
00005     public class Themes
00006     {
00007         #region Declarations
00008         //private static bool[] Darkmode = Enumerable.Repeat(false, Indices+1).ToArray();
00009
00010         public class Theme
00011         {
00015             private string _Name = String.Empty;
00016             public string Name
00017             {
00018                 get => this._Name;
00019                 internal set => this._Name = value;
00020             }
00021
00025             public int Id { get; internal set; }
00026
00030             private bool _DarkMode = false;
00031             public bool DarkMode
00032             {
00033                 get => _DarkMode;
00034                 set => _DarkMode = value;
00035             }
00036
00040             public int ColorCount { get; internal set; }
00041
00045             public Color[] ColorList { get; internal set; } = null!;
00046
00052             public Color GetColor(int Index)
```

```

00053     {
00054         if (Index >= ColorCount)
00055             Index = ColorCount - 1;
00056         if (Index < 0)
00057             Index = 0;
00058
00059         if (this.DarkMode)
00060         {
00061             //Reverse the regular Colorscheme to get the darkest colors first
00062             Color[] ColorListDark = (Color[])ColorList.Clone();
00063             Array.Reverse(ColorListDark);
00064             return ColorListDark[Index];
00065         }
00066         else
00067             return ColorList[Index];
00068     }
00069 }
00070
00071 #endregion
00072
00073 #region Theme List
00074 // make sure to build the colors in a theme starting from the lightest colors going up to the
00075 darkest color
00076 public static Theme[] ThemeList = new Theme[128]{
00077     new Theme() {
00078         Name = "Granite 1",
00079         Id = 1,
00080         DarkMode = false,
00081         ColorCount = 4,
00082         ColorList = new Color[] {
00083             Color.FromArgb(0xFF, 0xFF, 0xFF),
00084             Color.FromArgb(0xA5, 0xDE, 0xF1),
00085             Color.FromArgb(0x70, 0xAF, 0xCE),
00086             Color.FromArgb(0x3A, 0x7B, 0x99)
00087         }
00088     },
00089     new Theme() {
00090         Name = "Lagoon 2",
00091         Id = 2,
00092         DarkMode = false,
00093         ColorCount = 4,
00094         ColorList = new Color[] {
00095             Color.FromArgb(0xFF, 0xFF, 0xFF),
00096             Color.FromArgb(0xFD, 0xD0, 0x37),
00097             Color.FromArgb(0x12, 0xAD, 0xC1),
00098             Color.FromArgb(0x02, 0x95, 0xA9)
00099         }
00100     },
00101     new Theme() {
00102         Name = "Modernist 3",
00103         Id = 3,
00104         DarkMode = true,
00105         ColorCount = 3,
00106         ColorList = new Color[] {
00107             Color.FromArgb(0xFF, 0xFF, 0xFF),
00108             Color.FromArgb(0x1F, 0x26, 0x33),
00109             Color.FromArgb(0x4E, 0x33, 0xFF)
00110         }
00111     },
00112     new Theme() {
00113         Name = "Modernist 4",
00114         Id = 4,
00115         DarkMode = false,
00116         ColorCount = 3,
00117         ColorList = new Color[] {
00118             Color.FromArgb(0xFF, 0xFF, 0xFF),
00119             Color.FromArgb(0x33, 0x26, 0x1F),
00120             Color.FromArgb(0xFF, 0x33, 0x4E)
00121         }
00122     },
00123     new Theme() {
00124         Name = "Confetti 5",
00125         Id = 5,
00126         DarkMode = false,
00127         ColorCount = 4,
00128         ColorList = new Color[] {
00129             Color.FromArgb(0xFF, 0xFF, 0xFF),
00130             Color.FromArgb(0x9B, 0xA4, 0xC9),
00131             Color.FromArgb(0xFB, 0x46, 0x53),
00132             Color.FromArgb(0x17, 0x18, 0x1C)
00133         }
00134     },
00135     new Theme() {
00136         Name = "Confetti 6",
00137         Id = 6,
00138         DarkMode = false,
00139         ColorCount = 6,
00140     }
00141 }

```



```

00142         ColorList = new Color[] {
00143             Color.FromArgb(0xFF, 0xFF, 0xFF),
00144             Color.FromArgb(0x62, 0xCE, 0x9D),
00145             Color.FromArgb(0x3C, 0x89, 0xC7),
00146             Color.FromArgb(0xE5, 0xBC, 0x18),
00147             Color.FromArgb(0xE5, 0x6E, 0x24),
00148             Color.FromArgb(0x00, 0x00, 0x00)
00149         }
00150     },
00151     new Theme() {
00152         Name = "Confetti 7",
00153         Id = 7,
00154         DarkMode = true,
00155         ColorCount = 4,
00156         ColorList = new Color[] {
00157             Color.FromArgb(0xFF, 0xFF, 0xFF),
00158             Color.FromArgb(0xA7, 0x63, 0xF6),
00159             Color.FromArgb(0x69, 0x57, 0xE7),
00160             Color.FromArgb(0x3A, 0x2D, 0x80)
00161         }
00162     },
00163     new Theme() {
00164         Name = "Dribble Shot 8",
00165         Id = 8,
00166         DarkMode = false,
00167         ColorCount = 4,
00168         ColorList = new Color[] {
00169             Color.FromArgb(0xF5, 0xF7, 0xFB),
00170             Color.FromArgb(0x3D, 0xCF, 0xC2),
00171             Color.FromArgb(0x03, 0x65, 0xD1),
00172             Color.FromArgb(0x00, 0x00, 0x00)
00173         }
00174     },
00175     new Theme() {
00176         Name = "Dribble Shot 9",
00177         Id = 9,
00178         DarkMode = false,
00179         ColorCount = 4,
00180         ColorList = new Color[] {
00181             Color.FromArgb(0xF7, 0xF7, 0xF7),
00182             Color.FromArgb(0xDE, 0xF8, 0x87),
00183             Color.FromArgb(0x6F, 0xDA, 0xE8),
00184             Color.FromArgb(0x00, 0x00, 0x00)
00185         }
00186     },
00187     new Theme() {
00188         Name = "Dribble Shot 10",
00189         Id = 10,
00190         DarkMode = true,
00191         ColorCount = 4,
00192         ColorList = new Color[] {
00193             Color.FromArgb(0xEC, 0xEC, 0xEA),
00194             Color.FromArgb(0xF8, 0x55, 0x00),
00195             Color.FromArgb(0x14, 0x3E, 0xFD),
00196             Color.FromArgb(0x2A, 0x27, 0x28)
00197         }
00198     },
00199     new Theme() {
00200         Name = "Dribble Shot 11",
00201         Id = 11,
00202         DarkMode = false,
00203         ColorCount = 4,
00204         ColorList = new Color[] {
00205             Color.FromArgb(0xF5, 0xF5, 0xF5),
00206             Color.FromArgb(0x55, 0xAB, 0x83),
00207             Color.FromArgb(0xF0, 0x49, 0x24),
00208             Color.FromArgb(0x03, 0x03, 0x03)
00209         }
00210     },
00211     new Theme() {
00212         Name = "Dribble Shot 12",
00213         Id = 12,
00214         DarkMode = false,
00215         ColorCount = 4,
00216         ColorList = new Color[] {
00217             Color.FromArgb(0xF4, 0xF8, 0xFB),
00218             Color.FromArgb(0x1A, 0x9D, 0xFD),
00219             Color.FromArgb(0x2F, 0x9B, 0xA4),
00220             Color.FromArgb(0x0D, 0x0D, 0x0D)
00221         }
00222     },
00223     new Theme() {
00224         Name = "Blue 13",
00225         Id = 13,
00226         DarkMode = false,
00227         ColorCount = 4,
00228         ColorList = new Color[] {

```

```

00229         Color.FromArgb(0xFF, 0xFF, 0xFF),
00230         Color.FromArgb(0xFF, 0xFF, 0xFF),
00231         Color.FromArgb(0xFF, 0xFF, 0xFF),
00232         Color.FromArgb(0xFF, 0xFF, 0xFF)
00233     },
00234 },
00235 new Theme() {
00236     Name = "Orange 14",
00237     Id = 14,
00238     DarkMode = false,
00239     ColorCount = 4,
00240     ColorList = new Color[] {
00241         Color.FromArgb(0xFF, 0xFF, 0xFF),
00242         Color.FromArgb(0xFF, 0xFF, 0xFF),
00243         Color.FromArgb(0xFF, 0xFF, 0xFF),
00244         Color.FromArgb(0xFF, 0xFF, 0xFF)
00245     },
00246 },
00247 new Theme() {
00248     Name = "Green 15",
00249     Id = 15,
00250     DarkMode = false,
00251     ColorCount = 4,
00252     ColorList = new Color[] {
00253         Color.FromArgb(0xFF, 0xFF, 0xFF),
00254         Color.FromArgb(0xFF, 0xFF, 0xFF),
00255         Color.FromArgb(0xFF, 0xFF, 0xFF),
00256         Color.FromArgb(0xFF, 0xFF, 0xFF)
00257     },
00258 },
00259 new Theme() {
00260     Name = "Blue 16",
00261     Id = 16,
00262     DarkMode = false,
00263     ColorCount = 4,
00264     ColorList = new Color[] {
00265         Color.FromArgb(0xFF, 0xFF, 0xFF),
00266         Color.FromArgb(0xFF, 0xFF, 0xFF),
00267         Color.FromArgb(0xFF, 0xFF, 0xFF),
00268         Color.FromArgb(0xFF, 0xFF, 0xFF)
00269     },
00270 },
00271 new Theme() {
00272     Name = "Red 17",
00273     Id = 17,
00274     DarkMode = false,
00275     ColorCount = 4,
00276     ColorList = new Color[] {
00277         Color.FromArgb(0xFF, 0xFF, 0xFF),
00278         Color.FromArgb(0xFF, 0xFF, 0xFF),
00279         Color.FromArgb(0xFF, 0xFF, 0xFF),
00280         Color.FromArgb(0xFF, 0xFF, 0xFF)
00281     },
00282 },
00283 new Theme() {
00284     Name = "Holographic",
00285     Id = 18,
00286     DarkMode = false,
00287     ColorCount = 4,
00288     ColorList = new Color[] {
00289         Color.FromArgb(0xFF, 0xFF, 0xFF),
00290         Color.FromArgb(0xFF, 0xFF, 0xFF),
00291         Color.FromArgb(0xFF, 0xFF, 0xFF),
00292         Color.FromArgb(0xFF, 0xFF, 0xFF)
00293     },
00294 },
00295 new Theme() {
00296     Name = "Purple 19",
00297     Id = 19,
00298     DarkMode = false,
00299     ColorCount = 4,
00300     ColorList = new Color[] {
00301         Color.FromArgb(0xFF, 0xFF, 0xFF),
00302         Color.FromArgb(0xFF, 0xFF, 0xFF),
00303         Color.FromArgb(0xFF, 0xFF, 0xFF),
00304         Color.FromArgb(0xFF, 0xFF, 0xFF)
00305     },
00306 },
00307 new Theme() {
00308     Name = "Space Pink 20",
00309     Id = 20,
00310     DarkMode = false,
00311     ColorCount = 4,
00312     ColorList = new Color[] {
00313         Color.FromArgb(0xFF, 0xFF, 0xFF),
00314         Color.FromArgb(0xFF, 0xFF, 0xFF),
00315         Color.FromArgb(0xFF, 0xFF, 0xFF)

```

```

00316         Color.FromArgb(0x17, 0x11, 0x20)
00317     }
00318 },
00319 new Theme() {
00320     Name = "Grey 21",
00321     Id = 21,
00322     DarkMode = false,
00323     ColorCount = 4,
00324     ColorList = new Color[] {
00325         Color.FromArgb(0xFF, 0xFF, 0xFF),
00326         Color.FromArgb(0xF2, 0xF2, 0xF2),
00327         Color.FromArgb(0xD6, 0xD6, 0xD6),
00328         Color.FromArgb(0x00, 0x00, 0x00)
00329     },
00330 },
00331 new Theme() {
00332     Name = "Lime 22",
00333     Id = 22,
00334     DarkMode = false,
00335     ColorCount = 4,
00336     ColorList = new Color[] {
00337         Color.FromArgb(0xFF, 0xFF, 0xFF),
00338         Color.FromArgb(0xE0, 0xFF, 0x00),
00339         Color.FromArgb(0xE3, 0xE3, 0xE3),
00340         Color.FromArgb(0x1E, 0x20, 0x29)
00341     },
00342 },
00343 new Theme() {
00344     Name = "Multi 23",
00345     Id = 23,
00346     DarkMode = false,
00347     ColorCount = 5,
00348     ColorList = new Color[] {
00349         Color.FromArgb(0xE5, 0xFB, 0xFD),
00350         Color.FromArgb(0xE1, 0xF8, 0xF0),
00351         Color.FromArgb(0x33, 0xBB, 0x91),
00352         Color.FromArgb(0x2B, 0xCB, 0xE3),
00353         Color.FromArgb(0x1E, 0x20, 0x29)
00354     },
00355 },
00356 new Theme() {
00357     Name = "Red 24",
00358     Id = 24,
00359     DarkMode = true,
00360     ColorCount = 4,
00361     ColorList = new Color[] {
00362         Color.FromArgb(0x19, 0x02, 0x07),
00363         Color.FromArgb(0x5A, 0x08, 0x2D),
00364         Color.FromArgb(0x9D, 0x0B, 0x28),
00365         Color.FromArgb(0xFF, 0x00, 0x4D)
00366     },
00367 },
00368 new Theme() {
00369     Name = "Brown 25",
00370     Id = 25,
00371     DarkMode = true,
00372     ColorCount = 4,
00373     ColorList = new Color[] {
00374         Color.FromArgb(0xFF, 0xA8, 0x00),
00375         Color.FromArgb(0x9D, 0x5A, 0x0B),
00376         Color.FromArgb(0x5A, 0x39, 0x08),
00377         Color.FromArgb(0x1F, 0x31, 0x01)
00378     },
00379 },
00380 new Theme() {
00381     Name = "Blue 26",
00382     Id = 26,
00383     DarkMode = true,
00384     ColorCount = 4,
00385     ColorList = new Color[] {
00386         Color.FromArgb(0x90, 0xB8, 0xF8),
00387         Color.FromArgb(0x5F, 0x85, 0xDB),
00388         Color.FromArgb(0x35, 0x39, 0x41),
00389         Color.FromArgb(0x26, 0x28, 0x2B)
00390     },
00391 },
00392 new Theme() {
00393     Name = "Blue 27",
00394     Id = 27,
00395     DarkMode = true,
00396     ColorCount = 4,
00397     ColorList = new Color[] {
00398         Color.FromArgb(0xF4, 0xFC, 0xFE),
00399         Color.FromArgb(0x0E, 0xA2, 0xF6),
00400         Color.FromArgb(0x3D, 0x3A, 0x50),
00401         Color.FromArgb(0x30, 0x30, 0x41)
00402     }

```

```

00403     },
00404     new Theme() {
00405         Name = "Green 28",
00406         Id = 28,
00407         DarkMode = true,
00408         ColorCount = 4,
00409         ColorList = new Color[] {
00410             Color.FromArgb(0xF7, 0xF7, 0xF7),
00411             Color.FromArgb(0x0E, 0xF6, 0xBE),
00412             Color.FromArgb(0x3A, 0x50, 0x4B),
00413             Color.FromArgb(0x1E, 0x25, 0x28)
00414         }
00415     },
00416     new Theme() {
00417         Name = "Purple 29",
00418         Id = 29,
00419         DarkMode = true,
00420         ColorCount = 4,
00421         ColorList = new Color[] {
00422             Color.FromArgb(0xE3, 0xA2, 0xEE),
00423             Color.FromArgb(0xAC, 0x5F, 0xDB),
00424             Color.FromArgb(0x3C, 0x35, 0x41),
00425             Color.FromArgb(0x29, 0x26, 0x2B)
00426         }
00427     },
00428     new Theme() {
00429         Name = "Yellow 30",
00430         Id = 30,
00431         DarkMode = true,
00432         ColorCount = 4,
00433         ColorList = new Color[] {
00434             Color.FromArgb(0xEE, 0xEE, 0xEE),
00435             Color.FromArgb(0xF6, 0xC9, 0x0E),
00436             Color.FromArgb(0x3A, 0x47, 0x50),
00437             Color.FromArgb(0x25, 0x2C, 0x33)
00438         }
00439     },
00440     new Theme() {
00441         Name = "Multi 31",
00442         Id = 31,
00443         DarkMode = false,
00444         ColorCount = 4,
00445         ColorList = new Color[] {
00446             Color.FromArgb(0xFF, 0xFE, 0xFD),
00447             Color.FromArgb(0xFD, 0x76, 0xCB),
00448             Color.FromArgb(0xFF, 0xAC, 0x30),
00449             Color.FromArgb(0x00, 0x00, 0x00)
00450         }
00451     },
00452     new Theme() {
00453         Name = "Multi 32",
00454         Id = 32,
00455         DarkMode = false,
00456         ColorCount = 4,
00457         ColorList = new Color[] {
00458             Color.FromArgb(0xFF, 0xFF, 0xFF),
00459             Color.FromArgb(0xA1, 0xA7, 0xC8),
00460             Color.FromArgb(0x24, 0xF9, 0xF9),
00461             Color.FromArgb(0x6B, 0x9F, 0xED)
00462         }
00463     },
00464     new Theme() {
00465         Name = "Green 33",
00466         Id = 33,
00467         DarkMode = true,
00468         ColorCount = 4,
00469         ColorList = new Color[] {
00470             Color.FromArgb(0xC0, 0xF7, 0xB7),
00471             Color.FromArgb(0x5A, 0xC9, 0x94),
00472             Color.FromArgb(0x12, 0x41, 0x43),
00473             Color.FromArgb(0x14, 0x1B, 0x29)
00474         }
00475     },
00476     new Theme() {
00477         Name = "Brown 34",
00478         Id = 34,
00479         DarkMode = true,
00480         ColorCount = 4,
00481         ColorList = new Color[] {
00482             Color.FromArgb(0xFF, 0xFF, 0xFF),
00483             Color.FromArgb(0x82, 0x52, 0x60),
00484             Color.FromArgb(0x67, 0x3D, 0x4D),
00485             Color.FromArgb(0x11, 0x0D, 0x0E)
00486         }
00487     },
00488     new Theme() {
00489         Name = "Multi 35",

```

```

00490         Id = 35,
00491         DarkMode = false,
00492         ColorCount = 4,
00493         ColorList = new Color[] {
00494             Color.FromArgb(0xFF, 0xFF, 0xFF),
00495             Color.FromArgb(0x2E, 0xE0, 0x9A),
00496             Color.FromArgb(0xEF, 0x84, 0xF5),
00497             Color.FromArgb(0x15, 0x26, 0x5C)
00498         },
00499     },
00500     new Theme() {
00501         Name = "Multi 36",
00502         Id = 36,
00503         DarkMode = false,
00504         ColorCount = 5,
00505         ColorList = new Color[] {
00506             Color.FromArgb(0xF6, 0xF6, 0xF6),
00507             Color.FromArgb(0x6C, 0xC1, 0x64),
00508             Color.FromArgb(0xFF, 0x55, 0x45),
00509             Color.FromArgb(0x69, 0xBC, 0xEC),
00510             Color.FromArgb(0x00, 0x00, 0x00)
00511         },
00512     },
00513     new Theme() {
00514         Name = "Multi 37",
00515         Id = 37,
00516         DarkMode = false,
00517         ColorCount = 5,
00518         ColorList = new Color[] {
00519             Color.FromArgb(0xFF, 0xFF, 0xFF),
00520             Color.FromArgb(0x7E, 0xE7, 0xD6),
00521             Color.FromArgb(0xF9, 0xF4, 0x50),
00522             Color.FromArgb(0xC1, 0xBF, 0xFA),
00523             Color.FromArgb(0x1D, 0x1D, 0x1D)
00524         },
00525     },
00526     new Theme() {
00527         Name = "Blue 38",
00528         Id = 38,
00529         DarkMode = false,
00530         ColorCount = 5,
00531         ColorList = new Color[] {
00532             Color.FromArgb(0xFF, 0xFF, 0xFF),
00533             Color.FromArgb(0xC5, 0xC7, 0xD3),
00534             Color.FromArgb(0x00, 0xC0, 0xF7),
00535             Color.FromArgb(0x55, 0x3A, 0xFC),
00536             Color.FromArgb(0x2A, 0x32, 0x56)
00537         },
00538     },
00539     new Theme() {
00540         Name = "Blue 39",
00541         Id = 39,
00542         DarkMode = false,
00543         ColorCount = 5,
00544         ColorList = new Color[] {
00545             Color.FromArgb(0xE5, 0xF8, 0xFC),
00546             Color.FromArgb(0xD2, 0xD5, 0xDE),
00547             Color.FromArgb(0x16, 0x9B, 0xD6),
00548             Color.FromArgb(0x3F, 0x46, 0x85),
00549             Color.FromArgb(0x1F, 0x21, 0x2A)
00550         },
00551     },
00552     new Theme() {
00553         Name = "Blue 40",
00554         Id = 40,
00555         DarkMode = false,
00556         ColorCount = 5,
00557         ColorList = new Color[] {
00558             Color.FromArgb(0xFF, 0xFF, 0xFF),
00559             Color.FromArgb(0xE3, 0xE8, 0xED),
00560             Color.FromArgb(0x13, 0x8F, 0xFF),
00561             Color.FromArgb(0x86, 0xA2, 0xB9),
00562             Color.FromArgb(0x15, 0x2C, 0x5B)
00563         },
00564     },
00565     new Theme() {
00566         Name = "Multi 41",
00567         Id = 41,
00568         DarkMode = false,
00569         ColorCount = 5,
00570         ColorList = new Color[] {
00571             Color.FromArgb(0xFF, 0xF8, 0xF6),
00572             Color.FromArgb(0xDD, 0xD5, 0xFC),
00573             Color.FromArgb(0x13, 0x8F, 0xFF),
00574             Color.FromArgb(0x86, 0xA2, 0xB9),
00575             Color.FromArgb(0x15, 0x2C, 0x5B)
00576         },
00577     },

```

```

00577     },
00578     new Theme() {
00579         Name = "Green 42",
00580         Id = 42,
00581         DarkMode = false,
00582         ColorCount = 5,
00583         ColorList = new Color[] {
00584             Color.FromArgb(0xED, 0xF5, 0xE0),
00585             Color.FromArgb(0x8D, 0xE4, 0xAF),
00586             Color.FromArgb(0x5C, 0xDB, 0x94),
00587             Color.FromArgb(0x38, 0x95, 0x83),
00588             Color.FromArgb(0x05, 0x38, 0x6B)
00589         }
00590     },
00591     new Theme() {
00592         Name = "Multi 43",
00593         Id = 43,
00594         DarkMode = true,
00595         ColorCount = 7,
00596         ColorList = new Color[] {
00597             Color.FromArgb(0xFF, 0xFF, 0xFF),
00598             Color.FromArgb(0xF5, 0x4F, 0x74),
00599             Color.FromArgb(0x99, 0x72, 0x84),
00600             Color.FromArgb(0x3E, 0x3E, 0xB3),
00601             Color.FromArgb(0x56, 0x3D, 0x67),
00602             Color.FromArgb(0x24, 0x25, 0x82),
00603             Color.FromArgb(0x00, 0x00, 0x00)
00604         }
00605     },
00606     new Theme() {
00607         Name = "Multi 44",
00608         Id = 44,
00609         DarkMode = false,
00610         ColorCount = 7,
00611         ColorList = new Color[] {
00612             Color.FromArgb(0xFF, 0xFF, 0xFF),
00613             Color.FromArgb(0xE8, 0xA8, 0x7C),
00614             Color.FromArgb(0x85, 0xCD, 0xCA),
00615             Color.FromArgb(0xD9, 0x76, 0x94),
00616             Color.FromArgb(0xE2, 0x7D, 0x5F),
00617             Color.FromArgb(0x40, 0xB3, 0xA2),
00618             Color.FromArgb(0x00, 0x00, 0x00)
00619         }
00620     },
00621     new Theme() {
00622         Name = "Grey 45",
00623         Id = 45,
00624         DarkMode = true,
00625         ColorCount = 7,
00626         ColorList = new Color[] {
00627             Color.FromArgb(0xFF, 0xFF, 0xFF),
00628             Color.FromArgb(0xB0, 0xA2, 0x95),
00629             Color.FromArgb(0x93, 0x8E, 0x94),
00630             Color.FromArgb(0x73, 0x95, 0xAE),
00631             Color.FromArgb(0x55, 0x7A, 0x95),
00632             Color.FromArgb(0x5D, 0x5C, 0x61),
00633             Color.FromArgb(0x00, 0x00, 0x00)
00634         }
00635     },
00636     new Theme() {
00637         Name = "Green 46",
00638         Id = 46,
00639         DarkMode = false,
00640         ColorCount = 5,
00641         ColorList = new Color[] {
00642             Color.FromArgb(0xC5, 0xC6, 0xC8),
00643             Color.FromArgb(0x66, 0xFC, 0xF1),
00644             Color.FromArgb(0x46, 0xA2, 0x9F),
00645             Color.FromArgb(0x20, 0x28, 0x33),
00646             Color.FromArgb(0x0B, 0x0C, 0x10)
00647         }
00648     },
00649     new Theme() {
00650         Name = "Red 47",
00651         Id = 47,
00652         DarkMode = false,
00653         ColorCount = 5,
00654         ColorList = new Color[] {
00655             Color.FromArgb(0xE3, 0xE2, 0xDE),
00656             Color.FromArgb(0xE3, 0xAF, 0xBC),
00657             Color.FromArgb(0xEE, 0x4C, 0x7D),
00658             Color.FromArgb(0x9B, 0x17, 0x50),
00659             Color.FromArgb(0x5D, 0x00, 0x1D)
00660         }
00661     },
00662     new Theme() {
00663         Name = "Multi 48",

```

```

00664         Id = 48,
00665         DarkMode = false,
00666         ColorCount = 5,
00667         ColorList = new Color[] {
00668             Color.FromArgb(0xEF, 0xE1, 0xBA),
00669             Color.FromArgb(0xC5, 0xCB, 0xE3),
00670             Color.FromArgb(0xD7, 0x99, 0x22),
00671             Color.FromArgb(0xED, 0x3B, 0x17),
00672             Color.FromArgb(0x40, 0x56, 0xA1)
00673         },
00674     },
00675     new Theme() {
00676         Name = "Green 49",
00677         Id = 49,
00678         DarkMode = false,
00679         ColorCount = 5,
00680         ColorList = new Color[] {
00681             Color.FromArgb(0xCA, 0xFA, 0xFE),
00682             Color.FromArgb(0x40, 0xEE, 0xE5),
00683             Color.FromArgb(0x96, 0xCA, 0xEF),
00684             Color.FromArgb(0x55, 0xBD, 0xCA),
00685             Color.FromArgb(0xFC, 0x44, 0x44)
00686         },
00687     },
00688     new Theme() {
00689         Name = "Dark Neon 50",
00690         Id = 50,
00691         DarkMode = true,
00692         ColorCount = 4,
00693         ColorList = new Color[] {
00694             Color.FromArgb(0xEE, 0xED, 0xF0),
00695             Color.FromArgb(0xCB, 0x69, 0xC1),
00696             Color.FromArgb(0x6C, 0x72, 0xCB),
00697             Color.FromArgb(0x17, 0x18, 0x1F)
00698         },
00699     },
00700     new Theme() {
00701         Name = "Commerce 51",
00702         Id = 51,
00703         DarkMode = false,
00704         ColorCount = 4,
00705         ColorList = new Color[] {
00706             Color.FromArgb(0xFF, 0xFF, 0xFF),
00707             Color.FromArgb(0xFA, 0xFC, 0xFE),
00708             Color.FromArgb(0x57, 0xBE, 0x6C),
00709             Color.FromArgb(0x10, 0x10, 0x10)
00710         },
00711     },
00712     new Theme() {
00713         Name = "Blue 52",
00714         Id = 52,
00715         DarkMode = false,
00716         ColorCount = 4,
00717         ColorList = new Color[] {
00718             Color.FromArgb(0xFF, 0xFF, 0xFF),
00719             Color.FromArgb(0xF8, 0xF8, 0xFA),
00720             Color.FromArgb(0x34, 0x56, 0xFF),
00721             Color.FromArgb(0x07, 0x09, 0x28)
00722         },
00723     },
00724     new Theme() {
00725         Name = "Black&White 53",
00726         Id = 53,
00727         DarkMode = false,
00728         ColorCount = 3,
00729         ColorList = new Color[] {
00730             Color.FromArgb(0xFF, 0xFF, 0xFF),
00731             Color.FromArgb(0xE8, 0xE8, 0xEA),
00732             Color.FromArgb(0x00, 0x00, 0x00)
00733         },
00734     },
00735     new Theme() {
00736         Name = "Orange 54",
00737         Id = 54,
00738         DarkMode = false,
00739         ColorCount = 4,
00740         ColorList = new Color[] {
00741             Color.FromArgb(0xFF, 0xFF, 0xFF),
00742             Color.FromArgb(0xFA, 0xFA, 0xFB),
00743             Color.FromArgb(0xFD, 0x79, 0x4F),
00744             Color.FromArgb(0x13, 0x13, 0x13)
00745         },
00746     },
00747     new Theme() {
00748         Name = "Pastel 55",
00749         Id = 55,
00750         DarkMode = false,

```

```

00751         ColorCount = 4,
00752         ColorList = new Color[] {
00753             Color.FromArgb(0x40, 0x8A, 0x8B),
00754             Color.FromArgb(0xF3, 0xC6, 0xC0),
00755             Color.FromArgb(0xA5, 0x6F, 0x65),
00756             Color.FromArgb(0x32, 0x2D, 0x2A)
00757         },
00758     },
00759     new Theme() {
00760         Name = "Moss 56",
00761         Id = 56,
00762         DarkMode = false,
00763         ColorCount = 4,
00764         ColorList = new Color[] {
00765             Color.FromArgb(0xFF, 0xFF, 0xFF),
00766             Color.FromArgb(0x00, 0xE0, 0xC8),
00767             Color.FromArgb(0x00, 0x93, 0x93),
00768             Color.FromArgb(0x01, 0x61, 0x70)
00769         },
00770     },
00771     new Theme() {
00772         Name = "Cabana 57",
00773         Id = 57,
00774         DarkMode = false,
00775         ColorCount = 4,
00776         ColorList = new Color[] {
00777             Color.FromArgb(0xFF, 0xD9, 0x60),
00778             Color.FromArgb(0xD7, 0x41, 0x5D),
00779             Color.FromArgb(0x20, 0x95, 0xF2),
00780             Color.FromArgb(0x23, 0x6A, 0xD6)
00781         },
00782     },
00783     new Theme() {
00784         Name = "Multi 58",
00785         Id = 58,
00786         DarkMode = false,
00787         ColorCount = 5,
00788         ColorList = new Color[] {
00789             Color.FromArgb(0xCA, 0xD7, 0xD9),
00790             Color.FromArgb(0x41, 0xBE, 0xD4),
00791             Color.FromArgb(0x07, 0x89, 0xA4),
00792             Color.FromArgb(0xF1, 0xC3, 0x6D),
00793             Color.FromArgb(0x25, 0x42, 0x43)
00794         },
00795     },
00796     new Theme() {
00797         Name = "Brown 59",
00798         Id = 59,
00799         DarkMode = false,
00800         ColorCount = 5,
00801         ColorList = new Color[] {
00802             Color.FromArgb(0xF5, 0xCB, 0x63),
00803             Color.FromArgb(0xFE, 0xAA, 0x42),
00804             Color.FromArgb(0xC4, 0x84, 0x83),
00805             Color.FromArgb(0x7C, 0x6B, 0x7E),
00806             Color.FromArgb(0x30, 0x40, 0x4F)
00807         },
00808     },
00809     new Theme() {
00810         Name = "Green 60",
00811         Id = 60,
00812         DarkMode = false,
00813         ColorCount = 5,
00814         ColorList = new Color[] {
00815             Color.FromArgb(0xC3, 0xDD, 0xDE),
00816             Color.FromArgb(0xA5, 0xD1, 0xD4),
00817             Color.FromArgb(0x78, 0xA8, 0xAC),
00818             Color.FromArgb(0x2F, 0x77, 0x73),
00819             Color.FromArgb(0xFE, 0xB6, 0x00)
00820         },
00821     },
00822     new Theme() {
00823         Name = "Brown 61",
00824         Id = 61,
00825         DarkMode = false,
00826         ColorCount = 5,
00827         ColorList = new Color[] {
00828             Color.FromArgb(0xF3, 0xF3, 0xF3),
00829             Color.FromArgb(0xD9, 0xB6, 0xAA),
00830             Color.FromArgb(0xFA, 0xA9, 0x71),
00831             Color.FromArgb(0xE6, 0x76, 0x69),
00832             Color.FromArgb(0x85, 0x4C, 0x62)
00833         },
00834     },
00835     new Theme() {
00836         Name = "Brown 62",
00837         Id = 62,

```



```

00838         DarkMode = false,
00839         ColorCount = 5,
00840         ColorList = new Color[] {
00841             Color.FromArgb(0xEE, 0xEE, 0xDD),
00842             Color.FromArgb(0xED, 0xD9, 0xBB),
00843             Color.FromArgb(0xCA, 0xBC, 0x9D),
00844             Color.FromArgb(0xDC, 0x96, 0x66),
00845             Color.FromArgb(0x4D, 0x30, 0x14)
00846         },
00847     },
00848     new Theme() {
00849         Name = "Candle 63",
00850         Id = 63,
00851         DarkMode = false,
00852         ColorCount = 4,
00853         ColorList = new Color[] {
00854             Color.FromArgb(0xC1, 0xB9, 0xAE),
00855             Color.FromArgb(0xFC, 0xC6, 0x66),
00856             Color.FromArgb(0xFA, 0x8D, 0x70),
00857             Color.FromArgb(0x73, 0x60, 0x7C)
00858         },
00859     },
00860     new Theme() {
00861         Name = "Green 64",
00862         Id = 64,
00863         DarkMode = false,
00864         ColorCount = 4,
00865         ColorList = new Color[] {
00866             Color.FromArgb(0xFB, 0xFB, 0xF3),
00867             Color.FromArgb(0xD7, 0xCF, 0xFE),
00868             Color.FromArgb(0xC8, 0xF4, 0xC3),
00869             Color.FromArgb(0x0B, 0xAB, 0x7B)
00870         },
00871     },
00872     new Theme() {
00873         Name = "Multi 65",
00874         Id = 65,
00875         DarkMode = false,
00876         ColorCount = 4,
00877         ColorList = new Color[] {
00878             Color.FromArgb(0xE9, 0xED, 0xF1),
00879             Color.FromArgb(0xFF, 0xD8, 0x41),
00880             Color.FromArgb(0x3D, 0x73, 0xDD),
00881             Color.FromArgb(0x0E, 0x20, 0x45)
00882         },
00883     },
00884     new Theme() {
00885         Name = "Multi 66",
00886         Id = 66,
00887         DarkMode = false,
00888         ColorCount = 4,
00889         ColorList = new Color[] {
00890             Color.FromArgb(0xFC, 0xEE, 0xE3),
00891             Color.FromArgb(0xF8, 0xF1, 0xFF),
00892             Color.FromArgb(0xCC, 0xF9, 0xF2),
00893             Color.FromArgb(0x86, 0x49, 0xFC)
00894         },
00895     },
00896     new Theme() {
00897         Name = "Grey 67",
00898         Id = 67,
00899         DarkMode = true,
00900         ColorCount = 4,
00901         ColorList = new Color[] {
00902             Color.FromArgb(0xED, 0xEE, 0xF2),
00903             Color.FromArgb(0xFA, 0x5D, 0x3A),
00904             Color.FromArgb(0x4A, 0x4B, 0x4F),
00905             Color.FromArgb(0x40, 0x40, 0x41)
00906         },
00907     },
00908     new Theme() {
00909         Name = "Grey 68",
00910         Id = 68,
00911         DarkMode = false,
00912         ColorCount = 4,
00913         ColorList = new Color[] {
00914             Color.FromArgb(0xF6, 0xF6, 0xF6),
00915             Color.FromArgb(0xDC, 0xE6, 0xFF),
00916             Color.FromArgb(0xCD, 0xD0, 0xD9),
00917             Color.FromArgb(0xFF, 0x7A, 0x5B)
00918         },
00919     },
00920     new Theme() {
00921         Name = "Multi 69",
00922         Id = 69,
00923         DarkMode = false,
00924         ColorCount = 4,

```

```

00925         ColorList = new Color[] {
00926             Color.FromArgb(0xFF, 0xDA, 0xDA),
00927             Color.FromArgb(0xC6, 0xE6, 0xFF),
00928             Color.FromArgb(0xFF, 0x9A, 0x0D),
00929             Color.FromArgb(0x78, 0x6E, 0xF9)
00930         },
00931     },
00932     new Theme() {
00933         Name = "Grey 70",
00934         Id = 70,
00935         DarkMode = false,
00936         ColorCount = 4,
00937         ColorList = new Color[] {
00938             Color.FromArgb(0xE5, 0xE2, 0xDB),
00939             Color.FromArgb(0xDE, 0xE9, 0xE5),
00940             Color.FromArgb(0xFF, 0xD4, 0x81),
00941             Color.FromArgb(0x11, 0x11, 0x11)
00942         },
00943     },
00944     new Theme() {
00945         Name = "Grey 71",
00946         Id = 71,
00947         DarkMode = false,
00948         ColorCount = 4,
00949         ColorList = new Color[] {
00950             Color.FromArgb(0xF1, 0xF2, 0xED),
00951             Color.FromArgb(0xD5, 0xDF, 0xE8),
00952             Color.FromArgb(0x56, 0x75, 0x6D),
00953             Color.FromArgb(0x2F, 0x2F, 0x2F)
00954         },
00955     },
00956     new Theme() {
00957         Name = "Multi 72",
00958         Id = 72,
00959         DarkMode = false,
00960         ColorCount = 5,
00961         ColorList = new Color[] {
00962             Color.FromArgb(0xFF, 0x9B, 0x92),
00963             Color.FromArgb(0xFF, 0x70, 0x66),
00964             Color.FromArgb(0x55, 0xAF, 0xDC),
00965             Color.FromArgb(0x21, 0x74, 0xB4),
00966             Color.FromArgb(0x2A, 0x36, 0x5D)
00967         },
00968     },
00969     new Theme() {
00970         Name = "Multi 73",
00971         Id = 73,
00972         DarkMode = false,
00973         ColorCount = 5,
00974         ColorList = new Color[] {
00975             Color.FromArgb(0xF6, 0xF5, 0xEE),
00976             Color.FromArgb(0xFB, 0xEB, 0x45),
00977             Color.FromArgb(0xD0, 0xD0, 0xD0),
00978             Color.FromArgb(0xEA, 0xB4, 0xE3),
00979             Color.FromArgb(0x00, 0x00, 0x00)
00980         },
00981     },
00982     new Theme() {
00983         Name = "Grey 74",
00984         Id = 74,
00985         DarkMode = false,
00986         ColorCount = 5,
00987         ColorList = new Color[] {
00988             Color.FromArgb(0xF8, 0xF3, 0xED),
00989             Color.FromArgb(0xF6, 0xF5, 0xEE),
00990             Color.FromArgb(0xD0, 0xD0, 0xD0),
00991             Color.FromArgb(0xCA, 0x97, 0x5E),
00992             Color.FromArgb(0x00, 0x00, 0x00)
00993         },
00994     },
00995     new Theme() {
00996         Name = "Blue 75",
00997         Id = 75,
00998         DarkMode = false,
00999         ColorCount = 7,
01000         ColorList = new Color[] {
01001             Color.FromArgb(0xFF, 0xFF, 0xFF),
01002             Color.FromArgb(0xE1, 0xF0, 0xF9),
01003             Color.FromArgb(0xD3, 0xF8, 0xFB),
01004             Color.FromArgb(0x00, 0xCF, 0xDE),
01005             Color.FromArgb(0x1D, 0x3C, 0xD0),
01006             Color.FromArgb(0x01, 0x24, 0x5E),
01007             Color.FromArgb(0x00, 0x00, 0x00)
01008         },
01009     },
01010     new Theme() {
01011         Name = "Gotham 76",

```

```

01012         Id = 76,
01013         DarkMode = false,
01014         ColorCount = 4,
01015         ColorList = new Color[] {
01016             Color.FromArgb(0xEA, 0xEB, 0xED),
01017             Color.FromArgb(0xC2, 0x7D, 0xFC),
01018             Color.FromArgb(0x6E, 0x5D, 0xCF),
01019             Color.FromArgb(0x24, 0x3B, 0x67)
01020         },
01021     },
01022     new Theme() {
01023         Name = "Tangerine 77",
01024         Id = 77,
01025         DarkMode = false,
01026         ColorCount = 4,
01027         ColorList = new Color[] {
01028             Color.FromArgb(0xFF, 0xD1, 0x81),
01029             Color.FromArgb(0xFF, 0xAB, 0x41),
01030             Color.FromArgb(0xFF, 0x92, 0x01),
01031             Color.FromArgb(0xFF, 0x6D, 0x00)
01032         },
01033     },
01034     new Theme() {
01035         Name = "Grove 78",
01036         Id = 78,
01037         DarkMode = false,
01038         ColorCount = 4,
01039         ColorList = new Color[] {
01040             Color.FromArgb(0xE9, 0xEB, 0xED),
01041             Color.FromArgb(0xAE, 0xD3, 0x6C),
01042             Color.FromArgb(0x69, 0xAB, 0x3D),
01043             Color.FromArgb(0x29, 0x5D, 0x09)
01044         },
01045     },
01046     new Theme() {
01047         Name = "Embers 79",
01048         Id = 79,
01049         DarkMode = true,
01050         ColorCount = 4,
01051         ColorList = new Color[] {
01052             Color.FromArgb(0xFF, 0x96, 0x77),
01053             Color.FromArgb(0xF5, 0x47, 0x68),
01054             Color.FromArgb(0x97, 0x40, 0x63),
01055             Color.FromArgb(0x41, 0x43, 0x6A)
01056         },
01057     },
01058     new Theme() {
01059         Name = "Sunset 80",
01060         Id = 80,
01061         DarkMode = true,
01062         ColorCount = 4,
01063         ColorList = new Color[] {
01064             Color.FromArgb(0xFF, 0x75, 0x82),
01065             Color.FromArgb(0xC5, 0x6D, 0x86),
01066             Color.FromArgb(0x72, 0x5A, 0x7A),
01067             Color.FromArgb(0x35, 0x5C, 0x7D)
01068         },
01069     },
01070     new Theme() {
01071         Name = "Blue 81",
01072         Id = 81,
01073         DarkMode = false,
01074         ColorCount = 4,
01075         ColorList = new Color[] {
01076             Color.FromArgb(0xD9, 0xEF, 0xF7),
01077             Color.FromArgb(0x9B, 0xBB, 0xFC),
01078             Color.FromArgb(0xF9, 0xCE, 0x69),
01079             Color.FromArgb(0x47, 0x41, 0xA6)
01080         },
01081     },
01082     new Theme() {
01083         Name = "Green 82",
01084         Id = 82,
01085         DarkMode = false,
01086         ColorCount = 4,
01087         ColorList = new Color[] {
01088             Color.FromArgb(0xEA, 0xF9, 0xE7),
01089             Color.FromArgb(0xC0, 0xE6, 0xBA),
01090             Color.FromArgb(0x4C, 0xA7, 0x71),
01091             Color.FromArgb(0x01, 0x32, 0x37)
01092         },
01093     },
01094     new Theme() {
01095         Name = "Red 83",
01096         Id = 83,
01097         DarkMode = false,
01098         ColorCount = 4,

```

```

01099         ColorList = new Color[] {
01100             Color.FromArgb(0xFF6, 0xE8, 0xDF),
01101             Color.FromArgb(0xFE, 0xAE, 0x96),
01102             Color.FromArgb(0xFE, 0x97, 0x9C),
01103             Color.FromArgb(0x01, 0x32, 0x37)
01104         },
01105     },
01106     new Theme() {
01107         Name = "Multi 84",
01108         Id = 84,
01109         DarkMode = false,
01110         ColorCount = 4,
01111         ColorList = new Color[] {
01112             Color.FromArgb(0xFC, 0xDC, 0x73),
01113             Color.FromArgb(0xE7, 0x62, 0x86),
01114             Color.FromArgb(0x4F, 0xAD, 0xC0),
01115             Color.FromArgb(0x19, 0x39, 0x48)
01116         },
01117     },
01118     new Theme() {
01119         Name = "Blue 85",
01120         Id = 85,
01121         DarkMode = true,
01122         ColorCount = 4,
01123         ColorList = new Color[] {
01124             Color.FromArgb(0x60, 0xCB, 0xFF),
01125             Color.FromArgb(0x36, 0x64, 0xF4),
01126             Color.FromArgb(0xE8, 0x53, 0x53),
01127             Color.FromArgb(0x25, 0x25, 0x25)
01128         },
01129     },
01130     new Theme() {
01131         Name = "Multi 86",
01132         Id = 86,
01133         DarkMode = true,
01134         ColorCount = 4,
01135         ColorList = new Color[] {
01136             Color.FromArgb(0xF8, 0xB0, 0x97),
01137             Color.FromArgb(0x77, 0xAC, 0xB7),
01138             Color.FromArgb(0x3F, 0xC4, 0x95),
01139             Color.FromArgb(0x1A, 0x1A, 0x1A)
01140         },
01141     },
01142     new Theme() {
01143         Name = "Multi 87",
01144         Id = 87,
01145         DarkMode = true,
01146         ColorCount = 4,
01147         ColorList = new Color[] {
01148             Color.FromArgb(0xC2, 0xE7, 0xC9),
01149             Color.FromArgb(0xF2, 0xAC, 0x20),
01150             Color.FromArgb(0x42, 0x65, 0xD6),
01151             Color.FromArgb(0x29, 0x38, 0x55)
01152         },
01153     },
01154     new Theme() {
01155         Name = "Blue Gradient 88",
01156         Id = 88,
01157         DarkMode = true,
01158         ColorCount = 8,
01159         ColorList = new Color[] {
01160             Color.FromArgb(0x3A, 0xB5, 0xE2),
01161             Color.FromArgb(0x33, 0x9C, 0xC7),
01162             Color.FromArgb(0x2C, 0x84, 0xAB),
01163             Color.FromArgb(0x25, 0x6B, 0x90),
01164             Color.FromArgb(0x1F, 0x53, 0x74),
01165             Color.FromArgb(0x18, 0x3A, 0x59),
01166             Color.FromArgb(0x11, 0x22, 0x3D),
01167             Color.FromArgb(0x0A, 0x09, 0x22)
01168         },
01169     },
01170     new Theme() {
01171         Name = "Purple Gradient 89",
01172         Id = 89,
01173         DarkMode = true,
01174         ColorCount = 8,
01175         ColorList = new Color[] {
01176             Color.FromArgb(0xB5, 0x3A, 0xE2),
01177             Color.FromArgb(0x9C, 0x33, 0xC7),
01178             Color.FromArgb(0x84, 0x2C, 0xAB),
01179             Color.FromArgb(0x6B, 0x25, 0x90),
01180             Color.FromArgb(0x53, 0x1F, 0x74),
01181             Color.FromArgb(0x3A, 0x18, 0x59),
01182             Color.FromArgb(0x22, 0x11, 0x3D),
01183             Color.FromArgb(0x09, 0x0A, 0x22)
01184         },
01185     },

```

```

01186         new Theme() {
01187             Name = "Lime Gradient 90",
01188             Id = 90,
01189             DarkMode = true,
01190             ColorCount = 8,
01191             ColorList = new Color[] {
01192                 Color.FromArgb(0xB5, 0xE2, 0x3A),
01193                 Color.FromArgb(0x9C, 0xC7, 0x33),
01194                 Color.FromArgb(0x84, 0xAB, 0x2C),
01195                 Color.FromArgb(0x6B, 0x90, 0x25),
01196                 Color.FromArgb(0x53, 0x74, 0x1F),
01197                 Color.FromArgb(0x3A, 0x59, 0x18),
01198                 Color.FromArgb(0x22, 0x3D, 0x11),
01199                 Color.FromArgb(0x09, 0x22, 0x0A)
01200             }
01201         },
01202         new Theme() {
01203             Name = "Yellow Gradient 91",
01204             Id = 91,
01205             DarkMode = true,
01206             ColorCount = 8,
01207             ColorList = new Color[] {
01208                 Color.FromArgb(0xE2, 0xB5, 0x3A),
01209                 Color.FromArgb(0xC7, 0x9C, 0x33),
01210                 Color.FromArgb(0xAB, 0x84, 0x2C),
01211                 Color.FromArgb(0x90, 0x6B, 0x25),
01212                 Color.FromArgb(0x74, 0x53, 0x1F),
01213                 Color.FromArgb(0x59, 0x3A, 0x18),
01214                 Color.FromArgb(0x3D, 0x22, 0x11),
01215                 Color.FromArgb(0x22, 0x09, 0x0A)
01216             }
01217         },
01218         new Theme() {
01219             Name = "Purple Gradient 92",
01220             Id = 92,
01221             DarkMode = true,
01222             ColorCount = 8,
01223             ColorList = new Color[] {
01224                 Color.FromArgb(0xE2, 0x3A, 0xB5),
01225                 Color.FromArgb(0xC7, 0x33, 0x9C),
01226                 Color.FromArgb(0xAB, 0x2C, 0x84),
01227                 Color.FromArgb(0x90, 0x25, 0x6B),
01228                 Color.FromArgb(0x74, 0x1F, 0x53),
01229                 Color.FromArgb(0x59, 0x18, 0x3A),
01230                 Color.FromArgb(0x3D, 0x11, 0x22),
01231                 Color.FromArgb(0x22, 0x0A, 0x09)
01232             }
01233         },
01234         new Theme() {
01235             Name = "Red Gradient 93",
01236             Id = 93,
01237             DarkMode = true,
01238             ColorCount = 8,
01239             ColorList = new Color[] {
01240                 Color.FromArgb(0xE3, 0x41, 0x3B),
01241                 Color.FromArgb(0xC7, 0x39, 0x34),
01242                 Color.FromArgb(0xAC, 0x31, 0x2D),
01243                 Color.FromArgb(0x90, 0x29, 0x26),
01244                 Color.FromArgb(0x74, 0x21, 0x1E),
01245                 Color.FromArgb(0x58, 0x19, 0x17),
01246                 Color.FromArgb(0x3D, 0x11, 0x10),
01247                 Color.FromArgb(0x21, 0x09, 0x09)
01248             }
01249         },
01250         new Theme() {
01251             Name = "Green Gradient 94",
01252             Id = 94,
01253             DarkMode = true,
01254             ColorCount = 8,
01255             ColorList = new Color[] {
01256                 Color.FromArgb(0x41, 0xE3, 0x3B),
01257                 Color.FromArgb(0x39, 0xC7, 0x34),
01258                 Color.FromArgb(0x31, 0xAC, 0x2D),
01259                 Color.FromArgb(0x29, 0x90, 0x26),
01260                 Color.FromArgb(0x21, 0x74, 0x1E),
01261                 Color.FromArgb(0x19, 0x58, 0x17),
01262                 Color.FromArgb(0x11, 0x3D, 0x10),
01263                 Color.FromArgb(0x09, 0x21, 0x09)
01264             }
01265         },
01266         new Theme() {
01267             Name = "Blue Gradient 95",
01268             Id = 95,
01269             DarkMode = true,
01270             ColorCount = 8,
01271             ColorList = new Color[] {
01272                 Color.FromArgb(0x3B, 0x41, 0xE3),

```

```

01273         Color.FromArgb(0x34, 0x39, 0xC7),
01274         Color.FromArgb(0x2D, 0x31, 0xAC),
01275         Color.FromArgb(0x26, 0x29, 0x90),
01276         Color.FromArgb(0x1E, 0x21, 0x74),
01277         Color.FromArgb(0x17, 0x19, 0x58),
01278         Color.FromArgb(0x10, 0x11, 0x3D),
01279         Color.FromArgb(0x09, 0x09, 0x21)
01280     }
01281 },
01282 new Theme() {
01283     Name = "Green Gradient 96",
01284     Id = 96,
01285     DarkMode = true,
01286     ColorCount = 8,
01287     ColorList = new Color[] {
01288         Color.FromArgb(0x3B, 0xE3, 0x3B),
01289         Color.FromArgb(0x34, 0xC7, 0x36),
01290         Color.FromArgb(0x2D, 0xAC, 0x30),
01291         Color.FromArgb(0x26, 0x90, 0x2B),
01292         Color.FromArgb(0x1E, 0x74, 0x26),
01293         Color.FromArgb(0x17, 0x58, 0x21),
01294         Color.FromArgb(0x10, 0x3D, 0x1B),
01295         Color.FromArgb(0x09, 0x21, 0x16)
01296     }
01297 },
01298 new Theme() {
01299     Name = "Grey Gradient 97",
01300     Id = 97,
01301     DarkMode = true,
01302     ColorCount = 8,
01303     ColorList = new Color[] {
01304         Color.FromArgb(0xFE, 0xFE, 0xFE),
01305         Color.FromArgb(0xDA, 0xDA, 0xDA),
01306         Color.FromArgb(0xB6, 0xB6, 0xB6),
01307         Color.FromArgb(0x92, 0x92, 0x92),
01308         Color.FromArgb(0x6D, 0x6D, 0x6D),
01309         Color.FromArgb(0x49, 0x49, 0x49),
01310         Color.FromArgb(0x25, 0x25, 0x25),
01311         Color.FromArgb(0x01, 0x01, 0x01)
01312     }
01313 },
01314 new Theme() {
01315     Name = "Green Blue Gradient 98",
01316     Id = 98,
01317     DarkMode = false,
01318     ColorCount = 8,
01319     ColorList = new Color[] {
01320         Color.FromArgb(0x07, 0xCF, 0x00),
01321         Color.FromArgb(0x0A, 0xB1, 0x1E),
01322         Color.FromArgb(0x0D, 0x94, 0x3B),
01323         Color.FromArgb(0x10, 0x76, 0x59),
01324         Color.FromArgb(0x13, 0x59, 0x76),
01325         Color.FromArgb(0x16, 0x3B, 0x94),
01326         Color.FromArgb(0x19, 0x1E, 0xB1),
01327         Color.FromArgb(0x1C, 0x00, 0xCF)
01328     }
01329 },
01330 new Theme() {
01331     Name = "Green Red Gradient 99",
01332     Id = 99,
01333     DarkMode = false,
01334     ColorCount = 8,
01335     ColorList = new Color[] {
01336         Color.FromArgb(0x07, 0xCF, 0x00),
01337         Color.FromArgb(0x24, 0xB1, 0x04),
01338         Color.FromArgb(0x40, 0x94, 0x08),
01339         Color.FromArgb(0x5D, 0x76, 0x0C),
01340         Color.FromArgb(0x79, 0x59, 0x10),
01341         Color.FromArgb(0x96, 0x3B, 0x14),
01342         Color.FromArgb(0xB2, 0x1E, 0x18),
01343         Color.FromArgb(0xCF, 0x00, 0x1C)
01344     }
01345 },
01346 new Theme() {
01347     Name = "Blue Red Gradient 100",
01348     Id = 100,
01349     DarkMode = false,
01350     ColorCount = 8,
01351     ColorList = new Color[] {
01352         Color.FromArgb(0x1C, 0x00, 0xCF),
01353         Color.FromArgb(0x36, 0x00, 0xB5),
01354         Color.FromArgb(0x4F, 0x00, 0x9C),
01355         Color.FromArgb(0x69, 0x00, 0x82),
01356         Color.FromArgb(0x82, 0x00, 0x69),
01357         Color.FromArgb(0x9C, 0x00, 0x4F),
01358         Color.FromArgb(0xB5, 0x00, 0x36),
01359         Color.FromArgb(0xCF, 0x00, 0x1C)

```

```

01360     }
01361 },
01362 new Theme() {
01363     Name = "Ocean 101",
01364     Id = 101,
01365     DarkMode = true,
01366     ColorCount = 4,
01367     ColorList = new Color[] {
01368         Color.FromArgb(0xEE, 0xD8, 0xCB),
01369         Color.FromArgb(0x4A, 0xB4, 0xDE),
01370         Color.FromArgb(0x3B, 0x8A, 0xC3),
01371         Color.FromArgb(0x34, 0x5E, 0xA8)
01372     },
01373 },
01374 new Theme() {
01375     Name = "Petals 102",
01376     Id = 102,
01377     DarkMode = false,
01378     ColorCount = 4,
01379     ColorList = new Color[] {
01380         Color.FromArgb(0xF0, 0xD7, 0x94),
01381         Color.FromArgb(0xFF, 0xBA, 0xC4),
01382         Color.FromArgb(0xFF, 0xA1, 0x97),
01383         Color.FromArgb(0xFA, 0xBD, 0x70)
01384     },
01385 },
01386 new Theme() {
01387     Name = "Sailing 103",
01388     Id = 103,
01389     DarkMode = false,
01390     ColorCount = 4,
01391     ColorList = new Color[] {
01392         Color.FromArgb(0xE9, 0xEE, 0xF2),
01393         Color.FromArgb(0x78, 0xA6, 0xC8),
01394         Color.FromArgb(0x32, 0x67, 0x89),
01395         Color.FromArgb(0xE6, 0x5C, 0x4F)
01396     },
01397 },
01398 new Theme() {
01399     Name = "Multi 104",
01400     Id = 104,
01401     DarkMode = false,
01402     ColorCount = 5,
01403     ColorList = new Color[] {
01404         Color.FromArgb(0xF8, 0xED, 0x6D),
01405         Color.FromArgb(0xEF, 0x88, 0x5D),
01406         Color.FromArgb(0xB7, 0x3A, 0x5C),
01407         Color.FromArgb(0x6B, 0x2D, 0x70),
01408         Color.FromArgb(0x3B, 0xA6, 0xBA)
01409     },
01410 },
01411 new Theme() {
01412     Name = "Haze 105",
01413     Id = 105,
01414     DarkMode = false,
01415     ColorCount = 4,
01416     ColorList = new Color[] {
01417         Color.FromArgb(0xC8, 0xCE, 0xFF),
01418         Color.FromArgb(0x84, 0xB7, 0xFE),
01419         Color.FromArgb(0x25, 0x8B, 0xD6),
01420         Color.FromArgb(0x0F, 0x6B, 0xAC)
01421     },
01422 },
01423 new Theme() {
01424     Name = "Periwinkle 106",
01425     Id = 106,
01426     DarkMode = false,
01427     ColorCount = 4,
01428     ColorList = new Color[] {
01429         Color.FromArgb(0xAE, 0xED, 0xE2),
01430         Color.FromArgb(0xA3, 0xDC, 0xEF),
01431         Color.FromArgb(0xA2, 0xB9, 0xED),
01432         Color.FromArgb(0x9A, 0x9C, 0xE9)
01433     },
01434 },
01435 new Theme() {
01436     Name = "Slate 107",
01437     Id = 107,
01438     DarkMode = true,
01439     ColorCount = 4,
01440     ColorList = new Color[] {
01441         Color.FromArgb(0xEA, 0xEB, 0xED),
01442         Color.FromArgb(0x98, 0xDA, 0xD9),
01443         Color.FromArgb(0x58, 0x82, 0x91),
01444         Color.FromArgb(0x2E, 0x42, 0x4D)
01445     },
01446 },

```

```
01447     new Theme() {
01448         Name = "Orange 108",
01449         Id = 108,
01450         DarkMode = true,
01451         ColorCount = 4,
01452         ColorList = new Color[] {
01453             Color.FromArgb(0xFF, 0xFF, 0xFF),
01454             Color.FromArgb(0xF1, 0xED, 0xE5),
01455             Color.FromArgb(0xE4, 0x60, 0x36),
01456             Color.FromArgb(0x00, 0x00, 0x00)
01457         }
01458     },
01459     new Theme() {
01460         Name = "Multi 109",
01461         Id = 109,
01462         DarkMode = true,
01463         ColorCount = 4,
01464         ColorList = new Color[] {
01465             Color.FromArgb(0xF3, 0xEC, 0xE4),
01466             Color.FromArgb(0xED, 0x70, 0x61),
01467             Color.FromArgb(0x56, 0x9E, 0x34),
01468             Color.FromArgb(0x50, 0x5D, 0x7D)
01469         }
01470     },
01471     new Theme() {
01472         Name = "Grey 110",
01473         Id = 110,
01474         DarkMode = false,
01475         ColorCount = 4,
01476         ColorList = new Color[] {
01477             Color.FromArgb(0xFF, 0xFF, 0xFF),
01478             Color.FromArgb(0xDF, 0xDF, 0xDF),
01479             Color.FromArgb(0xAA, 0xAA, 0xAA),
01480             Color.FromArgb(0xFC, 0x57, 0x23)
01481         }
01482     },
01483     new Theme() {
01484         Name = "Gold 111",
01485         Id = 111,
01486         DarkMode = false,
01487         ColorCount = 4,
01488         ColorList = new Color[] {
01489             Color.FromArgb(0xE5, 0xE2, 0xDB),
01490             Color.FromArgb(0xDE, 0xE9, 0xE5),
01491             Color.FromArgb(0xFF, 0xD4, 0x81),
01492             Color.FromArgb(0x11, 0x11, 0x11)
01493         }
01494     },
01495     new Theme() {
01496         Name = "Blue 112",
01497         Id = 112,
01498         DarkMode = true,
01499         ColorCount = 4,
01500         ColorList = new Color[] {
01501             Color.FromArgb(0xF3, 0xEC, 0xDE),
01502             Color.FromArgb(0xEC, 0xDC, 0xF4),
01503             Color.FromArgb(0x1E, 0x56, 0xC3),
01504             Color.FromArgb(0x27, 0x29, 0x32)
01505         }
01506     },
01507     new Theme() {
01508         Name = "Neon 113",
01509         Id = 113,
01510         DarkMode = true,
01511         ColorCount = 4,
01512         ColorList = new Color[] {
01513             Color.FromArgb(0xF7, 0xF8, 0xF6),
01514             Color.FromArgb(0xD3, 0xDD, 0xDA),
01515             Color.FromArgb(0xD7, 0xF2, 0x66),
01516             Color.FromArgb(0x15, 0x15, 0x14)
01517         }
01518     },
01519     new Theme() {
01520         Name = "Multi 114",
01521         Id = 114,
01522         DarkMode = true,
01523         ColorCount = 4,
01524         ColorList = new Color[] {
01525             Color.FromArgb(0x8E, 0xC9, 0x7B),
01526             Color.FromArgb(0xF7, 0x91, 0x93),
01527             Color.FromArgb(0x24, 0x6B, 0xFD),
01528             Color.FromArgb(0x23, 0x19, 0x34)
01529         }
01530     },
01531     new Theme() {
01532         Name = "Gradient 115",
01533         Id = 115,
```



```

01534         DarkMode = false,
01535         ColorCount = 8,
01536         ColorList = GetThemeByGradient (Color.FromArgb(0x53, 0xC0, 0xEE), Color.FromArgb(0x52,
01537         0x33, 0xCE), 8).ColorList
01538     },
01539     new Theme() {
01540         Name = "Gradient 116",
01541         Id = 116,
01542         DarkMode = false,
01543         ColorCount = 8,
01544         ColorList = GetThemeByGradient (Color.FromArgb(0x8E, 0xEE, 0x53), Color.FromArgb(0x33,
01545         0x55, 0xCE), 8).ColorList
01546     },
01547     new Theme() {
01548         Name = "Gradient 117",
01549         Id = 117,
01550         DarkMode = false,
01551         ColorCount = 8,
01552         ColorList = GetThemeByGradient (Color.FromArgb(0xEE, 0xC3, 0x53), Color.FromArgb(0xEE,
01553         0x53, 0x53), 8).ColorList
01554     },
01555     new Theme() {
01556         Name = "Gradient 118",
01557         Id = 118,
01558         DarkMode = true,
01559         ColorCount = 8,
01560         ColorList = GetThemeByGradient (Color.FromArgb(0x01, 0x95, 0xB9), Color.FromArgb(0x0C,
01561         0x3E, 0x58), 8).ColorList
01562     },
01563     new Theme() {
01564         Name = "Gradient 119",
01565         Id = 119,
01566         DarkMode = true,
01567         ColorCount = 8,
01568         ColorList = GetThemeByGradient (Color.FromArgb(0x00, 0x4E, 0x92), Color.FromArgb(0x00,
01569         0x04, 0x28), 8).ColorList
01570     },
01571     new Theme() {
01572         Name = "Gradient 120",
01573         Id = 120,
01574         DarkMode = false,
01575         ColorCount = 8,
01576         ColorList = GetThemeByGradient (Color.FromArgb(0xBD, 0xC3, 0xC7), Color.FromArgb(0x2C,
01577         0x3E, 0x50), 8).ColorList
01578     },
01579     new Theme() {
01580         Name = "Gradient 121",
01581         Id = 121,
01582         DarkMode = true,
01583         ColorCount = 8,
01584         ColorList = GetThemeByGradient (Color.FromArgb(0x00, 0x4E, 0xB2), Color.FromArgb(0x00,
01585         0x04, 0x72), 8).ColorList
01586     },
01587     new Theme() {
01588         Name = "Gradient 122",
01589         Id = 122,
01590         DarkMode = false,
01591         ColorCount = 8,
01592         ColorList = GetThemeByGradient (Color.FromArgb(0x6D, 0xD5, 0xFA), Color.FromArgb(0x29,
01593         0x80, 0xB9), 8).ColorList
01594     },
01595     new Theme() {
01596         Name = "Gradient 123",
01597         Id = 123,
01598         DarkMode = false,
01599         ColorCount = 8,
01600         ColorList = GetThemeByGradient (Color.FromArgb(0x34, 0xE8, 0x9E), Color.FromArgb(0x0F,
01601         0x34, 0x43), 8).ColorList
01602     },
01603     new Theme() {
01604         Name = "Gradient 124",
01605         Id = 124,
01606         DarkMode = false,
01607         ColorCount = 8,
01608         ColorList = GetThemeByGradient (Color.FromArgb(0xEA, 0xEC, 0xC6), Color.FromArgb(0x2B,
01609         0xC0, 0xE4), 8).ColorList
01610     },
01611     new Theme() {
01612         Name = "Gradient 125",
01613         Id = 125,
01614         DarkMode = false,
01615         ColorCount = 8,
01616         ColorList = GetThemeByGradient (Color.FromArgb(0xEA, 0x52, 0xF8), Color.FromArgb(0x00,
01617         0x66, 0xFF), 8).ColorList
01618     },
01619     new Theme() {
01620         Name = "Gradient 126",

```

```

01610         Id = 126,
01611         DarkMode = false,
01612         ColorCount = 8,
01613         ColorList = GetThemeByGradient(Color.FromArgb(0xFF, 0xE0, 0x00), Color.FromArgb(0x79,
01614         0x9F, 0x0C), 8).ColorList
01615     },
01616     new Theme() {
01617         Name = "Gradient 127",
01618         Id = 127,
01619         DarkMode = true,
01620         ColorCount = 8,
01621         ColorList = GetThemeByGradient(Color.FromArgb(0x43, 0xC6, 0xAC), Color.FromArgb(0x19,
01622         0x16, 0x54), 8).ColorList
01623     },
01624     new Theme() {
01625         Name = "Gradient 128",
01626         Id = 128,
01627         DarkMode = true,
01628         ColorCount = 8,
01629         ColorList = GetThemeByGradient(Color.FromArgb(0xFC, 0xF6, 0xCF), Color.FromArgb(0xFF,
01630         0x88, 0x6A), 8).ColorList
01631     }
01632 };
01633 #endregion
01634
01635 #region Functions
01636 public static Theme GetThemeById(int Id)
01637 {
01638     Theme? theme = null;
01639     if (Id < 1)
01640         Id = 1;
01641     if (Id > ThemeList.Length)
01642         Id = ThemeList.Length;
01643     foreach (Theme _theme in ThemeList)
01644     {
01645         if (_theme.Id == Id)
01646         {
01647             theme = _theme;
01648             break;
01649         }
01650     }
01651     if (theme == null)
01652     {
01653         #if DEBUG
01654             throw new Exception("Exception: Theme is null");
01655         #else
01656             theme = ThemeList[0];
01657         #endif
01658     }
01659     return theme;
01660 }
01661
01662 public static Theme GetThemeByName(string Name)
01663 {
01664     Theme? theme = null;
01665     foreach (Theme _theme in ThemeList)
01666     {
01667         if (_theme.Name == Name)
01668         {
01669             theme = _theme;
01670             break;
01671         }
01672     }
01673     if (theme == null)
01674     {
01675         #if DEBUG
01676             throw new Exception("Exception: Theme is null");
01677         #else
01678             theme = ThemeList[0];
01679             theme.DarkMode = false;
01680         #endif
01681     }
01682     return theme;
01683 }
01684
01685 internal static Color GetThemeColor(Theme theme, int ColorIndex, bool DarkMode)
01686 {
01687     if (theme == null)
01688     {
01689         #if DEBUG
01690             throw new Exception("Exception: Theme is null");
01691         #else
01692             if (DarkMode)
01693                 return Color.Black;
01694             else
01695                 return Color.White;
01696         #endif
01697     }
01698 }

```

```

01712         }
01713         theme.DarkMode = DarkMode;
01714         return theme.GetColor(ColorIndex);
01715     }
01716
01725     public static Color GetThemeColor(string ThemeName, int ColorIndex, bool DarkMode)
01726     {
01727         Theme? theme = GetThemeByName(ThemeName);
01728         if (theme == null)
01729         {
01730             #if DEBUG
01731                 throw new Exception("Exception: Theme is null");
01732             #else
01733                 if (DarkMode)
01734                     return Color.Black;
01735                 else
01736                     return Color.White;
01737             #endif
01738         }
01739         return GetThemeColor(theme, ColorIndex, DarkMode);
01740     }
01741
01750     public static Color GetThemeColor(int ThemeId, int ColorIndex, bool DarkMode)
01751     {
01752         Theme? theme = GetThemeById(ThemeId);
01753         if (theme == null)
01754         {
01755             #if DEBUG
01756                 throw new Exception("Exception: Theme is null");
01757             #else
01758                 if (DarkMode)
01759                     return Color.Black;
01760                 else
01761                     return Color.White;
01762             #endif
01763         }
01764         return GetThemeColor(theme, ColorIndex, DarkMode);
01765     }
01766
01774     public static Color GetThemeColor(int ThemeId, int ColorIndex)
01775     {
01776         return GetThemeColor(ThemeId, ColorIndex, GetThemeById(ThemeId).DarkMode);
01777     }
01778
01786     public static Color GetThemeColor(string ThemeName, int ColorIndex)
01787     {
01788         Theme? theme = GetThemeByName(ThemeName);
01789         if (theme == null)
01790         {
01791             #if DEBUG
01792                 throw new Exception("Exception: Theme is null");
01793             #else
01794                 return GetThemeColor(1, 0, false);
01795             #endif
01796         }
01797         else
01798             return GetThemeColor(ThemeName, ColorIndex, theme.DarkMode);
01799     }
01800
01805     public static int GetThemeColorCount(string ThemeName)
01806     {
01807         Theme? theme = GetThemeByName(ThemeName);
01808         if (theme == null)
01809         {
01810             #if DEBUG
01811                 throw new Exception("Exception: Theme is null");
01812             #else
01813                 return -1;
01814             #endif
01815         }
01816         return theme.ColorCount;
01817     }
01818
01825     public static int GetThemeColorCount(int ThemeId)
01826     {
01827         Theme? theme = GetThemeById(ThemeId);
01828         if (theme == null)
01829         {
01830             #if DEBUG
01831                 throw new Exception("Exception: Theme is null");
01832             #else
01833                 return -1;
01834             #endif
01835         }
01836         return theme.ColorCount;
01837     }
01838
01845     public static List<string> GetThemeNames(bool SortAtoZ = true, bool ShowColorCount = true)
01846     {

```

```

01847         List<string> list = new List<string>();
01848         foreach (Theme theme in ThemeList)
01849         {
01850             if (ShowColorCount)
01851                 list.Add(theme.Name + " [" + theme.ColorCount.ToString() + "]");
01852             else
01853                 list.Add(theme.Name);
01854         }
01855         if (SortAtoZ)
01856             list.Sort((x, y) => string.Compare(x, y));
01857         return list;
01858     }
01859
01860     public static List<string> GetThemeNamesByColorCount(int ColorCount, bool ColorCountAsMin =
01861     false, bool SortAtoZ = true, bool ShowColorCount = true)
01862     {
01863         List<string> list = new List<string>();
01864         foreach (Theme theme in ThemeList)
01865         {
01866             if ((theme.ColorCount == ColorCount && !ColorCountAsMin) || (theme.ColorCount >=
01867             ColorCount && ColorCountAsMin))
01868             {
01869                 if (ShowColorCount)
01870                     list.Add(theme.Name + " [" + theme.ColorCount.ToString() + "]");
01871                 else
01872                     list.Add(theme.Name);
01873             }
01874         }
01875         if (SortAtoZ)
01876             list.Sort((x, y) => string.Compare(x, y));
01877         return list;
01878     }
01879
01880     public static Theme GetThemeByGradient(Color Start, Color End, int ColorCount, string Name =
01881     "GradientTheme 1")
01882     {
01883         //allow only between 2 and 8 theme colors
01884         if (ColorCount < 2)
01885             ColorCount = 2;
01886         if (ColorCount > 8)
01887             ColorCount = 8;
01888
01889         //create temporary color list starting with the start color
01890         List<Color> GradientColors = new List<Color>(ColorCount);
01891         GradientColors.Add(Start);
01892         for (int i = 1; i < ColorCount - 1; i++)
01893         {
01894             GradientColors.Add(Color.Black);
01895         }
01896
01897         //get RGB values of start and end color and calculate step size
01898         double SA = Start.A;
01899         double SR = Start.R;
01900         double SG = Start.G;
01901         double SB = Start.B;
01902         double EA = End.A;
01903         double ER = End.R;
01904         double EG = End.G;
01905         double EB = End.B;
01906         double StepA = Math.Abs(SA - EA) / (ColorCount - 1);
01907         double StepR = Math.Abs(SR - ER) / (ColorCount - 1);
01908         double StepG = Math.Abs(SG - EG) / (ColorCount - 1);
01909         double StepB = Math.Abs(SB - EB) / (ColorCount - 1);
01910
01911         //fill color list with gradient colors
01912         for (int i = 1; i < ColorCount - 1; i++)
01913         {
01914             if (EA > SA)
01915                 EA -= StepA;
01916             else
01917                 EA += StepA;
01918             if (ER > SR)
01919                 ER -= StepR;
01920             else
01921                 ER += StepR;
01922             if (EG > SG)
01923                 EG -= StepG;
01924             else
01925                 EG += StepG;
01926             if (EB > SB)
01927                 EB -= StepB;
01928             else
01929                 EB += StepB;
01930             GradientColors[ColorCount - i - 1] = Color.FromArgb((byte)EA, (byte)ER, (byte)EG,
01931             (byte)EB);
01932         }
01933     }
01934
01935

```

```

01946         //add end color as last color
01947         GradientColors.Add(End);
01948
01949         //return theme with gradient colors
01950         return new Theme()
01951         {
01952             Name = Name,
01953             Id = 1,
01954             DarkMode = false,
01955             ColorCount = ColorCount,
01956             ColorList = GradientColors.ToArray()
01957         };
01958     }
01959
01960     public static Theme GetThemeByGradient(Color Start, Color Center, Color End, string Name =
"GradientTheme 1")
01961     {
01962         //create temporary color list starting with the start color
01963         List<Color> GradientColors = new List<Color>(7);
01964         GradientColors.Add(Start);
01965         for (int i = 1; i < 6; i++)
01966         {
01967             GradientColors.Add(Color.Black);
01968         }
01969
01970         //get RGB values of start and center color and calculate step size
01971         double SA = Start.A;
01972         double SR = Start.R;
01973         double SG = Start.G;
01974         double SB = Start.B;
01975         double EA = Center.A;
01976         double ER = Center.R;
01977         double EG = Center.G;
01978         double EB = Center.B;
01979         double StepA = Math.Abs(SA - EA) / 3;
01980         double StepR = Math.Abs(SR - ER) / 3;
01981         double StepG = Math.Abs(SG - EG) / 3;
01982         double StepB = Math.Abs(SB - EB) / 3;
01983
01984         //fill color list with gradient colors
01985         for (int i = 1; i < 3; i++)
01986         {
01987             if (EA > SA)
01988                 EA -= StepA;
01989             else
01990                 EA += StepA;
01991             if (ER > SR)
01992                 ER -= StepR;
01993             else
01994                 ER += StepR;
01995             if (EG > SG)
01996                 EG -= StepG;
01997             else
01998                 EG += StepG;
01999             if (EB > SB)
02000                 EB -= StepB;
02001             else
02002                 EB += StepB;
02003             GradientColors[3 + i] = Color.FromArgb((byte)EA, (byte)ER, (byte)EG, (byte)EB);
02004         }
02005
02006         //add center color
02007         GradientColors[3] = Center;
02008
02009         //get RGB values of center and end color and calculate step size
02010         SA = Center.A;
02011         SR = Center.R;
02012         SG = Center.G;
02013         SB = Center.B;
02014         EA = End.A;
02015         ER = End.R;
02016         EG = End.G;
02017         EB = End.B;
02018         StepA = Math.Abs(SA - EA) / 3;
02019         StepR = Math.Abs(SR - ER) / 3;
02020         StepG = Math.Abs(SG - EG) / 3;
02021         StepB = Math.Abs(SB - EB) / 3;
02022
02023         //fill color list with gradient colors
02024         for (int i = 3; i < 6; i++)
02025         {
02026             if (EA > SA)
02027                 EA -= StepA;
02028             else
02029                 EA += StepA;
02030             if (ER > SR)
02031                 ER -= StepR;
02032             else
02033                 ER += StepR;
02034         }
02035     }

```

```
02040         else
02041             ER += StepR;
02042         if (EG > SG)
02043             EG -= StepG;
02044         else
02045             EG += StepG;
02046         if (EB > SB)
02047             EB -= StepB;
02048         else
02049             EB += StepB;
02050         GradientColors[8 - i] = Color.FromArgb((byte)EA, (byte)ER, (byte)EG, (byte)EB);
02051     }
02052
02053     //add end color as last color
02054     GradientColors.Add(End);
02055
02056     //return theme with gradient colors
02057     return new Theme()
02058     {
02059         Name = Name,
02060         Id = 1,
02061         DarkMode = false,
02062         ColorCount = 7,
02063         ColorList = GradientColors.ToArray()
02064     };
02065 }
02066
02067 #endregion
02068 }
02069 }
```

Index

C:/Git/ThemesLib/Themes/Themes.cs, [19](#)

ColorCount

Feuster.Themes.Theme, [10](#)

ColorList

Feuster.Themes.Theme, [10](#)

DarkMode

Feuster.Themes.Theme, [10](#)

Feuster, [7](#)

Feuster.Themes, [11](#)

GetThemeByGradient, [12](#)

GetThemeById, [13](#)

GetThemeByName, [13](#)

GetThemeColor, [14](#), [15](#)

GetThemeColorCount, [16](#)

GetThemeNames, [17](#)

GetThemeNamesByColorCount, [17](#)

ThemeList, [17](#)

Feuster.Themes.Theme, [9](#)

ColorCount, [10](#)

ColorList, [10](#)

DarkMode, [10](#)

GetColor, [9](#)

Id, [10](#)

Name, [11](#)

GetColor

Feuster.Themes.Theme, [9](#)

GetThemeByGradient

Feuster.Themes, [12](#)

GetThemeById

Feuster.Themes, [13](#)

GetThemeByName

Feuster.Themes, [13](#)

GetThemeColor

Feuster.Themes, [14](#), [15](#)

GetThemeColorCount

Feuster.Themes, [16](#)

GetThemeNames

Feuster.Themes, [17](#)

GetThemeNamesByColorCount

Feuster.Themes, [17](#)

Id

Feuster.Themes.Theme, [10](#)

Name

Feuster.Themes.Theme, [11](#)

ThemeList

Feuster.Themes, [17](#)