30 Tiles

Ein wesentliches Merkmal von Windows Phone ist die Personalisierung der Startoberfläche durch eine Vielzahl von Kacheln, auch Tiles genannt. Prägnante Benachrichtigungen, Texte und Bilder können auf Tiles abgebildet werden, um den Benutzer einzuladen, die App zu starten.

30.1 Allgemeines

Das Framework des Windows Phone stellt drei verschiedene Tile-Templates zur Verfügung. Jedes dieser Templates hat unterschiedliche Eigenschaften, die Sie zur Darstellung von Inhalten nutzen können.

Abb. 30–1 *Verfügbare Tile-Templates*

Tiles können in drei verschiedenen Größen auf dem Startbildschirm angezeigt werden.

Small	159px x 159px
Medium	336px x 336px
Wide	691px x 336px

Tab. 30–1 *Mögliche Größen von Tiles*

30.2 Erstellung eines Tiles

Benötigte Referenz

using Microsoft.Phone.Shell;

Um ein Tile auf dem Startbildschirm zu erzeugen, wird die Klasse *ShellTile* verwendet. Sie bietet die statische Methode *Create* an.

```
01 ShellTile.Create(targetUri,
02                  tileData,
03                  supportsWideTile);
```

Listing 30–1 *Erzeugung eines Tiles*

Der erste Parameter ist die URI, zu der nach dem Tippen auf das Tile navigiert werden soll. Diese URI kann wie gewohnt Parameter enthalten, die die Zielseite verarbeiten kann.

Der zweite Parameter *tileData* ist eine Instanz der *ShellTileData*-Klasse, die alle Information für das Tile enthält.

Je nachdem, welches Tile-Template Sie verwenden, übergeben Sie eine der unten aufgeführten Klassen.

Tile-Template	zu verwendende Klasse
Flip	FlipTileData
Iconic	IconicTileData
Cycle	CycleTemplate

Tab. 30–2 *Verfügbare TileData-Klassen*

Der letzte Parameter *supportsWideTile* (siehe Listing 30–1 Zeile 03) ist optional. Es ist ein Boolean, mit dem gesteuert wird, ob ein Tile in einer Auflösung von 691 px × 336 px (vgl. Tab. 30–1) dargestellt werden darf.

Die folgenden Kapitel gehen auf die Verwendung der jeweiligen Tile-Templates ein.

30.3 Flip-Tile

Das Flip-Tile verfügt über folgende Eigenschaften.

Title	Der »Title/BackTitle« eines Tiles wird am unteren, linken Rand angezeigt.
BackTitle	
Count	Spezifiziert die Zahl, die in dem schwarzen Kreis dargestellt wird.

BackgroundImage	Hintergrundbild, das auf der Vorderseite des Tiles angezeigt wird.
BackBackgroundImage	Hintergrundbild, das auf der Rückseite des Tiles angezeigt wird.
BackContent	Der Text auf der Rückseite des Tiles
WideBackgroundImage	Wenn das Flip-Tile in den Wide-Modus versetzt wird, wird das entsprechend optimierte Hintergrundbild verwendet.
WideBackBackgroundImage	
WideBackContent	Wenn das Flip-Tile in den Wide-Modus versetzt wird, wird dieser Text auf der Rückseite angezeigt.
SmallBackgroundImage	Spezifiziert das Bild des Tiles bei einer Darstellung von 159 px × 159 px.

Tab. 30–3 *Eigenschaften des Flip-Tiles*

Abbildung 30–2 bis Abbildung 30–3 zeigen, wie die Darstellung der genannten Eigenschaften aussieht.

Abb. 30–2 *Small-Tile*

Abb. 30–3 *Medium-Tile (Vorder- und Rückseite)*

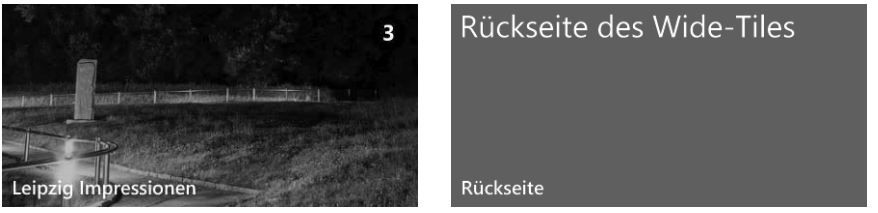

Abb. 30–4 *Wide-Tile (Vorder- und Rückseite)*

Den dazugehörigen Code finden Sie im sich anschließenden Listing.

```
01 var tile = ShellTile.ActiveTiles.FirstOrDefault();
02
03 if (tile != null)
04 {
05   var fliptile = new FlipTileData
06   {
07     Title = "Vorderseite",
08     Count = 9,
09     BackTitle = "Rückseite",
10     BackContent = "Inhalt Rückseite",
11     WideBackContent = "Rückseite des Wide-Tiles",
12     SmallBackgroundImage = new
         Uri("Assets/FlipTile/159.jpg",
         UriKind.Relative),
12     BackgroundImage = new
         Uri("Assets/FlipTile/336.jpg",
         UriKind.Relative),
13     WideBackgroundImage = new
         Uri("Assets/FlipTile/wide.jpg",
         UriKind.Relative)
14   };
15
16   tile.Update(fliptile);
17 }
```

Listing 30–2 *Beispiel zur Erstellung eines Flip-Tiles*

30.4 Iconic-Tile

Das Iconic-Tile verfügt über folgende Eigenschaften.

Title	Der »Title« eines Tiles wird am unteren, linken Rand angezeigt.
Count	Der Wert »Count« spezifiziert die Zahl, die in dem schwarzen Kreis dargestellt wird.
BackgroundColor	Die Rückseite des Iconic-Tiles kann verändert werden. Wird dieser Wert nicht gesetzt, wird automatisch die aktuelle Akzentfarbe des Windows Phone verwendet.
IconImage	Dieses Icon wird für Tiles in den Größen Medium und Wide verwendet.
SmallIconImage	Dieses Icon wird für Tiles in der Größe Small verwendet.
WideContent1	Zeile 1 für Inhalte des Wide-Tiles
WideContent2	Zeile 2 für Inhalte des Wide-Tiles
WideContent3	Zeile 3 für Inhalte des Wide-Tiles

Tab. 30–4 *Eigenschaften des Iconic-Tiles*

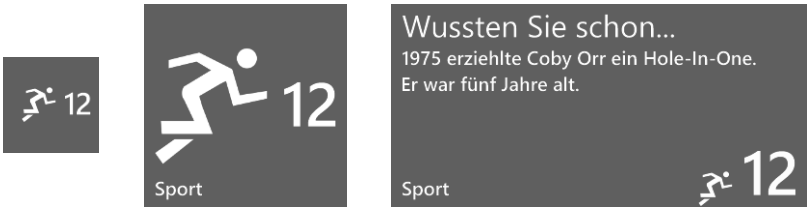

Abb. 30–5 *Repräsentation des Iconic-Tiles in verschiedenen Größen*

Hinweis

Windows Phone skaliert die Icons automatisch, dennoch ist es empfehlenswert, bei dem *SmallIconImage* eine Größe von 38 px × 38 px zu verwenden, damit die Eigenschaft Count vollständig dargestellt werden kann.

Für das Erstellen eines Iconic-Tiles kommt die Klasse *IconicTileData* zum Einsatz.

```
01 private void OnCreatingIconicTile(object sender,
                 GestureEventArgs e)
02 {
03   var icontile = new IconicTileData
04   {
05     Title = "Sport",
06     Count = 12,
07     IconImage =
         new Uri("Assets/IconicTile/202.png",
                 UriKind.Relative),
08     SmallIconImage =
         new Uri("Assets/IconicTile/110.png",
                 UriKind.Relative),
09     WideContent1 = "Wussten Sie schon...",
10     WideContent2 = "1975 erzielte Coby Orr ein
                       Hole-In-One.",
11     WideContent3 = "Er war fünf Jahre alt."
12   };
13
14   ShellTile.Create(new Uri("/MainPage.xaml",
                              UriKind.Relative),
                      icontile,
                      true);
15 }
```

Listing 30–3 *Beispiel zur Erstellung eines Iconic-Tiles*

30.5 Cycle-Tile

Die Besonderheit des Cycle-Tiles besteht darin, dass es bis zu 9 Bilder in einer Diashow anzeigen kann.

Des Weiteren können die nachstehend aufgeführten Eigenschaften gesetzt werden.

Title	Der »Title« eines Tiles wird am unteren, linken Rand angezeigt.
Count	Spezifiziert die Zahl, die in dem schwarzen Kreis dargestellt wird.
SmallBackgroundImage	Dieses Icon wird für Tiles in der Größe Small verwendet.
CycleImages	Eine Liste, die bis zu 9 Bilder verwaltet, die auf dem Tile angezeigt werden. Dies gilt nur für die Größen **Medium** und **Wide**.

Tab. 30–5 *Eigenschaften des Cycle-Tiles*

```
01 private void OnCreatingCycleTile(object sender,
                                GestureEventArgs e)
02 {
03   var cycleTile = new CycleTileData
04   {
05     Title = "Leipzig Impressionen",
06     Count = 3,
07     SmallBackgroundImage =
         new Uri("Assets/CycleTile/Tiles-2.jpg",
                 UriKind.Relative),
08     CycleImages = new List<Uri>
09     {
10       new Uri("Assets/CycleTile/Tiles-1.jpg",
                 UriKind.Relative),
11       new Uri("Assets/CycleTile/Tiles-2.jpg",
                 UriKind.Relative),
12       new Uri("Assets/CycleTile/Tiles-3.jpg",
                 UriKind.Relative),
13       new Uri("Assets/CycleTile/Tiles-4.jpg",
                 UriKind.Relative),
14       new Uri("Assets/CycleTile/Tiles-5.jpg",
                 UriKind.Relative),
15       new Uri("Assets/CycleTile/Tiles-6.jpg",
                 UriKind.Relative),
16       new Uri("Assets/CycleTile/Tiles-7.jpg",
                 UriKind.Relative),
17       new Uri("Assets/CycleTile/Tiles-8.jpg",
                 UriKind.Relative),
18       new Uri("Assets/CycleTile/Tiles-9.jpg",
         UriKind.Relative),
19     }
20   };
21
```

```
22  ShellTile.Create(
        new Uri("/MainPage.xaml?shellTemplateType=
                                    CycleTemplate",
23              UriKind.Relative),
24              cycleTile,
25              true);
26 }
```

Listing 30–4 *Beispiel zur Erstellung eines Cycle-Tiles*

30.6 Tiles anhand ihrer URI aktualisieren

Möchte man ein bestehendes Tile mit der *ShellTile.Update*-Methode aktualisieren, muss man dessen Instanz laden. Anhand der URI kann ein Tile identifiziert werden. Es ist eine eindeutige Identifikation, da zu einer URI nur ein Tile existieren darf, sonst wird durch das Framework eine *Exception* geworfen.

```
01 public static ShellTile GetByUri(Uri targetUri)
02 {
03   var tile = ShellTile.ActiveTiles
                 .FirstOrDefault(u =>
                    u.NavigationUri.Equals(targetUri));
04
05   return tile;
06 }
```

Listing 30–5 *Ein Shell-Tile anhand der URI laden*

Mithilfe des übergebenen Lambda-Ausdrucks in Zeile 03 kann aus der *IEnumerable ActiveTiles* das richtige Tile geladen werden.

30.6.1 Exist-Erweiterung

Die Methode *GetByUri* aus Listing 30–5 kann als Grundlage für eine weitere Hilfsmethode verwendet werden, um zu erfahren, ob ein Tile existiert, woraufhin man beispielsweise in einer Einstellungen-Seite einen optischen Hinweis platzieren kann, dass die Tiles aktiviert wurden.

Abb. 30–6 *Beispiel einer Seite zur Einstellung von Tiles*

```
01 public static bool Exists(Uri targetUri)
02 {
03   return GetByUri(targetUri) != null;
04 }
```

Listing 30–6 *Exist-Methode für die Prüfung eines Tiles*

30.7 Tiles löschen

Die Methode *GetByUri* aus Listing 30–5 leistet auch für das Löschen gute Dienste.

Nach Aufruf dieser Methode bekommt man eine Instanz des Tiles zurück und kann die *Delete*-Methode der Klasse *ShellTile* nutzen.

```
01 public static void Remove(Uri targetUri)
02 {
03   var tile = GetByUri(targetUri);
04   if (tile != null)
05   {
06     tile.Delete();
07   }
08 }
```

Listing 30–7 *Methode zum Löschen eines Tiles*

30.8 Typisierung von Tiles

Möchten Sie Ihrem Benutzer die Wahl lassen, welchen Tile-Typ er verwendet? Dann liefert Ihnen dieser Abschnitt einen Lösungsansatz.

Es besteht das Problem, dass je URI nur ein Tile existieren darf. Um etwaige Konflikte zwischen Tiles zu vermeiden, die auf die gleiche Seite verweisen, kann der Tile-URI bei der Erstellung des Tiles ein Parameter angehangen werden.

Jede Tile-Kategorie bekommt einen anderen Wert für den Parameter. So werden die Tiles wieder ein-eindeutig.

Folgendes Listing zeigt eine *CreateOrUpdate*-Methode für Tiles, in der die Typisierung vorgenommen wird.

```
01 public static void CreateOrUpdate<T>
                          (this ShellTile tile,
02                         Uri targetUri,
03                         T tileData,
04                         bool supportsWideTile)
                           where T : ShellTileData
05 {
06   if (tile == null)
07   {
08     ShellTile
         .Create(targetUri.TypeForTileTemplate<T>(),
                 tileData,
                 supportsWideTile);
09   }
10   else
```

```
11   {
12     tile.Update(tileData);
13   }
14 }
```

Listing 30–8 *Typisierung eines Tiles anhand der URI*

Listing 30–8 zeigt eine generische Methode. Der generische Parameter *T* ist nötig, um je nach *ShellTileData*-Instanz (Flip, Iconic oder Cycle) die Typisierung vorzunehmen. Dies erfolgt in Zeile 08 mit der Erweiterungsmethode *TypeForTileTemplate*. Diese prüft den Typ der übergebenen *ShellTileData*-Instanz und setzt den entsprechenden Parameter.

```
01 public static Uri TypeForTileTemplate<T>
                     (this Uri uri)
                      where T : ShellTileData
02 {
03   string pathAndQuery = uri.OriginalString;
04
05   pathAndQuery += pathAndQuery.EndsWith(".xaml")
06       ? TileParameters.ParameterInitializer
07       : TileParameters.ParameterConcatenator;
08
09   if (typeof(FlipTileData) == typeof(T))
10   {
11     pathAndQuery += TileParameters.FlipTileParameter;
12   }
13   else if (typeof(IconicTileData) == typeof(T))
14   {
15     pathAndQuery +=
         TileParameters.IconicTileParameter;
16   }
17   else if (typeof(CycleTileData) == typeof(T))
18   {
19     pathAndQuery +=
         TileParameters.CycleTileParameter;
20   }
21   else
22   {
       throw new ArgumentException(
         String.Format("The ShellDataType '{0}'
                       is not supported.",
                       typeof (T).Name));
26   }
23
24   return new Uri(pathAndQuery,
                    UriKind.RelativeOrAbsolute);
25 }
```

Listing 30–9 *Erweiterungsmethode zur Typisierung eines Tiles*

Die URI wird in Zeile 03 in einen String umgewandelt, um zu prüfen, ob die URI bereits Parameter enthält oder ob der Typisierungsparameter der Erste ist (Zeile 05 ff.). Dabei wird die Klasse *TileParameters* verwendet. Sie besitzt statische Eigenschaften vom Typ *String*, die die Werte der Parameter enthalten. Diese Klasse wurde eingeführt, um bei späterer Wiederverwendung Tippfehler zu vermeiden und die Lesbarkeit des Codes zu verbessern.

Falls eine *ShellTileData*-Klasse als generischer Operator übergeben wird, die die Erweiterungsmethode nicht unterstützt, wird eine *ArgumentException* geworfen.