# Resource

Met resources kunt u data opslaan die u later in de programma kunt hergebruiken. U kunt een resource hergebruiken met twee types:

* Static Resource
* Dynamic Resource

**Waarom resources**Omdat veel stijlen en opmaak hetzelfde moeten worden gehouden zijn er variabelen nodig om het hierin op te slaan. Ook moet het meteen duidelijk zijn in de code welke opmaak er op is toegepast. Dit wordt gedaan met behulp van resources.   
  
Resources kunt u aanmaken in XAML of C#. Deze resources moeten dan eerst wel in een ResourceDictionary worden opgeslagen zodat het in blokken kan worden aangeroepen.  
  
**Noodzakelijk voor resources**Omdat elke resource moet worden geïdentificeerd moet deze een unieke key hebben. Dit wordt gedaan met static- of dynamic resource. Hierdoor weet het programma welke resource kan worden hergebruikt. Deze resource key geeft u als type mee in de resource tag.  
  
**Variabel**Het belangrijkste van een resource is het feit dat de resources variabel zijn. Hierbij kunt u eigenschappen aan resources toepassen en deze wijzigen. Wel moet er dan rekening gehouden worden met static resources.

## Static resource

Static resources worden voor het compileren uitgevoerd. Dit betekent dat static resources eenmaal wordt uitgevoerd. Tijdens het uitvoeren van de applicatie kunnen hierbij dan geen veranderingen worden aangepast. Static resources kunnen hierbij dan niet worden gerefereerd. De volgende XAML code laat zien hoe u een static resource aanroept:



**StaticResource:** *De resource type die u aangeeft voor het aanroepen van een bepaalde resource.*

**Key:** *De identiteit van de resource die u aanroept.*

**SNELLE OPZET**Een static resource zet u als volgt op:

1. Maak een nieuwe **.XAML** aan in **Visual Studio**
2. Maak een **ResourceDictionary tag** aan

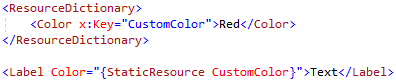


1. Maak in de **ResourceDictionary** een resource aan. Dit mag elke resource zijn die u wilt. In dit geval wordt er een **resource** **Color** aangemaakt met een **x:Key=”CustomColor”.**



1. Als laatst roept u de resource aan met **static resource**.

Eindresultaat

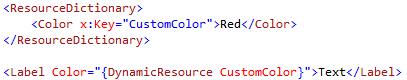


## Dynamic resource

Met dynamic resources kunt u een resource wijzigen tijdens het uitvoeren van de applicatie. Wat bij static resources het geval was dat je voor het compileren resources kunt uitvoeren en wijzigen is bij dynamic resources niet het geval.   
  
Deze kunt u dan alleen maar voor gebruik maken tijdens uitvoering. Het aanroepen van een dynamic resource lijkt heel veel als het aanroepen van de static resource. Hierbij is het enige verschil dat u in plaats van StaticResource een DynamicResource aanroept. De volgende opzet XAML code laat zien hoe.



**Refereer in C#**Het refereren met dynamic resource werkt wel en bij static resources niet omdat deze tijdens het uitvoeren van de applicatie kan worden gewijzigd. Hierbij wordt er van de volgende XAML code vanuit gegaan:



Om de CustomColor te refereren doet u als volgt:

  
Hiermee kunt u gemmakelijk de resources hergebruiken op elke gepaste object in code.

## Waarom static- of dynamic resource

Omdat een static resource alleen voor uitvoering van de applicatie wordt uitgevoerd is het vaak handig om met behulp van static resources een template te maken.   
  
Dit omdat er in templates niets veranderd hoeft te worden omdat deze resources niet dynamisch zijn. Het grote voordeel hierbij is dat de snelheid wordt behouden. Dit is niet het geval bij dynamic resources. Wel kunt u de applicatie updaten met behulp van dynamic resources. Hiermee wordt bedoeld dat u controls kunt maken die u wanneer u wilt kan aanpassen.  
  
Met static resources met behulp van ResourceDictionary kunt u verschillende thema’s maken. Als u dit met dynamic resources probeert kunnen de thema’s er heel anders uitzien omdat er tijdens uitvoering van de applicatie resources kunnen worden gewijzigd en niet statisch zijn.

Het beste is om altijd te beginnen met static resources. Alleen als u zeker weet dat u dynamic resources nodig heeft gebruikt u deze.   
  
De volgende voor- en nadelen nogmaals op een rijtje:  
  
**Static resource**

|  |  |
| --- | --- |
| + | - |
| Snelheid | Niet aan te passen tijdens uitvoering van de applicatie |
| Geschikt voor templates | Kan langer duren voordat de applicatie wordt opgestart |

**Dynamic resource**

|  |  |
| --- | --- |
| + | - |
| Updaten van de programma | Snelheid |
|  | Kan geen uitvoeringen doen voor het compileren |
|  | Niet geschikt voor templates |