

# Qt ve QSettings

## Kaya Oğuz

<[kaya@kuzeykutbu.org](mailto:kaya@kuzeykutbu.org)>

Bu belge ve içerisindeki kodlar GPL lisanslıdır.

---

## QSettings Nedir?

Kullandığınız her programın bir "SETTINGS" bölümü vardır. Buradan programla ilgili ayarları yaparsınız. Önemli olan bu ayarları bir şekilde program tekrar başladığında uygulamaktır.

Bunun için birçok yöntem var. Örneğin Linux altında birçok program home dizininizde gizli dosyalar oluşturur. Bunlar genelde rc ile bitirler ve programınızın ayarlarını tutarlar. Yapı olarak programcının izlediği bir biçimde olsalar da şekilleri

```
use_font = Sans, 10
show_line_numbers = 1
tab_space = 3
```

gibi ayarların karşılıklarında değerleri duracak şekildedir. Bunları program içinden okutmak ve ayarlar değiştiğinde tekrar kaydetmeniz gerekecek, bunun için metodlar yazmanız gerekecek, dosyadaki bir hata belki bütün ayarların bozulmasına neden olacaktır.

Bütün bunların üstesinden gelmek için QSettings denen sınıfı kullanabilirsiniz! QSettings, her platforma özel şekilde, ayarlarınızı tutmak, düzenlemek, çağırmak ve kaydetmek gibi bütün işlemleri gören bir sınıf. Linux altında yukarıdakine benzer şekilde ayarları tutarken, Windows altında bu değerleri Registry'de tutmaktadır. Mac altında ne yapıyor bilmiyorum :) Ama sonuçta programın ayarlarını kontrol etmek için QSettings işimizi çok rahat bir şekilde halledecektir.

## Nasıl Kullanacağım?

QSettings'i kullanmadan önce karar vermeniz gereken şirket adıdır. Bir şirket adınız yoksa benim gibi Java'daki paket mantığı ile alan adını kullanabilirsiniz. Bir de programınızın adı gerekiyor elbette. Bu sayede programınıza özel ayarları yaratıyor, ya da ulaşıyorsunuz:

```
#include <QSettings>
...
QSettings ayarlar("KuzeyKutbu.Org", "QMetin");
```

Bunu programınızda istediğiniz yerde kullanabilirsiniz. Ama en mantıklısı program ilk yüklenirken, ana pencerenin kurucusunda (constructor) kullanmanız daha iyi olacaktır. Hazır pencere kurulurken gerekli ayarları uygulayabilirsiniz. Nasıl

uygulayacaksınız peki? Yukardaki koda devam edelim:

```
bool showLineNumbers = ayarlar.value("editor/lineNumbers", true).toBool();
int tabSpace = ayarlar.value("editor/tabSpace", 3).toInt();
```

Neler oluyor? Öncelikle QSettings ile dönen değerler QVariant tipindedir. "O ne?" dersanız, void değişken tipi birşey, yani aslında çok benzemese de benzediği yer şu: Dönen değeri istediğiniz bir tipe "cast" edebilirsiniz. Cast etmek nedir diyorsanız, değişken tipini başka bir değişken tipine çevirmek diyebilirim. Tabii bir float değerini int'e cast ederseniz, diyelim 2.12 sayısını, o zaman o nokta 12 gider, 2 kalır. Ama int'i float'a cast ederseniz, diyelim 2'yi, o zaman 2.0 gibi olur. QVariant diğer bütün sınıflara değer kaybı olmadan cast edilebilen geniş bir sınıftır, içiniz rahat rahat kullanabilirsiniz, hatta kullanınız :)

Peki value() metodundaki ikinci değişken nedir? Bu öntanımlı, ya da varsayılan deneniştir. Yani ayarlarda böyle bir değer yoksa bu değeri kullan demektir.

Yeni değerleri nasıl kaydedeceğiz? Bunu daha çok ayarlar penceresindeki "Tamam" tuşuna basıldıktan sonra yapacaksınız :) Tek yapmanız gereken ayarları setValue ile kaydetmek:

```
ayarlar.setValue("editor/lineNumbers", false);
ayarlar.setValue("editor/tabSpace", 4);
```

Burada false ve 4 yazan yerlere yüksek ihtimalle arayüzden gelen değerleri koyacaksınız. Peki şu editor/tabSpace derken editor/ ne işe yarıyor?

Bir programın birden çok farklı ayarı olabilir. Düzenleyici ile ilgili ayarlar, programın genel ayarları, veritabanı ayarları vs. Bunları istediğiniz başlıklar altında toplayabilirsiniz.

## Daha fazla bilgi...

Daha fazla bilgi için Assistant'ın QSettings maddesine bakabilir, daha önce bu özelliği kullanan programlara [\[1\]](#) bakabilirsiniz :)