

# HISSE ANALIZ UYGULAMASI

## GUNCELLEME REHBERI

Her yeni ozellik veya duzeltme icin bu adimlari takip et.

### ADIM 1: Kod Degisikliklerini Yap ve Test Et

- Visual Studio'da kodunu yaz
- F5** ile test et (Debug mode)
- Tum ozelliklerin calistigindan emin ol
- Hatalari duzelt

### ADIM 2: Versiyon Numarasini Artir

- Solution Explorer** → **HisseAnalizUygulamasi.csproj** dosyasini ac
- Versiyon numaralarini degistir:

**Onceki:**

```
<AssemblyVersion>1.0.0.0</AssemblyVersion> <FileVersion>1.0.0.0</FileVersion>  
<Version>1.0.0</Version>
```

**Yeni (ornek):**

```
<AssemblyVersion>1.0.1.0</AssemblyVersion> <FileVersion>1.0.1.0</FileVersion>  
<Version>1.0.1</Version>
```

- Ctrl+S** ile kaydet

Degisiklik Turu	Ornek
Kucuk bug fix	1.0.0 → 1.0.1
Yeni ozellik	1.0.1 → 1.1.0
Buyuk degisiklik	1.1.0 → 2.0.0

## ADIM 3: update.xml Dosyasini Guncelle

1. Proje root klasorunde **update.xml** dosyasini ac

2. Su 3 yeri degistir:

```
<version>1.0.1.0</version> <url>...v1.0.1/HisseAnalizUygulamasi-v1.0.1.zip</url>  
<changelog>...tag/v1.0.1</changelog>
```

### Degisen yerler:

- <version> tag'i: **1.0.1.0**
- URL'deki tag: **v1.0.1**
- ZIP dosya adi: **HisseAnalizUygulamasi-v1.0.1.zip**
- Changelog tag'i: **v1.0.1**

3. **Ctrl+S** ile kaydet

## ADIM 4: Release Build Yap

İki yontemden birini secebilirsin:

### YONTEM A: release.ps1 Script ile (Hizli)

1. **PowerShell** veya **Terminal** ac
2. Proje klasorune git
3. Script'i calistir

```
cd C:\Users\[KULLANICI]\source\repos\HisseAnalizUygulamasi .\release.ps1 -version "1.0.1"
```

**NOT:** Versiyon numarasini .csproj'daki ile ayni yaz (son .0'i at)

#### Bu script yapacaklar:

- Eski publish klasorunu temizler
- Release build yapar
- ZIP olusturur
- Git commit yapar
- Tag olusturur
- GitHub'a push yapar

Script bittiginde ADIM 5'e gec.

### YONTEM B: Manuel Komutlar (PS1 kullanmadan)

1. **Eski publish klasorunu temizle**

```
Remove-Item -Path "publish" -Recurse -Force -ErrorAction SilentlyContinue
```

2. **Release build yap**

```
dotnet publish -c Release -r win-x64 --self-contained -p:PublishSingleFile=true -o publish/
```

3. **ZIP olustur**

```
cd publish Compress-Archive -Path * -DestinationPath ../HisseAnalizUygulamasi-v1.0.1.zip -Force cd ..
```

4. **Git commit ve tag**

```
git add . git commit -m "v1.0.1 - [Aciklama]" git tag v1.0.1
```

5. **GitHub'a push**

```
git push origin main git push origin v1.0.1
```

#### Sonuc:

- **HisseAnalizUygulamasi-v1.0.1.zip** olusturuldu

- Git'e commit ve tag yapıldı
- GitHub'a push yapıldı

## ADIM 5: GitHub'da Release Olustur

### 5.1 GitHub'a Git

<https://github.com/faikalbayrak/HisseAnalizUygulamasi/releases>

### 5.2 Yeni Release Olustur

"Draft a new release" butonuna tikla (sag ustte yesil buton)

### 5.3 Bilgileri Doldur

Choose a tag:

Dropdown'dan **v1.0.1** sec

Release title:

v1.0.1 - [Kisa aciklama]

Ornekler:

- v1.0.1 - Banka Uyari Sistemi
- v1.0.1 - Bug Duzeltmeleri
- v1.0.1 - Performans iyilestirmeleri

Describe this release:

```
## Degisiklikler

### Yeni Ozellikler:
- [Yeni ozellik 1]
- [Yeni ozellik 2]

### Duzeltmeler:
- [Bug duzeltmesi 1]
- [Bug duzeltmesi 2]

### Gelistirmeler:
- [Iyilestirme 1]
- [Iyilestirme 2]
```

### 5.4 ZIP Dosyasini Yukle

Proje klasorundeki **HisseAnalizUygulamasi-v1.0.1.zip** dosyasini surukle birak

### 5.5 Yayinla

"Publish release" yesil butonuna tikla

## ADIM 6: Kontrol Et

### 6.1 Release'in Yayinlandigini Dogrula

- GitHub releases sayfasinda **v1.0.1** görünmeli
- ZIP dosyasi "Assets" altinda görünmeli
- Tikla ve indirebiliyor musun kontrol et

### 6.2 update.xml'in Dogru Oldugunu Kontrol Et

Tarayicida sunu ac:

```
https://raw.githubusercontent.com/faikalbayrak/HisseAnalizUygulamasi/main/update.xml
```

## ADIM 7: Arkadasin Test Etsin

### Arkadasinin Yapmasi Gerekenler:

1. Eski versiyonu calistir
2. Guncelleme pop-up'i gelecek
3. **"Update"** tikla
4. Otomatik indirir
5. Uygulamayi kapatir
6. Guncellemeyi kurar
7. Uygulamayi yeniden baslatir
8. **Yeni versiyon calisiyor!**

TAMAMLANDI!

Guncelleme basariyla yayinlandi ve otomatik guncelleme sistemi calisiyor!

## Hizli Kontrol Listesi

### YONTEM A: release.ps1 Script Kullanarak

#	Gorev
1	Kod degisikligi yap ve test et
2	.csproj'da versiyon artir (orn: 1.0.1.0)
3	update.xml'de 3 yeri guncelle
4	.\release.ps1 -version "1.0.1" calistir
5	GitHub'da "Draft a new release" tikla
6	Tag sec (v1.0.1) ve bilgileri doldur
7	ZIP yukle ve "Publish release" tikla
8	Arkadasin test etsin

### YONTEM B: Manuel Komutlarla (PS1 olmadan)

#	Gorev
1	Kod degisikligi yap ve test et
2	.csproj'da versiyon artir (orn: 1.0.1.0)
3	update.xml'de 3 yeri guncelle
4	Remove-Item publish (eski temizle)
5	dotnet publish (build)
6	Compress-Archive (ZIP olustur)
7	git commit + git tag
8	git push (main ve tag)
9	GitHub'da release olustur ve ZIP yukle
10	Arkadasin test etsin





## Sik Sorulan Sorular

### Versiyon numaralari nereye yaziliyor?

Dosya	Format	Ornek
.csproj	X.X.X.0	1.0.1.0
update.xml	X.X.X.0	1.0.1.0
PS1 script	X.X.X	1.0.1
Git tag	vX.X.X	v1.0.1
GitHub Release	vX.X.X	v1.0.1

### Script calismiyor, ne yapmaliyim?

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser .\release.ps1  
-version "1.0.1"
```

### Tag zaten var diyor?

```
git tag -d v1.0.1 git push origin :refs/tags/v1.0.1 git tag v1.0.1 git push origin  
v1.0.1
```

### Guncelleme gelmiyor?

- update.xml versiyon numarasini kontrol et
- GitHub release'in "Published" oldugunu kontrol et
- ZIP dosyasinin dogru yuklendigini kontrol et
- Uygulamayi **tamamen kapat** ve yeniden ac
- Internet baglantisini kontrol et

### PS1 yerine hangi yontemi secmeliyim?

Kriter	PS1 Script	Manuel
Hiz	Hizli (tek komut)	Yavas (5 komut)
Kontrol	Otomatik	Tam kontrol
Hata Riski	Dusuk	Orta
Onerilen	Yeni baslayanlar	Deneyimli

## Sonraki Guncelleme Icin

Bu dokumani **favorilere ekle** veya **yazdir**.

Her guncelleme yapacaginda bu adimlari takip et!

Basarilar!