

# Belgelendirme Çalışma Grubu Elkitabı

Hazırlayan:  
**Nilgün Belma Bugüner**

<nilgun (at) belgeler-gen-tr>

Son güncelleme: Aralık 2006

## Özet

Bu bölüm çalışma grubumuzun görev alanı, sanalyöremizin yapısı hakkında bilgi vermek, çeviri ekipleri, çevirmen, yazar olarak çalışmalara katılan ya da katılmak isteyenlere çalışmalarında yardımcı olmak ve yönlendirmek, onların çalışmalarının izlenmesini sağlamak gibi bilgilendirme yanında çalışmalarımızın izlenmesini kolaylaştırmak amacıyla hazırlanmıştır.

## İçindekiler

<b>1. Linux Belgelendirme Çalışma Grubu</b>	3
1.1. Görev Alanı	3
1.2. Görev Dağılımı	3
<b>2. Sanalyöremiz</b>	4
2.1. Site içeriğinden Diğer Dosya biçimleri Nasıl Elde Edilir?	4
2.1.1. Salt Metin Belgeler Nasıl Üretilir?	4
2.2. Linux Kitaplığı	5
2.2.1. Çapraz Bağ Özelliği	5
2.2.1.1. Sözlüğün Kullanımı	5
2.2.2. Genişleme özelliği	6
2.2.3. Yansılama kolaylığı	6
<b>3. Ne NASIL yapılır?</b>	6
3.1. Belgelerin Kabulü	6
3.1.1. Genel Kurallar	6
3.1.2. Çeviri Yaparken Nelere Dikkat Etmeli?	8
3.1.3. Yeni Belge Hazırlarken Nelere Dikkat Etmeli?	9
<b>4. Linux Belgeleri NASIL</b>	11
4.1. Giriş	11
4.2. Bir Belge Hazırlamaya Başlamadan Önce	11
4.3. Bir XML Dosyasının Yapıtaşları	13
4.3.1. Etiketler	13
4.3.2. Açıklamalar, Yorumlar	14
4.3.3. İlk satır ve Belge Türünün Belirtilmesi	14
4.4. Docbook Elemanları	15
4.5. Hata Ayıklama	19
<b>A. Sample-HOWTO.xml dosyası</b>	21

## Teşekkür

"Ne Nasıl Yapılır" kısmındaki "Çeviri Yaparken ve Yeni Belge Hazırlarken Nelere Dikkat Edilmeli" bölümlerini yazan **Deniz Akkuş**'a teşekkür ederim.

-- NBB

## Biz kimiz?

Biz, Türk Linux Kullanıcılarının Linux sistemlerini anadilindeki kaynaklardan yararlanarak kullanmaları için çalışmalar yapan gönüllüleriz.

# 1. Linux Belgelendirme Çalışma Grubu

## 1.1. Görev Alanı

### 1. Belgelerin dilimize çevrilmesi

Bu proje kapsamında, her türlü belgenin dilimize çevrilmesi öngörülmüştür. Bu kapsamda:

- Nasıl Belgelerinin dilimize çevrilmesi,
- Linux Rehber Kitaplarının dilimize çevrilmesi,
- Kılavuz (man) Sayfalarının dilimize çevrilmesi,
- Info belgelerinin dilimize çevrilmesi,
- Bunların dışında kalan her türlü yardım belgesinin dilimize çevrilmesi,
- Yukarıda anılan belge sınıflandırmalarına girmeyen ancak Linux kullanıcılarına faydalı olabilecek her türlü belgenin çevirilerinin yapılması

öngörülmüştür.

### 2. Linux Kullanıcılarının hazırladıkları belgeleri yayınlamak.

### 3. Yukarıda anılanlardan daha önce yapılmış çalışmaları biraraya toplamak.

### 4. Daha önce başlamış belge çeviri projelerini aynı çatı altına toplamak ve bir havuz oluşturmak.

### 5. Belgelerin biçim dönüşümleri için alt yapı çalışmaları yapmak.

SGML, Man, Info gibi belge biçimlerinin XML biçime ve XML biçimden Man, Info, HTML, PS, PDF v.s. belge biçimine dönüştürülmesi için sanalyöremizin alt yapısına uygun yazılım geliştirme çalışmaları yapmak ve bu tür araçları sağlayan yazılımlar hakkında bilgilendirmek.

### 6. Linux Bilişim Terimleri Sözlüğünün geliştirilmesi.

### 7. Tüm çalışmaların sanalyöremizde yayınlanması.

## 1.2. Görev Dağılımı

### Grup Koordinatörleri

#### Nilgün Belma Bugüner <nilgun (at) belgeler-gen-tr>

Sitenin Docbook–XML alt yapısının mimarı, grubun kurucusu ve yöneticisidir. Sanalyöremizin güncellenmesinden ve güncel tutulmasından sorumludur. Özgeçmişi hakkındaki bilgileri [kişisel sayfasında](#)<sup>(B1)</sup> bulabilirsiniz.

#### Deniz Akkuş <deniz (at) arayan.com>

GNU–TR Projesi koordinatörü, Turkish–HOWTO dahil pek çok belgenin yazarı ve çevirmeni ve grubumuzun yöneticisidir. Resmi yansılarımızın koordinasyonu, dağıtım paketlerinin hazırlanması gibi daha çok alt yapıya özgü çalışmaları yürütür. Özgeçmişi hakkındaki bilgileri [kişisel sayfasında](#)<sup>(B2)</sup> bulabilirsiniz.

Grubumuz sürekli kendini geliştirmeyi ilke edinmiştir. Bu bağlamda grup koordinatörlerinin yükünü hafifletecek alt görevleri kendileri tespit edip, bunları uygulamaya sokabilecek özgüdüye sahip üyelerimize bu görev alanları için kendilerine şans tanıyacağız ve eğer kabul ederlerse onlara yardımcı olmaktan zevk alacağız.

Grubumuza katılmak için [belgeler-grubu \(at\) lists.sourceforge.net](mailto:belgeler-grubu@lists.sourceforge.net) listesine üye olmanız yeterlidir. Üye olmak için [buraya](#)<sup>(B3)</sup> tıklayınız ve listemize kendinizi kaydediniz.

## 2. Sanalyöremiz

Sanalyöremiz [GNU/Linux Kılavuz Sayfaları Çeviri Ekibinin Çalışmaları](#)<sup>(B4)</sup>nı içeren bölüm hariç, salt HTML biçimindedir.

Linux Kitaplığı bir Docbook–XML *set*'idir. Okumakta olduğunuz bu bölüm ise bir Docbook–XML *book*'udur. HTML dosyalar [KİTAPLIK](#)<sup>(B5)</sup> ve bu sayfayı içeren *kitap* dizinleri altında bir xslt dönüşümü ile otomatik olarak bu kodlardan üretilir.

Sanalyöremizin XML taslakları [DocBook XML V4.1.2 DTD](#)<sup>(B6)</sup> paketi sanalyöremiz için özelleştirilerek oluşturulmuştur. XSL dönüşümü için *xsl–stylesheet–1.49* paketinden faydalanılmış ancak bu paket sanalyöremiz için özelleştirilmiştir. Sanalyöremizin kodu standart *xsl–stylesheets–1.49*<sup>(B7)</sup> paketiyle çalışmaz. Sadece bol bol hata üretir. Bu bakımdan denemek isterseniz, bu paketi anonim CVS erişimimizi kullanarak almanızı öneririz.

Sanalyöremizi oluşturan taslaklar *cvs* ile alınabilir. Sanalyöremizin XML taslakları projemize alt yapısını sunan SourceForge CVS sunucusunda bulunmaktadır. Bu taslaklara *webcvs* üzerinden erişebileceğiniz gibi anonim CVS erişimi ile doğrudan makinanıza indirebilirsiniz. İnternet'e bağlı değilseniz bağlanın ve konsolda (ya da *xterm*'de):

```
cvs -d:pserver:anonymous@belgeler.cvs.sourceforge.net:/cvsroot/belgeler login
```

komutunu girdikten sonra sizden parola istenecek. Birşey yazmadan <Enter> tuşuna basın ve:

```
cvs -z3 -d:pserver:anonymous@belgeler.cvs.sourceforge.net:/cvsroot/belgeler \  
co -P sitesrc
```

komutunu verin. Kısa bir süre sonra dosyalar makinanıza akmaya başlayacak. İşlem tamamlandığında bulunduğunuz dizinin içinde bir dizin, *sitesrc* dizinini göreceksiniz. Sanalyöremizi oluşturan tüm kodu bu dizinin altında bulacaksınız.

Sanalyöremizi yansılamak için halihazırda *rsync* kullanıyoruz. Resmi yansımızı sunmak isterseniz, grup koordinatörlerimizle iletişime geçiniz.

Proje sayfamızın adresi <http://sourceforge.net/projects/belgeler> olup burada sayamadığımız ancak SourceForge tarafında projelerin izlenmesine yönelik pek çok özelliği kullanarak çalışmalarımız hakkında bilgi alabilirsiniz.

### 2.1. Site içeriğinden Diğer Dosya biçimleri Nasıl Elde Edilir?

Salt metin biçimine dönüşüm yaparken HTML dosyalarından yararlanıyoruz. PS ve PDF belgelere dönüşüm için ise XML belgeleri önce TeX belgelere sonra PS veya PDF belgelere dönüştürüyoruz. Kitaplığımızdaki belgeleri, HTML olmazsa istenen verimin alınamayacak olanları ile zaten bir süre sonra tamamen atıl duruma gelecek olanları hariç size PDF olarak zaten sunuyoruz. Bunları <http://pdf.belgeler.org> adresinde bulabilirsiniz. Eğer bu siteyi Acrobat Reader eklentili Firefox ile ziyaret ederseniz, HTML içerikli bir siteyi gezer gibi bu PDF sitesini gezebilirsiniz (KPDF bu satırlar yazılırken site içi bağları henüz çalıştıramıyordu).

#### 2.1.1. Salt Metin Belgeler Nasıl Üretilir?

Bu işlem için bir http istemcisi kullanacağız. Bu istemcinin özelliği salt metin uçbirim üzerinde çalışması, evet, Links'ten bahsediyorum. Links'i komut satırından çalıştırarak herhangi bir HTML dosyasını bir metin dosyasına dönüştürebilirsiniz.

```
links -dump Turkish-HOWTO.html > Turkish-HOWTO.txt
```

Lynx'de kullanabilirsiniz ama tabloları desteklemediğinden sonuç pek iyi olmayacaktır.

Bir sonraki bölümde Linux Kitaplığının oluşumundan ve özelliklerinden bahsedilecektir.

## 2.2. Linux Kitaplığı

Linux Kitaplığı, en tepede bulunan docbook'un `set` elemanı altındaki çok sayıda `book` elemanından ve onların altındaki çok sayıda sayfadan oluşur. Bu yapı bize pek çok kolaylık sağlamaktadır. Bu kolaylıkların neler olduğuna kısaca değinelim.

### 2.2.1. Çapraz Bağ Özelliği

Bu yapının başlıca özelliği bir belgeden ister aynı kitap içindeki ister başka bir kitap içindeki bir belgeye ya da bir sayfaya ya da bir paragrafa kolayca bağlar oluşturulabilmesidir.

Bunu yaparken URL'yi ayrı, metni ayrı yazmak gerekmemekte, sadece bağ verilecek elemanın ID parametresinin değerini bir parametrede belirtmek yeterli olmaktadır. Bu ID'ye sahip elemanın bulunması URL'nin ve tıklanacak başlığın oluşturulmasını kod gerçekleştirmektedir.

Bu işlemin doğru çalışabilmesi için bir elemana belirtilecek ID parametresinin değerinin kitaplık içinde eşsiz olması gerekmektedir. Bunu sağlamak için basit bir yöntem kullanılmaktadır. Her belgenin tepe elemanına eşsiz bir ID verilmekte, alt elemanlarına verilecek ID'lerde bu ID önek olarak kullanılmaktadır. Tepe elemana eşsiz ID'nin verilmesi çok kolaydır. Tepe elemana ID belirtilir ve kod derlenir. Hata oluşmuyorsa eşsiz ID bulunmuş demektir. Çünkü kod ID'lerin eşsizliğini test etmektedir.

Bu özellik, bize kitaplığın içine bir sözlük yerleştirme ayrıcalığı da vermektedir. Büyük bir şansla bu yapıya uyan bir sözlüğümüz vardı. Şans demek pek doğru değil, aslında, bu kitaplık mevcut sözlüğü kullanabilsin diye oluşturuldu. Kodu bu kitaplığın yazılmasından önce yazılan sözlüğün iç çapraz bağları bu kitaplığın oluşturulması fikrinin doğmasına sebep oldu.

#### 2.2.1.1. Sözlüğün Kullanımı

Kitaplığımız içinde sözlüğün varlığı bir sözcüğü ya da bir terimi ki, bu terim çok sayıda sözcükten de oluşabilir; sözlükteki bir terimle kolayca ilişkilendirmeyi sağlar. Bu özellik, çevirilerde Türkçe karşılığı yeni oluşturulan terimlere yabancılığı ortadan kaldırmak ve sözcüğü basit bir karşılık gibi değil, tam tanımıyla hatta örnekleriyle öğrenme şansını verir. Bu şansı verebilmesi sözlüğün yapısından ileri gelmektedir. Sözlük basit bir karşılıklar listesi değil, bir ansiklopedik sözlük olarak tasarlanmıştır.

Ancak sözlüğün yapı içinde yer alması eşsiz ID'lerin çok da basitçe oluşturulamaması sonucunu doğurmaktadır. Bu durum sözlük içinde bulunması imkansız ID'ler oluşturulmasını gerektirmiştir. Çünkü sözlük içindeki bir terime verilen bağlarda ID olarak doğrudan doğruya terimin kendisi kullanılmaktadır. Terimin dosya ismi `.html` uzantısı olmaksızın aşağıdaki gibi kullanılabilir.

Örneğin: `<link linkend="Adres">adres</link>`

"adres" terimini cümle içinde geçtiği yerde `link` elemanı ile sarmalamak yetmektedir. Kod kitaplık içindeki `linkend` parametresi ile verilen değeri ID'ler arasında bu değeri aramakta, bulunduğu bu bağ oluşturulmaktadır.

Bulamazsa ne olacak sorusu hemen akla gelebilir. Bu duruma uygun bir önleme sahibiz. Birisi terimi sözlüğe yerleştiriverir :).

Bu özellik, kitaplığımızı oluştururken bize bir olanak daha sunar. Başka bir belgeye kitaplıkta olsun ya da olmasın yazım sırasında tahmini bağlar oluşturmaya imkan verir.

Bu üstün özelliği ;-)) kullanabilmek, ID değerlerinin bazı özelliklere sahip olmasıyla sağlanabilir. Kök ID değerleri, örneğin bir NASIL belgesi için belgenin dosya isminden oluşturulmaktadır. Bu ID'ler HTML dosyalarının ismini oluşturduğundan çevirmen bu özelliği kolayca kullanabilir.

Bu ID'lerin dosya isimleri olması nedeniyle, tahmin gibi bir belirsizlik ortadan kalkmaktadır. Ancak beraberinde ek bir efor gerektirmektedir. Bir belgenin orijinali, kitaplığa yerleştirilmeden önce elemanlarındaki ID değerlerinin yeniden düzenlenmesi gerekmektedir. Böylece çevirmen başka bir belgedeki ilgili konuya ID değerini bilerek bağ verebilmektedir.

ID'lerin dosya ismi olarak kullanılması ayrıca, dosya isimlerinin değişmezliğini de beraberinde getirerek, belgelerin daima aynı isim ve adresle bulunabilmesini sağlamaktadır. Bu özellik, bir referans sanalyöre için vazgeçilmezdir.

### 2.2.2. Genişleme özelliği

Kitaplık, tepe elemanı olarak `set` seçilerek olabilecek her türlü genişlemeye imkan sağlanmıştır. Kitaplığın teorik olarak kitap sayısı sınırsız olabilir. Ayrıca XML alt yapı günün şartlarına göre mevcut veriyi bozmadan her türlü değişikliğe izin vermektedir.

### 2.2.3. Yansılama kolaylığı

Yansılama kolaylığı kitaplığın oluşturulma amacının bir yan ürünüdür denebilir. Kitaplığın oluşturulma amacı daha önce de belirtildiği gibi son kullanıcının makinasında bulundurabileceği bir kitaplık oluşturmaktır. Bunu rsync sunucumuzla sağlıyoruz. Başsayfamızın hemen başında bu bilgi var.

Kitaplığımızın yansılanmasını özellikle tercih ediyoruz. Yansılama hem sanalyöremizin bulunduğu makinanın yükünü azaltacak hem de kitaplık kodunun farklı makinalarda saklanarak, belgelerimizin arşivlenmesini sağlayacaktır. Bu kitaplığımızın güvenliği açısından ayrı bir önem arz etmektedir.

## 3. Ne NASIL yapılır?

### 3.1. Belgelerin Kabulü

Yayınlanmak üzere gönderilen ya da başka kaynaklardan toplanan belgeler 3. şahıslara kopyalama ve dağıtma izni verilmesi önkoşuluyla Kitaplığımızın altındaki "Belge Arşivi"nde bir öndeğerlendirmeden sonra yayınlanabilir.

Belge Arşivi'ndeki belgeler burada belirtilen kurallar dahilinde onay kurulumuz tarafından değerlendirmeye alınır. Onay kurulumuzca alınan tavsiye kararlarına uygun olarak belge düzeltildiğinde işbu belge kitaplığımızın uygun bir kitabına dahil edilir. Bu işlem genellikle belge arşive girmeden yazar ile iletişim aşamasında gerçekleşmekte ve belge doğrudan kitaplığımıza eklenmektedir.

Yazarı bulunamayan ya da tavsiye kararına uygun güncellemesi yapılmayan tüm belgeler ile güncelliğini yitirmiş belgeler Belge Arşivinde kalmaya devam eder.

Bu bölümün devamında bir belgede aranılacak özellikler anlatılmıştır. Belge gönderen bu bölümdeki kuralları okumuş ve kabul etmiş sayılır.

### 3.1.1. Genel Kurallar

Bu bölümde belgeleri hazırlarken dikkat edilecek bazı genel kurallardan söz edilecektir.

1. Kitaplık kapsamında hazırlanacak tüm belgeler ister bir makale ister bir kitap olsun, Docbook–xml ile yazılmalıdır. Docbook–xml belgelerin nasıl hazırlanacağını *Linux Belgeleri NASIL* (sayfa: 11)'dan öğrenebilirsiniz. Çeviri belgeler ise başka bir biçimde hazırlanmış olsalar bile docbook–xml'e dönüştürüldükten sonra çevrilmelidir.
2. Belgelerdeki `book`, `part`, `chapter`, `sect1`, `article` ya da ilk `section` elemanlarının ID parametrelerini boş bırakmayın. Bu ID'ler HTML dosyalarının ismi olarak kullanıldığından önemlidir.

Kitaplıkta kullanılacak bir belgenin tepe elemanının ID parametresini alt elemanlarında önek olarak kullanırsanız başkalarını ID seçimi konusunda daha serbest bırakmış olursunuz. Dosya isimlerinin uzun olmasından çekinmeyin. ID'lerin eşsiz olması daha önemlidir.

3. Tüm ID parametrelerinin değerlerini küçük harflerle yazın.
4. Belge içi bağlarda kullanacağınız ID değerlerini daima belgenin tepe elemanının ID değerini önek olarak kullanarak oluşturun.

Örneğin, Turkish–HOWTO.xml dosyasındaki tepe elemanı `article`'dir ve ID parametresi

```
<article id="turkish-howto">
```

olarak verilmiştir. Belge içinde Türkçe F klavye dosyasına bağ verirken

```
<link linkend="turkish-howto_trf">trf.kmap.gz</link>
```

kullanılmıştır. Burada dikkat ederseniz `linkend` parametresinde kullanılan ID parametresi `article` elemanının ID değerinin sonuna `trf` eki getirilerek oluşturulmuştur. Bu bağın hedefindeki elemana bakarsak:

```
<sect1 id="turkish-howto_trf">
```

olduğunu görürüz.

5. `programlisting`, `screen`, `literallayout` elemanları ile sarmalanmış metin parçaları sabit genişlikli yazı tipi ile boşlukları korunarak gösterilirler. Bu metinler XML dosya içinde yazılırken satır başlangıcından ne kadar uzakta ise gösterilirken kendi bölgesinin sol kenarından o kadar içerde gösterilirler. Dosyayı biçimli yazmak isterseniz buna özellikle dikkat etmelisiniz. Bu alanlardaki metinleri düzenlerken **kesinlikle sekme (tab) tuşunu kullanmayın** Sekme karakterinin içerdiği boşluk sayısı kişisel seçimlere bağlı olarak değişiklik gösterir ve bu belge üzerinde bir başkası çalıştığında herşey farklı görünebilir. Bu bakımdan sadece boşluk karakterini kullanmalısınız.
6. Belgeyi yazarken kullandığınız karakter kodlamasını XML dosyasının başlığında belirtin:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Sistemlerini `LANG="tr_TR"` çevre değişkeni ile belirtilen kodlama ile kullananlar için bu başlık:

```
<?xml version="1.0" encoding="ISO-8859-9"?>
```

`LANG="tr_TR.UTF-8"` çevre değişkeni ile belirtilen kodlama ile kullananlar için:

```
<?xml version="1.0" encoding="UTF-8"?>
```

olacaktır. Bu değer, XML dosyasının XSL dönüşümü yapılırken, yani HTML biçime dönüştürülürken dosya içindeki karakterlerin hangi dil kodlamasına göre değerlendirilmesi gerektiğini belirtmek için kullanılır. Oluşturulan HTML dosyasının dil kodlaması için bu değer kullanılmaz. O dönüşüm sırasında belirlenmektedir. Siz XML dosyayı hangi dil kodlamasıyla yazarsanız yazın HTML dosyalar ISO–8859–9 kodlama ile oluşturulur.

7. Satırsonu karakterlerinin UNIX türünde olmasına dikkat edin. Yani satırsonu karakteri sadece `LF` karakterinden oluşmalıdır, Windows'daki `CR+LF` ve Mac'deki `CR` karakterlerinden değil. Bu bakımdan dosya üzerinde Linux dışında bir ortamda çalışıyorsanız, dosyanın UNIX türünde olduğundan emin olun.

ID'ler ile ilgili kurallardan, belge çok büyük ve değiştirilmesi büyük emek harcanmasını gerektiriyorsa vazgeçilebilir. Bu tür belgeler çoğu zaman kitap formundadırlar. Bunlar kitaplık kapsamına alınmaz ve kitaptan ayrı yayınlanır. Buna en iyi örnek olarak, yaklaşık 15MB'lık boyutuyla PHP-Manual gösterilebilir. Bu tür belgelerin sözlükten ve kitaplık kapsamındaki belgelerden ancak sabit bağlarla yaralanabileceği, iç çapraz bağlantıları kullanmayacağı açıktır.

### 3.1.2. Çeviri Yaparken Nelere Dikkat Etmeli?

#### Orjinaline yakınlık

Çevirilerin mümkün olduğu kadar orjinaline yakın olması beklenmektedir. Çevirilerde hedef, çeviriyi okuyan kişinin orjinaline ihtiyaç duymamasıdır. Dolayısıyla bir belgenin çeviri olabilmesi için kısaltılmadan, cümlelerin bazı bölümleri atlanmadan yani serbest çeviri olarak değil, **tamamen** çevrilmesi gerekir.

Bilgisayar dünyası hızlı gelişen bir dünya olduğu için zaman zaman çeviri yapılan bir belgede orjinalinde olmayan güncellemeler veya bazı ek açıklamalar yapma gereği çıkabilir. Bu tarz, çevirmen katkıları, ayrıca `<footnote>` elemanları ile sarmalanarak [Ç.N.: bla bla] şeklinde dip notlarda gösterilmelidir. Docbook yapısı bu dipnotları kendiliğinden oluşturmakta ve dipnotlara gidip geri dönmek bir sorun teşkil etmemektedir.

#### Üslup

Çeviri belgelerde orjinalinin üslubu esastır. Dil yapısının farklılığının gerektirdikleri haricinde, orjinalinin yazarı ne yazmış ise, bu çevrilmelidir. Örneğin 1. tekil şahıs ile samimi ve araya şakalar serpiştirerek yazılmış bir belge, üçüncü şahısta, bu şakalar olmaksızın, ciddi bir üslupla çevrilemez.

Buna rağmen, çevirinin Türkçesi akıcı olmalı ve **çeviri kokmamalıdır**. Örneğin çeşitli deyimler birebir çevrilmemeli, aynı manayı veren Türkçe deyimlerle değiştirilmelidir. Çevirmenin o an aklına gelmeyen Türkçe sözcüklerin yerine İngilizce orjinallerini kullanmaması, (örnek: *Şimdi "daemon"larınızı "restart" edin*), Türkçe'de olmayan cümle yapılarını kullanması (örnek: *o kullanıcılar ki, her gün makınayı yeniden başlatırlar, onlara yapılacak birşey yoktur*) gibi **dublaj Türkçesinden sakınmak gereklidir**.

#### Dil

Çeviri belgeler yabancı dil bilmeyen veya az bilen Türkler tarafından okunacaktır. Dolayısıyla belgenin dili Türkçe olmalıdır.

#### Terimler

Genelde karşılaşılan sorun, bazı terimlerin Türkçe'sinin henüz olmaması veya ilgili çevirmen tarafından terimin Türkçe karşılığının bilinmemesidir. Bu noktada hemen kolaycılığa kaçmak ve terimin orjinalini olduğu gibi kullanmak (daemon'ları restart edin) belgenin Türkçe'den başka bir dil bilmeyen ana hedef kitlesi tarafından okunduğu zaman anlaşılmasına, dolayısıyla hedefine ulaşamamasına yol açacaktır.

Bu noktada çevirmen, kendi sözcük dağarcığını zenginleştirmek ve gerek sözlük kullanmak, gerek [linux-yerellestirme \(at\) liste.linux.org.tr](mailto:linux-yerellestirme@liste.linux.org.tr) listesinden yardım istemek sureti ile ilgili terimin Türkçe'sini bulmak ve kullanmakla mükelleftir. Gerçekten Türkçe'de ilk defa kullanılacak olan bir terim ise, uygun Türkçe karşılığının ne olması gerektiği tartışılmalı ve ilgili sözlüklere ilave edilmelidir. Çeviri belge üretmenin ikincil hedefi de bir bilgisayar dili olarak Türkçe'nin zenginleşmesi ve kullanılmasını sağlamaktır.

Kimi zaman dilimize ilk olarak orjinali ile kısıtlı miktarda geçmiş, fakat zaman içerisinde Türkçe'si de türetilmiş olan terimlerle karşılaşılabilir. (Örnek: proxy/vekil, kernel/çekirdek vb) Burada, eğer çevirmen Türkçe'sinin yaygın olmadığından kuşku duyar ise, terimi ilk kullandığı zaman orjinalini yanında vermeyi



tercih edebilir (örneğin: vekil sunucular [İng.: proxy] faydalı şeylerdir). Çevirmen, belgenin kalan kısmında Türkçe'sini kullanarak terimin yaygınlaşmasını sağlamakla mükelleftir. Terim ilk kullanıldığı zaman yanında orijinalini bir sefere mahsus göstermek, bir kural değil, bir istisnadır ve yalnızca gerçekten İngilizce bilmeyen kullanıcılar tarafından da orijinal terimin bilinme ihtimali olan durumlarda başvurulacak bir yöntemdir.

### Yabancı kökenli sözcükler kullanmak

Her ne kadar bilgisayar terimleri konusunda, uzmanlık sahası gereği olarak, terim oluşturmak ve kullanmak yolu ile Türke'ye bir katkı sağlamak çevirmenin ve belge hazırlayıcılarının ikincil hedefleri arasında olsa da, genel anlamda dil değişiklikleri bu grubun uzmanlık sahasının dışına çıkmaktadır.

Dolayısıyla bir belgede terimlerle ilgili Türkçe kullanımı konusunda değerlendirmeler yaparken, çevirmenin *çeviri* yerine *tercüme* (veya tam tersi) gibi her ikisi de günlük yaşayan Türkçe'de bulunan sözcükler arasında yaptığı tercihlere karışılmaz. Fakat, bu noktada, özellikle son yıllarda yaygınlaşan, herhangi bir yabancı sözcüğün karşılığı olup olmadığı düşünülmezsizin birebir ithal edivermek kolaylığına düşülmemelidir. Örneğin *çeviri* veya *tercüme* kullanılabilir, fakat *tradüksiyon* kullanılmamalıdır. *Çeviri* veya *tercüme* örneğine tekrar geri dönersek, gönlümüz *çeviri*'den yanadır.

Mümkün olduğu kadar öz Türkçe kullanımı esastır, fakat çevirmenin bunu bir düstur olarak kabul etmesi beklenmekte, çeviri onay mekanizmasının bu noktada elde TDK sözlüğü, teker teker her sözcük üzerinde inceleme yapması beklenmemektedir.

Bu noktada da çizgi, yine dublaj Türkçe'sinden geçmektedir. Bir yakınımın, altyazılı bir filmde Cibuti'yi "Djibouti" olarak altyazılarda gördüğü zaman dediği gibi "Allah bilir altyazıyı hazırlayananın dedesi orada şehit olmuştu..." tarzından serzenişlere yer bırakmayacak şekilde dil kullanımı yapılmalıdır. Yer isimleri Türkçe karşılıkları ile çevrilmelidir (örneğin Aleppo, Halep'tir. Morocco, "Morako" olarak değil, Fas olarak çevrilir gibi).

Hedefimiz, akıcı ve anlaşılır bir Türkçe kullanımını teşvik etmektir.

### Onay

Yukarıdaki özelliklere uygun olan bir çeviri belge onaylanır. Çevirilerde esas, orijinalini, akıcı ve anlaşılır bir Türkçe ile yansıtmasıdır.

### 3.1.3. Yeni Belge Hazırlarken Nelere Dikkat Etmeli?

#### İçerik

##### Hedef Kitle

Belgelerde esas, çeşitli seviyelerdeki bilgisayar kullanıcılarına faydalı olmasıdır. Bu sebeple, belge yazarı belgenin kime, ne fayda sağlayacağına dair bilgiyi belgenin başında vermelidir. Bu bilgi, belgenin genel değerlendirmesinde büyük önem kazanacaktır.

Örneğin eğer başlangıç kullanıcılarına yönelik bir belge ise, bu kullanıcının çekirdeğe bir modül yazabileceğini varsaymak, belgenin bir işe yaramamasına yol açacaktır.

##### Tarih, Bağlam

Bilgisayar dünyası hızla gelişmekte ve dünün problemleri kısa zamanda problem olmaktan çıkmaktadır. Dolayısıyla yazılacak bütün belgelerde belgenin tarihi, kullanılan protokollerin sürümleri, işletim sistemi sürümü, vb gibi problem ve imkanların varlığını bir zaman çizgisine oturtmaya yarayacak bilgiler açık olmalıdır.

Örneğin Linux'un dinamik filtreleme yapamayacağından bahseden bir güvenlik duvarı belgesi, eğer bunu "XX tarihinde, ipchains sürüm YYY ile dinamik filtreleme yapılamaz" şeklinde belirtirse, dinamik filtrelemenin her dağıtımda öntanımlı olarak var olduğu 2002 yılında bunu okuyan bir kullanıcı bu zaman aşımına uğramış bilgi yüzünden yanlış yönlendirilmeyecektir.

### Örnekleme

Mümkün olduğu kadar belgeler her dağıtım, her sürümde üzerinde uygulanabilecek genel kavram ve bilgiler içermelidir. Fakat örnekler, mümkün olduğunca spesifik uygulamalar üzerinde verilmeli, kullanıcının bunu kendisinin uygulayabilmesi için gereken ek şartlar belirtilmelidir.

### Bilginin sadeleştirilmesi

Özellikle başlangıç kullanıcılarına yönelik bilgiler belli bir ayrıntı seviyesinin altına inemez. Fakat, okuyucunun, ayrıntılarını henüz öğrenemediği daha büyük bir dünyanın varlığından haberdar olması açısından, bir genelleme veya geçiştirme olduğu zaman, bu belirtilmelidir.

Örneğin DNS'in kabaca ne iş yaptığını anlatan bir belgede, DNS sorgularının arka planının çok ayrıntısına inilmesi uygun olmayabilir. Fakat bunu *yerel DNS sunucusu ana DNS sunucusunu sorgular ve bilgiyi ondan alır* demek yerine *yerel DNS sunucusu ana DNS sunucusunu sorgular ve bilgiyi ondan alır (asıl sorgu mekanizması bundan biraz daha ayrıntılıdır, fakat bu belgenin kapsamı dışındadır)* diye belirtmek, okuyucunun yanlış kavramlar oluşturmasını engeller.

### Ek kaynaklar

Okuyucunun daha fazla bilgi için başvurabileceği (gerek Türkçe, gerek başka dillerde) kaynakları belirtmek faydalı bir işlemdir.

### Dil

Çeviri belgeler için belirtilen dile özgü hususlar özgün belgeler için de geçerlidir.

### Onay mekanizması

Özgün belgelerde:

1. Hedef kitle için uygunluk.
2. Genel akıcılık.
3. Verilen bilgilerin doğruluğu.

aranır. Özellikle verilen bilgilerin doğruluğu, eğer bu konuda bilgiye sahip bir çalışma grubu üyesi yok ise, sorun doğurabilir. Böyle durumlarda belge yayınlanır.

### Düzeltilme/Ekleme

Herhangi bir belgeye gelen düzeltmeler, eklemeler, belli bir zaman aralığında gerçekleştirilmelidir. Kamusal ortamda, bireylerin ortak bilgi kümesinden faydalanma ve bir kişinin ortaya çıkaramayacağı derinlikte ve genişlikte belge üretilmesi ancak bu şekilde gerçekleşebilir.

Herhangi bir belge yazarı, belgesini yayınlanmak üzere gönderdiği zaman peşinen bu ekleme ve düzeltmeleri kabul etmiştir.

Ekleme ve düzeltmelerde bulunan kişilerin adları, katkının ne olduğuna dair ufak bir not ile belgenin başında yazar bölümünde yayınlanır. Belgenin içerisinde bu katkıları ayrıca farklı gösterilmez, işaret edilmez. Pek çok kişinin katkı ve eklemesi olan bir belgede, okuyucu yormaksızın tek bir belge ortaya çıkarabilmek ancak bu şekilde mümkün olacaktır.

Ekleme ve düzeltmeler çalışma grubunun belirleyeceği kurallar dahilinde, çalışma grubunun atayacağı bir editör aracılığı ile yapılabileceği gibi, belgenin ilk yazarı/çevirmeni de bu işi üstlenebilir.

Burada esas, bu ekleme ve düzeltmelerin katkıda bulunmak isteyen kişinin şevkini kırmayacak bir zaman aralığında gerçekleştirilmesidir.

Bu süreç, belgenin doğruluğunu değerlendirmek açısından da, genelin bilgi kümesinden faydalanmak yolunu açacağından dolayı, belgelerin doğru, faydalı ve güncel kalmasını sağlayacaktır.

## 4. Linux Belgeleri NASIL

### 4.1. Giriş

Web sayfası hazırlamış olanlar XML yapısına kolayca uyum sağlayabilir. Tipik bir HTML kodu aşağıdaki gibidir.

```
<html>
  <head>
    <title>Başlık</title>
  </head>
  <body>
    <h3>Sayfam</h3>
    <p>Sayfam güzeldir.</p>
  </body>
</html>
```

Şimdi bu sayfayı bir de XML ile yazalım:

```
<article>
  <articleinfo>
    <title>Başlık</title>
  <articleinfo>
  <sect1>
    <title>Sayfam</title>
    <para>
      Sayfam güzeldir.
    </para>
  </sect1>
</article>
```

Yapıları ne kadar benziyor değil mi?

Diyebilirsiniz ki, HTML biliyorum, ben sayfayı HTML ile yazayım, bitsin, gitsin. O zaman hazırladığınız belgeyi sadece web tarayıcınızla okuyabilirsiniz. Halbuki XML ile yazdığınız bir belge HTML'ye dönüştürülebildiği gibi, TXT, PDF, vb. belge biçemlerine dönüştürülebilir ortak bir yapı sunar. Bir HTML sayfa tek bir sayfadır ama Docbook olarak hazırlanmış bir belge ile değil kitap, bir kitaplık yazabilirsiniz. Ve hatta tek bir komutla bu kitaplıktan yüzlerce HTML sayfası üretebilirsiniz. Üstelik HTML çıktısını beğenmezseniz görüntüsünü değiştirmek için HTML dosyayı yeniden yazmak gibi dertlerle de uğraşmazsınız. Ayrıca, bir sürü belge biçemi bilmek zorunda da değilsiniz. Hiç çıktının görünümü ile uğraşmadan kendinizi sadece bu belgelerle aktaracağınız bilginin üretilmesine verebilirsiniz. sitemizdeki belgeler, tabii bu belge ve sözlük bu sistemle üretilmiştir.

### 4.2. Bir Belge Hazırlamaya Başlamadan Önce

Önce sisteminizde tam bir Türkçe desteği olmalı. Bunun nasıl yapılacağı kitaplığımızdaki [Türkçe–NASIL](#)<sup>(B11)</sup> belgesinden öğrenebilirsiniz.

Belgeleri hazırlayabilmek için basit bir metin düzenleyici yeterli. Gedit, Emacs, Kwrite ya da Kate işinizi görür. Ben Kate kullanmayı tercih ediyorum, çok sayıda belge ile çalışabilme ve XML dosyaların yazımını kontrol etme ve dosyanın karakter kodlamasını değiştirebilme özelliği bana çekici geliyor.

Kate kullanmayı tercih etmemin başlıca sebebi XML dosyaları yazmak için bir eklentiye sahip olması. Kate'in bu eklentisi ile birlikte kullanmak üzere sitemizin DTD'sini bir sıkıştırılmış bir dosya olarak dağıtıyoruz. Kate'in bu eklentisinin ve belgeler.dtd.xml dosyasının nasıl kullanılacağını [Hata Ayıklama](#) (sayfa: 19) bölümünde anlatacağım.

Bir Docbook kılavuzuna ihtiyacınız olacak. Benim bir belge hazırlarken faydalandığım kılavuzu [docbook.org](#)<sup>(B13)</sup>'dan alabilirsiniz. İsmi 'Docbook: Definitive Guide'

Bir XML dosyası yazacak araçlarımız artık var. Yazdığımız belgenin derlenmesi lazım ki, bir yerde hata yapmış mıyız bilelim. Bir HTML çıktısını da görmek istersiniz değil mi? O halde belgemizi derlemek için bazı programlara, yardımcı kodlara ihtiyacımız olacak.

Docbook kodunu oluşturan eleman ve parametrelerin birbirleriyle ilişkilerinin düzenlendiği bir kod tanımlarına (XML-DTD), HTML koda dönüşümün nasıl yapılacağını belirleyebileceğimiz bir dönüşüm diline (XSL) ve tüm bunları harmanlayıp HTML kodunu üretecek bir yardımcı programa (xsltproc) ihtiyacımız olacak. Bu araçlar hemen her Linux dağıtımda zaten bulunuyor. Docbook'un standart elemanları dışında elemanlar kullanmazsanız ve görsel olarak farklı bir şey istemiyorsanız XSL dönüşümlerini yapacak kodlar da hazır olarak var. Bu kitaplığı hazırlarken kullandığım paketleri sizler de kullanırsanız, bir uyumsuzluk sorunu yaşamayız. Ayrıca sorunlarınızı ilettiğinizde aynı paketleri kullandığımızdan anlaşmak da kolay olur.

Şimdi bu paketlerin hangileri olduğuna nereden alınabileceğine ve dosya sisteminizde bulunabilecekleri yerlere bakalım.

Bu paketler ve sürümleri benim kullandıklarım, sanırım daha üst sürümleri de sorun çıkarmaz.

```
xml-dtd-4.1.2(B14): /usr/share/docbook/xml-dtd-4.1.2(B15)
```

```
xsl-stylesheets-1.49(B16): /usr/share/docbook/xsl-stylesheets-1.49(B17)
```

Bu paketler derlenmeden kullanılıyor. İndirip paketi açmanız yeterli. RPM'leri de var. Ancak bu paketleri indirip kurmak yerine size daha iyi bir önerim var. Web sitemizin kaynak kodlarını biliyorsunuz dağıtıyoruz. Anonim CVS erişimi ile tüm kaynak kodlarımızı alabilirsiniz. İçinde Docbook-XML-DTD ve Docbook-XSL zaten var. Bu paket üzerinde çalışırsanız sayfaları sitemizdeki gibi göreceksiniz. Çünkü Docbook-XSL içindeki kod özelleştirilmiştir. Ama aşağıdaki iki paket sisteminizde kurulu olmalı yoksa paket içeriğinden HTML dosyaları üretemezsiniz.

```
libxslt-1.0.13(B18)
```

```
libxml2-2.4.17(B19)
```

Bu iki paket ise Gnome'un standart paketleri ve derlenmeleri gerekiyor. Derlemek yerine RPM paketlerinden kurmayı da tercih edebilirsiniz.

Tüm temel gereksinimlerimizi karşıladığımızı göre belge hazırlama ortamımıza son rötuşları yapalım. Ev dizininizde çalışmalarınızı tutabileceğiniz bir dizin oluşturun. Örneğin: `/home/nilgun/belgeler` dizine geçin. İnternet'e bağlı değilseniz bağlanın ve konsolda (ya da xterm'de):

```
cvs -d:pserver:anonymous@belgeler.cvs.sourceforge.net:/cvsroot/belgeler login
```

yazdıktan sonra sizden parola istenecek. Birşey yazmadan <Enter> tuşuna basın ve:

```
cvs -z3 -d:pserver:anonymous@belgeler.cvs.sourceforge.net:/cvsroot/belgeler \  
co -P sitesrc
```

komutunu verin. Kısa bir süre sonra dosyalar makinanıza akmaya başlayacak. İşlem tamamlandığında bulunduğunuz dizinin içinde bir dizin, `sitesrc` dizinini göreceksiniz. Bu dizini çalışma dizininiz olarak kullanmayın. Bu dizini örneğimizdeki `/home/nilgun/belgeler` dizini altında `source` ismiyle kopyalayın. Böylece içeriği aynı olan iki dizin elde etmiş olacaksınız. Şimdi diyeceksiniz ki, ikisi de aynıysa neden bunu yaptık. `sitesrc` dizininin içeriğini arasıra güncelleyebilirsiniz. Eğer bu dizini çalışma dizininiz olarak kullanırsanız, bir cvs güncellemesi sırasında bizim yaptıklarımız, sizin yaptıklarınızın üzerine yazılıp çalışmanızı kaybetmenize sebep olacaktır.

Siz şimdi "bu güncellemenin nasıl yapılacağını da söyle de bu iş tamam olsun" diyececeksiniz. :-)

Güncelleme yapmak için `sitesrc` dizinine geçin ve

```
cvs -z3 update
```

komutunu girin. Gördüğünüz gibi güncelleme işlemi ilk verdiklerimiz kadar karmaşık değil. Tek koşul bu komutu `sitesrc` altında değiştirmeden tuttuğumuz dizin içindeyken kullanmaktır.

Ancak hiç olmazsa ayda bir kere, sitesrc dizininin bulunduğu dizinde bu dizini ilk olarak indirdiğimiz yukarıdaki uzun komutları kullanarak da güncelleniz gereklidir.

Bu güncellemeler farklıdır. Uzun komutla yapılan yeni dizinleri de kopyalar. Kısa komutla yapılan güncelleme ise sadece mevcut dizinleri günceller.

sitesrc/docbook dizini altındaki iki dizin Docbook–XML–DTD ve Docbook–XSL kodlarını barındırır. Diğer dizinler ise sitemizi oluşturan XML taslaklar ile kitaplığımızın uygun şekilde oluşmasını sağlayan kodları içerir.

Bu erişim yöntemi sadece CVS deposundaki dosyaları makinanıza indirmek için kullanılabilir. CVS depomuzla etkileşimli çalışmak için yapılması gereken işlemler farklıdır. CVS yazılımının belgelerini okuyarak, veya kitaplığımızdaki [CVS ile Açık Kod Yazılım Geliştirme<sup>\(B20\)</sup>](#) NASIL belgesinden bu bilgileri edinebilirsiniz.

### 4.3. Bir XML Dosyasının Yapıtaşları

XML, HTML gibi bir biçimleme dili değildir. Sadece veriyi kullanım amacına uygun olarak etiketlemek için kullanılan bir dildir. Zaten isminin açılımı da Genişletilebilir Etiketleme Dili'dir. XML dosyaları HTML, salt metin ya da PDF biçimli dosyalar haline getirmek için bir biçimleme dili olan XSL kullanılır. XML hakkında burada dolaylı olarak yüzeysel bazı bilgiler bulacaksınız. Daha fazla bilgi edinmek isterseniz, aşağıdaki belgeleri okumanızı öneririm:

- [Extensible Markup Language](#) (XML) 1.0 (Second Edition)<sup>(B21)</sup>
- [Namespaces in XML](#)<sup>(B22)</sup>
- [XML Information Set](#)<sup>(B23)</sup>
- [XML Schema Part 0: Primer](#)<sup>(B24)</sup>
- [XML Schema Part 1: Structures](#)<sup>(B25)</sup>
- [XML Schema Part 2: Datatypes](#)<sup>(B26)</sup>

Örnek bir XML belgesini Ekler bölümünde bulabilirsiniz: [Sample–HOWTO.xml](#) (sayfa: 21)

#### 4.3.1. Etiketler

XML dilinin olmazsa olmaz bazı özellikleri var. HTML ve benzeri etiketli dillerde olduğu gibi metin etiketler arasında veriliyor.

`<title>bir konu başlığı</title>` gibi. Ancak bu etiketleme de dikkat edeceğimiz bazı kurallar var:

- Etiket isimleri daima < ve > karakterleri arasında olmalı.
- Etiket ve değişken isimleri daima bir harfle başlamalıdır.
- Kullandığımız XML–Docbook için etiket ismi sadece küçük harflerden oluşmaktadır. SGL ve HTML için etiket ismini oluşturan harflerin büyüklüğünün önemi yoksa da XML için harf büyüklüğü önemlidir. Bu bakımdan standart olmamakla beraber etiket isimlerinde sadece küçük harf kullanma alışkanlığı edinmeniz yararlı olacaktır. XML'deki bu özellik temel aldığı kodlamadan kaynaklanmaktadır. UTF–8 ve UTF–16.
- Bir etiket tanımında, içinde veri içermesi öngörülürse, bir açılış etiketini daima bir kapanış etiketi izlemeli. Yukarıdaki `<title>` etiketi gibi. HTML ve SGML'deki bazı durumlarda bitiş etiketi verilmeyebilirse de XML için bitiş etiketleri zorunludur.
- Etiket sadece bir işlem gerçekleştirmek içinse kapanış etiketi kullanılmaz. Örnek: `<sbr/>` Bu etiket, Docbook'ta kendinden sonra gelen metni satırbaşından başlatmak için kullanılır.

- Bir etiketin bitiş belirleyicisi / karakteridir. Bir veriyi sarmalayan etiketlerde, bu im, etiket isminin başına, işlem etiketlerinde sona konur.
- Dosya içinde etiketler haricinde < karakteri kullanılmaz. Bu imi kullanmanız gerekiyorsa `&lt;` ile belirtmelisiniz.

& karakterini de yazamazsınız. Bu im XML için < gibi bir dil ögesidir. `&lt;`'nin yazıldığı gibi görünmesi için `&amp;lt;` yazmalısınız.

### 4.3.2. Açıklamalar, Yorumlar

Programlama dillerinde açıklama ve yorum satırları nasıl bir kurala bağlıysa, XML için de kurallar vardır. Bu kural HTML'deki ile aynıdır. Bir açıklama `<!--` ile başlar ve `-->` ile biter. Bu dizgecikler açıklamaların başlangıç ve bitiş etiketleri gibi düşünülmelidir. Yani her başlangıç etiketi için bir bitiş etiketi olmalıdır. Bu, bir yorum içinde, bitiş etiketi olmayan bir başlangıç etiketinin bulunamayacağı anlamına gelir. Bu kurala uyan her türlü açıklama ve yorum dosyanın herhangi bir yerinde kullanılabilir. Açıklama ve yorumlar XML dosyanın derlenmesi sırasında derleyici tarafından yorumlanmazlar.

### 4.3.3. İlk satır ve Belge Türünün Belirtilmesi

Belgeler sitemizi oluşturan temel dosya olan `belgeler.xml` dosyasını inceleyerek bir belge nasıl hazırlanır öğrenirken sitemizin yapısı hakkında da fikir edinebileceksiniz. `belgeler.xml` dosyasının ilk satırlarından başlayalım:

```
<?xml version="1.0" encoding="UTF-8"?> ①
<!DOCTYPE set PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN" ②
    "docbook/dtd/docbookx.dtd" [ ③
    <!ENTITY howtos.xml SYSTEM "source/howtos/howtos.xml"> ④
    <!-- Türkçe --> ⑤
    <!ENTITY belge-NASIL.xml SYSTEM "source/howtos/belge-NASIL.xml">
    <!ENTITY apps.xml SYSTEM "source/apps/apps.xml">
    <!ENTITY iptables-usage.xml SYSTEM "source/apps/iptables-usage.xml">

    <!ENTITY others.xml SYSTEM "source/others/others.xml">
    <!ENTITY lgpl.xml SYSTEM "source/others/lgpl.xml">
    <!ENTITY gpl.xml SYSTEM "source/others/gpl.xml">
]>
```

Bu satırlar pek sık karşılaşmayacağınız karmaşıklığa sahip, ama gözünüz korkmasın, onları da ben yazdım. Bildiklerimi şu an sizinle paylaşmaya devam ediyorum.

- ① Bu satır, dosyanın bir XML dosyası olduğunu, 1.0 sürümüne göre yazıldığını, ve dosyadaki karakter kodlamasının UTF-8 olduğunu belirtiyor. Bir XML dosya için en azından `<?xml version="1.0"?>` satırı bulunmalıdır. XML'in güncel sürümü 1.0'dır. Bu değer değiştirilirse dosya işlenemeyebilir.

Satırın `encoding` parçası dosya içeriği işlenirken önem kazanır. Dosyada, belirtilen karakter kodlamasında bulunmayan bir karaktere ya da karakter dizisine rastlandığında hata üretilecektir. Burada belirtilen kodlama türü, çıktının kodlama türü değil, başlığı barındıran dosyanın karakter kodlamasıdır.

XML bildiriminin yapıldığı `<? ve ?>` etiketleri XML'de işlem komutlarının etiketleridir. İşlem komutları başlangıç etiketine bitişik yazılırlar. Burada `<?xml` bir işlem komutudur.

XML bildirim satırında kullanılacak diğer parametrelerden burada bahsedilmeyecektir. Onları, [bölüm başında](#) (sayfa: 13) isimlerini ve bağlarını verdiğim belgelerde bulabilirsiniz.

- ② Bu satır ve devamındaki satırlarda, dosyada işlenen belgenin türü, belgeyi oluşturan etiketlerin tanımlandığı paketin ismi ve sürümü, nerede bulunabileceği, belgeyi oluşturan ek dosyalar varsa bunların tanımları gibi bilgiler belirtilir. Bu tanımların başlangıç etiketi `<!DOCTYPE` ve bitiş etiketi `>` karakteridir. İçindeki herşey başlangıç etiketi olan etiketin parametreleri olarak belirli bir biçimde verilmek zorundadır.

`<!DOCTYPE` bildirimindeki ilk parameter olarak görünen `set` Docbook'un belge türü elemanlarından biridir. `set` elemanı bir kitaplar kümesi içerebilir. Bizim belgeler sitemiz de bir miktar kitap içeriyor. Bu alanda belirtilebilecek diğer belge türü elemanları, `book`, `part`, `article`, vs. olabilir. Docbook'un elemanları ilerde tanıtılacaktır.

Hemen ardından gelen `PUBLIC` bir tanımlayıcıdır. Belirtilmesi isteğe bağlıdır. Kendinden sonra gelen dizge ile belgede kullanılan eleman ve parametrelerin hangi DTD ile ilişkili olduğunu belirtmekte kullanılır. Burada `<!DOCTYPE` bildiriminin burada kullanılmamış olan özelliklerinden bahsedilmeyecektir. Bu bilgileri [Docbook Guide<sup>\(B29\)</sup>](#) da bulabilirsiniz.

- ③ Bu satır, `<!DOCTYPE` bildiriminde bulunması zorunlu olan sistem tanımlayıcıdır. Burada, belgenin oluşturulmasında kullanılan DTD'nin adresi verilmelidir. Bu adres bir URL olabildiği gibi burada olduğu gibi bir sistem dizini de olabilir. Hemen ardından gelen `[ ]`'ler içindeki bölüm aşağıda anlatılmıştır.
- ④ `<!ENTITY` bildiriyle belgemizi oluşturan diğer dosyaları tanıtıyoruz. Bunun için bir değişken ismi ve `SYSTEM` belirteciyle de dosyanın bulunduğu yeri belirtiyoruz. Belittiğimiz değişken isimlerini dosya içinde diğer dosyaların bağlanacağı noktalarda başına bir `&` ve sonuna da bir `;` getirerek kullanıyoruz. `belgeler.xml` dosyasının sonundaki gibi:

```
&howtos.xml;  
&apps.xml;  
&others.xml;  
</set>
```

- ⑤ Bu satır bir açıklama satırıdır. Bilgi vermek amacıyla kullanılmıştır. Dosyanın derlenmesi sırasında bu ve benzeri satırlar yorumlanmazlar.

`<!DOCTYPE` bildiri ek dosyalarda belirtilmez. Diğer dosyaları incerseniz, bu dosyalarda bu satırların yorumlanmaması için açıklama elemanları ile sarmalanmış olduklarını göreceksiniz. İptal edildiği halde dosya içinde bulunmalarının sebebi bu dosyalar bağımsız olarak işlendiğinde bu bildirimlerin tekrar etkin kılınarak kullanılabilmesi içindir.

## 4.4. Docbook Elemanları

Docbook elemanlarını her biri diğerlerini içeren 8 grupta inceleyebiliriz:

### 1. Kitaplıklar (`set`)

Bir `set` elemanı en azından bir `book` içerir. Sitemizin kitaplık bölümü bir miktar `book`'tan oluşan bir `set`'tir

### 2. Kitaplar (`book`)

Bir `book` genellikle en üst seviye elemandır. Bir kitap serisi planlamadığınız sürece en tepe Docbook elemanıdır. Aşağıdaki elemanların bir karışımından oluşur.

#### **Adama Bölümü** (`dedication`)

Genellikle kitabın en başında yer alır.

#### **Gezinme Elemanları**



İçindekiler (`toc`), Çizelge ve Şekil Listeleri (`lot`), Fihrist (`index`) gibi kitap bölümleridir.

### Oylumlar

Bir kitabın üst seviye hiyerarşik parçalarıdır. (`part` ve `reference`) Ancak bir kitap bu elemanlar olmadan da oluşturulabilir.

### Kısımlar

Oylumların ya da kitabın alt kısmı olarak kullanılabilen `chapter` etiketi ile tanımlanan kitap bölümleridir.

### Bölümler

Bu kitabın içindeki belgeleri oluşturan parçalar (`article` ve `sect1`) bu bölümlere iyi bir örnektir. Her `article` çok sayıda alt bölüm (`sect1`, `section`) içerebilir.



### Uyarı

`article` ve `chapter` etiketleri eşdeğer etiketlerdir. Yani bir `article`, bir `chapter` içinde bulunamaz.

### 3. Oylumlar (`part`)

Kitapların alt bölümü olarak yukarıda bahsedildi.

### 4. Kısımlar (`chapter`)

Kitapların alt bölümü olarak yukarıda bahsedildi.

### 5. Bölümler (`article`, `sect1`)

Kitapların alt bölümü olarak yukarıda bahsedildi.

### 6. Alt Bölümler (`section`)

Docbook'ta alt bölümleri oluşturan elemanlar çok çeşitlidir:

#### `sect1...sect5` Elemanları

Bunlar hemen tüm belgelerde sıklıkla kullanılan numaralı bölüm elemanlardır. `sect1`'den başlayarak iç içe 5 seviyeli bölümlenme yapılabilir.

#### `section` Elemanları

`section` elemanlarının `sect*` elemanlarından farkı sonsuz iç içelikte bölümlenme yapılabilmesidir.

#### `simplesection` Elemanı

Genellikle diğer `sect*` elemanlarının içinde yer alır. `simplesect` elemanının altına başka `sect*` elemanı yerleştirilemez. Bu bakımdan sonlandırıcı `sect*` elemanı gibi görev yapar.

#### `bridgehead` Elemanı

Sadece bir bölüm başlığı üretir.

#### `refsect1...refsect3` Elemanları

Bu elemanlar sadece `refentry` elemanlarının içinde bulunabilir. `sect*` elemanları gibidir, ama sadece 3 seviyeli bölümlenme yapılabilir.

#### `glossdiv`, `bibliodiv` ve `indexdiv` Elemanları

Sözcük dağarcığı (`glossdiv`), kaynakça (`bibliodiv`) ve fihrist (`indexdiv`) bölümlerini oluşturmakta kullanılırlar. `sect*` elemanlarının tersine bunlar iç içe seviyeli olamazlar.



## 7. Ön bilgi elemanları

`bookinfo`, `partinfo`, `articleinfo` gibi ön bilgi verme bölümleridir. Bu bölümlerde başlık, yazara ilişkin bilgiler, sürümlerin gelişimi belge hakkında özet bilgi, içindekiler, anahtar sözcükler gibi bilgiler verilir.

## 8. Blok elemanları

Blok elemanlarına en iyi örnek paragraflardır (`para`). Listeler tablolar, şekiller ve resimlerde blok elemanlarıdır.

### Listeler

Docbook'ta 7 tane liste elemanı vardır.

`calloutlist`

Genellikle bir program ya da ekran görüntüsü üzerinde bilgi vermek için kullanılır. Bir önceki bölümde XML dosyasının ilk satırlarını incelerken bu elemandan faydalanmışım.

`glosslist`

Sözcük dağarcığındaki terimler ve tanımları için kullanılır.

`itemizedlist`

Sıra numarasız imli listeler için kullanılır.

`orderedlist`

Sıra numaralı listeler için kullanılır. Bu bölümdeki 1 den 8 e kadar giden liste gibi.

`segmentedlist`

Öğelerinin isimlerinin tekrarlandığı listeler için kullanılır.

`simplelist`

Öğelerinin sıra numarası almadığı imsiz bir liste türüdür. Öğeleri satır ve sütunlar halinde düzenlenebilir.

`variablelist`

Parametre, seçenek açıklamaları türünden listelerde kullanılır. Burada 7 liste elemanını tanıttığım liste böyle bir listedir.

### Dikkat Çekme Elemanları

Docbook'ta 5 tür dikkat çekme elemanı vardır: `caution`, `important`, `note`, `tip`, `warning`

### Paragrafların içinde kullanılan elemanlar

HTML'deki `<BR>` elemanının Docbook'da ki karşılığı `<sbr/>` en basit örnek olarak verilebilir.

Başlıcaları:

`address`

Posta adreslerini belirtirken kullanılır. Satır içi kullanımda ise isim ve adresleri etiketlemek için kullanılır.

`literallayout`

İçerdiği metne sabit yazı tipi uygular ama `programlisting` ve `screen` gibi belli bir zemin rengi yoktur. Boşluklar korunduğundan ASCII-art gösterimler için faydalıdır. (Bu standart uygulama değildir. Bu kitaplık için xsl dönüşümü yapılırken elemana bu özellik verilmiştir.)

`programlisting` ve `screen`

Program listeleri, konsol çıktıları vb. metinlerin oldukları gibi görünmesini sağlayan bloklama elemanlarıdır. Bu kitaplık için özelleştirilmiş ve zemin rengi atanmıştır. İçeriğine sabit genişlikli yazı tipi uygular.

`screenshot`

Grafik ekran görüntüleri için kullanılır.

`synopsys`

Komut ve işlev bildirimlerini belirlemek için kullanılır.

### Örnekler, Şekiller ve Tablolar

Örnekler için `example` ve `informalexample`, şekiller için `figure` ve `informalfigure`, tablolar için `table`, ve `informaltable` elemanları kullanılabilir. `informal*` elemanları türlerinin başlıksız olanlarıdır.

### Paragraflar

Hemen her elemanın içinde yer alabilen blok elemanlarıdır. `para`, `formalpara`, `simplepara` gibi farklı amaçlara yönelik türleri vardır. `formapara` elemanı başlıklı bir paragraf türü belirtir.

### İfadeler

Matematiksel ifadelerin gösterilmesinde kullanılır. `equation` ve `informalequation` elemanları blok, `inlineequation` ise satır içi kullanım elemanlarıdır.

### Resimler

Resimler için kullanılacak elemanlar `figure` ve `screenshot` elemanlarıdır. Satır içi kullanım için `inlinegraphic` elemanı kullanılır. Ayrıca `video`, `ses`, `resim` ve `metin` veriler için `mediaobject`, satır içi kullanımda `inlinemediaobject` elemanları kullanılabilir.

### Soru-Cevap Elemanları

Sıkça Sorulan Sorular bölümlerini oluşturmak için `qandaset`, `question`, `answer` elemanları kullanılabilir.

## 9. İç elemanlar

Bunlar bölümler ve paragraflar içinde yer alan bağ, karakter biçimleme, uyarı vb. elemanlarıdır. Burada çok sık kullanılan bazı elemanlar tanıtılmıştır.

`abbrev`

Bir kısaltmayı etiketler. Kısaltmayı bir nokta izler.

`acronym`

Bir deyim ya da ismin baş harfleriyle oluşturulan bir kısaltmanın anlatımı için kullanılır.

`emphasis`

Etiketlediği metnin eğik karakterlerle gösterileceğini belirtmek için kullanılır. Bu kitap için özelleştirilerek, `role` parametresi ile bu etikete kalın karakter gösterimi de eklenmiştir: `<emphasis role="bold">`. Hem kalın hem eğik istenirse elemanlar birbirinin içinde kullanılmalıdır.

`footnote`

Dipnotlar için kullanılır.

`trademark`

Bir ticari markayı etiketlemekte kullanılır.

`link`

Belge içi çapraz bağlar için kullanılır. Bağ hedefi olan elemanın `id` parametresi, `linkend` parametresine belirtilir.

`ulink`

Dış hedefler için bağ belirtir. Hedefin adres bir URL olarak `url` parametresinde belirtilir.

`literal`

Satır içinde özgün hali korunması ve vurgulanması gereken metin parçaları için kullanılır.

`sgmltag`

Metin içinde geçen etiket isimlerini etiketlemede kullanılır.

`replaceable`

Metin içinde kullanıcının değiştirmesi beklenen ya da istenen metin parçalarını etiketlemekte kullanılır.

Bu bölümde bahsettiğim elemanlar Docbook'un 300'den fazla eleman içerdiği gözönüne alındığında çok azdır. Ama en çok kullanılanlardan bahsetmeye çalıştım. Buradaki bilgiler tek başlarına bir belge yazmak için yeterli değildir. Burada sadece Docbook Guide'ın Docbook oluşturulması ile ilgili bölümünü kısmen çevirdim, kısmen de ben bir şeyler kattım. Docbook Guide, her eleman için bu sayfa gibi birer sayfa içeriyor. Bu bakımdan belge hazırlayacaksanız bu kılavuz daima farenizin imlecinin altında bulunmalıdır. Ya da hazır bir belgenin sadece çevirisiyle ilgilenmelisiniz.

## 4.5. Hata Ayıklama

Hata ayıklamaktansa hatasız yazmayı tercih etmelisiniz. Bu Kate kullanınca hiç zor değil. Kate XML dosyaları yazmak için bir eklenti içeriyor. Bu eklenti sayesinde etiketleri yazmak zorunda kalmıyorum, basitçe bir menüden seçiyorum. Dolayısıyla dosyada bir hata olmuyor. Bu özelliğinden yararlanabilmek için kullanılan DTD'nin Kate'e tanıtılması gerekiyor. Sitemizin DTD'sini Kate ile kullanılabilir biçime dönüştürdüm. Bu dosyayı proje sayfamıza indirilebilirsiniz: [belgeler.dtd.xml-1.0.bz2<sup>\(B30\)</sup>](#) (19kB). Belgeler-DTD, ihtiyaçlar doğrultusunda geliştirilmektedir. Bu bakımdan yeni sürümler hakkında otomatik olarak bilgilendirilmek isterseniz, paket bağının yanındaki küçük zarfa tıklayın ve istenen bilgileri verin.

Bu dosyayı Kate ile nasıl kullanacağız?

Önce Red Hat dağıtımını kullandığımı belirtmeliyim. Çünkü bu dosyayı kopyalayacağınız yer sizin dağıtımınızda farklı bir yerde olabilir. Bunzip2 ile önce dosyayı açın, yaklaşık 1.1 MB.lık bir dosya oluşacak. Bu dosyayı ([belgeler.dtd.xml](#)) `/usr/share/apps/katexmltools/` dizinine kopyalayın.

Kate'i çalıştırın. Menüünden sırayla `Ayarlar -> Kate'i Yapılandır -> Eklentiler -> Yönetici` üzerinden `Kate XML Eklentisi`'ni seçip `Yüklenmiş Ekler` penceresine aktarın. Bu işlem Kate menü çubuğuna yeni bir menü ekleyecek: `XML Eklentisi`. Artık bir XML dosyası yazmaya hemen başlayabilirsiniz. İlk etiketimizi kendimiz yazacağız. Örneğin `<article>` etiketini yazın ve hemen ardından `ctrl+enter` tuşlarına basın. Bir uyarı iletisi alacaksınız. `Evet` düğmesine tıklayıp DTD yüklenmesini seçin ve açılan dosya yükleme penceresinde `belgeler.dtd.xml` dosyasını seçin. Kısa sürede Kate bu dosyayı yükleyecek.

Ne zaman bir etikete ihtiyacınız olursa yapacağınız şey `ctrl+enter` tuşlamak olacak. Etiketleri kullandıkça, açılan DTD menüsünün içeriğinin değiştiğini farkedeceksiniz. Bu DTD'nin bir özelliğidir. Etiketler, bir diğerinin yorum aralığında farklı etiketler kabul eder. Bu özellik sayesinde yazdığınız XML dosya, DTD'ye uyumlu olur ve derleme sırasında hata vermez. Ayrıca, imleç bir etiketin yorum alanında değil de etiketin içindeyken menü isterseniz bu kez etiketle birlikte kullanılabilir parametrelerin listesini alacaksınız.

Kate'i bu eklentisi ile kullandığınızda tamamen hatasız bir dosya üreteceğinizi sanmayın. İnsanız, mutlaka bir yerde hata yaparız. Bu bakımdan okumayı burada bırakmayıp belgenin devamını okuyun.

Belgemizi hazırlarken ve/veya hazırladıktan sonra hataları ayıklamamız ve sunuma hazırlamamız lazım. Bunun için, konsolda basit bir komut yazıyoruz:

```
$ xsltproc -o index.html belgem.xsl belgem.xml
```

Burada ilk parametre olan `-o` ile belgemizin HTML çıktısının ismini belirtiyoruz: `index.html`. Sıradaki parametre hazırladığımız belgenin HTML olarak derlenmesini sağlayan XSL dönüşüm betiği, son parametre de hazırladığımız dosyadır. Bu komutu çalıştırdığınızda `belgem.xml` hatasızsa bir dizi HTML dosya üretilenektir. Komut, dosyaların hepsinin aynı dizinde olduğu varsayılarak hazırlanmıştır. Her terim için görelî dizin isimleri belirtebilirsiniz.

Burada XSLT betiği olarak `belgeler-html.xsl` betiğini kullanabilirsiniz.

Kullandığımız komut derleme sırasında dosyada bir hata olduğunda bir sürü hata üretir. İlk belirtilen hatayı giderdikten sonra komutu tekrar çalıştırın. Diğer hatalar, dosya ardışık işlendiği için ardışık üretilen hatalardır. Daima ilk hatayı gidererek kısa sürede sonuca ulaşırsınız. Dosyayı yazarken hep derlenebilecek şekilde yazmanızı ve yazdıkça hatalar bakımından denetlemenizi öneririm. Böylece hatayı hep son yazdığınız kısımlarda arama lüksünüz olur. Aksi takdirde hatayı bulmak çok güç olabilir.

Belgeleri istenen biçimde biçimlemekte kullanılan XSL dili hakkında burada bilgi verilmeyecektir. Bu dil hakkında fikir edinmek için `xsl-stylesheet` belgelerine ve koduna bakabilir, daha ayrıntılı bilgi edinmek için aşağıdaki adresleri ziyaret edebilirsiniz:

[Extensible Stylesheet Language \(XSL\)<sup>\(B31\)</sup>](#)

[XSL Transformations \(XSLT\) Version 1.0<sup>\(B32\)</sup>](#)

[Associating Style Sheets with XML documents<sup>\(B33\)</sup>](#)

[XML Path Language \(XPath\)<sup>\(B34\)</sup>](#)

Belgeleri, SGML araçları ile denetlemenizi önermeyeceğim. Diğer belge biçimlerini elde ederken SGML araçlarını kullanmadığımız için belge taslaklarında SGML uyumluluğu aranmamaktadır. XML uyumluluğu yeterlidir.

Bu belgenin oluşturulmasında faydalanılan tüm kaynaklar belge içinde yeri geldikçe belirtilmiştir.

## A. Sample–HOWTO.xml dosyası

Bu dosya Linux Documentation Project (LDP)'in bir parçasıdır. Dosyanın aslına [linuxdoc.org](http://linuxdoc.org)<sup>(B35)</sup> adresinden erişebilirsiniz.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "http://docbook.org/xml/4.1.2/docbookx.dtd" []>

<article lang="tr">

  <articleinfo>

    <!-- Use "HOWTO", "mini HOWTO", "FAQ" in title, if appropriate -->
    <title>Sample XML HOWTO</title>

    <author>
      <firstname>your_firstname</firstname>
      <surname>your_surname</surname>
      <affiliation>
        <!-- Valid email...spamblock/scramble if so desired -->
        <address><email>xxx@xxx.xxx</email></address>
      </affiliation>
    </author>

    <!-- All dates specified in ISO "YYYY-MM-DD" format -->
    <pubdate>2002-04-19</pubdate>

    <!-- Most recent revision goes at the top; list in descending order -->
    <revhistory>
      <revision>
        <revnumber>1.0</revnumber>
        <date>2002-04-19</date>
        <authorinitials>xx</authorinitials>
        <revremark>first official release</revremark>
      </revision>
      <revision>
        <revnumber>0.9</revnumber>
        <date>2002-04-15</date>
        <authorinitials>xx</authorinitials>
        <revremark>first draft proposed</revremark>
      </revision>
    </revhistory>

    <!-- Provide a good abstract; a couple of sentences is sufficient -->
    <abstract>
      <para>
        This is a sample DocBook XML HOWTO which has been
        constructed to serve as a document template.
      </para>
    </abstract>

  </articleinfo>
```

```
<!-- Content follows...include introduction, license information, feedback -->

<sect1 id="intro">
  <title>Introduction</title>

  <para>
    Provide a good overview and anything else you may think is needed in
    this area. Looking at other HOWTOs may assist in providing ideas.
  </para>

  <!-- Legal Sections -->
  <sect2 id="copyright">
    <title>Copyright and License</title>

    <!-- The LDP recommends, but doesn't require, the GFDL -->
    <para>
      This document, <emphasis>Sample XML HOWTO</emphasis>,
      is copyrighted (c) 2002 by <emphasis>author_name</emphasis>.
      Permission is granted to copy, distribute and/or modify this
      document under the terms of the GNU Free Documentation
      License, Version 1.1 or any later version published
      by the Free Software Foundation; with no Invariant Sections,
      with no Front-Cover Texts, and with no Back-Cover Texts.
      A copy of the license is available at
      <ulink url="http://www.gnu.org/copyleft/fdl.html">
      http://www.gnu.org/copyleft/fdl.html</ulink>.
    </para>

    <para>
      Linux is a registered trademark of Linus Torvalds.
    </para>
  </sect2>

  <sect2 id="disclaimer">
    <title>Disclaimer</title>

    <para>
      No liability for the contents of this document can be accepted.
      Use the concepts, examples and information at your own risk.
      There may be errors and inaccuracies, that could be damaging to
      your system. Proceed with caution, and although this is highly
      unlikely, the author(s) do not take any responsibility.
    </para>

    <para>
      All copyrights are held by their by their respective owners,
      unless specifically noted otherwise. Use of a term in this
      document should not be regarded as affecting the validity of any
      trademark or service mark. Naming of particular products or
      brands should not be seen as endorsements.
    </para>
  </sect2>

  <!-- Give credit where credit is due...very important -->
  <sect2 id="credits">
    <title>Credits / Contributors</title>
```

```
<para>
  In this document, I have the pleasure of acknowledging:
</para>

<!-- Please scramble addresses; help prevent spam/email harvesting -->
<itemizedlist>
  <listitem>
    <para>
      Individual 1 <email>someone1 (at) somewhere.org</email>
    </para>
  </listitem>
  <listitem>
    <para>
      Individual 2 <email>someone2 (at) somewhere.org</email>
    </para>
  </listitem>
</itemizedlist>
</sect2>

<!-- Feedback -->
<sect2 id="feedback">
  <title>Feedback</title>

  <para>
    Feedback is most certainly welcome for this document. Send
    your additions, comments and criticisms to the following
    email address : <email>xxx@xxx.xxx</email>.
  </para>
</sect2>

<!-- Translations -->
<sect2 id="translations">
  <title>Translations</title>

  <para>
    Pointers to available translations are nice.
    Also your translators tend to give very important inputs.
  </para>

  <itemizedlist>
    <listitem>
      <para>
        <ulink url="http://tldp.org/">French Translation</ulink>
        provided by Individual <email>someone (at) somewhere.fr</email>
      </para>
    </listitem>
    <listitem>
      <para>
        <ulink url="http://tldp.org/">German Translation</ulink>
        provided by Individual <email>someone (at) somewhere.de</email>
      </para>
    </listitem>
  </itemizedlist>
</sect2>

</sect1>
```

```
<!-- rest of document follows... -->
</article>
```

## Notlar

- a) Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler buldukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.
- b) Konsol görüntüsünü temsil eden sarı zeminli alanlarda metin genişliğine sığmayan satırların sığmayan kısmı `▮` karakteri kullanılarak bir alt satıra indirilmiştir. Sarı zeminli alanlarda `▮` karakteri ile başlayan satırlar bir önceki satırın devamı olarak ele alınmalıdır.

(B1) <http://nilgun.buguner.name.tr>

(B2) <http://www.arayan.com/da/>

(B3) <https://lists.sourceforge.net/lists/listinfo/belgeler-grubu>

(B4) [../manpages/](http://manpages/)

(B5) [/KiTAPLIK/](http://kitaplik.com/)

(B6) <http://www.oasis-open.org/docbook/xml/4.1.2/docbkx412.zip>

(B7) <http://prdownloads.sourceforge.net/docbook/docbook-xsl-1.49.tar.gz>

(B11) [../howto/turkish-howto.pdf](http://howto.turkish-howto.com/pdf)

(B13) <http://docbook.org/tdg/en/tdg-en-html-2.0.4.zip>

(B14) <http://www.oasis-open.org/docbook/xml/4.1.2/docbkx412.zip>

(B15) <file:///usr/share/sgml/docbook/>

(B16) <http://prdownloads.sourceforge.net/docbook/docbook-xsl-1.49.tar.gz>

(B17) <file:///usr/share/sgml/docbook/>

(B18) <ftp://fr.rpmfind.net/linux/gnome.org/pre-gnome2/sources/>

(B19) <ftp://fr.rpmfind.net/linux/gnome.org/pre-gnome2/sources/>

(B20) [../howto/cvs-acikkod-nasil.pdf](http://howto.cvs-acikkod-nasil.com/pdf)

(B21) <http://www.w3.org/TR/REC-xml>

(B22) <http://www.w3.org/TR/REC-xml-names>

(B23) <http://www.w3.org/TR/xml-infoset>

(B24) <http://www.w3.org/TR/xmlschema-0/>

(B25) <http://www.w3.org/TR/xmlschema-1/>



(B26) <http://www.w3.org/TR/xmlschema-2/>

(B29) <http://docbook.org/tdg/en/tdg-en-html-2.0.4.zip>

(B30) <https://sourceforge.net/projects/belgeler/>

(B31) <http://www.w3.org/TR/xsl/>

(B32) <http://www.w3.org/TR/xslt>

(B33) <http://www.w3.org/TR/xml-style-sheet>

(B34) <http://www.w3.org/TR/xpath>

(B35) <http://linuxdoc.org>

Bu dosya (GroupManual.pdf), belgenin XML biçiminin  $\TeX$ Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

27 Şubat 2007