

# XML-Namensräume (NSP)

## **Lernziele**

- Sie verstehen Prinzip und Sinn des Namensraumkonzepts von XML.
- Sie wissen, was ein Qualified Name, ein Namensraum-Präfix und ein Namensraum-URI sind und wozu Sie verwendet werden.
- Sie können Namensräume deklarieren und in Ihren XML-Dokumenten verwenden.
- Sie kennen den Gültigkeitsbereich von deklarierten Namensräumen.
- Sie sind in der Lage, einen Standard-Namensraum für ein XML-Dokument zu deklarieren.
- Sie kennen den Zusammenhang zwischen Validierung und Namensräumen.
- Sie setzen das erworbene Wissen praktisch in den Übungen ein!

# Was sind Namensräume?

*They aren't a cure of cancer, they aren't a way to win the lottery, and they aren't a direct cause of world peace.*

*Ronald Bourret über Namensräume*

*Namespaces in XML is one of the greatest disasters that XML's creators have inflicted upon XML*

*Simon St. Laurent*

Der **Namensraum** (*namespace*) ist ein Konzept des W3C. Es handelt sich dabei um eine Ergänzung zur XML-Spezifikation, die vom W3C im Januar 1999 als Empfehlung (*Recommendation*) verabschiedet wurde. Sie enthält Regelungen, wie XML-Elemente und -Attribute zu bezeichnen sind, wenn sie aus unterschiedlichen Zusammenhängen stammen.

## **Ziel:**

Vermeidung von Namenskonflikten und Mehrdeutigkeiten.

Wenn man bestehende Definitionen aus verschiedenen DTDs übernehmen oder segmentierte DTDs zusammenfügen möchte, läuft man Gefahr, Mehrdeutigkeiten in der neuen DTD zu haben.

Das Namensraumkonzept erlaubt es, den Kontext eines Elements in einem Namensraum zu bestimmen.

## **Beispiel NSP-1: Namenskonflikte und Mehrdeutigkeit**

```
<mitarbeiter name="Heckmann">

  <!-- Adresse 1 -->

  <adresse>
    <strasse>Schneller Weg</strasse>
    <hausnummer>99</hausnummer>
    <plz>12435</plz>
    <ort>Berlin</ort>
  </adresse>

  <!-- Adresse 2 -->

  <rechner>
    <domain>firma.de</domain>
    <name>mitarbeiterrechner</name>
```

```
<adresse>168.111.222.333</adresse>
</rechner>

</mitarbeiter>
```

Wollte man die unterschiedlichen gemeinten Adressen in einem Dokument verwenden - und bei technischer Dokumentation wäre das ja zumindest möglich -, wäre völlig unklar, was ein Element `adresse` beinhalten darf und soll. Es wäre *mehrdeutig*. Sollte der Name `adresse` in einem Dokument vorkommen, welches von einem XML-Prozessor auf Gültigkeit geprüft wird, käme es zu einem *Namenskonflikt (name collision)*.

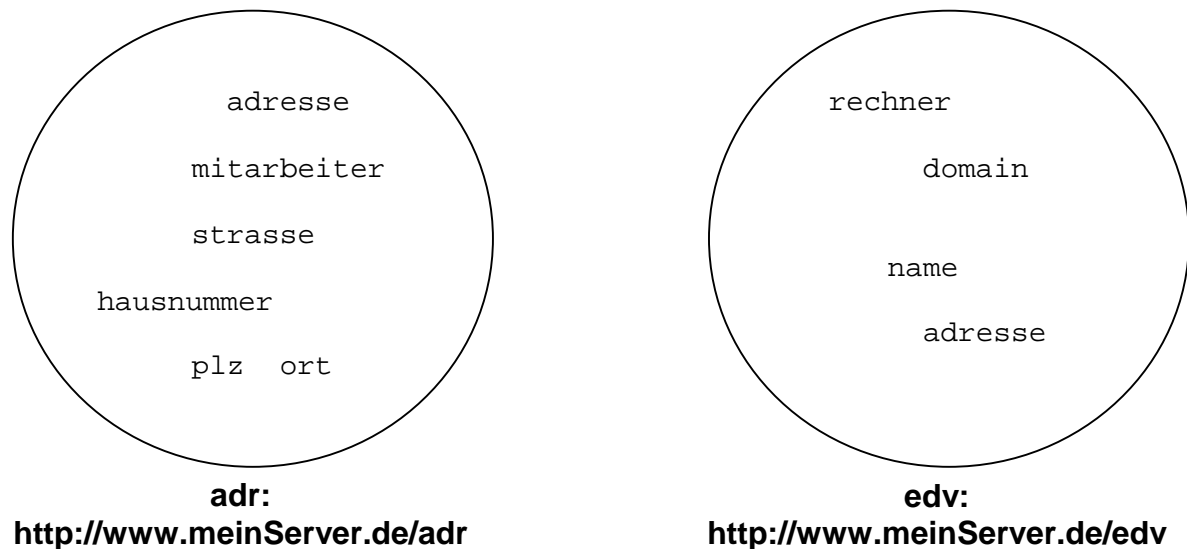
#### **Beispiel NSP-2: Rechnung mit Namen aus verschiedenen Domänen**

```
<Rechnung>
  <Kunde kundenID="765432-87">
    <Name>Maier</Name>
    <Vorname>Dieter</Vorname>
  </Kunde>
  <Produkt produktID=543216688">
    <Name>Schachbrett</Name>
    <Beschreibung>
      <html>
        <head>
          <title>Produktbeschreibung</title>
        </head>
        <body>
          <p>
            30x30 großes Teakholz-Schachbrett. Schön
            verziert
          </p>
        </body>
      </html>
    </Beschreibung>
    <Preis>56543 Euro</Preis>
  </Produkt>
</Rechnung>
```

**XML-Namensräume** sind eine Zusammenstellung von Namen, die in XML-Dokumenten als Namen für Elementtypen und Attribut-Namen verwendet werden.

# Identifikation von Namensräumen

Um Namensräume voneinander zu unterscheiden, benötigt man eindeutige Bezeichner. Das XML-Namensraumkonzept verwendet dazu URIs.



*Abbildung NSP-1: Zwei Namensräume*

Denkbar wäre also eine Namensbezeichnung wie z.B:

### **Beispiel NSP-3: Namensraumidentifikation über URIs**

```
<{http://www.meinServer.de/adr}adresse>  
...  
</{http://www.meinServer.de/adr}adresse>
```

Zwei Gründe sprechen allerdings gegen die direkte Verwendung von Namensraum-URIs in XML-Namen:

- Die Länge der URIs
- URIs enthalten Zeichen, die in XML-Namen nicht zulässig sind

Im XML-Namensraumkonzept werden deshalb Aliase für die URI verwendet, die XML-Namen sein müssen. Der Doppelpunkt (:) fungiert dabei als Namensraum-Delimiter.

#### **Beispiel NSP-4: Namensraumverwendung**

```
<adr:mitarbeiter>
  <adr:name>Dieter Thoma</adr:name>
  <adr:adresse>
    <adr:strasse>Schneller Weg</adr:strasse>
    <adr:hausnummer>99</adr:hausnummer>
    <adr:plz>12435</adr:plz>
    <adr:ort>Berlin</adr:ort>
  </adr:adresse>
  <edv:rechner>
    <edv:domain>firma.de</edv:domain>
    <edv:name>mitarbeiterrechner</edv:name>
    <edv:adresse>168.111.222.333</edv:adresse>
  </edv:rechner>
</adr:mitarbeiter>
```

Der Namensraum-URI heißt **Namensraumname** und der Alias für den Namensraumnamen wird **Namensraum-Präfix** genannt.

# Qualified Names

Kern des Namensraum-Konzepts sind die sogenannten **Qualified Names (qualifizierten Namen)**. Elementnamen und Attribut-Namen werden mit einem Präfix versehen, über welches der Namensraum identifizierbar ist, zu dem das Element oder Attribut gehört. Ein Doppelpunkt trennt das Präfix vom **lokalen Teil** des Qualified Name.

## **Qualified Name**

*Namensraum-Präfix + : + lokaler Name*

### **Beispiel NSP-5: Qualified Name**

```
<buch:titel>XML in 21 Tagen</buch:titel>
```

Für die Namensraum-Präfixe und den lokalen Teil des qualifizierenden Namens gelten die XML-Namensregeln mit der Ausnahme, dass sie selbst keinen Doppelpunkt enthalten dürfen.

# Namensraum-Deklarationen

Namensraum-Deklarationen stellen die Verbindung zwischen Namensraum-Präfix und Namensraumnamen her.

Namensräume werden im Start-Tag eines Elements durch Verwendung eines reservierten Attributs deklariert. Der Attribut-Name beginnt mit `xmlns:` und endet mit dem Namensraum-Präfix. Der **Name** eines Namensraums ist sein URI. Das Namensraum-Präfix dient in den Qualified Names als Abkürzung für den eigentlichen Namen (den URI) des Namensraums.

## **Namensraum-Deklaration**

```
<präfix:element xmlns:präfix="Namensraum-URI">...</element>
```

### **Beispiel NSP-6: Namensraum-Deklaration**

```
<buch:buch buch:auflage="199.000"
  xmlns:buch="http://meinverlag.de/">
  <buch:titel xmlns:buch="http://meinverlag.de/">
    XML in 21 Tagen
  </buch:titel>
  <buch:autor xmlns:buch="http://meinverlag.de/">
    Der XML-Guru
  </buch:autor>
</buch:buch>
```

### **Beachten Sie:**

*Der Namensraum-URI muss nicht auf ein existierendes Dokument einer bestimmten Form verweisen. Er dient lediglich als weltweit eindeutiger Bezeichner!*

Über das Namensraum-Präfix kann nun durch "Nachschlagen" in der Deklaration (**Namespace Lookup**) die Namensraumzugehörigkeit des Elements ermittelt werden. Namensraumname plus Elementname bezeichnet man auch als **Expanded Name**. Diesen Expanded Name benutzt jedoch nur der XML-Prozessor intern, er taucht niemals im XML-Dokument auf.

## Gültigkeitsbereich von Namensraum-Deklarationen

Die Namensraum-Deklaration gilt für das Element, in dem sie angegeben ist, und für alle Elemente im Inhalt dieses Elements (alle Kindelemente), objekt-orientiert ausgedrückt: sie wird vererbt.

### **Beispiel NSP-7: Vererbung bei Namensraum-Deklarationen**

```
<buch:buch buch:auflage="199.00"  
  xmlns:buch="http://meinverlag.de/">  
  <buch:titel>XML in 21 Tagen</buch:titel>  
  <buch:autor>Der XML-Guru</buch:autor>  
</buch:buch>
```

### **Beachten Sie:**

*Attribute erhalten nicht automatisch den Namensraum des zugehörigen Elements.*

### **Beispiel NSP-8: Namensraum von Attributen**

```
<buch:buch auflage="199.00"  
  xmlns:buch="http://meinverlag.de/">  
  <buch:titel>XML in 21 Tagen</buch:titel>  
  <buch:autor>Der XML-Guru</buch:autor>  
</buch:buch>
```

In diesem Falle gehört das Attribut `auflage` des Elements nicht zum Namensraum `buch` des Elements `buch`.

## Mehrere Namensraum-Deklarationen in einem XML-Dokument

Sie dürfen auch mehrere Deklarationen in einem Element verwenden. Auf diese Weise können Sie z.B. HTML-Code direkt im XML-Code verwenden.

### **Beispiel NSP-9: Mehrere Deklarationen in einem Element**

```
<html:html xmlns:html="http://www.w3.org/TR/REC-html40"  
  xmlns:buch="http://meinverlag.de/">  
  <html:head>  
    <html:title>Literaturverzeichnis</html:title>  
  </html:head>  
  <html:body>
```



```

<buch:buch buch:auflage="199.00">
  <html:table>
    <html:tr>
      <html:td>
        <buch:titel>XML in 21 Tagen</buch:titel>
      </html:td>
      <html:td>
        <buch:autor>Der XML-Guru</buch:autor>
      </html:td>
    </html:tr>
  </html:table>
</buch:buch>
</html:body>
</html:html>

```

## Deklaration eines Standard-Namensraums

Elemente lassen sich auch ohne Präfix - über die Deklaration eines Standard-Namensraums an Namensräume binden. Zu diesem Zweck deklariert man den Namensraum mit `xmlns` ohne Präfix.

### ***Deklaration eines Standard-Namensraums***

```
<element xmlns="Namensraum-URI">... </element>
```

### ***Beispiel NSP-10: Deklaration eines Standard-Namensraums***

```

<html xmlns="http://www.w3.org/TR/REC-html40">
  <head>
    <title>Weiterleitung</title>
  <body>
    <p align=justify">Wir sind umgezogen:
      <a href="http://ww.spielzeug.de">besuchen Sie
uns</a>.
    </p>
  </body>
</html>

```

Natürlich kann man einen Standard-Namensraum auch mit expliziten Namensräumen kombinieren. So können Sie z.B. in obigem Beispiel den HTML-Namensraum als Standard-Namensraum deklarieren und den `buch`-Namensraum explizit angeben. Dann mischen Sie einfach ihre eigenen `<buch>`-Elemente mit HTML-Elementen, ohne jedes mal das Namensraum-Präfix vor jeden HTML-Elementnamen schreiben zu müssen.

**Beispiel NSP-11: Kombination Standard-Namensraum mit expliziter Namensraum-Deklaration**

```
<html xmlns="http://www.w3.org/TR/REC-html40"
      xmlns:buch="http://meinverlag.de/">
  <head>
    <title>Literaturverzeichnis</title>
  </head>
  <body>
    <buch:buch buch:auflage="199.00">
      <table>
        <tr>
          <td>
            <buch:titel>XML in 21 Tagen</buch:titel>
          </td>
          <td>
            <buch:autor>Der XML-Guru</buch:autor>
          </td>
        </tr>
      </table>
    </buch:buch>
  </body>
</html>
```

**Beachten Sie:**

Standard-Namensräume gelten nicht für Attribute.

## Umdeklaration von Namensräumen

Die Namensraum-Deklaration kann durch eine andere Namensraum-Deklaration, die dasselbe Präfix benutzt, überschrieben werden. In einem solchen Fall wird also ein Namensraum umdeklariert.

**Beispiel NSP-12: Überschreiben von Namensraum-Deklarationen**

```
<buch:buch buch:auflage="199.00"
      xmlns:buch="http://meinverlag.de/">
  <buch:titel xmlns:buch="http://deinverlag.fr/">XML in
  21 Tagen</buch:titel>
  <buch:autor>Der XML-Guru</buch:autor>
</buch:buch>
```

## Undeklaration von Standard-Namensräumen

Es ist möglich, wenn auch in den seltensten Fällen zu empfehlen, einen deklarierten Standard-Namensraum aufzuheben:

### **Beispiel NSP-13: Undeklaration eines Standard-Namensraums**

```
<buch auflage="199.00"
  xmlns="http://meinverlag.de/">
  <titel>XML in 21 Tagen</titel>
  <autor>Der XML-Guru</autor>
  <kritik xmlns="">Nicht zu empfehlen</kritik>
</buch>
```

### **Beachten Sie:**

Ein mit einem Präfix assoziierter Namensraum kann in der Version 1.0 der Namensraum-Empfehlung nicht undeklariert werden.

### **Beispiel NSP-14: Unzulässige Undeklaration eines Namensraums**

```
<buch:buch auflage="199.00"
  xmlns="http://meinverlag.de/">
  <buch:titel>XML in 21 Tagen</buch:titel>
  <buch:autor>Der XML-Guru</buch:autor>
  <buch:kritik xmlns:buch="">
    Nicht zu empfehlen
  </buch:kritik>
</buch:buch>
```

# Komponenten des Namensraumkonzepts

## Beispiel NSP-15: Namensraum-Deklaration

```
<buch:autor xmlns:buch="http://meinVerlag.de/buch">  
  <!-- Inhalt des Elements -->  
</buch:autor>
```

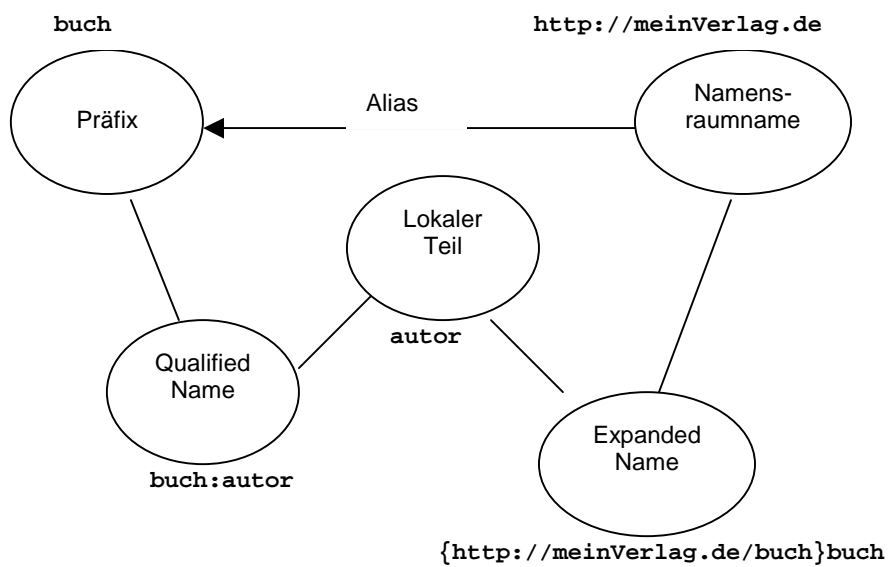


Abbildung NSP-2: Zusammenhang der Komponenten des Namensraumkonzepts

# Fallen im Umgang mit Namensräumen

## Was Namensräume nicht sind

- Ressourcen, die real irgendwo existieren
- Ein Teil der XML-1.0-Syntax
- Direkt verknüpft mit DTDs oder XML-Schemas
- Eine Redefinition des Gültigkeitskonzepts von XML
- Etwas mit einer semantischen Bedeutung
- Ein Objekt, eine Schnittstelle oder irgendetwas anderes mit einer internen Struktur
- Eine Technik, um zwei Dokumente zusammenzuführen, die verschiedene DTDs nutzen
- Eine Technik, um XML-Dokumente zu validieren, die XML-Namensräume verwenden
- Eine Technik, um Elementtyp-Deklarationen und Attributlisten-Deklarationen in einer DTD mit XML-Namensräumen zu assoziieren

## Namensräume und Attribute

- Ein Attribut-Name ohne Präfix gehört explizit zu keinem Namensraum (auch nicht zum Namensraum des Elements!).
- Standard-Namensraum-Deklarationen haben keinen Effekt auf irgendeinen Attribut-Namen.
- Kein Start-Tag darf zwei Attribute mit dem gleichen Namen enthalten. Dies gilt unabhängig davon, ob der Attributname ein Qualified Name ist oder nicht.
- Kein Start-Tag darf zwei qualifizierte Attribut-Namen enthalten, die den gleichen lokalen Teil besitzen und verschiedene Präfixe, die zum selben Namensraumnamen gehören.
- Attribute dürfen nur einen oder keinen Doppelpunkt enthalten (:). Enthält ein Attribut-Name einen Doppelpunkt, so wird er als Namensraum-

Delimiter interpretiert, der den lokalen Teil vom Namensraum-Präfix trennt.

- Einige Attributwerte können als Typ deklariert werden, die XML-Namen enthalten müssen (ID, IDREF, IDREFS, ENTITY, ENTITIES, NOTATION). Attribute dieser Typen dürfen keine Doppelpunkte in ihren Werten enthalten. Dies kann allerdings nur durch einen validierenden Parser sichergestellt werden.

## Ausblick: Namensräume in XML 1.1

Zur Zeit arbeitet die XML-Arbeitsgruppe des W3C an einer Version der Namensraum-Empfehlung, die für XML 1.1 gelten soll.

Folgende Spezifikationen sind in diesem Zusammenhang relevant

- *Namespaces in XML Requirements 1.1* (<http://www.w3.org/TR/xml-names11-req/>)
- *Namespaces in XML 1.1*

**Namespaces in XML 1.1 liegt zur Zeit als Candidate Recommendation vor.**

**URL:**

<http://www.w3.org/TR/xml-names11/>

## Undeklaration von Namensräumen

Im Gegensatz zur ersten Version der Namensraum-Empfehlung ist es nun erlaubt, die Deklaration von Namensräumen mit Präfixen rückgängig zu machen. Dies geschieht, indem man dem zugehörigen Namensraum-Präfix eine leere Zeichenkette als Namensraumname zuweist.

### **Beispiel NSP-16: Undeklaration von Namensräumen**

```
<buch:buch auflage="199.00"
  xmlns="http://meinverlag.de/">
  <buch:titel>XML in 21 Tagen</buch:titel>
  <buch:autor>Der XML-Guru</buch:autor>
  <buch:kritik xmlns:buch="">
    Nicht zu empfehlen
  </buch:kritik>
</buch:buch>
```

Eine solche Undeklaration von Namens war bisher nur für Standard-Namensräume möglich.

## IRIs

### IRI = Internationalized Resource Identifiers

In *Namespaces in XML 1.1* übernehmen IRIs den Platz von URIs. D.h., dass Namensraum-Namen nun IRIs sind.

**Eine IRI-Spezifikation wurde vom W3C entworfen und liegt als Internet-Draft der IETF vor.**

**URL:**

<ftp://ftp.nordu.net/internet-drafts/draft-duerst-iri-04.txt>

<http://www.w3.org/International/iri-edit/draft-duerst-iri-04.txt>

Der Unterschied zwischen URIs und IRIs besteht hauptsächlich im zulässigen Zeichensatz. Während in URIs lediglich eine Untermenge von ASCII-Zeichen zulässig sind, stützen sich die URIs auf das Universal Character Set (UCS) von Unicode bzw. ISO/IEC 10646.

**Beispiel NSP-17: Ein IRI als Namensraumname**

```
<buch:buch buch:auflage="199.00"
  xmlns:buch="http://meinverlag.de/§342">
  <buch:titel>XML in 21 Tagen</buch:titel>
  <buch:autor>Der XML-Guru</buch:autor>
</buch:buch>
```



## Namensräume und Validierung

Das Namensraumkonzept ist in der XML-Community heftig umstritten, ist jedoch in allen gängigen XML-Prozessoren integriert und nicht mehr wegzudenken aus der XML-Welt.

Dennoch gibt es einige Dinge, die Sie beachten müssen, wenn Sie Namensräume in Dokumenten verwenden möchten, die gegen eine DTD validiert werden sollen.

So können Sie z.B. einem gültigen XML-Dokument nicht einfach Elemente und Attribute eines anderen Namensraumes hinzufügen, ohne dass das Dokument ungültig wird.

**Grund:** Die hinzugefügten Elemente und Attribute sind nicht in der DTD deklariert. Zudem ist der Namensraum-URI lediglich ein Name zur eindeutigen Identifikation des Namensraums. Ein XML-Prozessor lädt keine DTD oder ein Schema von einer in der Namensraum-Deklaration zugewiesenen Internet-Adresse.

### Namensräume in gültigen XML-Dokumenten

Sollen Ihre XML-Dokumente, die Namensräume verwenden, gültig sein, müssen die Elementtypen und Attribute, welche einem Namensraum zugeordnet sind, über einen Qualified Name deklariert werden (d.h. mit Namensraum-Präfix). Zudem müssen auch die `xmlns`-Attribute, die zur Kennzeichnung von Namensraum-Deklarationen dienen, in der DTD deklariert werden.

#### **Beispiel NSP-18: Ungültiges XML-Dokument mit Namensräumen**

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

<!DOCTYPE notiz [
  <!ELEMENT notiz (#PCDATA)>
  <!ATTLIST notiz style CDATA #IMPLIED>
]>

<notiz html:style="font-color:red"
  xmlns:html="http://www.w3.org/TR/REC-html40">
  Dieses Dokument ist leider nicht gültig!
</notiz>
```

### Beispiel NSP-19: Gültiges XML-Dokument mit Namensräumen

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

<!DOCTYPE notiz [
  <!ELEMENT notiz (#PCDATA)>
  <!ATTLIST notiz html:style CDATA #IMPLIED
                xmlns:html CDATA #FIXED
  "http://www.w3.org/TR/REC-html40">
]>

<notiz html:style="font-color:red"
       xmlns:html="http://www.w3.org/TR/REC-html40">
  Dieses Dokument ist gültig!
</notiz>
```

Dem Wurzelement `notiz` ist das Attribut `style` aus dem `html`-Namensraum zugeordnet. Das Attribut wird über das Präfix `html` qualifiziert und der Namensraum im Wurzelement deklariert. In der DTD ist dieses Attribut *mit Präfix* zu deklarieren: `<!ATTLIST notiz html:style CDATA #IMPLIED>`. Darüber hinaus muss auch das `xmlns`-Attribut, welches zur Deklaration des Namensraums verwendet wird, in der DTD deklariert werden. Dies kann mittels einer Attribut-Vorgabe in Kombination mit dem Schlüsselwort `#FIXED` erfolgen. Dadurch lässt sich sicherstellen, dass im Wurzelement `notiz` immer eine Namensraum-Deklaration für den HTML-Namensraum erfolgt, auch wenn sie nicht explizit angegeben wird.

Ein Beispiel aus der Praxis:

### Beispiel NSP-20: Praxisbeispiel - gültiges XML-Dokument mit DTD

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

<!DOCTYPE ma:mitarbeiterliste [
  <!ELEMENT ma:mitarbeiterliste (ma:mitarbeiter+)>
  <!ATTLIST ma:mitarbeiterliste
            xmlns:ma CDATA #FIXED
            "http://www.meinServer.de/adresse"
            xmlns:edv CDATA #FIXED
            "http://www.meinServer.de/rechner"
  >
  <!ELEMENT ma:mitarbeiter (ma:name, ma:adresse,
edv:rechner)>
  <!ELEMENT ma:adresse (ma:strasse, ma:ort)>
  <!ELEMENT edv:rechner (edv:name, edv:adresse)>
  <!ELEMENT ma:name (#PCDATA)>
  <!ELEMENT ma:strasse (#PCDATA)>
  <!ELEMENT ma:ort (#PCDATA)>
```

```

    <!ELEMENT edv:name (#PCDATA)>
    <!ELEMENT edv:adresse (#PCDATA)>
  ]>

  <ma:mitarbeiterliste
    xmlns:ma="http://www.meinServer.de/adresse"
    xmlns:edv="http://www.meinServer.de/edv">
    <ma:mitarbeiter>
      <ma:name>Dieter Thomas Heck</ma:name>
      <ma:adresse>
        <ma:strasse>Hitstr. 7</ma:strasse>
        <ma:ort>12435 Berlin</ma:ort>
      </ma:adresse>
      <edv:rechner>
        <edv:name>Saturn</edv:name>
        <edv:adresse>127.98.76.35</edv:adresse>
      </edv:rechner>
    </ma:mitarbeiter>
    <ma:mitarbeiter>
      <ma:name>Fritz Walter</ma:name>
      <ma:adresse>
        <ma:strasse>Am Stadion 5</ma:strasse>
        <ma:ort>10115 Berlin</ma:ort>
      </ma:adresse>
      <edv:rechner>
        <edv:name>Pluto</edv:name>
        <edv:adresse>156.97.234.98</edv:adresse>
      </edv:rechner>

    </ma:mitarbeiter>
  </ma:mitarbeiterliste>

```

### **Beachten Sie:**

Für gültige Dokumente fordert die XML-Empfehlung, dass der DTD-Name mit dem Namen des Wurzelements übereinstimmt. Ist auch der Name des Wurzelements über ein Präfix qualifiziert, so muss auch der DTD-Name dieses Präfix berücksichtigen.

## Namensräume in der DTD

In der DTD werden Namensräume nicht als solche erkannt. Für einen XML-Prozessor sind die Namensraum-Präfixe in einer DTD nicht Stellvertreter des Namensraum-URI, sondern lediglich ein Teil des Element- oder Attribut-Namen. Mit anderen Worten: eine DTD, in der Elementtypen aus zwei Na-

mensräumen deklariert werden, "weiss" nichts davon, dass sie es mit zwei Namensräumen zu tun hat. Das folgendes Beispiel verdeutlicht diesen Sachverhalt:

**Beispiel NSP-21: Namensräume in der DTD**

```
<?xml version="1.0"?>
  <!DOCTYPE ns1:notiz [
    <!ELEMENT ns1:notiz (ns2:notiz)>
    <!ATTLIST ns1:notiz xmlns:ns1 CDATA #FIXED
"http://www.meinServer.de">
    <!ELEMENT ns2:notiz EMPTY>
    <!ATTLIST ns2:notiz xmlns:ns2 CDATA #FIXED
"http://www.meinServer.de">
  ]>

<ns1:notiz xmlns:ns1="http://www.meinServer.de">
  <ns2:notiz xmlns:ns2="http://www.meinServer.de"/>
</ns1:notiz>
```

Die beiden in der DTD deklarierten Elementtypen besitzen den gleichen lokalen Namen `notiz`, unterscheiden sich jedoch im Präfix (`ns1` und `ns2`). Trotzdem gehören Sie zum selben Namensraum, da das Präfix nur ein Stellvertreter (Abkürzung) der Namensraum-URI ist. Entscheidend für die Identität von Namensräumen ist der Namensraum-URI und der ist für beide Elemente gleich. Dies gilt jedoch nicht innerhalb von DTDs! Aus Sicht der DTD werden zwei unterschiedliche Elementtypen (nämlich `ns1:notiz` und `ns2:notiz`) deklariert. Somit ist dieses Dokument gültig. Würde innerhalb einer DTD eine Zuordnung von Namensraum-Präfixen zu Namensraum-URIs erfolgen, würde ein validierender Parser beide Element-Namen als gleich betrachten und das Dokument für ungültig erklären, da ein Elementtyp in einer DTD nur einmal deklariert werden darf.

## **Wann werden Namensräume wirklich benötigt?**

- bei potentiellen Namenskonflikten
- beim Einsatz von mehreren XML-Anwendungen in einem Dokument (also eigentlich fast immer!)