

## LEERBOEK bij [avenue.quark](https://www.avenue.quark.nl)

 **Inhoud**

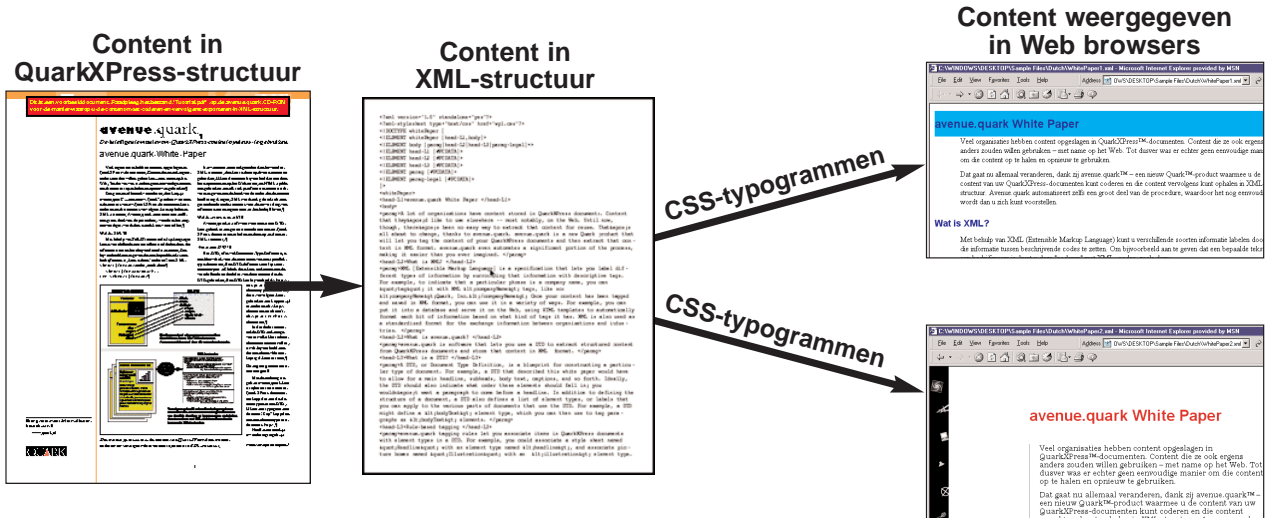
Inleiding	3
Vorbereidingen voor het werken met avenue.quark	5
Werken met codeerregels	6
Uitgepakte XML content bekijken – alleen voor Windows	13
Een codeerregel toevoegen aan een set met codeerregels	17
Tekstkaders combineren tot een fragment	22
Coderen met fragmenten en een nieuwe codeerregel	25
Een illustratie en een onderschrift van een code voorzien	27

## Inleiding

In dit Leerboek gaan we het volgende doen:

- 1 Een voorbeeld van een QuarkXPress Passport™-document openen.
- 2 Met behulp van `avenue.quark™`, samen met een meegeleverde DTD (Document Type Definitie), de meeste content in dit voorbeelddocument coderen.
- 3 De gecodeerde content bewaren in XML (Extensible Markup Language)-structuur.
- 4 De content bekijken in een Web browser met behulp van Cascading Style Sheets (alleen in Windows).
- 5 Het coderingsproces stroomlijnen door eerst de set met codeerregels bij te werken om meer soorten content te kunnen coderen en vervolgens verscheidene tekstkaders combineren tot een *fragment* dat als één geheel kan worden gecodeerd.
- 6 Een illustratie en een onderschrift van een code voorzien.

Het voordeel van avenue.quark is de mogelijkheid content in QuarkXPress Passport-structuur van een code te voorzien, deze content te exporteren in de gestandaardiseerde XML-structuur, waarna die content op verschillende manieren opnieuw kan worden gebruikt. In dit Leerboek bijvoorbeeld gaan we ons oorspronkelijke voorbeeldocument (links onder op de afbeelding) openen, zijn content met behulp van avenue.quark exporteren in XML-structuur, waarna we die content gaan bekijken in een Web browser aan de hand van twee volledig verschillende sets CSS (Cascading Style Sheet)-typogrammen (rechts onderaan in de afbeelding).



QuarkXPress Passport content bewaard als XML en bekeken in een Web browser die CSS ondersteunt

!!! Raadpleeg hoofdstuk 1, “De basisprincipes van avenue.quark” in “avenue.quark in vogelvlucht” (“avenue.quark in vogelvlucht.pdf”) voor een uitgebreide inleiding tot XML, DTD’s en avenue.quark. Wilt u meer informatie over codeerregels raadpleeg dan hoofdstuk 5 “Codeerregels” in hetzelfde bestand.

## Vorbereidingen voor het werken met avenue.quark

Voordat u begint met de oefeningen in dit Leerboek moet u eerst de volgende stappen uitvoeren:

- 1** | Controleer of u QuarkXPress Passport 4.04 of een hogere versie heeft.
- 2** | Controleer of de avenue.quark QuarkXTensions™-module en de Item Sequence QuarkXTensions-module in de “XTension”-map in uw QuarkXPress Passport-programmamap staan. Voor aanwijzingen bij de installatie verwijzen wij u naar het installatieboekje dat met avenue.quark wordt meegeleverd of naar het bestand “Installation Instructions.pdf”.
- 3** | Controleer of het bestand met de naam “WhitePaper - Nederlands.xmt” in de “Templates”-map in uw QuarkXPress Passport-programmamap staat.
- 4** | Kopieer de map “avenue.quark Tutorial” naar het bureaublad van uw computer, zodat u er gemakkelijk en snel bijkunt.

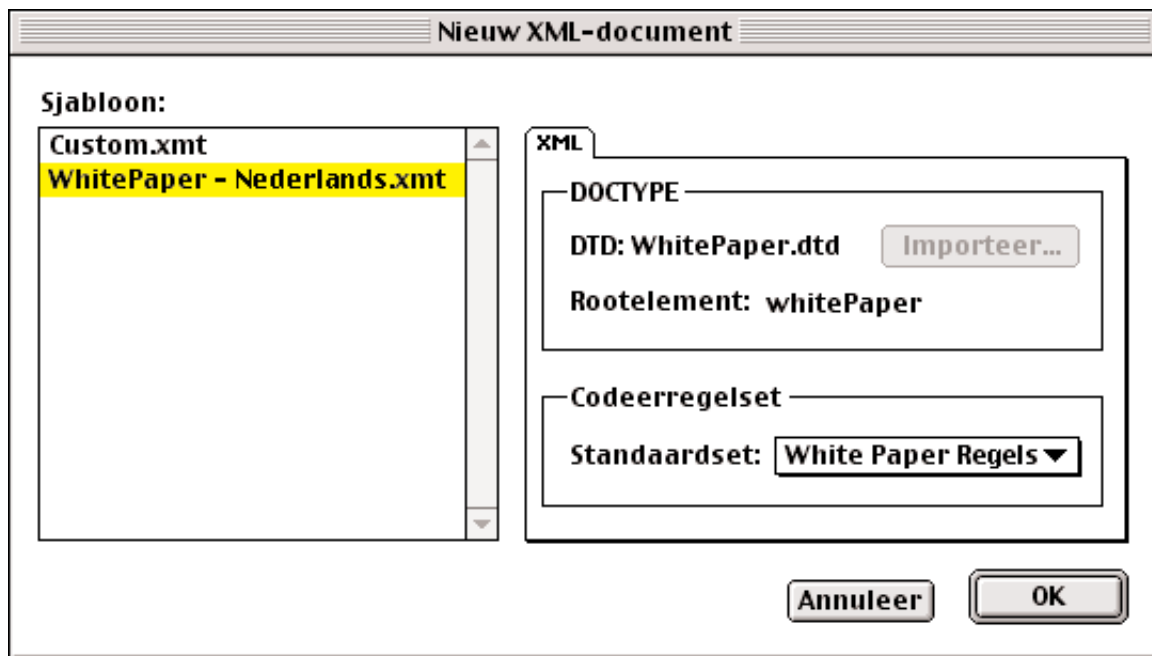
Een goed idee is dit Leerboek te printen, zodat u gemakkelijk even kunt terugkijken wanneer u de diverse stappen uitwerkt.

## Werken met codeerregels

In dit hoofdstuk zult u zien hoe u een functie moet gebruiken die veel van het coderen in XML automatiseert en die we *op regels gebaseerd coderen* noemen. Het coderen aan de hand van regels vindt plaats via een *set met codeerregels*, die eigenlijk een lijst met richtlijnen is voor het automatisch coderen van content in QuarkXPress Passport, gebaseerd op de typogrammen en andere vormgevingsspecificaties in deze content.

We gaan een reeds gemaakte set met codeerregels gebruiken om de content van een tekstkader in een voorbeelddocument te coderen:

- 1 Start QuarkXPress Passport.
- 2 Kies **Archief** (Mac OS) of **Bestand** (Windows) → **Open**; selecteer het bestand met de naam "SampleDoc.qxt" in de Nederlandse taalmap in de map "avenue.quark Tutorial" op uw bureaublad.
- 3 Kies **Archief** (Mac OS) of **Bestand** (Windows) → **Nieuw** → **XML** om een nieuw XML-document te maken. Het dialoogvenster **Nieuw XML-document** verschijnt.



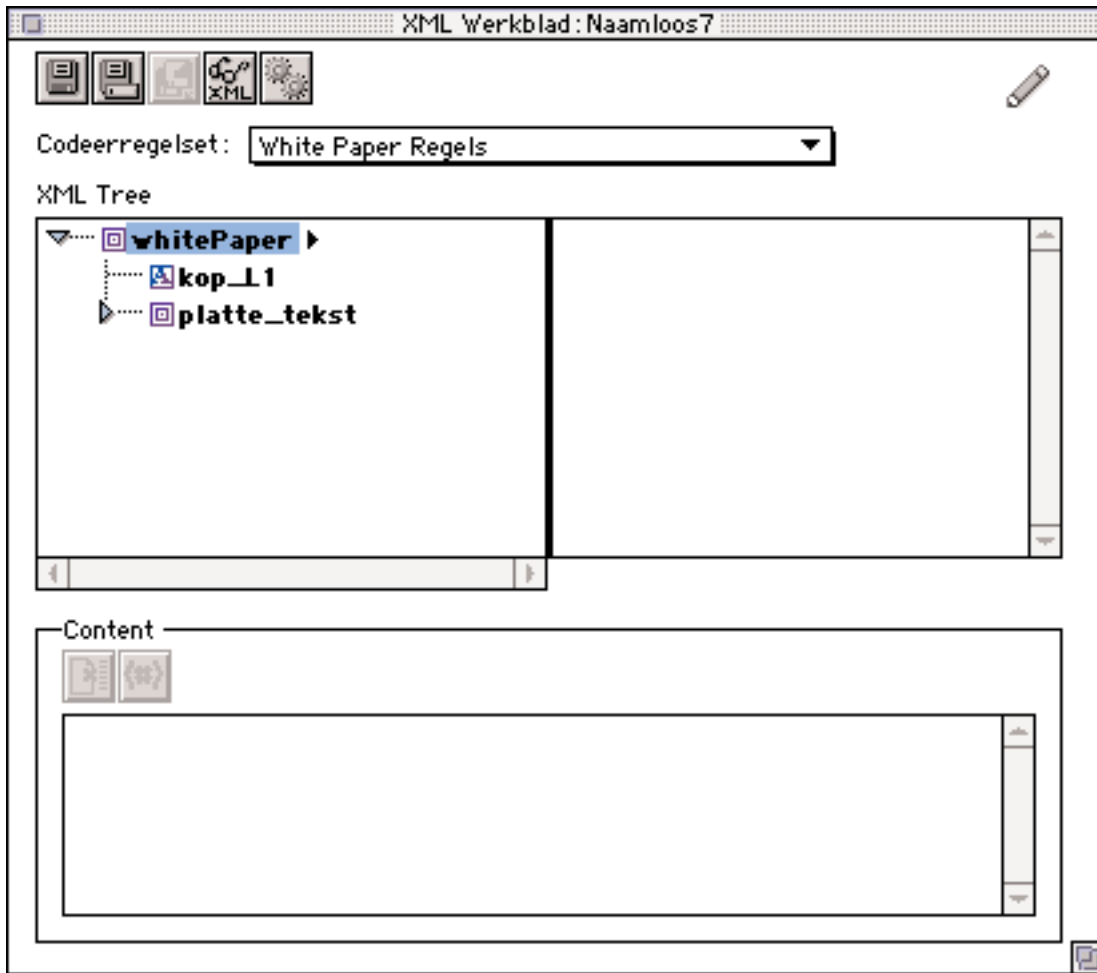
Het dialoogvenster **Nieuw XML-document**

!!! Als de XML-opdracht niet verschijnt in het submenu **Archief** (Mac OS) of **Bestand** (Windows) → **Nieuw** is avenue.quark misschien niet goed geïnstalleerd. Zie het bestand “Installation Instructions.pdf” voor aanwijzingen wat betreft de manier waarop de software moet worden geïnstalleerd.

4 Selecteer **WhitePaper - Nederlands.xmt** in de **Sjabloon**-lijst om aan te geven dat u het nieuwe XML-document wilt baseren op het sjabloon met de naam “WhitePaper - Nederlands.xmt.” In dit sjabloon wordt aangegeven dat u gaat werken met een DTD met de naam “WhitePaper.dtd”, met `<whitePaper>` als het rootelement en dat u gebruik maakt van een reeds geconfigureerde set codeerregels met de naam “White Paper Regels.”

💡 **Wat is een DTD?** Een DTD, ofte wel Document Type Definitie, definieert een set elementtypen (codedefinities) en een set structuurregels voor een specifiek type document. Een DTD voor een nieuwsbericht kan bijvoorbeeld specificeren dat in elk artikel een stuk tekst moet worden gecodeerd als een `<kop>`-element, dat elk artikel wel of niet een `<subkop>`-element kan hebben, dat iedere illustratie onmiddellijk moet worden gevolgd door precies één `<onderschrift>`-element enzovoort. Raadpleeg voor meer informatie over DTD's hoofdstuk 1 “De basisprincipes van avenue.quark” in het bestand “avenue.quark in vogelvlucht.pdf.”

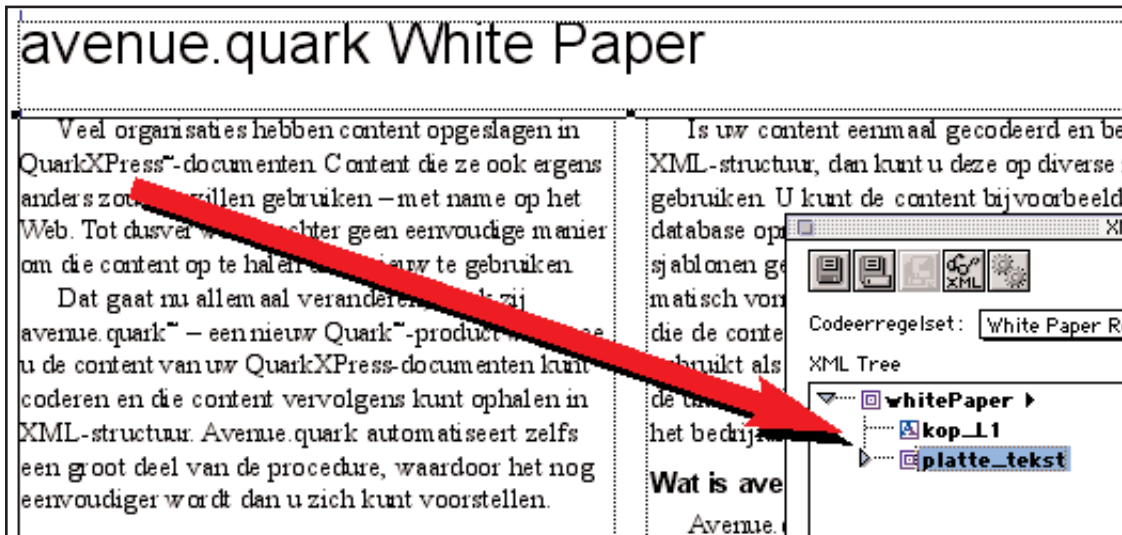
- 5 Klik op OK. Het XML Werkblad-palet verschijnt met alle verplichte elementen in de WhitePaper DTD (in dit geval `<whitePaper>`, `<kop_L1>` en ten minste één `<platte_tekst>`) die automatisch worden toegevoegd aan de XML Tree-lijst.



Het XML Werkblad-palet



- 6 Selecteer het tekstkader dat begint met “Veel organisaties hebben content...” Druk vervolgens op ⌘ (Mac OS) of Ctrl (Windows) en sleep het tekstkader op het `<platte_tekst>`-element in de XML Tree-lijst. (Namen van elementen worden vet weergegeven in de XML Tree-lijst, bijvoorbeeld: **platte\_tekst**.) Avenue.quark codeert automatisch de content in het tekstkader en maakt waar nodig nieuwe elementen aan.



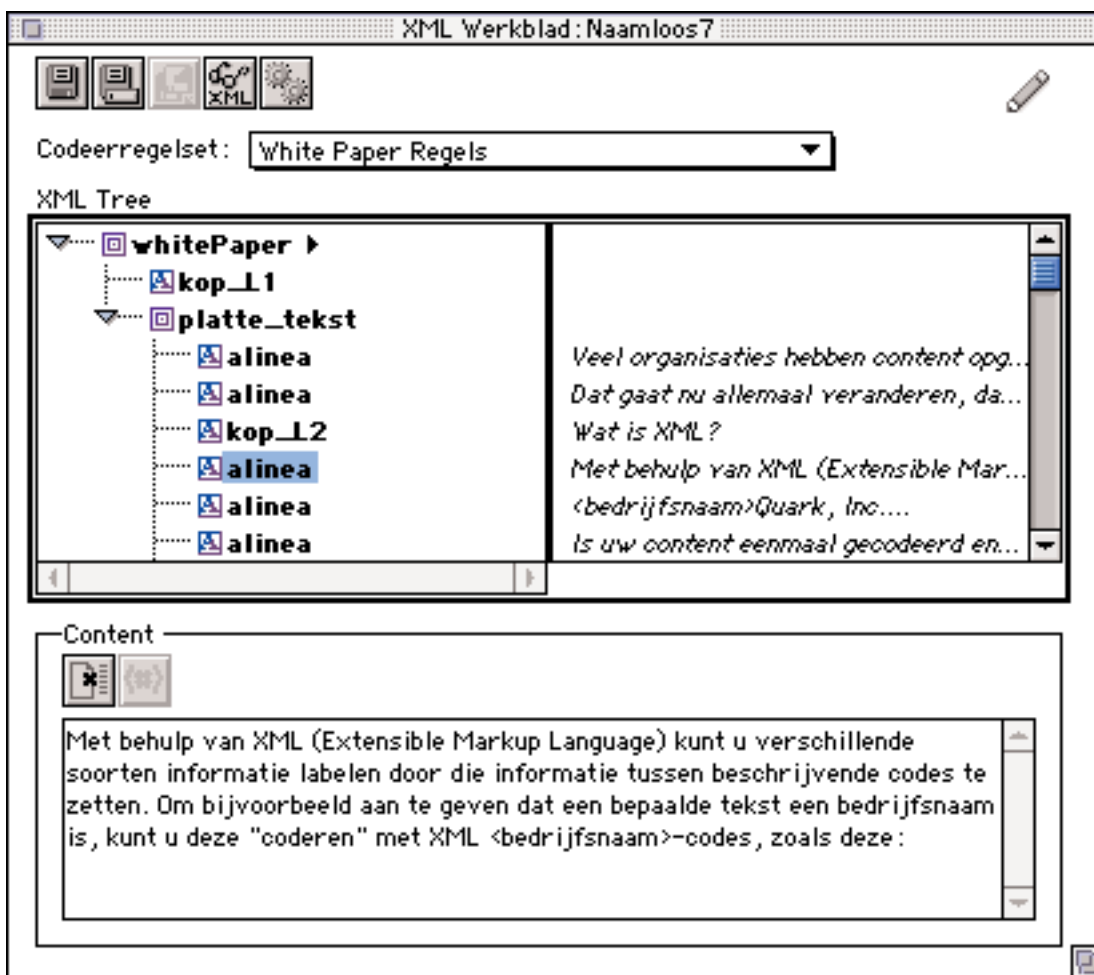
#### Coderen aan de hand van codeerregels

Wat is er zojuist gebeurd? Avenue.quark heeft net gebruik gemaakt van de codeerregelset “White Paper Regels” om daarmee de content in het tekstkader dat u zo juist op het `<platte_tekst>`-element heeft geslept te coderen. U krijgt meer informatie over codeerregelsets in het hoofdstuk “Een codeerregel toevoegen aan een set met codeerregels” verderop in dit Leerboek.

Hé, wacht even! Het werkte niet! Als het dialoogvenster **Kies codeerregel/positie** verschijnt wanneer u het tekstkader op het `<platte_tekst>`-element sleept, klik dan op **Stop**. Kies vervolgens **Wijzig** (Mac OS) of **Bewerk** (Windows) → **Voorkeuren** → **avenue.quark**; zorg ervoor dat het vakje **Voeg repeterende elementen altijd toe aan het eind van de huidige aftakking** is aangekruist; klik op **OK**; sluit vervolgens het XML Werkblad-palet en ga terug naar stap 3.

Als er iets anders onverwachts plaatsvond toen u het tekstkader op het `<platte_tekst>`-element in de XML Tree-lijst sleepte, hoeft u zich daarover geen zorgen te maken. Sluit gewoon het XML Werkblad-palet en ga terug naar stap 3. U kunt nog even controleren of het bestand "WhitePaper - Nederlands.xmt" in de map "avenue.quark Tutorial" in de map "Templates" in uw QuarkXPress Passport-programmamap staat.

**7** Blader door de XML Tree-lijst (in het XML Werkblad-palet) waarbij u kunt zien hoe de content van het "white paper" ofte wel witboek is gecodeerd met de juiste elementtypes. Subkoppen zijn gecodeerd als `<kop_L2>`-elementen en de alinea's met platte tekst hebben als code het `<alinea>`-element meegekregen. U kunt de volledige inhoud van een element zien door te klikken op de elementnaam en in het Content-veld te kijken.



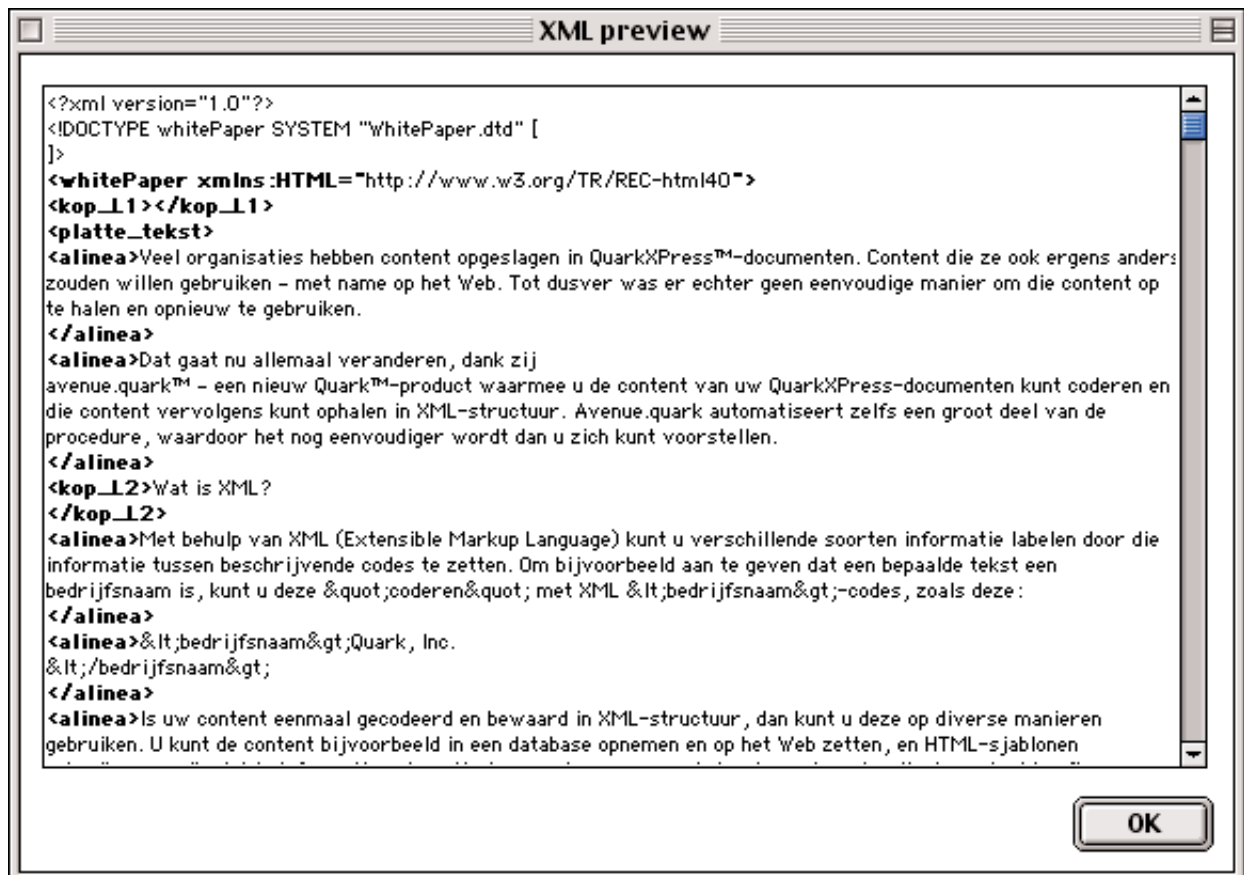
Het XML Werkblad-palet met het Content-veld

!!! Vier dingen die *niet* werden gecodeerd zijn de kop (“avenue.quark White Paper”), de illustratie, het onderschrift bij de illustratie en de juridische tekst die u kunt vinden in de laatste twee alinea’s op pagina 2. We gaan straks in dit Leerboek kijken hoe u deze op eenvoudige wijze kunt coderen.

8 Klik op de knop **XML preview** bovenin het **XML Werkblad**-palet om het dialoogvenster **XML preview** te openen. Hierin kunt u zien hoe het geëxporteerde XML-bestand eruit zal zien.



Knop **XML preview**



Het dialoogvenster **XML Preview**



**Waar kijk ik naar?** In het dialoogvenster **XML preview** ziet u hoe uw content eruit zal zien in XML-structuur. Elk stukje informatie in een XML-bestand staat tussen codes met punthaken, die beschrijven wat die content is. Bijna bovenaan het voorgaande schermvoorbeeld kunt u zien dat de eerste alinea in het document tussen een begincode `<alinea>` en een eindcode `</alinea>` staat. Hierdoor kunnen diverse andere programma's de platte tekst in het artikel identificeren en het op de juiste wijze weergeven – door het bijvoorbeeld op het scherm weer te geven in een 12-punts schreeffont.

**9** Klik op **OK** om het dialoogvenster **XML preview** te sluiten.

**1** Klik op de **Bewaar**-knop bovenin het **XML Werkblad**-palet en bewaar het document in de map “avenue.quark Tutorial” als “MijnDoc.xml”. (Laat het venster-/ uitrolmenu **Tekenset** op **UTF8** staan en kruis het vakje **Bewaar XML als standalone bestand** aan.) U gaat in het volgende hoofdstuk “Uitgepakte XML content bekijken – alleen voor Windows” het bestand bekijken in een Web browser.





**Bewaar**-knop

**11** Klik op het sluitvakje op het **XML Werkblad**-palet om het XML-document te sluiten.

## Uitgepakte XML content bekijken – alleen voor Windows

Heeft u eenmaal de content van een QuarkXPress Passport-document uitgepakt en deze opgeslagen in een XML-bestand, dan kunt u een heleboel dingen met die content doen. U kunt deze bijvoorbeeld met Cascading Style Sheets (CSS) in een Web browser bekijken als vormgegeven document. In dit hoofdstuk gaan we eerst een paar voorbeeld XML-bestanden bekijken met CSS-vormgeving, waarna we u laten zien hoe u CSS-vormgeving kunt toevoegen aan het XML-document dat u in het vorige hoofdstuk heeft bewaard.

 **Wat is CSS?** CSS is een manier om typogrammen toe te kennen aan gecodeerde content. Deze methode is in feite ontwikkeld voor HTML, maar kan ook met XML worden gebruikt. Het is een handig systeem, omdat u de vormgeving van een groot aantal documenten kunt wijzigen via het updaten van één CSS-bestand.

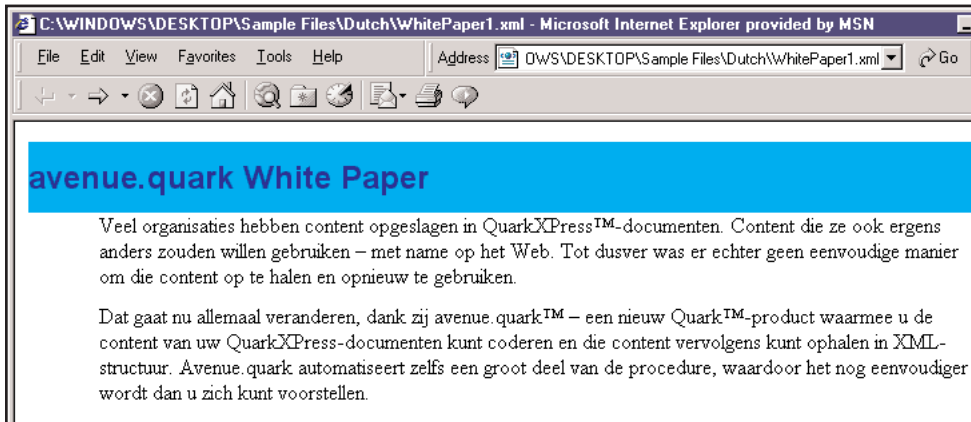
 Op het moment dat deze tekst werd geschreven, werkte dit alleen met Microsoft Internet Explorer 5.0, dat alleen verkrijgbaar is voor Windows. Tegen de tijd dat u dit hoofdstuk leest, zijn er vast wel andere Web browsers (zoals Internet Explorer 5.0 voor Mac OS) die XML met CSS-typogrammen kunnen weergeven. Als u niet beschikt over een browser die XML met CSS-typogrammen kan weergeven, sla dit hoofdstuk dan over en ga naar het hoofdstuk “Een codeerregel toevoegen aan een set met codeerregels”.

### Een XML-bestand met CSS-typogrammen bekijken

We hebben dit Leerboek aangevuld met twee voorbeelden van XML-bestanden, zodat u gemakkelijk kunt zien hoe een XML-bestand eruitziet wanneer ze met CSS worden bekeken. Deze bestanden zijn gegenereerd aan de hand van hetzelfde voorbeelddocument waarmee u in het vorige hoofdstuk heeft gewerkt. Ze zien er identiek uit, met één uitzondering: Elk bestand gebruikt voor de weergave een ander CSS-bestand. Om de bestanden te kunnen bekijken en vergelijken, moet u het volgende doen:

- 1 Start een Web browser die XML met CSS-typogrammen kan weergeven, zoals Microsoft Internet Explorer 5.x.

- 2** Open het voorbeeldbestand “WhitePaper1.xml.” De browser toont het XML-bestand met de CSS-typogrammen uit het bestand “wp.css.” (Uw geëxporteerde XML-bestand moet er precies zo uitzien wanneer u het bekijkt met dit CSS-bestand in de volgende paragraaf.)



De XML-versie van het witboek, weergegeven met de CSS-typogrammen in “wp.css”

- 3** Open het voorbeeldbestand “WhitePaper2.xml” in de Web browser. De browser geeft dezelfde XML content weer, maar deze keer worden de CSS-typogrammen uit het bestand “wp2.css” gebruikt. (Uw geëxporteerde XML-bestand moet er precies zo uitzien als u het bekijkt met dit CSS-bestand in de volgende paragraaf.)



Hetzelfde XML-bestand weergegeven met de CSS-typogrammen in “wp2.css”

## Uw XML-bestand bekijken met CSS-typogrammen

We gaan het XML-bestand bekijken dat u in het eerste hoofdstuk heeft gemaakt:

- 1 Open in een text editor of ander tekstverwerkingsprogramma dan Microsoft Word het bestand “MijnDoc.xml” (het document dat u heeft bewaard in stap 10 van het eerste hoofdstuk).

**!!!** **Waarom niet in Word?** Sommige versies van Microsoft Word vervangen automatisch twee divisies door een halve kastlijn. Gebruik daarom een “platte tekst”-tekstverwerker als SimpelTekst (Mac OS) of WordPad (Windows). We gaan het bestand openen in WordPad: Start WordPad; kies **Bestand** → **Openen**; kies in het uitrolmenu **Bestandstypen** de optie **Alle documenten (\*.\*)**; selecteer “MijnDoc.xml” en klik vervolgens op **Openen**.

- 2 Voeg na de eerste regel in het XML-bestand onderstaande regel toe. Deze regel geeft aan dat het XML-document moet worden bekeken met het typogrammenbestand “wp.css”. Zorg ervoor dat u deze regel exact zo overneemt:

```
<?xml-stylesheet type="text/css" href="wp.css"?>
```

- 3 Voeg na de regel die u zojuist heeft ingetikt de volgende regel toe. Deze geeft het begin van een commentaar aan; we gebruiken hem als “comment-out” van de DTD die in dit bestand is opgenomen. Dit is nodig omdat sommige Web browsers moeite hebben interne DTD’s te lezen.

```
<!--
```

- 4 Voeg onderstaande regel toe onmiddellijk achter de regel met als inhoud “]”. Deze regel geeft het eind aan van de “commented-out” DTD.

```
-->
```

- 5 Bewaar het bestand in ASCII-structuur (tekst zonder vormgeving). Zorg ervoor dat u de documentnaam ongewijzigd laat.

- 6** Zorg ervoor dat u in dezelfde map als “MijnDoc.xml” een kopie van het bestand “wp.css” heeft staan (meegeleverd met het Leerboek bij avenue.quark).
- 7** Open “MijnDoc.xml” in uw Web browser. De CSS-typogrammen die zijn gedefinieerd in “wp.css” worden aan het XML-bestand toegekend.
- 8** Om te kijken hoe het XML-bestand eruit ziet wanneer u dit bekijkt met de CSS-typogrammen uit “wp2.css,” moet u dit opnieuw openen in uw text editor of tekstverwerker en “wp.css” wijzigen in “wp2.css” en wel als volgt:
- ```
<?xml-stylesheet type="text/css" href="wp2.css"?>
```
- 9** Bewaar het bestand in ASCII-structuur (tekst zonder vormgeving). Zorg ervoor dat u de documentnaam ongewijzigd laat.
- 10** Zorg ervoor dat u in dezelfde map als “MijnDoc.xml” een kopie van het bestand “wp2.css” heeft staan (meegeleverd met het Leerboek bij avenue.quark).
- 11** Klik op de **Refresh**-knop in de Web browser om het XML-bestand opnieuw te laden. De CSS-typogrammen gedefinieerd in “wp2.css” worden toegekend aan het XML-bestand. Bekijk de verschillen; u ziet hoe het XML-bestand er heel anders kan uitzien als het op een andere manier wordt getoond.
- 12** Sluit uw Web browser en uw text editor of tekstverwerker.

**!!!** Als u de WhitePaper DTD bekijkt in een text editor en u weet ongeveer hoe een DTD in elkaar zit, ziet u waarschijnlijk dat het `<whitePaper>`-element een attribuut heeft met de naam “xmlns:HTML” met als vaste waarde “http://www.w3.org/TR/REC-html40”. Dit attribuut wordt automatisch opgenomen in alle XML-bestanden die op deze DTD zijn gebaseerd. Het doel is om Microsoft Internet Explorer versie 5.0 of hoger erop te wijzen dat het document dezelfde HTML “namespace” gebruikte. Aangezien het document de HTML namespace gebruikt, wordt de code `<HTML:img>` geïnterpreteerd als een HTML `<IMG>`-code. Resultaat: U kunt afbeeldingen weergeven waarnaar in het XML-bestand wordt verwezen.



## Een codeerregel toevoegen aan een set met codeerregels

In het eerste hoofdstuk zijn we uitgegaan van een bestaande set met codeerregels om het grootste deel van een voorbeelddocument automatisch te coderen. De juridische tekst aan het eind van het document was echter *niet* gecodeerd. Een van de redenen is dat in de meegeleverde codeerregelset geen regel zat voor tekst waaraan het alineatypogram “Juridische tekst” was toegekend. In dit hoofdstuk laten wij u zien hoe dit probleem kan worden opgelost door een nieuwe codeerregel toe te voegen.

!!! Wanneer in dit Leerboek sprake is van “alineatypogrammen” wordt bedoeld QuarkXPress Passport-typogrammen en niet CSS-typogrammen.

We gaan een nieuwe codeerregel toevoegen aan de codeerregelset “White Paper Regels”:

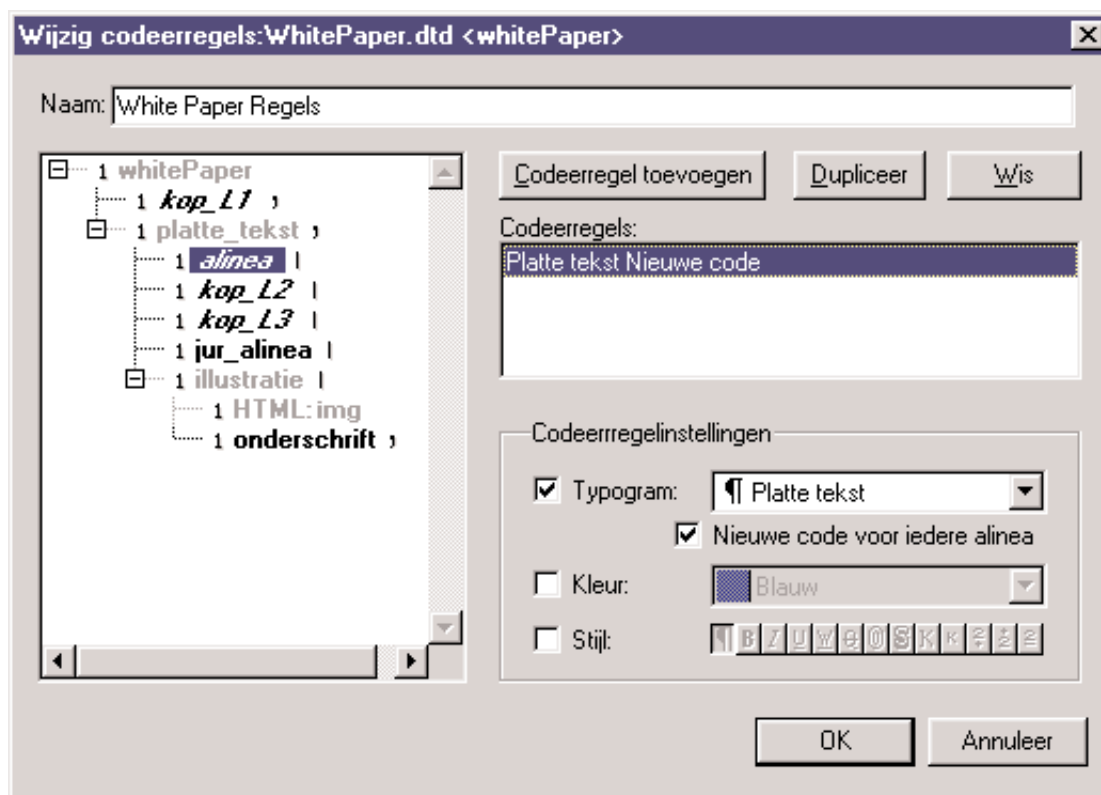
- 1 Start QuarkXPress Passport. Als QuarkXPress Passport al actief is, moet u openstaande XML **Werkblad**-paletten sluiten.
- 2 Open “SampleDoc.qxt” als u dat bestand nog niet heeft geopend.
- 3 Kies **Archief** (Mac OS) of **Bestand** (Windows) → **Nieuw** → **XML** om een nieuw XML-document te maken. Het dialoogvenster **Nieuw XML-document** verschijnt. Selecteer in de **Sjabloon**-lijst de optie **WhitePaper - Nederlands**, klik vervolgens op **OK** om het XML **Werkblad**-palet te openen.

- 4 Kies **Wijzig** (Mac OS) of **Bewerk** (Windows) → **Codeerregels**. Het dialoogvenster **Codeerregels** verschijnt.







Het dialoogvenster **Codeerregels**

- 5 Selecteer de codeerregelset met de naam “White Paper Regels”. (Dit is de codeerregelset die we hebben gebruikt in het eerste hoofdstuk.) Klik vervolgens op **Wijzig** om deze codeerregelset via het dialoogvenster **Wijzig codeerregels** te wijzigen.



#### Het dialoogvenster **Wijzig codeerregels**

- 6 De lijst aan de linkerkant van het dialoogvenster toont de WhitePaper DTD als een hiërarchische boomstructuur. U kunt de inhoud van container elementen en attributen tonen en verbergen door te klikken op resp. het - en het -symbool (Mac OS) of het - en -symbool (Windows). Om de regels in deze codeerregelset te bekijken, klikt u op de elementnamen “kop\_L1”, “alinea”, “kop\_L2” en “kop\_L3” in deze lijst. De codeerregels die aan elk elementtype zijn toegekend, ziet u in de **Codeerregels**-lijst helemaal rechts boven in het dialoogvenster.



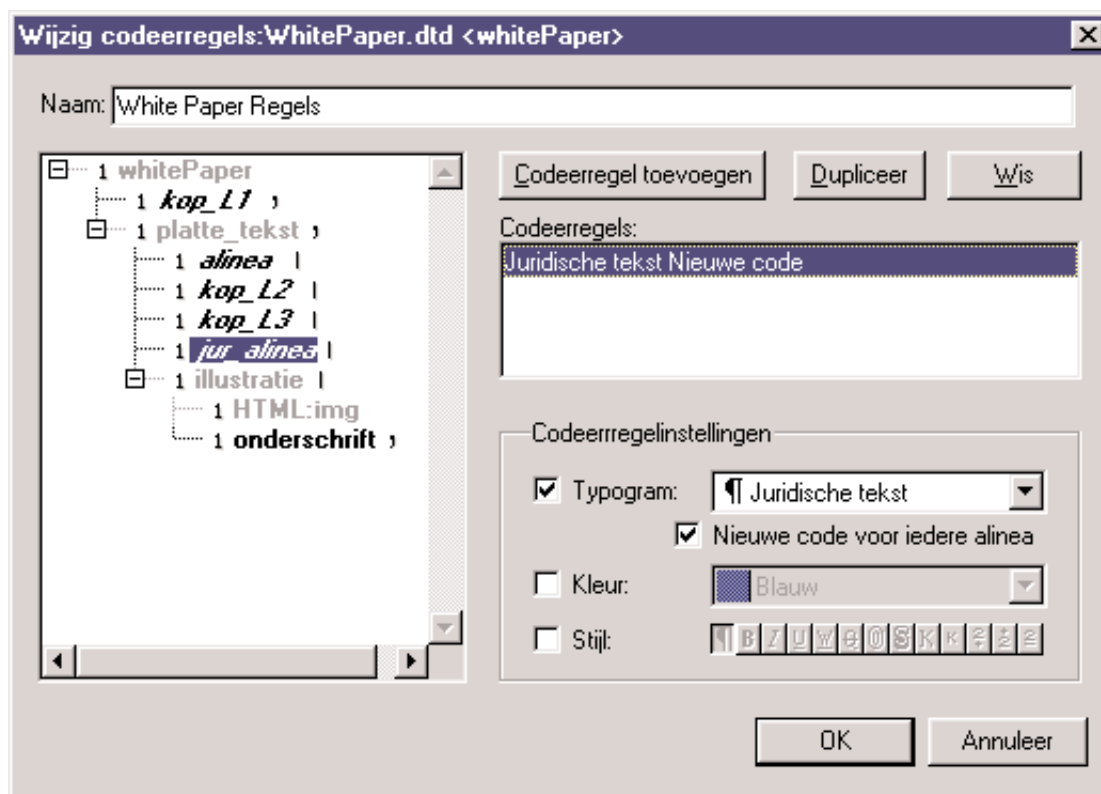
**Wat doen codeerregels eigenlijk?** Via codeerregels laat u avenue.quark weten dat u de content die op een specifieke wijze is vormgegeven wilt coderen met een specifiek elementtype. Bovenstaande codeerregel geeft bijvoorbeeld aan dat alinea's met het "Platte tekst"-typogram moeten worden gecodeerd als `<alinea>`-elementen.

**7** Klik op het element met de naam `<jur_alinea>` in de lijst aan de linkerkant en klik vervolgens op **Codeerregel toevoegen** om voor dit elementtype een nieuwe codeerregel toe te voegen. Dit geeft aan dat u een nieuwe codeerregel wilt maken voor het elementtype `<jur_alinea>`.



**Waarom "jur\_alinea"?** U kunt XML-codes veelzeggende namen geven (binnen bepaalde grenzen). In deze DTD betekent een `<jur_alinea>`-element dat het hier gaat om een tekstalinea, terwijl een `<alinea>`-element alleen maar een alinea met juridische tekst kan bevatten.

- 8** Kruis het vakje **Typogram** aan en kies in het venster-/uitrolmenu **Typogram** de optie **Juridische tekst**. Kruis het vakje **Nieuwe code voor iedere alinea** aan, dit om aan te geven dat u wilt dat het `<jur_alinea>`-elementtype moet worden toegekend aan alle alinea's die gebruik maken van het typogram "Juridische tekst" en dat u wilt dat iedere alinea met het typogram "Juridische tekst" moet worden gecodeerd als afzonderlijk element. N.B. De **Codeerregels**-lijst (rechts bovenaan) wordt bijgewerkt en u ziet daar uw selecties verschijnen.



Het dialoogvenster **Wijzig codeerregels** met een nieuwe codeerregel voor het `<jur_alinea>`-elementtype

- 9** Klik op **OK** om de gemaakte wijzigingen in de codeerregelset "White Paper Regels" te bewaren.
- 10** Klik op **Bewaar** om het dialoogvenster **Codeerregels** te sluiten.

**!!!** Zorg ervoor dat u het actieve **XML Werkblad**-palet en het actieve QuarkXPress Passport-document open laat staan voordat u verder gaat – u heeft deze in de volgende hoofdstukken nog nodig.

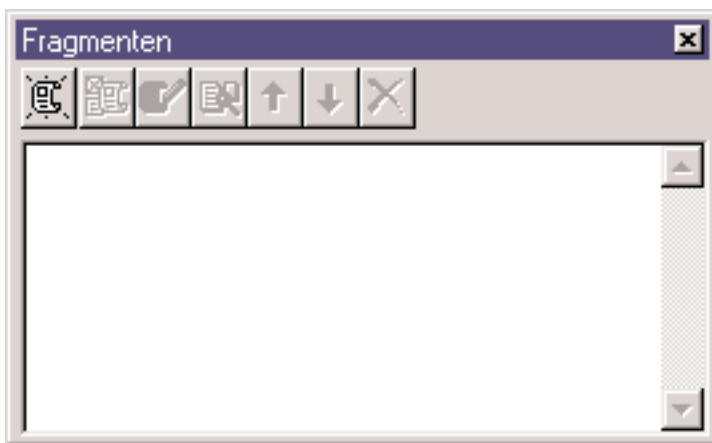
## Tekstkaders combineren tot een fragment

U heeft al gezien dat avenue.quark u kan helpen de inhoud van één tekstkader vlug en eenvoudig te coderen en uit te pakken. We maken het coderingsproces voor u zelfs nog eenvoudiger door de tekstkaderketens waaruit het witboek is samengesteld te combineren tot een *fragment*, waarna we dit fragment vervolgens gaan coderen als één eenheid.

!!! Fragmenten zijn onderdeel van de functionaliteit van de Item Sequence QuarkXTensions-software, die met avenue.quark wordt meegeleverd. Om met fragmenten te kunnen werken, moet de Item Sequence QuarkXTensions-software zijn geïnstalleerd in de "XTension"-map in uw QuarkXPress Passport-programmamap.

We gaan een tekstkaderketen combineren tot een fragment:

- 1 Kies **Beeld** → **Toon fragmenten** om het **Fragmenten**-palet te openen.



Via het **Fragmenten**-palet kunt u van tekstkaders fragmenten maken die u gewoon een naam kunt geven.

- 2 Klik op de knop **Nieuw fragment** om een nieuw fragment te maken.

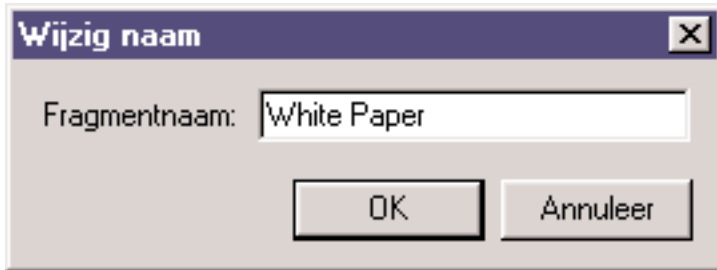


Knop **Nieuw fragment**

- 3** Klik op de knop **Wijzig naam** om het dialoogvenster **Wijzig naam** te openen. Tik in het **Naam**-veld als naam **White Paper** in en klik vervolgens op OK.



#### Knop **Wijzig naam**



Via het dialoogvenster **Wijzig naam** kunt u het geselecteerde fragment een andere naam geven.

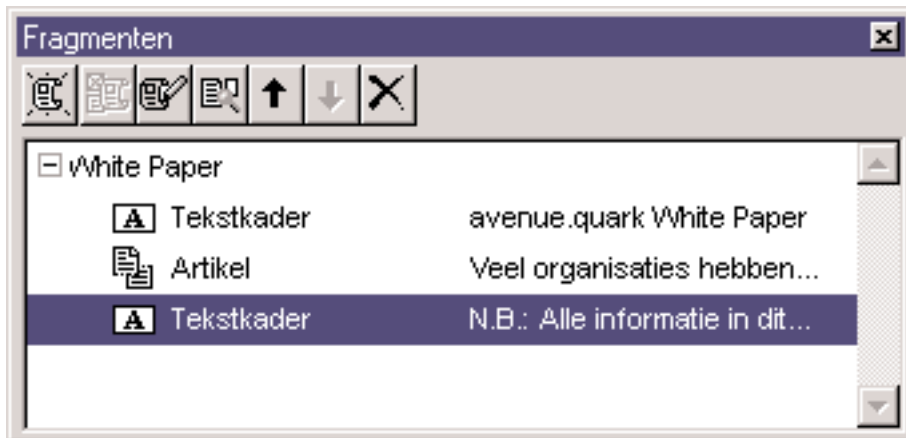
- 4** Klik op het tekstkader met de kop (“avenue.quark White Paper”) om dit te selecteren.
- 5** Klik op de knop **Voeg item toe** om het tekstkader aan het “White Paper”-fragment toe te voegen.



#### Knop **Voeg item toe**

- 6** Selecteer het tekstkader dat begint met “Veel organisaties hebben content...” Klik vervolgens op de knop **Voeg item toe** om dit, meteen na het kader met de kop, aan het fragment toe te voegen.

- 7** Ga naar pagina 2 en selecteer het tekstkader rechts onder op de pagina. In dit tekstkader staat de juridische tekst voor het document. Klik op de knop **Voeg item toe** om dit kader aan het fragment toe te voegen, precies achter het kader met het grootste gedeelte van de content van het witboek.



Het **Fragmenten**-palet toont een fragment als een lijst met benoemde kaders.

De tekstkaders waaruit het witboek is samengesteld, zijn nu, in de goede volgorde, met elkaar gecombineerd tot één fragment met de naam “White Paper”. Hierdoor kunt u ermee werken als één geheel. In het volgende hoofdstuk gaan we het fragment exporteren in XML-structuur.


- !!!** Zorg ervoor dat, voordat u verder gaat, het **Fragmenten**-palet, het actieve **XML Werkblad**-palet en het actieve QuarkXPress Passport-document open blijven staan – u heeft ze straks nog nodig in het volgende hoofdstuk.



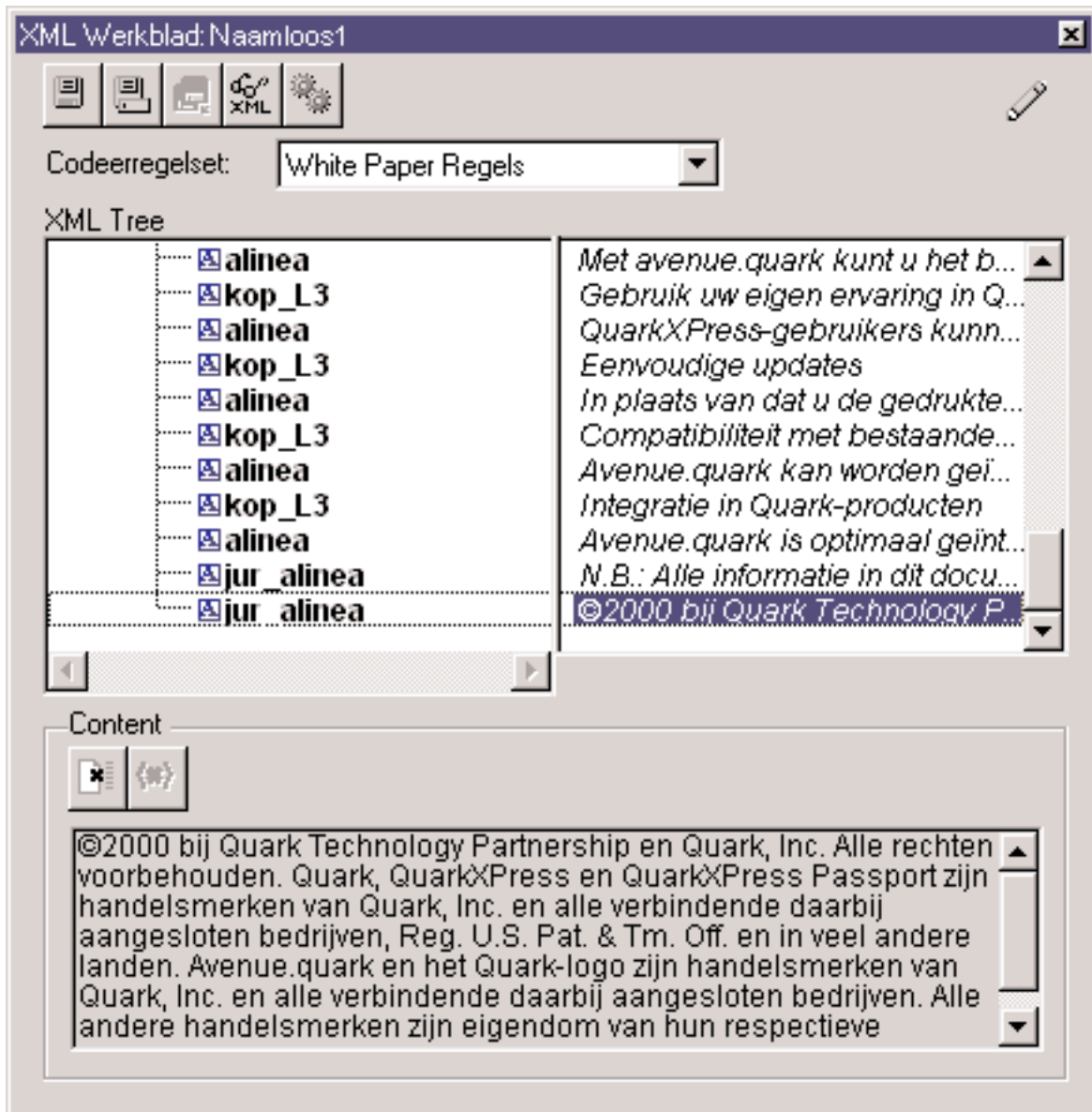
## Coderen met fragmenten en een nieuwe codeerregel

Nu u een nieuwe codeerregel heeft toegevoegd voor de juridische tekst aan het eind van het witboek en alle tekstkaders waaruit het witboek is samengesteld, heeft gecombineerd tot één fragment, zult u zien hoe eenvoudig het is alle belangrijke tekst in het document in één keer te coderen.

We gaan het “Witboek”-fragment coderen:

- 1 Selecteer het fragment met de naam “White Paper”.
  - 2 Druk op ⌘ (Mac OS) of Ctrl (Windows) en sleep het fragment op het `<whitePaper>`-element in het **XML Werkblad**-palet. Avenue.quark codeert automatisch de content in alledrie tekstkaders en maakt daarbij meteen de noodzakelijke elementen aan.
-  **Hé, het werkte niet bij mij!** Als er iets onverwachts gebeurde toen u het fragment op het `<whitePaper>`-element in de **XML Tree**-lijst sleepte, hoeft u zich vooralsnog geen zorgen te maken. Sluit voor nu het **XML Werkblad**-palet en ga terug naar Stap 1.

- 3** Blader door de XML Tree-lijst totdat u helemaal onderaan staat. U ziet dat deze keer avenue.quark alle alinea's waaraan het alineatypogram "Juridische tekst" is toegekend, correct heeft gecodeerd.



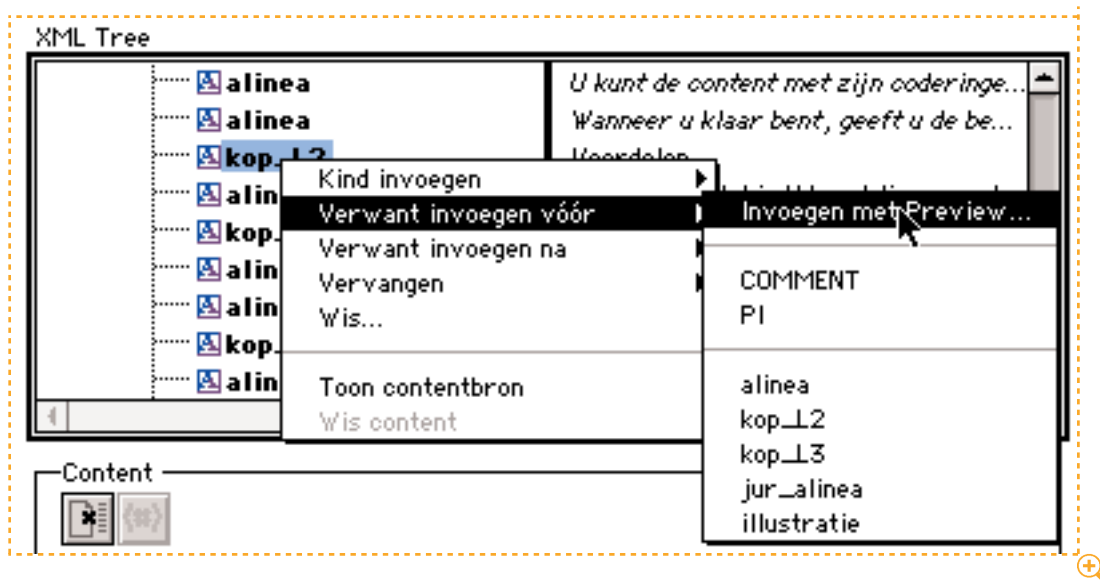
Compleet gecodeerd document met de juridische tekst gecodeerd als <jur\_alinea>

- 4** Laat zowel het QuarkXPress Passport-witboekdocument als het nieuwe XML-document openstaan; we gaan er straks in het volgende hoofdstuk mee verder.

## Een illustratie en een onderschrift van een code voorzien

De enige belangrijke onderdelen van het witboek die we nog niet hebben gecodeerd, zijn de illustratie en het onderschrift links onder op pagina 1. In dit hoofdstuk gaat u een nieuw `<illustratie>`-element in het XML-document zetten, waarna u de illustratie met het onderschrift gaat toevoegen aan de kinderen van dat element.

- 1 Blader in het XML Werkblad-palet door de XML Tree-lijst totdat het `<kop_L2>`-element zichtbaar wordt dat het woord "Voordelen" bevat.
- 2 Control+klik (Mac OS) of rechts+klik (Windows) op dit element om een venster-/uitrolmenu te openen. Kies daarin **Verwant invoegen vóór** om een submenu te openen en selecteer vervolgens de optie **Invoegen met Preview...**. Het dialoogvenster **Verwant invoegen vóór** verschijnt.



Via het venster-/uitrolmenu **XML Tree** kunt u een item invoegen vóór het item dat u heeft geselecteerd in de **XML Tree**-lijst.

- 3** Selecteer in de **Items**-lijst de optie **illustratie** om aan te geven dat u een `<illustratie>`-element wilt invoegen. De **Preview XML Tree**-lijst toont nu een preview van de huidige boomstructuur in het XML-document nadat het `<illustratie>`-element is ingevoegd. Blader in de **Preview XML Tree**-lijst naar beneden totdat u het vetgedrukte `<illustratie>`-element ziet staan.

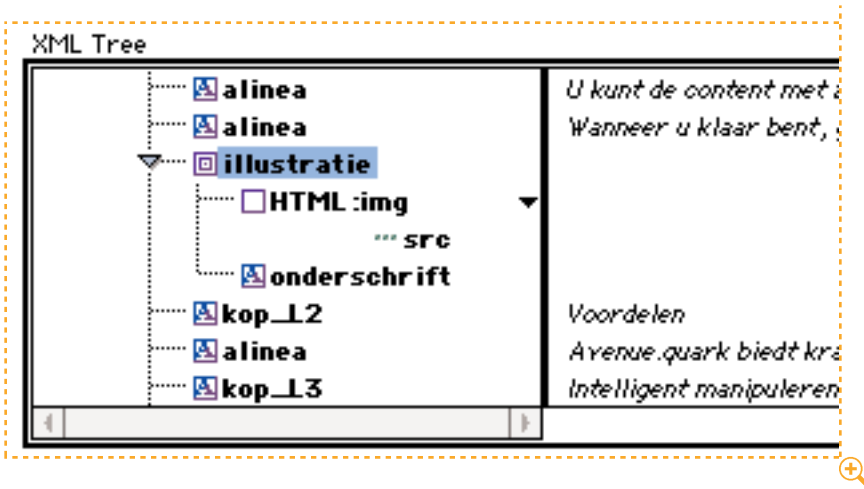


Via het dialoogvenster **Verwant invoegen vóór** kunt u de verplichte kinderen bekijken van het item dat u heeft ingevoegd.



In de **Preview XML Tree**-lijst kunt u de verplichte kinderen zien van het element dat u wilt toevoegen. De verplichte kinderen van het `<illustratie>`-element zijn een `<HTML:img>`-element en een `<onderschrift>`-element.

- 4** Klik op OK. Het `<illustratie>`-element wordt toegevoegd vóór het geselecteerde `<kop_L2>`-element.

- 5 Klik in de XML Tree-lijst op het ►-symbool naast het `<HTML:img>`-element om het `src`-attribuut weer te geven dat bij dat element hoort.



Het zojuist ingevoegde `<illustratie>`-element heeft twee kinderen: een `<HTML:img>`-element en een `<onderschrift>`-element. Het `<HTML:img>`-element heeft een attribuut met de naam `src`.

- 6 Selecteer het tekstkader onder de illustratie links onder op pagina 1. Selecteer vervolgens het **verplaatsgereedschap**  en sleep het kader op het element met de naam `<onderschrift>` in de XML Tree-lijst. De tekst van het onderschrift wordt naar het `<onderschrift>`-element gekopieerd.
- 7 Selecteer het illustratiekader boven het kader met het onderschrift. Gebruik vervolgens het **verplaatsgereedschap**  om het illustratiekader op het attribuut te slepen met de naam "src" in de XML Tree-lijst. De naam van het illustratiebestand wordt gekopieerd naar het `src`-attribuut van het `<HTML:img>`-element.



**Wat is er nu precies gebeurd?** U heeft avenue.quark net verteld om de naam van het illustratiebestand te zetten in het `src`-attribuut voor het `<HTML:img>`-element. Alleen de bestandsnaam is ingevoegd, niet de illustratie zelf. Dit betekent dat als u het illustratiebestand wilt bekijken dit in dezelfde map moet staan als het XML-bestand, anders wordt de illustratie niet in de browser weergegeven.

**8** Klik op de **Bewaar**-knop in het **XML Werkblad**-palet en bewaar het document in de map “avenue.quark Tutorial” op uw harde schijf als “MijnDoc2.xml.” (Ook hier moet u de instelling in het venster-/uitrolmenu **Tekenset** op **UTF8** laten staan en het vakje **Bewaar XML als standalone bestand** aankruisen.)



**Bewaar**-knop

**9** Gebruik de procedure die is beschreven onder “Een XML-bestand met CSS-typogrammen bekijken” (in het hoofdstuk “Uitgepakte XML content bekijken – alleen voor Windows”) om het bestand “MijnDoc2.xml” te bekijken. Zorg ervoor dat het bestand met de naam “diagram1.gif” in dezelfde map staat als “MijnDoc2.xml” als u wilt dat de illustratie op uw beeldscherm zichtbaar is.



U heeft nu een paar dingen gezien die avenue.quark kan doen. Raadpleeg voor meer informatie “avenue.quark in vogelvlucht” (“avenue.quark in vogelvlucht.pdf”), die bij dit Leerboek bij avenue.quark wordt meegeleverd.

©2000 Quark Technology Partnership en Quark, Inc. wat betreft de content en vormgeving van dit materiaal. Alle rechten voorbehouden.

©2000 Quark Technology Partnership en Quark, Inc. wat betreft de technologie. Alle rechten voorbehouden. Amerikaanse en buitenlandse patenten aangevraagd.

Quark, QuarkXPress, QuarkXPress Passport en QuarkXTensions zijn handelsmerken van Quark, Inc. en alle verbindende daarbij aangesloten bedrijven, Reg. U.S. Pat. & Tm. Off., en in veel andere landen. Avenue.quark en het Quark-logo zijn handelsmerken van Quark, Inc. en alle verbindende daarbij aangesloten bedrijven.

Microsoft is een geregistreerd handelsmerk van Microsoft Corporation.

Alle andere handelsmerken zijn eigendom van hun respectieve eigenaren.