



Gebruikershandleiding cXML

VERSIE 1.1

APRIL 2000

Ariba Inc. (Ariba) verleent u hierbij een permanente, niet-exclusieve, mondiale licentie, vrij van royalty's, voor het gebruik van de cXML-specificatie (nader te noemen de "Specificatie") onder auteursrechten van Ariba in de Specificatie, om de Specificatie te gebruiken, te kopiëren, te publiceren, te wijzigen en te distribueren. Ariba verleent u verder een licentie, vrij van royalty's, onder de toepasselijke intellectuele eigendomsrechten van Ariba, om de cXML-codes en schemarichtlijnen die in de Specificatie zijn opgenomen te implementeren en te gebruiken, met het doel computerprogramma's te maken die gebruikmaken van deze richtlijnen. De voorwaarde bij deze licentie is dat u ermee instemt geen intellectuele eigendomsrechten te laten gelden tegenover Ariba of andere bedrijven voor hun implementatie van de Specificatie. Ariba behoudt uitdrukkelijk alle andere rechten voor die betrekking hebben op het materiaal en het onderwerp van deze Specificatie. Ariba geeft uitdrukkelijk geen enkele garantie met betrekking tot deze Specificatie, daaronder mede begrepen de garantie dat deze Specificatie of implementaties hiervan de rechten van derden niet schendt. Deze Specificatie wordt niet-gereconditioneerd geleverd zonder enige expliciete of impliciete garantie. Wanneer u de Specificatie publiceert, kopieert of distribueert, dient deze melding van auteursrechten te worden aangehecht. Wanneer u deze Specificatie echter wijzigt, mag de naam van de gewijzigde Specificatie niet de term "cXML" bevatten. Wanneer u op- of aanmerkingen of suggesties naar Ariba verzendt en Ariba wijzigt cXML op grond van uw aanwijzingen, is Ariba eigenaar van de gewijzigde cXML-versie.

De informatie in dit document kan zonder voorafgaande kennisgeving worden gewijzigd.

Inhoudsopgave

Voorwoord	v
Doelgroep en vereisten	v
Welke hoofdstukken zijn voor u van belang?	v
 Hoofdstuk 1	
cXML, inleiding	1
De mogelijkheden van cXML	1
Catalogi	2
Punchout	3
Inkooporders	4
Toepassingstypen die gebruikmaken van cXML	5
Inkooptoepassingen	5
Netwerkplatforms voor e-commerce	5
Punchout-catalogi	6
Systemen voor orderontvangst	6
Strategie voor levering van de inhoud	7
Validatie met DTD's	8
DTD's voor cXML verkrijgen	8
Validatie uitvoeren	8
Profieltransactie	9
Hulpprogramma's voor XML	10

Hoofdstuk 2

Een punchout-site implementeren	11
Vereisten voor punchout	11
Inkooporganisaties	11
Leveranciers	13
Volgorde van gebeurtenissen bij punchout	16
Stap 1 & 2: Punchoutaanvraag	16
Stap 3: Selectie van producten	17
Stap 4: Check Out	18
Stap 5: Verzending van de inkooporder	19
Punchout-documenten	21
Punchout, indexcatalogus	21
PunchOutSetupRequest	23
PunchOutSetupResponse	28
PunchOutOrderMessage	29
Wijzigingen in uw webpagina's	31
Launch Page	31
Start Page	35
Sender Page	35
Order Receiver Page	39
Suggesties bij punchout-websites	39
Richtlijnen voor de implementatie	39
Cookies van inkoper en leverancier	40
Aanpassing	41

Hoofdstuk 3

cXML-inkooporders ontvangen	43
Inkooporderproces	43
Inkooporders ontvangen	44
OrderRequest	44
OrderResponse	47
Bijlagen bij orders aanvaarden	47

Appendix A	
Specificatie van cXML	49
Protocolspecificatie	50
Request/response-model	50
XML-conventies	51
cXML-envelop	52
Inpaklagen	54
Header	56
Request	58
Response	58
One-way-model (asynchroon)	61
Basiselementen	66
Type-eenheden	66
Basiselementen	67
Profieltransactie	68
ProfileRequest	68
ProfileResponse	69
Orderdefinities	70
OrderRequest	70
Response op een OrderRequest	79
Punchout-transactie	79
PunchOutSetupRequest	79
PunchOutSetupResponse	82
PunchOutOrderMessage	83
Latere statuswijzigingen	87
DocumentReference	88
StatusUpdateRequest	88
Catalogusdefinities	89
Supplier	90
Index	92
Contract	94
Definities inschrijvingenbeheer	96
Leveranciersgegevens	96
Catalogusinschrijvingen	99
Definities voor ophalen van berichten	102
GetPendingRequest	102
GetPendingResponse	103

Appendix B

Nieuwe functies in cXML 1.1 105

Algemene wijzigingen in cXML	105
Verbeterde ondersteuning voor het gebruik van meerdere talen	105
Gecentraliseerde DTD's	106
Nieuwe profieltransactie	106
Nieuwe statuscodes	107
Nieuw type attribuut voor leden van de marketplace	108
Wijzigingen in Extrinsic-elementen	108
Nieuw element Contact	108
Attribuut RequisitionID ondersteund	109
Overzicht van Extrinsic-informatie die is verplaatst	110
Extrinsics in de kop	110
Verbeteringen in punchout-transacties	111
Verbeterd PunchOutSetupRequest	111
Element SelectedItem	111
Leeg document PunchOutOrderMessage	112
Nieuw verborgen veld cXML-base64	112
Nieuwe inkooporderfuncties	113
Nieuw attribuut lineNumber	113
Bijlagen bij inkooporders	113
Nieuw attribuut shipComplete	114
Nieuw element ShortName	114
Nieuwe statustransactie van inkooporders	115
Nieuw element OrderReference	115
Nieuwe transactie StatusUpdateRequest	116
Nieuw element Followup	116

Index 117

Voorwoord

In dit document wordt beschreven hoe cXML (commerce eXtensible Markup Language) wordt gebruikt voor de gegevensoverdracht met binnen de e-commerce.

Doelgroep en vereisten

Dit document is bedoeld voor programmeurs die toepassingen met cXML-ontwerpen en leveranciers die hun e-commercewebsites willen inrichten voor punchout.

cXML is een open, veelzijdige taal voor de transactievereisten van:

- Elektronische productcatalogi
- cXML-punchout-catalogi
- Inkooptoepassingen
- Inkoopgroepen

Lezers dienen praktische kennis te hebben van e-commerceconcepten en de HTTP-standaard voor webcommunicatie.

In dit document wordt niet beschreven hoe u bepaalde inkooptoepassingen of hubs van e-commercenetwerken gebruikt.

Welke hoofdstukken zijn voor u van belang?

- **e-commerce Business Managers** - Lees Hoofdstuk 1, “cXML, inleiding.” voor een overzicht van cXML-mogelijkheden.
- **Webprogrammeurs** - Webprogrammeurs die e-commercesites implementeren, dienen alle hoofdstukken te lezen.
- **Beheerders van punchoutsites** - Webengineers die ervaring hebben met punchoutwebsites dienen Appendix B, “Nieuwe functies in cXML 1.1.” te lezen.

Hoofdstuk 1

cXML, inleiding

In dit hoofdstuk vindt u een bespreking van cXML (commerce eXtensible Markup Language) voor e-commerce transacties.

Dit hoofdstuk biedt een overzicht van cXML. De volgende onderwerpen komen aan de orde:

- De mogelijkheden van cXML
- Toepassingstypen die gebruikmaken van cXML
- Strategie voor levering van de inhoud
- Validatie met DTD's
- Profieltransactie
- Hulpprogramma's voor XML

De mogelijkheden van cXML

cXML stelt inkopers, leveranciers, aggregators en tussenpersonen in staat te communiceren met één open standaardtaal.

Geslaagde e-commerceportalen voor business-to-business (B2B e-commerce) zijn afhankelijk van een flexibel, algemeen geaccepteerd protocol. cXML is de sleutel om de toegankelijkheid tot uw producten en diensten zo uitgebreid mogelijk te maken, omdat dit een goed gedefinieerde, krachtige taal is die speciaal is ontworpen voor B2B e-commerce. cXML is de keuze van grote leveranciers en inkooporganisaties.

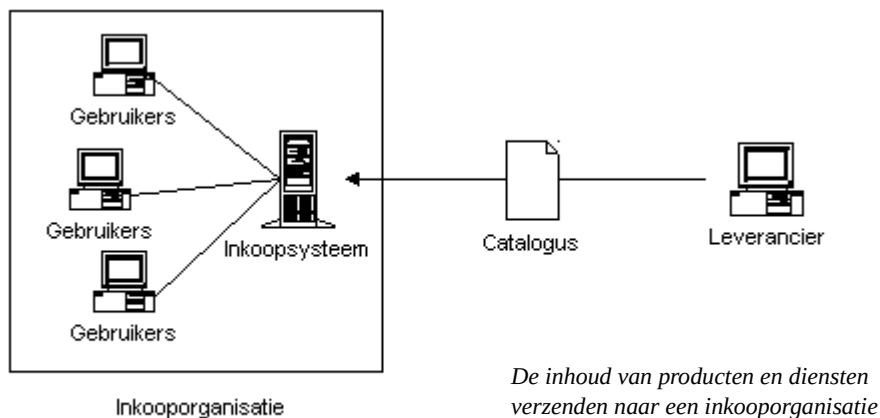
cXML-transacties bestaan uit *documenten*. Dit zijn eenvoudige tekstbestanden met een goed gedefinieerde indeling en inhoud. De meeste typen cXML-documenten zijn te vergelijken met papieren documenten die traditioneel in de zakenwereld werden gebruikt.

In de volgende subsecties vindt u een beschrijving van de belangrijkste typen cXML-documenten.

Catalogi

Catalogi zijn bestanden waarin producten en diensten worden aangeboden aan inkooporganisaties. In catalogi beschrijft u de producten en diensten die u aanbiedt en de prijzen die u berekent. Dit is het belangrijkste communicatiekanaal van u naar uw klanten.

Organisaties die gebruikmaken van de inkooptoepassingen kunnen in catalogi zien welke producten en diensten u aanbiedt en vervolgens bij u inkopen. De catalogi worden in deze toepassingen gelezen en intern in de database opgeslagen. Nadat een inkooporganisatie uw catalogi heeft goedgekeurd, is de inhoud zichtbaar voor gebruikers. Deze gebruikers kunnen vervolgens items uitkiezen en opnemen in inkoopaanvragen.



U kunt catalogi maken voor elk product of elke dienst, ongeacht de maat, de prijs of de leveringswijze.

Elk item in een catalogus dient bepaalde basisgegevens te bevatten. Extra informatie kan facultatief worden toegevoegd, waardoor uitgebreide catalogusfuncties kunnen worden gebruikt, zoals beschrijvingen in verschillende talen.

Punchout

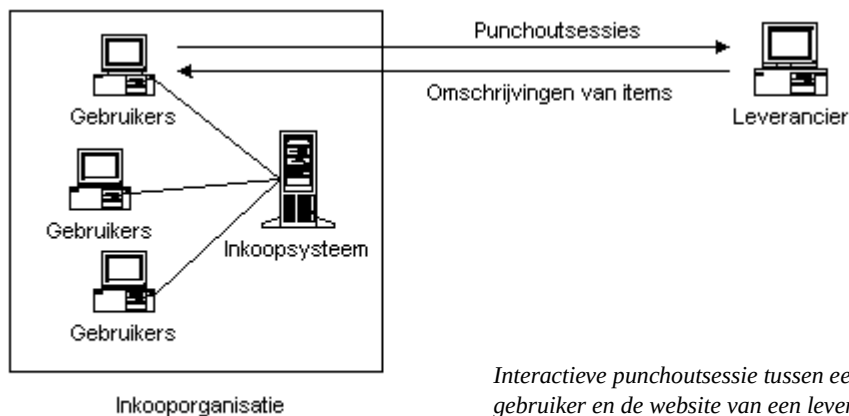
Punchout vormt een alternatief voor statische catalogusbestanden. Punchout-sites zijn live, interactieve catalogi op uw website.

Wanneer u een website voor e-commerce hebt, kunt u deze zodanig wijzigen dat punchout wordt ondersteund. De communicatie tussen punchout-sites en inkoopsystemen vindt plaats met cXML via het Internet.

Zie

Hoofdstuk 2,
"Een punchout-site
implementeren."
voor meer informatie.

In inkooptoepassingen wordt voor punchoutsites een knop weergegeven in plaats van product- of prijsdetails. Wanneer een gebruiker op de knop klikt, geeft de webbrowser pagina's weer van uw lokale website. Afhankelijk van de wijze waarop u de pagina's hebt geïmplementeerd, kan de gebruiker productopties bekijken, configuraties opgeven en leveringsmethoden selecteren. Als de gebruiker klaar is met de selectie van items, klikt deze op een knop waarmee de ordergegevens worden geretourneerd naar de inkooptoepassing. De volledig geconfigureerde producten en de bijbehorende prijzen worden vervolgens weergegeven op de inkoopaanvragen van de gebruiker.



Interactieve punchoutsessie tussen een gebruiker en de website van een leverancier

Op uw website kunt u vooraf overeengekomen contractproducten en -prijzen aanbieden.

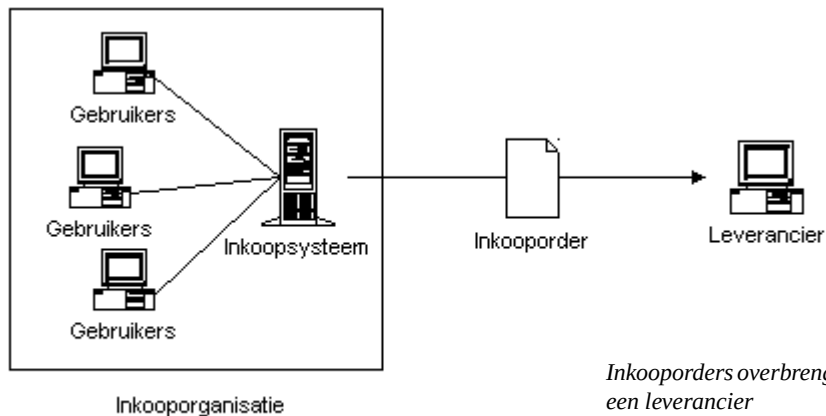
Inkooporders

Inkooporganisaties verzenden inkooporders naar leveranciers om uitvoering van een contract te vragen.

Deze inkooporders kunnen naar leveranciers worden gerouteerd via een platform van een e-commercenetwerk, zoals Ariba Network.

Zie

Hoofdstuk 3,
"cXML-inkooporders
ontvangen." voor
meer informatie.



cXML is slechts een van de mogelijke indelingen voor inkooporders. Andere indelingen zijn e-mail, fax en EDI (X.12 Electronic Data Interchange). cXML is echter de ideale indeling voor inkooporders. Deze indeling is flexibel, goedkoop te implementeren, ondersteunt het grootste bereik aan gegevens en bijlagen en kan worden gebruikt in combinatie met geautomatiseerde systemen.

Toepassingstypen die gebruikmaken van cXML

cXML kan worden gebruikt in elke e-commercetoepassing. Deze indeling wordt op dit moment gebruikt door inkooporganisaties, verticale en horizontale inkoopgroepen, leveranciers en bedrijven die toepassingen verkopen.

In de volgende subsecties vindt u een beschrijving van de belangrijkste toepassingstypen waarbij op dit moment wordt gebruikgemaakt van cXML.

Inkooptoepassingen

In inkooptoepassingen, zoals Ariba Buyer en Ariba IBX (Internet Business eXchange), wordt cXML gebruikt voor externe transacties.

Ariba Buyer is een bedrijfstoepassing die dienst doet bij grote organisaties en die werknemers via een intranet gebruiken.

Ariba IBX is een op het Internet gebaseerde service waarmee inkoopgroepen kunnen worden gemaakt die uit een groot aantal kleine en middelgrote bedrijven bestaan.

Via deze toepassingen kunnen groepen gebruikers bij leveranciers contractproducten en -diensten inkopen die zijn goedgekeurd door hun inkoopbeheerders. Aangevraagde inkopen worden eerst goedgekeurd door beheerders binnen de groep en de goedgekeurde inkooporders worden via een aantal mogelijke kanalen, waaronder cXML via het Internet, naar leveranciers verzonden.

Netwerkplatforms voor e-commerce

Netwerkplatforms voor e-commerce, zoals Ariba Network, zijn services via het web om inkopers en leveranciers met elkaar in contact te brengen.

Deze webservices bieden functies als validatie van catalogi en bestandsbeheer, publicatie van catalogi en inschrijvingen op catalogi, geautomatiseerde orderrouting en inkooporderhistorie.

De communicatie tussen deze webservices, de toepassingen van inkopers en van leveranciers kan volledig per cXML via het Internet plaatsvinden.

Punchout-catalogi

Zie

Hoofdstuk 2,
"Een punchout-site
implementeren." voor
meer informatie.

Zoals hiervoor werd beschreven, zijn punchout-catalogi interactieve catalogi op websites van leveranciers. Punchout-catalogi zijn webservertoepassingen, geschreven in een programmeertaal zoals ASP (Active Server Pages), JavaScript of CGI, waarmee de punchout-sessies van de inkoper worden beheerd.

Punchout-catalogi aanvaarden punchout-aanvragen vanuit inkooptoepassingen, stellen de identiteit van de inkooporganisatie vast en geven de toepasselijke producten en prijzen weer in HTML-indeling. Vervolgens selecteren gebruikers bepaalde items, configureren deze en selecteren opties indien dit van toepassing is.

Aan het einde van de punchout-sessie worden beschrijvingen van de selecties die de gebruiker heeft gemaakt vanuit de punchout-site in cXML-indeling naar de inkooptoepassingen verzonden.

Systemen voor orderontvangst

Zie

Hoofdstuk 3,
"cXML-inkooporders
ontvangen." voor
meer informatie.

Systemen voor orderontvangst zijn toepassingen op leverancierslocaties waarmee inkooporders die inkooporganisaties hebben verzonden, worden aanvaard en verwerkt. Een systeem voor orderontvangst kan geautomatiseerd zijn, zoals systemen voor voorraadbeheer, orderuitvoering of orderverwerking.

Het is eenvoudig om informatie uit cXML-inkooporders te halen, daarom is het relatief gemakkelijk om de aanpassingen aan te brengen waarmee het bestaande systeem voor orderverwerking de inkooporders kan aanvaarden.

Strategie voor levering van de inhoud

In inkooptoepassingen wordt de inhoud van producten en diensten aan gebruikers gepresenteerd. Leveranciers willen de manier waarop hun klanten de producten en diensten bekijken onder controle houden, omdat de presentatie van cruciaal belang is voor het verkoopproces. Inkooporganisaties willen dat de inhoud gemakkelijk toegankelijk en te doorzoeken is, om zo de geschiktste contracten af te kunnen sluiten.

Inkooporganisaties en leveranciers kunnen kiezen uit verschillende methoden voor de levering van de inhoud van producten en diensten. De gebruikte methode wordt in overleg tussen leverancier en inkooporganisatie bepaald en is afhankelijk van de aard van de producten of diensten in kwestie.

In de volgende tabel worden categorieën van algemeen ingekochte producten en diensten weergegeven en de bijbehorende gewenste leveringsmethode van de inhoud.

Product	Eigenschappen	Methode voor levering van de inhoud
Kantoorbenodigdheden, Interne benodigdheden	Statische inhoud, stabiele prijzen	Statische catalogi
Laboratoriumbenodigdheden, MRO (Onderhoud, reparatie en exploitatie), Electronische onderdelen	Vereist normalisatie om nuttig te zijn	Punchout naar een verticaal productportaal
Boeken, Chemicaliën	Groot aantal regelitems	Punchout naar een site van een leverancier
Computers, Netwerkapparatuur, Randapparatuur	Veel configuratiemogelijkheden	Punchout naar een configuratiehulpprogramma van een leverancier
Dienstverlening, Gedrukte materialen	Inhoud heeft sterk variabele attributen	Punchout naar een elektronisch formulier op de site van een leverancier

Inkooporganisaties kunnen inhoud lokaal binnen de organisatie opslaan of via punchout op het Internet bekijken. Bij cXML-catalogi worden beide strategieën ondersteund.

Zoals de voorgaande tabel aangeeft, biedt punchout een flexibel kader waarbinnen leveranciers, afhankelijk van hun product of klant, een aangepaste inhoud kunnen bieden. Het doel van deze inhoudsstrategie is dat inkopers en leveranciers catalogusgegevens kunnen uitwisselen op de zinvolste manier.

Validatie met DTD's

cXML is een XML-taal en wordt daarom zorgvuldig gedefinieerd door een set Document Type Definitions (DTD's). Deze DTD's zijn tekstbestanden waarin de precieze syntaxis en volgorde van de cXML-elementen worden beschreven. De cXML die in toepassingen wordt gelezen of geschreven, kan worden gevalideerd met DTD's.

Het is niet noodzakelijk cXML-documenten in cXML-toepassingen te valideren, maar dit verdient wel aanbeveling.

DTD's voor cXML verkrijgen

DTD's voor alle versies van cXML zijn verkrijgbaar op consistente locaties op cxml.org:

`http://xml.cXML.org/schemas/cXML/<version>/cXML.dtd`

waarbij `<version>` het volledige nummer is van de cXML-versie, zoals 1.1.007.

Validatie uitvoeren

Deze DTD's kunnen in toepassingen worden gebruikt voor de validatie van alle inkomende en uitgaande cXML-documenten. Toepassingen voor XML-validatie zijn beschikbaar op het web. Microsoft Internet Explorer 5 heeft een ingebouwde functie voor XML-validatie.

Valideer alle ontvangen cXML-documenten wanneer u transacties zo uitgebreid mogelijk wilt behandelen. Geef de juiste foutcode door als u fouten ontdekt, zodat de zender het document opnieuw kan verzenden.

De beste resultaten worden bereikt wanneer cXML-clients niet steeds DTD's halen bij het parseren van cXML-documenten, maar naar de cXML-versie in de documentkop kijken en DTD's halen die nog niet lokaal zijn opgeslagen.

Profieltransactie

Met de Profieltransactie wordt basisinformatie over cXML-servers doorgegeven. Dit is de enige transactie die alle cXML-servers moeten ondersteunen.

Deze transactie bestaat uit twee documenten: ProfileRequest en ProfileResponse. Met deze twee documenten wordt informatie opgehaald over de mogelijkheden van de server, waaronder de ondersteunde cXML-versie, ondersteunde transacties en opties bij deze transacties.

Opmerking: alle cXML 1.1-servers **moeten** de profieltransactie ondersteunen.

Clients kunnen de profieltransactie ook gebruiken om servers te 'pingen' en zo te controleren of deze beschikbaar zijn.

ProfileRequest

Het document ProfileRequest heeft geen inhoud maar routeert eenvoudigweg naar de opgegeven cXML-server.

ProfileResponse

De server reageert met het document ProfileResponse waarin transacties worden weergegeven die worden ondersteund door de cXML-server, de locatie van deze transacties en eventuele benoemde opties met een stringwaarde.

Hulpprogramma's voor XML

Hulpprogramma's voor bewerking en validatie van XML-bestanden zijn beschikbaar op het web, zowel te koop als gratis. In het volgende overzicht wordt een aantal van deze hulpprogramma's beschreven:

- **Internet Explorer 5** van Microsoft. Een webbrowser met XML waarmee XML-bestanden met DTD's kunnen worden gevalideerd.
www.microsoft.com/windows/ie/default.htm
- **XML Notepad** van Microsoft. Een eenvoudige XML-editor.
msdn.microsoft.com/xml/notepad/intro.asp
- **XML Authority** van Extensibility. Een op Java gebaseerde XML DTD-editor, met hiërarchische en grafische weergaven.
www.extensibility.com
- **XML Spy** van Icon Information Systems. Een hulpprogramma voor het onderhoud van DTD's en XML-bestanden, met een raster-, bron- en browserweergave.
www.icon-is.com
- **XMetaL** van Softquad Software. Een authoringprogramma voor XML dat kan worden aangepast.
www.softquad.com
- **CLIP** van Techno2000 USA. Een gebruiksvriendelijk authoringprogramma voor XML, met begeleiding bij bewerkingen.
www.t2000-usa.com
- **XMLwriter** van Wattle Software. Een grafisch authoringprogramma voor XML, speciaal ontworpen voor het beheer van XML-projecten.
www.xmlwriter.net

Op de volgende websites vindt u meer hulpprogramma's voor XML:

www.xmlsoftware.com
www.xml.com/pub/pt/Editors

Hoofdstuk 2

Een punchout-site implementeren

Met punchout hebben gebruikers van inkooptoepassingen toegang tot leverancierscontracten voor producten of diensten die op uw website worden weergegeven. Daardoor is het niet meer nodig volledige catalogi naar inkooporganisaties te verzenden. In plaats daarvan zendt u slechts korte bestanden met een beschrijving van uw bedrijfsopzet, van productcategorieën of van producten.

In dit hoofdstuk wordt besproken hoe een website kan worden gewijzigd om punchout te ondersteunen. De volgende onderwerpen komen aan de orde:

- Vereisten voor punchout
- Volgorde van gebeurtenissen bij punchout
- Punchout-documenten
- Wijzigingen in uw webpagina's
- Suggesties bij punchout-websites

Vereisten voor punchout

Voordat inkooporganisaties hun inkooptoepassingen configureren voor punchout of leveranciers punchoutwebsites implementeren, dienen beide partijen de voordelen en vereisten van punchout te beoordelen.

Inkooporganisaties

In minder dan een dag kunnen cXML-compatibele inkooptoepassingen voor contact met de punchout-site van een leverancier worden opgezet en getest.

De barrières van technische integratie zijn laag voor inkooporganisaties. Daarom dient de beslissing om punchout te gaan gebruiken te worden gebaseerd op de bedrijfsvoering en het soort producten dat wordt ingekocht. (Zie "Strategie voor levering van de inhoud" op pagina 7 voor een lijst van producten die geschikt zijn voor punchout.)

Zakelijke kwesties

Inkooporganisaties dienen na te denken over de volgende vragen:

- Hebben de medewerkers die de inkoopaanvragen doen en goedkeuren, toegang tot het Internet? Als dit niet het geval is, kan een gereguleerde toegang tot het Internet worden toegestaan?
- Wil de inkooporganisatie dat de leveranciers catalogusinhoud (inclusief prijzen) maken en onderhouden?
- Kopen medewerkers die inkoopaanvragen doen op dit moment goederen in op het Internet? Als dit het geval is, vereisen deze goederen een configuratiehulpprogramma van de leverancier of bevatten deze goederen unieke attributen die niet kunnen worden opgenomen in een model met statische inhoud?
- Maakt de inkooporganisatie gebruik van inhoudsaggregators voor catalogi (bijvoorbeeld: Aspect, TPN Register of Harbinger)?
- Koopt de inkooporganisatie op dit moment diensten in (bijvoorbeeld: consultants, tijdelijke krachten of onderhoud) via het Internet?
- Maakt de inkooporganisatie op dit moment gebruik van on line-bronnen?

Wanneer het antwoord op een van de voorgaande vragen bevestigend is, kan punchout geschikt zijn voor deze inkooporganisatie.

Technische kwesties

Inkooporganisaties dienen aan de volgende technische vereisten te voldoen:

- Direct toegang tot het Internet - Gebruikers binnen inkooporganisaties dienen direct toegang te hebben tot het Internet. Punchout steunt op regelmatige browsersessies waarbij de gebruiker in wisselwerking staat met de live website van de leverancier. Deze communicatie vindt plaats via de normale infrastructuur van het Internet, niet via de inkooptoepassing.
- Betrouwbare verbinding met het Internet - De toegang tot het Internet dient voortdurend operationeel en betrouwbaar te zijn. Wanneer gebruikers geen producten kunnen inkopen vanwege onderbrekingen in de verbinding met het Internet, is dit van negatieve invloed op het inkoopbeleid.
- Contracten met punchout-leveranciers - Inkoopagenten dienen contracten te hebben met leveranciers die gebruikmaken van punchout. Alleen bekende, geverifieerde inkooporganisaties hebben toegang tot de punchoutwebsites.

Leveranciers

De term *leverancier* in de context van punchout omvat meer dan de traditionele definitie van de term. Het punchout-protocol werd ontworpen als flexibel raamwerk waardoor gegevens kunnen worden verzonden over vrijwel elke soort product of dienst van elke soort leverancier, distributeur, aggregator of fabrikant.

Voorbeelden van producten en diensten zijn onder meer:

- Computers, direct van fabrikant of wederverkoper
- Chemicaliën en reagens van een aggregator
- Kantoorbenodigdheden van een distributeur
- Contractdiensten van een uitzendbureau

U hebt wellicht al een transactieve website waarop inhoud wordt weergegeven en inkooporders kunnen worden ontvangen. Wanneer dit zo is, dient u bij de vraag of u wel of niet punchout gaat implementeren, uw bedrijfsvoering en technische hulpmiddelen nader te bekijken.

Zakelijke kwesties

Leveranciers dienen na te denken over de volgende vragen:

- Verkoopt u op dit moment uw producten of diensten via het Internet? Als dit het geval is, biedt u via uw website inhoud aan die specifiek op een bepaalde klant is gericht (contractprijzen)?
- Vallen uw producten en diensten in een van de punchoutcategorieën die zijn omschreven in de tabel in "Strategie voor levering van de inhoud" op pagina 7? Deze categorieën omvatten:
 - Producten met een hoge configuratiegraad (zoals computers)
 - Groot aantal regelitems (zoals boeken)
 - Unieke productattributen (zoals chemicaliën)
 - Genormaliseerde gegevens (zoals MRO-aanbod)
- Geeft u de voorkeur aan de ontvangst van inkooporders en/of betalingen via uw website?

Wanneer het antwoord op een van de voorgaande vragen bevestigend is, kan punchout geschikt zijn voor uw organisatie.

Technische kwesties

Leveranciers dienen aan de volgende technische vereisten te voldoen:

- Betrouwbare verbinding met het Internet - De infrastructuur van de webserver en de verbinding met het Internet dienen uitzonderlijk betrouwbaar te zijn. Wanneer gebruikers op afstand geen toegang hebben tot de inhoud, gaan deze wellicht naar een andere leverancier.
- Deskundige websitebeheerders - De punchoutwebsite en de ondersteunende toepassingen vereisen periodiek onderhoud en regelmatige wijzigingen. De behoeften van gebruikers en uw aanbiedingen veranderen, dus hebt u personeel nodig die de infrastructuur van de punchout wijzigen.
- Ondersteuning voor basistransacties - Punchout-websites hoeven niet de volledige cXML-functionaliteit te ondersteunen, maar ondersteuning van de volgende transacties is noodzakelijk:

Profieltransactie
PunchOutSetupRequest
PunchOutSetupResponse
PunchOutOrderMessage

Raming van de werkzaamheden

In de volgende tabel vindt u een raming van de werkzaamheden die zijn vereist voor de integratie van cXML-punchout, gebaseerd op schattingen van leveranciers:

Niveau van de reeds bestaande infrastructuur	Geschatte tijd voor voltooiing
Transactieve site met XML-infrastructuur	3 weken met intern IT-personeel 3-4 weken met contractanten
Transactieve site zonder XML-infrastructuur	4 weken met intern IT-personeel 4-5 weken met contractanten

Inzicht in XML

De eerste stap naar de implementatie van punchout is inzicht in XML. XML is een taal voor de beschrijving van talen. cXML-documenten worden geconstrueerd op basis van DTD's (Document Type Definitions). Deze DTD's worden als sjabloon gebruikt om inhoudmodellen te definiëren binnen een cXML-document (bijvoorbeeld voor de geldige volgorde en het nesten van elementen) en de gegevenstypen van attributen.

Info XML

XML (eXtensible Markup Language) is een standaardsysteem voor het doorgeven van gegevens tussen Internet-toepassingen. XML-documenten bevatten gegevens in de vorm van paren code/waarde. XML heeft een structuur die te vergelijken is met die van HTML (hypertext markup language), maar bij Internet-toepassingen kunnen gegevens gemakkelijker uit XML-documenten worden gehaald en gebruikt dan uit gewone HTML-documenten.

Naarmate Internet-toepassingen wijder verspreid raken, neemt het gebruik van XML toe.

Wanneer u een punchout-website wilt implementeren, dient u fundamenteel inzicht te hebben in de manier waarop XML-gegevens worden gemaakt, geparseerd, gezocht, ontvangen en verzonden van en naar een externe bron.

De basishulpprogramma's waarmee XML-documenten worden verwerkt, zijn XML-parsers. Deze parsers zijn gratis beschikbaar bij Microsoft en andere bedrijven (in Microsoft Internet Explorer 5 bijvoorbeeld is standaard een XML-parser opgenomen). Zie "Hulpprogramma's voor XML" op pagina 10 voor een lijst van XML-hulpprogramma's.

Volgorde van gebeurtenissen bij punchout

Een punchout-sessie bestaat uit een aantal afzonderlijke stappen.

Stap 1 & 2: Punchoutaanvraag

Gebruikers melden zich aan bij een inkooptoepassing en openen nieuwe inkoopaanvragen. Zij vinden de gewenste items door de lokale catalogi te doorzoeken op product, leverancier of productbeschrijving. Wanneer zij een punchout-item selecteren, wordt in de inkooptoepassing een nieuw browservenster geopend en worden de gebruikers aangemeld bij de eigen accounts op uw website.

De volgende afbeelding illustreert de stappen van de punchout-aanvraag:



Hoe werkt het? Wanneer een gebruiker op een punchout-item klikt, wordt via de inkooptoepassing het cXML-document `PunchOutSetupRequest` verzonden naar een hub van een e-commercenetwerk. De hub handelt als vertrouwde derde, accepteert de aanvraag, verifieert de inkooporganisatie en stuurt de aanvraag door naar uw punchout-website.

Opmerking: Alle cXML-documenten die via het Internet worden verzonden, gaan via beveiligde HTTPS-verbindingen die zijn gecodeerd met SSL 3.0 (Secure Socket Layer).

Het doel van deze aanvraag is uw website op de hoogte te stellen van de identiteit van de inkoper en om door te geven dat de bewerking moet worden uitgevoerd. Ondersteunde bewerkingen zijn onder meer de volgende:

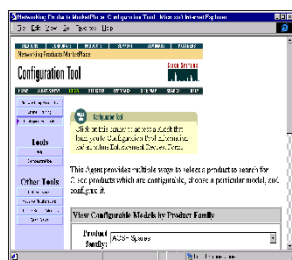
- **maken** - Hiermee wordt een nieuwe punchout-sessie gestart.
- **bewerken** - Hiermee wordt een punchout-sessie opnieuw geopend voor bewerking.
- **inspecteren** - Hiermee wordt een punchout-sessie opnieuw geopend voor inspectie. Er kunnen geen wijzigingen worden aangebracht in de gegevens.

Nadat een aanvraag op uw website wordt ontvangen, wordt een PunchOutSetupResponse teruggezonden waarin een URL is opgenomen die de inkooptoepassing aangeeft naar welke locatie moet worden gegaan om een browsersessie op uw website te starten.

In de inkooptoepassing wordt een nieuw browservenster geopend, waarin een sessie in een account op uw website wordt weergegeven. Deze account kan specifiek zijn voor een bepaalde regio, een bedrijf, een afdeling of een gebruiker.

Stap 3: Selectie van producten

Gebruikers selecteren items uit uw voorraad met behulp van alle functies en diensten op uw website:



3. Aanvrager zoekt en configureert producten via site van leverancier.

Afhankelijk van het product of de klant, kunnen deze functies onder meer de volgende zijn:

- Configuratiehulpprogramma's voor de opbouw van aangepaste producten (bijvoorbeeld computers, organische mengsels of persoonlijk gerichte producten).
- Zoekmachines voor het zoeken naar gewenste producten in grote catalogi.
- Weergaven van genormaliseerde gegevens voor de vergelijking van producten op grond van prijs, functies of beschikbaarheid (bijvoorbeeld MRO-producten).
- Weergaven van attributen die uniek zijn voor een bepaald product (bijvoorbeeld afgedrukte materialen, chemicaliën en reagens, of diensten).

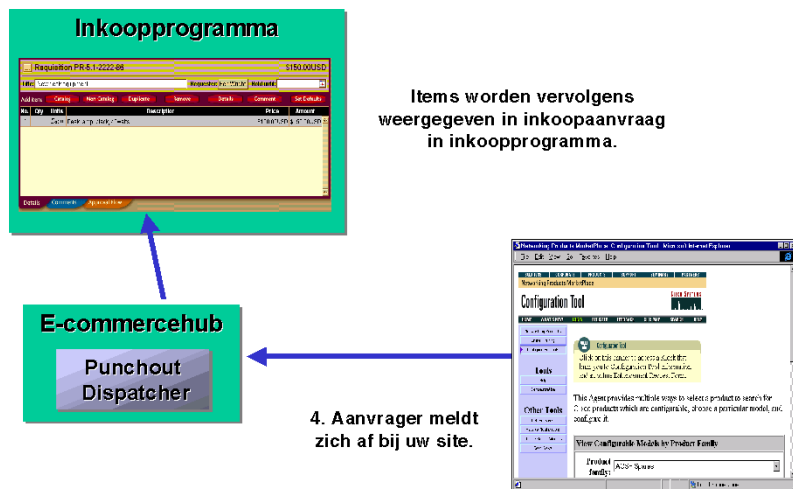
- Real-time prijzen, voorraad en controle van beschikbaarheid.
- Automatische berekening van belasting en vrachtkosten, gebaseerd op verzendbestemming, afmetingen of hoeveelheid van items (niet noodzakelijk om tijdens de punchout-sessie te berekenen).

Hoe werkt het? Nadat de gebruikers via de inkooptoepassing naar uw website zijn gedirigeerd, gaat het winkelen op dezelfde manier als wanneer zij direct aanmelden bij uw website. Geen enkele van de hiervoor genoemde functies en diensten hoeft dus te worden gewijzigd.

Stap 4: Check Out

Op uw website worden de totale kosten van de selecties van de gebruiker berekend, waaronder belasting, vracht en specifieke kortingen. De gebruiker klikt vervolgens op de knop Check Out van uw website om de inhoud van het winkelwagentje te verzenden naar de inkoopaanvragen binnen de inkooptoepassing.

De volgende afbeelding illustreert de stappen van de Check Out:



Hoe werkt het? Wanneer de gebruiker op de knop Check Out klikt, wordt vanuit uw website het cXML-document PunchOutOrderMessage, dat productdetails en prijzen bevat, naar de inkooptoepassing gezonden. U kunt ook verborgen leverancierscookies verzenden, die op een later tijdstip items kunnen verbinden met een bepaalde inkoopessie.

In feite hebt u een notering gemaakt van de aangevraagde items. U hebt nog geen inkooporder ontvangen, dus u kunt deze order ook nog niet boeken.

Wanneer gebruikers later een item in een inkoopaanvraag willen bewerken, kunt u hen toestaan re-punchout uit te voeren op uw website. Via de inkooptoepassing wordt de inhoud van de oorspronkelijke winkelwagen teruggezonden naar de website en de gebruiker kan ter plaatse wijzigingen aanbrengen. Na de check out worden de items vanuit de website teruggestuurd naar de inkoopaanvraag.

Uw website is de informatiebron voor alle punchout-items. Wijzigingen in de hoeveelheid of toevoeging van nieuwe items aan de inkoopaanvraag kunnen wijzigingen in de kosten van belasting of verzending tot gevolg hebben, waardoor er op de website een nieuwe berekening moet worden gemaakt. Daarom moeten wijzigingen in de oorspronkelijke items op de website worden aangebracht, niet in de inkooptoepassing en dat verklaart de noodzaak van de re-punchout. Een re-punchout is eenvoudigweg een PunchOutSetupRequest waarmee een bewerking wordt uitgevoerd.

Stap 5: Verzending van de inkooporder

Nadat de inhoud van het winkelwagentje vanuit de website is doorgestuurd naar de inkoopaanvraag van de gebruiker, wordt in de inkooptoepassing het goedkeuringsproces gestart. Wanneer de inkoopaanvraag wordt goedgekeurd, wordt de aanvraag omgezet in een inkooporder. De inkooporder wordt teruggezonden naar de website om te kunnen worden uitgevoerd. Tegelijk met de order kunnen inkoopkaartgegevens worden verzonden of u kunt de order bij levering factureren.

De volgende afbeelding illustreert de verzending van de inkooporder:



Hoe werkt het? Vanuit de inkooptoepassing worden alle inkooporders in cXML-indeling naar de hub van een e-commercenetwerk verzonden. Vanuit de hub worden deze vervolgens naar u gerouteerd via de gewenste methode voor orderrouting. Wanneer u de ontvangst van de inkooporder bevestigt, hebt u de order in feite geboekt.

De methode van orderrouting dient cXML te zijn voor leveranciers die punchout gebruiken en wel om de volgende redenen:

- Bij cXML-inkooporders kan ingesloten cookie-informatie van de leverancier naar u worden teruggezonden. De leverancierscookie kan elk gegevenstype zijn, daarom is het niet eenvoudig deze via een andere methode voor orderrouting, zoals e-mail, fax of EDI, te laten lopen.
- Leveranciers met punchout-sites gebruiken cXML. Het vergt dus weinig extra moeite om cXML-inkooporders te accepteren.

In Hoofdstuk 3, "cXML-inkooporders ontvangen." worden inkooporders gedetailleerd besproken.

Punchout-documenten

Er zijn vier typen cXML-documenten:

- Punchout, indexcatalogus
- PunchOutSetupRequest
- PunchOutSetupResponse
- PunchOutOrderMessage

Punchout, indexcatalogus

Indexcatalogi van punchouts zijn bestanden waarin punchout-items worden weergegeven en die verwijzen naar uw punchout-website.

In het volgende voorbeeld wordt een dergelijke indexcatalogus weergegeven:

Type cXML-document en URL van DTD	<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.10,007/cXML.dtd"> <Index> <SupplierID domain="DUNS">83528721</SupplierID> <IndexItem> <IndexItemPunchout> <ItemID> <SupplierPartID>5555</SupplierPartID> </ItemID> <PunchoutDetail> <Description xml:lang="en-US">Desk Chairs</Description> <Description xml:lang="fr-FR">Chaises de Bureau</Description> <URL>http://www.workchairs.com/punchout.asp</URL> <Classification domain="UNSPSC">5136030000</Classification> </PunchoutDetail> </IndexItemPunchout> </IndexItem> </Index></pre>
Uw identificatie voor het punchout-item	
URL voor uw punchout-website (startpagina)	

SupplierID Hiermee wordt de leveranciersorganisatie geïdentificeerd. U kunt elk identificatiedomein gebruiken, maar de aanbevolen domeinen zijn DUNS (Dun & Bradstreet Universal Naming System) en Netwerk-ID. Zie www.dnb.com voor meer informatie over DUNS-nummers.

Description Dit is de tekst die in de productcatalogi staat en wordt weergegeven in de inkooptoepassing. U kunt de omschrijving in verschillende talen opgeven. In de inkooptoepassing wordt vervolgens de juiste taal voor de gebruiker weergegeven.

Classification Hiermee wordt de productgroep van het regelitem aangegeven. Al uw producten en diensten dienen te zijn geordend en gestandaardiseerd volgens het UNSPSC-schema. Bij indexcatalogi van punchout-items bepaalt de Classification de locatie van het punchout-item binnen catalogi die voor gebruikers worden weergegeven. Zie www.unspsc.com voor een lijst van UNSPSC-codes.

Indexcatalogi maken en publiceren

Maak de catalogi en publiceer deze op een e-commercehub. De catalogusbeheerder binnen een inkooporganisatie downloadt de catalogi en slaat deze op voor gebruik in inkooptoepassingen.

Gebruikers zien de inhoud van uw indexcatalogi voor punchout-items naast de normale, statische catalogusitems.

Granulatie punchout-items

U kunt catalogi maken op winkelniveau, op middenpadniveau of op productniveau.

- In catalogi op winkelniveau worden alle producten en diensten weergegeven. Gebruikers moeten zoeken om het gewenste item te vinden.
- Catalogi op middenpadniveau geven verwante producten en diensten weer.
- Catalogi op productniveau geven slechts één product of dienst weer. De gebruiker hoeft niet te zoeken.

Het niveaubreedte van uw punchout-items is afhankelijk van uw bedrijfsmodel, de samenstelling van het aanbod van uw producten en diensten en de structuur van uw punchout-website.

Hoe meer zoek- en configuratiehulpmiddelen u op uw website hebt, des te breder kunt u de punchout-items in uw catalogus maken.

PunchOutSetupRequest

Een punchout-sessie wordt gestart wanneer een gebruiker uw punchout-item selecteert. In de inkooptoepassing wordt het document PunchOutSetupRequest gegenereerd en verzonden naar een e-commercehub, vanwaar dit wordt doorgestuurd naar uw punchout-website.

Dit is een voorbeeld van een PunchOutSetupRequest-document:

		<pre><?xml version="1.0"?> <!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd"> <cXML version="1.1.007" xml:lang="en-US" payloadID="933694607118.1869318421@jlee" timestamp="2000-08-15T08:36:47- 07:00"> <Header> <From> <Credential domain="DUNS"> <Identity>65652314</Identity> </Credential> </From> <To> <Credential domain="DUNS"> <Identity>83528721</Identity> </Credential> </To> <Sender> <Credential domain="AribaNetworkUserId"> <Identity>sysadmin@ariba.com</Identity> <SharedSecret>abracadabra</SharedSecret> </Credential> <UserAgent>Ariba ORMS 6.1</UserAgent> </Sender> </Header> <Request> <PunchOutSetupRequest operation="create"> <BuyerCookie>1CX3L4843PPZO</BuyerCookie> <Extrinsic name="CostCenter">610</Extrinsic> <Extrinsic name="User">john_smith</Extrinsic> <BrowserFormPost> <URL>https://aribaorms:26000/punchout.asp</URL> </BrowserFormPost> <SupplierSetup> <URL>http://www.workchairs.com/punchout.asp</URL> </SupplierSetup> <SelectedItem> <ItemOut quantity="1"> <ItemID> <SupplierPartID>5555</SupplierPartID> </ItemID> </ItemOut> </SelectedItem> </PunchOutSetupRequest> </Request> </cXML></pre>
Originator (inkooporganisatie)		
Destination (leverancier)		
Previous relaying entity (in dit geval Ariba Network)		
Soort aanvraag		
Bestemming voor uiteindelijke PunchOutOrderMessage		
Door gebruiker geselecteerd item		

```
        </ItemOut>
      </SelectedItem>
    </PunchOutSetupRequest>
  </Request>
</cXML>
```

De attributen `payloadID` en `timestamp` vlakbij het begin worden door cXML-clients gebruikt om documenten te volgen en duplicaten van documenten te traceren.

Met de elementen `From`, `To` en `Sender` kunnen ontvangende systemen partijen identificeren en verifiëren. De elementen `From` en `To` in een document worden niet gewijzigd. Tijdens de verzending van het document naar de bestemming, wordt het element `Sender` in de tussenliggende knooppunten (zoals Ariba Network) wel gewijzigd.

Maken, bewerken en inspecteren

Met het attribuut `operation` wordt het type sessie aangegeven dat de inkoper start. Dit kunnen de volgende acties zijn: maken, bewerken of inspecteren.

- **maken** Met deze sessies worden nieuwe winkelwagentjes gegenereerd die corresponderen met nieuwe inkoopaanvragen.
- **bewerken** In deze sessies worden eerder gemaakte winkelwagentjes opnieuw geopend om deze te kunnen bewerken. In inkooptoepassingen worden regelitemgegevens verzonden als onderdeel van de `PunchOutSetupRequest`. Op de punchout-website worden deze gegevens gebruikt om het winkelwagentje dat tijdens de oorspronkelijke sessie is gemaakt, opnieuw te concretiseren.
- **inspecteren** In deze sessies worden eerder gemaakte winkelwagentjes opnieuw geopend, maar alleen om te bekijken. Net als bij bewerken worden in de inkooptoepassing regelitemgegevens verzonden als onderdeel van de `PunchOutSetupRequest`. Nadat de winkelwagen opnieuw is geconcretiseerd, kan de inhoud van de punchout-website echter niet worden gewijzigd.

In het volgende voorbeeld ziet u een aanvraag voor bewerken:

```
<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US" payloadID="933695135608.677295401@jlee"
timestamp="2000-08-15T08:45:35-07:00">
  <Header>
    <From>
      <Credential domain="DUNS">
        <Identity>65652314</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>83528721</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="AribaNetworkUserId">
        <Identity>sysadmin@ariba.com</Identity>
        <SharedSecret>abracadabra</SharedSecret>
      </Credential>
      <UserAgent>Ariba ORMS 6.1</UserAgent>
    </Sender>
  </Header>
  <Request>
    <PunchOutSetupRequest operation="bewerken">
      <BuyerCookie>1CX3L4843PPZO</BuyerCookie>
      <Extrinsic name="CostCenter">610</Extrinsic>
      <Extrinsic name="User">john_smith</Extrinsic>
      <BrowserFormPost>
        <URL>https://aribaorms:26000/punchout.asp</URL>
      </BrowserFormPost>
      <SupplierSetup>
        <URL>http://www.workchairs.com/punchout.asp</URL>
      </SupplierSetup>
      <ItemOut quantity="2">
        <ItemID>
          <SupplierPartID>220-6338</SupplierPartID>
          <SupplierPartAuxiliaryID>E000028901
          </SupplierPartAuxiliaryID>
        </ItemID>
      </ItemOut>
    </PunchOutSetupRequest>
  </Request>
</cXML>
```

Wanneer de gebruiker de bewerkssessie heeft gestart door de selectie van een catalogusitem, bevat de PunchOutSetupRequest een element SelectedItem, zoals bij maken.

Verificatie door een e-commercehub

Alle PunchOutSetupRequest-documenten worden via een e-commercehub gerouteerd voor verificatie en om de URL van uw punchout-website op te zoeken. De stappen zijn als volgt:

1. In de hub wordt het document PunchOutSetupRequest van de gebruiker ontvangen.
2. In de hub wordt de ID van de inkoper geverifieerd (Van en Gedeeld geheim) aan de hand van de e-commerceaccount van de inkoper. De aangevraagde leverancier wordt eveneens geïdentificeerd (Aan).
3. In de hub wordt het gedeeld geheim van uw account opgezocht en dit (Gedeeld geheim) wordt ingevoegd in het element Sender.
4. De URL van uw punchout-website wordt eveneens opgezocht en het PunchOutSetupRequest-document wordt naar deze URL verzonden.
5. Het cXML-document wordt ontvangen op uw website. Het is duidelijk dat het document is geverifieerd, want het bevat uw eigen gedeeld geheim.
6. Op de website wordt de informatie in het element From gebruikt om de aanvrager op bedrijfsniveau te identificeren (bijvoorbeeld: acme.com).
7. U kunt de gegevens van de elementen Contact en extrinsic in de body van de aanvraag gebruiken om de gebruiker te identificeren (bijvoorbeeld: Jan Smit, financiële afdeling van acme.com).

De documenten PunchOutSetupRequest en PunchOutSetupResponse worden voor verificatie door de e-commercehub geleid. Het document PunchOutOrderMessage (waarmee de inhoud van het winkelwagentje wordt geretourneerd naar de inkooptoepassing) reist direct tussen de website en de gebruiker via standaard HTTP of HTTPS.

Supplier Setup-URL en SelectedItem

In eerdere versies van cXML kon de URL van uw punchout-website alleen worden opgegeven in het element `SupplierSetup`. Vanaf cXML versie 1.1 is de e-commercehub reeds op de hoogte van de URL van uw punchout-website.

Eveneens vanaf cXML versie 1.1 kan in inkooptoepassingen het element `SelectedItem` worden gebruikt om punchout op winkelniveau, middenpadniveau of productniveau op te geven.

Het element `SupplierSetup` is afgekeurd. Uw punchout-website moet echter beide methoden kunnen hanteren totdat alle punchout-websites het element `SelectedItem` herkennen.

Contact-gegevens voor Extrinsic-gegevens en identificatie van de gebruiker

Het document `PunchoutSetupRequest` kan gedetailleerde gebruikersinformatie bevatten in het element `Contact`, die op de website kan worden gebruikt om gebruikers te verifiëren en te begeleiden. Dit kunnen de volgende gegevens zijn:

- Naam en functie van de gebruiker
- E-mailadres

Daarnaast kan de `PunchOutSetupRequest` ook *extrinsic*-gegevens bevatten die u kunt gebruiken om gebruikers nader te identificeren, zoals:

- Kostenplaats en subaccount van de gebruiker
- Regio
- Supervisor
- Standaardvaluta

Inkooporganisaties configureren inkooptoepassingen zodanig dat deze Contact- en extrinsic-gegevens invoegen. Vraag uw klanten welke gegevens u kunt verwachten.

PunchOutSetupResponse

Nadat het document PunchOutSetupRequest op uw website is ontvangen, wordt een PunchOutSetupResponse verzonden. Het document PunchOutSetupResponse heeft twee functies:

- Het geeft aan of de PunchOutSetupRequest met succes is verzonden.
- Het verschaft de inkooptoepassing een doorstuur-URL naar uw startpagina.

Het document bevat een <URL>-element dat de URL van de startpagina aangeeft. Deze URL wordt doorgestuurd naar de webbrowser van de gebruiker zodat de interactieve browsersessie kan worden gestart. In deze URL moet voldoende informatie zijn opgenomen om verbinding te kunnen leggen met een sessiecontext op de website, zoals de identiteit van de aanvrager en de inhoud van het element BuyerCookie.

In het volgende voorbeeld wordt een PunchOutSetupResponse-document weergegeven:

```
<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US" payloadID="933694607739"
timestamp="2000-08-15T08:46:00-07:00">
  <Response>
    <Status code="200" text="success"></Status>
    <PunchOutSetupResponse>
      <StartPage>
        <URL>
          http://xml.workchairs.com/retrieve?reqUrl=20626;Initial=TRUE
        </URL>
      </StartPage>
    </PunchOutSetupResponse>
  </Response>
</cXML>
```

PunchOutOrderMessage

Nadat de gebruiker items heeft geselecteerd op de website, deze heeft geconfigureerd en op de knop Check Out heeft geklikt, verzendt u een PunchOutOrderMessage om de inhoud van het winkelwagentje door te sturen naar de inkooptoepassing van de inkoper. Dit document kan veel meer gegevens bevatten dan andere documenten, omdat de inhoud van elke denkbare winkelwagen volledig onder woorden moet worden gebracht. Dit document volgt het Request/Response-model niet strikt. Het gebruik van dit document wordt nog gedetailleerd besproken.

In het volgende voorbeeld wordt een PunchOutOrderMessage weergegeven:

```
<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US" payloadID="933695160894"
timestamp="2000-08-15T08:47:00-07:00">
  <Header>
    <From>
      <Credential domain="DUNS">
        <Identity>83528721</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>65652314</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="workchairs.com">
        <Identity> website 1</Identity>
      </Credential>
      <UserAgent>Workchairs cXML Application</UserAgent>
    </Sender>
  </Header>
  <Message>
    <PunchOutOrderMessage>
      <BuyerCookie>1CX3L4843PPZO</BuyerCookie>
      <PunchOutOrderMessageHeader operationAllowed="bewerken">
        <Total>
          <Money currency="USD">763.20</Money>
        </Total>
      </PunchOutOrderMessageHeader>
      <ItemIn quantity="3">
        <ItemID>
          <SupplierPartID>5555</SupplierPartID>
          <SupplierPartAuxiliaryID>E000028901
          </SupplierPartAuxiliaryID>
        </ItemID>
      </ItemDetail>
    </PunchOutOrderMessage>
  </Message>
</cXML>
```

```
<UnitPrice>
  <Money currency="USD">763.20</Money>
</UnitPrice>
<Description xml:lang="en">
  <ShortName>Excelsior Desk Chair</ShortName>
  Leather Reclining Desk Chair with Padded Arms
</Description>
<UnitOfMeasure>EA</UnitOfMeasure>
<Classification domain="UNSPSC">5136030000
</Classification>
</ItemDetail>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>
```

Door gebruik van een BuyerCookie kan in de inkooptoepassing een gegeven PunchOutOrderMessage in verband worden gebracht met de oorspronkelijke PunchOutSetupRequest. Daarom moet dit element, indien aanwezig, altijd vanuit de website worden geretourneerd. Gebruik de BuyerCookie niet om punchout-sessies te volgen, want dit element verandert bij elke sessie, van maken naar inspecteren, naar bewerken.

De SupplierPartAuxiliaryID werkt als een leverancierscookie. Via dit veld kunt u extra gegevens verzenden, zoals quote number of een ander cXML-document. Vanuit de inkooptoepassing wordt dit teruggezonden in een volgende PunchOutSetupRequest bij bewerken of inspecteren, en in de uiteindelijke cXML-inkooporder. U kunt de leverancierscookie gebruiken om items in een inkoopaanvraag te associëren met de corresponderende items in een winkelwagentje op uw website.

In UnitOfMeasure wordt beschreven hoe het product wordt verpakt of verzonden. Deze beschrijving moet conform de UN/CEFACT Unit of Measure Common Codes zijn. Zie www.unece.org/cefact voor een lijst van UN/CEFACT codes.

In Classification wordt de commodity-code van UNSPSC (United Nations Standard Product and Service Code) voor elk geselecteerd item weergegeven. Deze codes worden door back-endsystemen binnen inkoop- en leveranciersorganisaties gebruikt voor de boekhouding en de generering van rapporten. Zie www.unspsc.org voor een lijst van UNSPSC-codes.

Wijzigingen in uw webpagina's

Wanneer u de drie cXML-punchout-documenten wilt ontvangen of verzenden, is het waarschijnlijk nodig vier pagina's op uw website te maken of te wijzigen:

- Launch Page
- Start Page
- Sender Page
- Order Receiver Page

Ter illustratie van hoe u deze pagina's zou kunnen implementeren, wordt gebruikgemaakt van eenvoudige voorbeelden van ASP-code (Active Server Page) en de XML-parser van Microsoft Internet Explorer 5. De werkelijke implementatie van deze pagina's kan hiervan afwijken, afhankelijk van de ontwikkelomgeving van de leverancier (bijvoorbeeld: CGI, JavaScript of WebObjects).

Launch Page

Op de Launch Page worden alle geverifieerde PunchOutSetupRequest-documenten ontvangen vanuit de e-commercehub. De HTTP-stroom die vanuit de hub wordt verzonden, wordt gelezen en de cXML-aanvraag die binnen de stroom is ingesloten, wordt gevalideerd aan de hand van de cXML-DTD (in het geval van ASP, met behulp van method-oproepen naar de XML-parser van Internet Explorer 5).

Na validatie worden op de Launch Page elementen uit het document gehaald om:

1. De gebruiker te identificeren en te bepalen naar welke pagina de gebruiker wordt verwezen.
2. Een PunchOutSetupResponse-document op te stellen en dit te retourneren aan de zender.

Op de Launch Page worden de volgende gegevens opgeslagen voor gebruik op de Start Page:

- Identiteit van de aanvrager (Sender)
- De taal van de gebruiker (xml:lang) zodat u vertaalde inhoud kunt weergeven
- Type van de aanvraag (maken, bewerken of inspecteren)
- Extrinsic-gegevens die de gebruiker en de gebruikerslocatie nader identificeren.

U ziet hierna een voorbeeld van een Launch Page. Deze code maakt geen gebruik van een XML-parser om de PunchOutSetupResponse dynamisch te genereren, maar van een statisch XML-sjabloon waarin regelitemgegevens zijn ingevuld. **Deze code is uitsluitend bedoeld ter illustratie.**

```
script language=JScript RUNAT=Server>
function elementValue(xml, elem)
{
    var begidx;
    var endidx;
    var retStr;

    begidx = xml.indexOf(elem);
    if (begidx > 0) {
        endidx = xml.indexOf('</', begidx);
        if (endidx > 0)
            retStr = xml.slice(begidx+elem.length,
                               endidx);
        return retStr;
    }
    return null;
}

function twoChar( str )
{
    var retStr;
    str = str.toString();
    if ( 1 == str.length ) {
        retStr = "0" + str;
    } else {
        retStr = str;
    }
    return retStr;
}

function timestamp( dt )
{
    {
        var str;
        var milli;
        str = dt.getFullYear() + "-" + twoChar( 1 + dt.getMonth() ) + "-";
        str += twoChar( dt.getDate() ) + "T" + twoChar( dt.getHours() ) + ":";
        str += twoChar( dt.getMinutes() ) + ":" + twoChar( dt.getSeconds() ) + ".";
        milli = dt.getMilliseconds();
        milli = milli.toString();
        if ( 3 == milli.length ) {
            str += milli;
        } else {
            str += "0" + twoChar( milli );
        }
        str += "-08:00";
        return str;
    }
}
```



```

function genProlog( cXMLvers, randStr )
{
    var dt;
    var str;
    var vers, sysID;
    var nowNum, timeStr;
    if ( 1.1 > parseFloat( cXMLvers )) {
        vers = "1.0";
        sysID = "cXML.dtd";
    } else {
        vers = "1.1.007";
        sysID = "http://xml.cXML.org/schemas/cXML/" + vers + "/cXML.dtd";
    }
    dt = new Date();
    nowNum = dt.getTime();
    timeStr = timestamp( dt );
    str = '<?xml version="1.1.007" encoding="UTF-8"?>\n';
    str += '<!DOCTYPE cXML SYSTEM "' + sysID + '">\n';
    str += '<cXML version="' + vers + '" payloadID="' + nowNum + '".';
    str += randStr + '@' + Request.ServerVariables("LOCAL_ADDR");
    str += '" timestamp="' + timeStr + '">';
    return str;
}
</script>
REM, Gegevens voor maken, noodzakelijk in proloog.
%<
Randomize
randStr = Int( 100000001 * Rnd )
prologStr = genProlog( "1.0", randStr )
Response.ContentType = "text/xml"
Response.Charset = "UTF-8"
%>
<%
REM Dit ontvangt de PunchOutSetup request die vanuit de e-commercehub binnenkomt.
REM Het neemt de ORMSURL en buyercookie en hecht deze aan de URL van de Start
Page,
REM en stuurt de response terug naar de aanvrager.
REM punchoutredirect.asp?bc=2133hfefe&url="http://workchairs/com/..&redirect="
Dim ret
Dim punch
Dim statusText
Dim statusCode
Dim cookie
Dim url
Dim xmlStr
Dim fromUser
Dim toUser
cookie = ""
url = ""
xmlStr = ""

```

```

dir = ""
path = Request.ServerVariables("PATH_INFO")
dir = Left(path, InstrRev(path, "/"))
if IsEmpty(dir) then
    dir = "/"
end if

REM Met deze opdracht wordt de binnenkomende HTTP-cXML-aanvraag gelezen
xml = Request.BinaryRead(Request.TotalBytes)
for i = 1 to Request.TotalBytes
    xmlstr = xmlstr + String(1, AscB(MidB(xml, i, 1)))
Next
cookie = elementValue(xmlstr, "<BuyerCookie>")
url = elementValue(xmlstr, "<URL>")
fromUser = elementValue(xmlstr, "<Identity>")
newXMLStr = Right(xmlstr, Len(xmlstr) - (Instr(xmlstr, "<Identity>") +
Len("<Identity>")))
toUser = elementValue(newXMLStr, "<Identity>")
%>
REM Hiermee wordt de cXML-PunchOutSetupResponse ingedeeld
<% if IsEmpty(cookie) then %>
<%= prologStr %>
<Response>
    <Status code="400" Text="Bad Request">Invalid Document. Unable to extract
    BuyerCookie.</Status>
</Response>
</cXML>
<% else %>
<%= prologStr %>
<Response>
    <Status code="200" text="OK"/>
    <PunchOutSetupResponse>
        <StartPage>
            <URL>http://<%=
Request.ServerVariables("LOCAL_ADDR")%>/<%= dir%>/punchoutredirect.asp?bc=<%=
cookie%>&amp;url="<%= url%>"&amp;from=<%= fromUser%>&amp;to=<%=
toUser%>&amp;redirect=<%= StartPage%></URL>
        </StartPage>
    </PunchOutSetupResponse>
</Response>
</cXML>
<%end if%>

```

Vanuit de Launch Page dient een URL voor de StartPage te worden geretourneerd die uniek is voor de betreffende punchout-sessie. Bovendien moet deze URL slechts voor een beperkte tijdsperiode geldig zijn. Door deze URL te deactiveren, maakt u het onbevoegde gebruikers moeilijker toegang te verkrijgen tot uw Start Page.

Vergeet niet de functionaliteit te implementeren voor de volgende sessies bewerken en inspecteren. Gebruikers kunnen binnen de inkooptoepassing geen orderdetails voor punchout-items (zoals hoeveelheid) wijzigen. Zij kunnen dit alleen doen met een sessie bewerken. Het is voor gebruikers bijzonder handig wanneer bij sessies inspecteren, die plaatsvinden nadat u de order hebt ontvangen, de orderstatus wordt weergegeven.

Start Page

Op de Start Page wordt de aanvrager aangemeld bij een account op uw website. Vanuit de Start Page kunnen de gebruikers gaan winkelen. Deze pagina kan al op uw website aanwezig zijn, maar u dient deze zodanig te wijzigen dat de gebruikersnaam en het wachtwoord in het PunchOutSetupRequest-document kunnen worden opgezocht.

Zorg ervoor dat alleen bevoegde gebruikers toegang hebben tot de Start Page. Wanneer u met de verificatie wacht totdat de gebruiker de site verlaat, bent u niet in staat vertrouwelijke prijzen of voorwaarden te beschermen.

Wanneer u gebruikmaakt van HTTP-browsercookies om de voorkeuren van een gebruiker vast te stellen, dient u deze te vernietigen nadat de PunchOutOrderMessage naar de inkoper is verzonden. Met het vernietigen van deze cookies voorkomt u dat privileges worden aangeboden aan onbevoegde gebruikers.

Sender Page

Vanuit de Sender Page wordt de inhoud van het winkelwagentje van de gebruiker naar de gebruiker verzonden. Zoals hiervoor is beschreven, klikken gebruikers op de knop Check Out wanneer zij de winkelwagen hebben gevuld.

U ziet hierna een eenvoudige ASP-implementatie van deze functie. Deze code maakt geen gebruik van een XML-parser om de PunchOutSetupMessage dynamisch te genereren, maar van een statisch XML-sjabloon waarin regelitemgegevens zijn ingevuld. **Deze code is uitsluitend bedoeld ter illustratie.**

Dit is een gedeelte van de productpagina op de website van een leverancier:

```
<!--#include file="punchoutitem.inc"-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from
url=(0093)https://secure1.shore.net/wbird/cgi/vsc.cgi/wbird/houses/urban.htm?L+wbird+w
adt4101+928011405 -->

<TABLE border=0>
  <TBODY>
    <TR>
      <TD><IMG src="UrbanHouses_files/uhjm.gif"> </TD>
      <TD><STRONG>Jefferson Memorial</STRONG>- A birdfeeder with a
rotunda! This famous American monument will be a unique addition to any garden or yard.
It attracts small to medium sized birds and its dimensions are 11" x 9 1/2" x 8" H.
    </TD>
    </TR>
  </TBODY>
</TABLE><BR>
-Jefferson Memorial<STRONG>
$139.95</STRONG><BR>
<% AddBuyButton 139.95,101,"Bird Feeder, Jefferson Memorial",5 %>
<BR>
<HR>
```

Met de functie AddBuyButton wordt de PunchOutOrderMessage teruggezonden naar de gebruiker.

De volgende weergave is het include-bestand (punchoutitem.inc) waarnaar hiervoor werd verwezen:

```
<%
REM Deze asp wordt opgenomen in items.asp, waarin de parameters van de items worden
gespecificeerd,
REM een cXML-document wordt ingedeeld, waardoor de gebruiker kan doorgaan met de
checkout van een item.
function CreateCXML(toUser, fromUser, buyerCookie, unitPrice, supPartId, desc)
%>
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE cXML SYSTEM
"http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" payloadID="<%= Now &"@ "&
Request.ServerVariables("LOCAL_ADDR")%>" timestamp="<%= Now
%>">
  <Header>
    <From>
      <Credential domain="ariba.com">
        <Identity><%= toUser%></Identity>
      </Credential>
```

```

        </From>
        <To>
            <Credential domain="ariba.com">
                <Identity><%= fromUser%></Identity>
            </Credential>
        </To>
        <Sender>
            <Credential domain="ariba.com">
                <Identity><%= toUser%></Identity>
            </Credential>
            <UserAgent>PunchoutSite</UserAgent>
        </Sender>
    </Header>
    <Message>
        <PunchOutOrderMessage>
            <BuyerCookie><%= buyerCookie%></BuyerCookie>
            <PunchOutOrderMessageHeader
                operationAllowed="edit">
                <Total>
                    <Money currency="USD"><%=
                        unitPrice%></Money>
                </Total>
            </PunchOutOrderMessageHeader>
            <ItemIn quantity="1">
                <ItemID>
                    <SupplierPartID><%= supPartId%></SupplierPartID>
                    <SupplierPartAuxiliaryID><%= supPartAuxId%>
                    </SupplierPartAuxiliaryID>
                </ItemID>
                <ItemDetail>
                    <UnitPrice>
                        <Money currency="USD"><%= unitPrice%>
                    </Money>
                    </UnitPrice>
                    <Description xml:lang="en"><%= desc%>
                    </Description>
                    <UnitOfMeasure>EA</UnitOfMeasure>
                    <Classification
                        domain="SupplierPartID"><%= supPartId%>
                    </Classification>
                </ItemDetail>
            </ItemIn>
        </PunchOutOrderMessage>
    </Message>
</cXML>
<% end function

function AddBuyButton(unitPrice, supPartId, supPartAuxId, desc)

toUser = Session("toUser")

```

```

fromUser = Session("fromUser")

buyerCookie = Session("buyercookie")
url = Session("urlToPost")
if not IsEmpty(buyerCookie) then
    %>
    <FORM METHOD=POST ACTION=<%= url%>>
    <INPUT TYPE=HIDDEN NAME="cxml-urlencoded" VALUE="<%= CreateCXML
toUser, fromUser, buyerCookie, unitPrice, supPartId, supPartAuxId, desc%>">
    <INPUT TYPE=SUBMIT value=BUY>
    </FORM>
<%else%>
</p>
<%
end if
end function
%>

```

De functie AddBuyButton bevat de FORM POST waarmee de URL-gecodeerde PunchOutOrderMessage wordt teruggezonden naar de gebruiker.

HTTP Form Encoding

Wanneer u een PunchOutOrderMessage verzendt, gebruikt u *HTML form encoding*, een ander transportmodel dan het traditionele HTTP-request/responsemiddel. Deze andere transportmethode zorgt voor een gemakkelijkere integratie tussen uw website en de inkooptoepassing. Bovendien kunnen inkooporganisaties nu XML-gegevens ontvangen zonder een webserver met een brandscherm nodig te hebben.

In plaats van een PunchOutOrderMessage direct naar de inkooptoepassing te verzenden, wordt deze op de website gecodeerd als verborgen HTML Form-veld en naar de URL gezonden die is opgegeven in het element BrowserFormPost van de PunchOutSetupRequest. Het verborgen HTML Form-veld moet een van de volgende namen krijgen: cxml-urlencoded of cxml-base64 (deze namen zijn niet hoofdlettergevoelig).

Dankzij deze codering kunt u een checkoutwebpagina ontwerpen die het cXML-document bevat. Wanneer gebruikers op de knop Check Out klikken, worden de gegevens (onzichtbaar voor gebruikers) vanuit de website aan de inkooptoepassing gepresenteerd als een HTML Form Submit.

Punchout annuleren

U wilt wellicht een knop Annuleren toevoegen aan uw pagina's, zodat gebruikers de punchout-sessie kunnen annuleren. Met de knop Annuleren wordt een lege `PunchOutOrderMessage` verzonden waarmee bij de inkooptoepassing wordt aangegeven dat er geen items worden geretourneerd en dat aanwezige punchout-items uit de aanvraag moeten worden verwijderd. U kunt de knop ook gebruiken voor huishoudelijke bewerkingen op de website, zoals de winkelwagen legen en de gebruikerssessie sluiten.

Order Receiver Page

Op de Order Receiver Page worden cXML-inkooporders geaccepteerd die zijn verzonden door inkooporganisaties. Deze pagina kan lijken op de hiervoor besproken Launch Page.

Zie Hoofdstuk 3, "cXML-inkooporders ontvangen." voor informatie over de ontvangst van inkooporders.

Suggesties bij punchout-websites

Neem de volgende suggesties in overweging wanneer u overweegt een punchout-website te implementeren.

Richtlijnen voor de implementatie

Volg deze richtlijnen bij de ontwikkeling van uw punchout-website:

- Bestudeer de cXML-specificatie.
- Gebruik een XML-parser en valideer documenten aan de hand van de cXML-DTD.
- Gebruik de eigenschap `xml:lang=` om de taal van de gebruiker vast te stellen zodat u vertaalde inhoud kunt weergeven.
- Gebruik de referentie `From` om inkooporganisaties te identificeren.
- Verzend een unieke, tijdelijke URL voor de sessie.
- Houd niet vast aan browsercookies.
- Overlaad uw klanten niet met eisen voor extrinsic-gegevens.

- Gebruik voor elk regelitem UNUOM (United Nations Units of Measure) en UNSPSC (United Nations Standard Product and Service Codes).
- Geef uw klanten waardevolle informatie. Geef de beschikbaarheid van producten weer, de status van orders en speciale aanbiedingen.
- De wijze waarop de site kan worden verlaten, moet intuïtief en gemakkelijk te begrijpen zijn. In het gunstigste geval hoeft de gebruiker slechts op drie knoppen te klikken om te kunnen inkopen.
- Breng een code aan voor volgende sessies van bewerken en inspecteren. Binnen de inkooptoepassing kunnen gebruikers geen orderdetails (zoals hoeveelheid) voor punchout-items wijzigen. Hiervoor is een bewerkingssessie nodig.
- U bent uw klanten zo goed mogelijk ter wille wanneer bij inspectiesessies de orderstatus wordt weergegeven.
- Test uw punchout-website. Neem de tijd om de site te testen in combinatie met de inkooptoepassingen van uw klanten.
- Bij punchout-transacties worden noteringen geproduceerd, geen inkooporders. Implementeer een pagina voor de ontvangst van cXML-inkooporders waarop u orders kunt accepteren.

Cookies van inkoper en leverancier

Met de cookies van inkoper en leverancier kunnen zowel inkopers als leveranciers de eigen regelitemgegevens opnieuw concretiseren voor hun back-endsystemen.

- Stuur de cookie van de inkoper (BuyerCookie) terug wanneer u deze ontvangt. Breng geen wijzigingen aan.
- Maak gebruik van de leverancierscookie (SupplierPartAuxiliaryID).

De inkoperscookie is analoog aan een inkoopaanvraagnummer. Met deze cookie kan het systeem van de inkooporganisatie de relatie onderhouden tussen een aanvraag en een winkelwagen.

Op dezelfde manier is de leverancierscookie analoog aan een quote number. Met deze cookie kunt uw systeem een relatie onderhouden tussen een winkelwagentje en de inkoopaanvraag en inkooporder van de inkoper. Vanuit de inkooptoepassingen wordt de leverancierscookie teruggestuurd in een volgende sessie van punchout, bewerken of inspecteren, en in de uiteindelijke inkooporder. Door de leverancierscookie vervalt eveneens de noodzaak zichtbare, leveranciersspecifieke gegevens vanuit de website terug te sturen naar de inkoper.

Aanpassing

In de kop van de `PunchOutSetupRequest` wordt de inkooporganisatie altijd geïdentificeerd, maar de aanvraag kan ook Contact- en Extrinsic-gegevens bevatten (zoals de kostenplaats of locatie van de gebruiker of de productcategorie) die u kunt gebruiken om de dynamische URL ten dienste van de gebruiker te bepalen.

Hoewel niet alle inkooporganisaties extrinsic-gegevens verzenden, kunt u deze gebruiken om uw webwinkel nauwkeuriger te organiseren. U kunt bijvoorbeeld een aparte webwinkel maken voor elke kostenplaats binnen de inkooporganisatie (of voor elke productcategorie of gebruiker).

U kunt ook de eerdere noteringen van een gebruiker opslaan en weergeven. Ook kunt u gebruikers noteringen opnieuw laten gebruiken, de status van orders laten controleren en rapporten maken over activiteiten in het verleden. Sla de historie van noteringen uitsluitend per gebruiker op om beveiligingsproblemen te voorkomen.

Een van de belangrijkste overwegingen tijdens de planning is de hoeveelheid werk die het kost een sterk dynamische en aangepaste punchout-website te implementeren. U dient evenwicht te vinden tussen aanpassing en complexiteit – een complexe website kost meer tijd om te implementeren en te onderhouden, maar kan van grotere waarde zijn voor de gebruikers. Het verdient aanbeveling te beginnen met een eenvoudige punchout-website en deze langzamerhand uit te breiden.

Hoofdstuk 3

cXML-inkooporders ontvangen

In dit hoofdstuk wordt beschreven hoe een website kan worden ingericht voor de ontvangst van inkooporders in cXML-indeling en hoe u statusberichten van inkooporders naar inkooporganisaties of marketplaces kunt verzenden.

Inkooporderproces

In inkooptoepassingen worden goedgekeurde inkoopaanvragen omgezet in een of meer inkooporders. Een inkooporder is een formele aanvraag van een inkooporganisatie naar een leverancier om een contract uit te voeren.

cXML is slechts een mogelijke indeling voor de verzending van inkooporders. Andere veelgebruikte indelingen zijn e-mail, fax en EDI (X.12 Electronic Data Interchange). cXML is de beste indeling voor inkooporders omdat hiermee de orderverwerking gemakkelijk kan worden geautomatiseerd. Dankzij de goed gedefinieerde structuur van cXML kunnen systemen voor orderverwerking de elementen binnen een inkooporder gemakkelijk interpreteren. Met weinig of geen menselijke tussenkomst kunnen de toepasselijke gegevens binnen een inkooporder worden gerouteerd naar de gewenste afdeling, zoals Levering, Facturering en Verkoop.

Daarnaast kunnen met de cXML-methode voor orderrouting leverancierscookies (SupplierPartAuxiliaryID) en bijlagen bij inkooporders worden verzonden.

Wanneer u uw account configureert op een hub van een e-commercenetwerk, geeft u een URL op waarnaar alle cXML-inkooporders worden verzonden. Na ontvangst van een inkooporder, stuurt u deze door naar het interne systeem voor orderbeheer en voert de order op traditionele wijze uit. Vanuit uw website moet ook een Order Response-bevestiging naar de hub van het e-commercenetwerk worden teruggezonden, waardoor de inkoper op de hoogte wordt gesteld dat u de inkooporder met succes hebt ontvangen en geparseerd.

U hebt geen punchout-website nodig om cXML-inkooporders te kunnen ontvangen. Punchout en orderontvangst via cXML zijn twee verschillende zaken. De infrastructuur en de toepassingen die zijn vereist voor de ondersteuning van punchout zijn echter ook vereist voor de ontvangst van cXML-inkooporders.

Inkooporders ontvangen

Voor de overdracht van inkooporders worden twee typen cXML-documenten gebruikt. Vanuit de inkooptoepassing wordt een OrderRequest verzonden en u reageert met een OrderResponse. Deze documenten gaan via de hub van het e-commercenetwerk.

OrderRequest

Het document OrderRequest is analoog aan een inkooporder. In het volgende voorbeeld ziet u een OrderRequest voor een item:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cx.xml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US" payloadID="93369535150910.10.57.136"
timestamp="2000-08-03T08:49:11+07:00">
  <Header>
    <From>
      <Credential domain="AribaNetworkUserId">
        <Identity>admin@acme.com</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>114315195</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="AribaNetworkUserId">
        <Identity>sysadmin@ariba.com</Identity>
        <SharedSecret>abracadabra</SharedSecret>
      </Credential>
      <UserAgent>Ariba Network V1.1</UserAgent>
    </Sender>
  </Header>
  <Request>
    <OrderRequest>
      <OrderRequestHeader orderID="DO102880"
orderDate="2000-08-03T08:49:09+07:00" type="new">
        <Total>
```

```

    <Money currency="USD">4688.00</Money>
  </Total>
  <ShipTo>
    <Address isoCountryCode="US" addressID="1000467">
      <Name xml:lang="en">Acme, Inc.</Name>
      <PostalAddress name="default">
        <DeliverTo>John Q. Smith</DeliverTo>
        <DeliverTo>Buyers Headquarters</DeliverTo>
        <Street>123 Main Street</Street>
        <City>Mountain View</City>
        <State>CA</State>
        <PostalCode>94089</PostalCode>
        <Country>United States</Country>
      </PostalAddress>
      <Email name="default">john_smith@acme.com</Email>
      <Phone name="work">
        <TelephoneNumber>
          <CountryCode isoCountryCode="US">1
          </CountryCode>
          <AreaOrCityCode>800</AreaOrCityCode>
          <Number>5555555</Number>
        </TelephoneNumber>
      </Phone>
    </Address>
  </ShipTo>
  <BillTo>
    <Address isoCountryCode="US" addressID="12">
      <Name xml:lang="en">Acme Accounts Payable</Name>
      <PostalAddress name="default">
        <Street>124 Union Street</Street>
        <City>San Francisco</City>
        <State>CA</State>
        <PostalCode>94128</PostalCode>
        <Country isoCountryCode="US">US</Country>
      </PostalAddress>
      <Phone name="work">
        <TelephoneNumber>
          <CountryCode isoCountryCode="US">1
          </CountryCode>
          <AreaOrCityCode>415</AreaOrCityCode>
          <Number>6666666</Number>
        </TelephoneNumber>
      </Phone>
    </Address>
  </BillTo>
  <Shipping>
    <Money currency="USD">12.34</Money>
    <Description xml:lang="en-us">FedEx 2-day</Description>
  </Shipping>
  <Tax>

```

```

        <Money currency="USD">10.74</Money>
        <Description xml:lang="en">CA State Tax</Description>
    </Tax>
    <Payment>
        <PCard number="1234567890123456" expiration="2002-03-12"/>
    </Payment>
</OrderRequestHeader>
<ItemOut quantity="2" >
    <ItemID>
        <SupplierPartID>220-3165</SupplierPartID>
        <SupplierPartAuxiliaryID>E000028901</SupplierPartAuxiliaryID>
    </ItemID>
    <ItemDetail>
        <UnitPrice>
            <Money currency="USD">2344.00</Money>
        </UnitPrice>
        <Description xml:lang="en">Laptop Computer Notebook Pentium® II
processor w/AGP, 300 MHz, met 12.1" TFT XGA-scherm
        </Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">43171801</Classification>
        <URL>http://www.supplier.com/Punchout.asp</URL>
        <Extrinsic name="ExtDescription">Enhanced keyboard</Extrinsic>
    </ItemDetail>
    <Distribution>
        <Accounting name="DistributionCharge">
            <Segment type="Account" id="7720"
description="Office Supplies"/>
            <Segment type="CostCenter" id="610"
description="Engineering Management"/>
        </Accounting>
        <Charge>
            <Money currency="USD">4688.00</Money>
        </Charge>
    </Distribution>
</ItemOut>
</OrderRequest>
</Request>
</cXML>

```

OrderResponse

Met het document OrderResponse bevestigt u dat u de inkooporder hebt ontvangen en dat deze correct kan worden geparseerd. Dit is geen toezegging dat de inkooporder wordt uitgevoerd, maar uitsluitend een bevestiging van de ontvangst en de geldigheid van het cXML-document.

```
<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" payloadID="8/3/2000 8:49:30 PM@205.180.14.45"
timestamp="2000-08-03T08:49:30+07:00">
  <Response>
    <Status code="200" text="OK"/>
  </Response>
</cXML>
```

Bijlagen bij orders aanvaarden

Inkopers lichten hun inkooporders vaak toe met memo's, tekeningen of faxen. Zij kunnen bestanden van elk type aan cXML-inkooporders hechten met behulp van MIME (Multipurpose Internet Mail Extensions).

cXML bevat alleen referenties naar externe MIME-delen die binnen één meerdelige MIME-envelop worden verzonden (bij het cXML-document, bij een e-mailbericht of gezamenlijk per fax).

De bijlagen worden in de hubs van e-commercenetwerken ontvangen en kunnen worden doorgestuurd naar de leverancier of worden opgeslagen zodat deze on line kunnen worden opgehaald.

Zie "Verzending van bijlagen" op pagina 55 voor meer informatie over bijlagen bij inkooporders.

Bezoek de volgende websites voor meer informatie over de MIME-standaard:

www.hunnysoft.com/mime
www.rad.com/networks/1995/mime/mime.htm

Appendix A

Specificatie van cXML

In deze bijlage vindt u een beschrijving van de protocol- en gegevensindelingen van cXML (commerce eXtensible Markup Language). Deze bijlage bevat alle informatie die u nodig hebt om de ondersteunde transacties vanuit client- of vanuit serversysteemperspectief te implementeren. Zowel de protocolinteracties als de bedrijfsdocumenten binnen de transacties worden uitgebreid behandeld.

Het gebruik van cXML wordt tevens toegelicht aan de hand van praktijkvoorbeelden.

Deze bijlage bestaat uit de volgende secties:

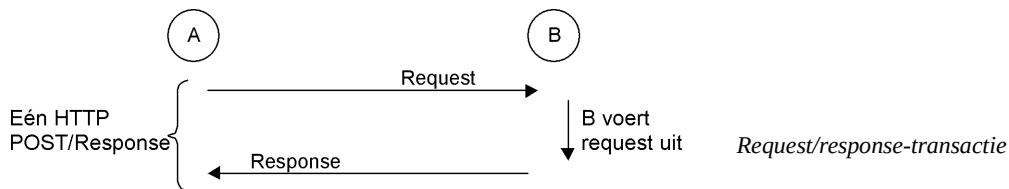
- Protocolspecificatie
- Basiselementen
- Profieltransactie
- Orderdefinities
- Punchout-transactie
- Latere statuswijzigingen
- Catalogusdefinities
- Definities inschrijvingenbeheer
- Definities voor ophalen van berichten

Protocolspecificatie

Er zijn twee communicatiemodellen voor cXML-transacties: request/response en one-way. Omdat de vereiste bewerkingen nauwkeurig zijn omschreven, zijn met deze twee modellen eenvoudige implementaties mogelijk. Beide modellen zijn vereist, omdat er situaties kunnen optreden waarbij een van de twee modellen niet voldoet.

Request/response-model

Request/response-transacties kunnen alleen via een HTTP-verbinding worden uitgevoerd. In de volgende afbeelding ziet u een voorbeeld van de stappen in een request/response-interactie tussen deelnemers A en B.



Deze transactie bestaat uit de volgende stappen:

1. A initieert een HTTP/1.x-verbinding met B op een vooraf bepaalde URL die het adres van B vertegenwoordigt.
2. A gebruikt een POST-bewerking om het cXML-document te verzenden via de HTTP-verbinding.
3. A wacht totdat een antwoord wordt teruggezonden via de HTTP-verbinding.
4. B bezit een HTTP/1.x-compatibele server die de HTTP-aanvraag verzendt naar de bron die is opgegeven in de URL die in stap 1 is gebruikt. Deze bron kan elke geldige locatie zijn die bekend is bij de HTTP-server van B, zoals een CGI-programma of een ASP-pagina.
5. De bron van B die in stap 4 is bepaald, leest de inhoud van het cXML-document en wijst de aanvraag aan de juiste handler voor die aanvraag toe.
6. De handler van B voor de cXML-aanvraag voert het in de aanvraag opgegeven werk uit en maakt een cXML-document op als antwoord.
7. B verzendt het cXML-antwoord naar A via de HTTP-verbinding die in stap 1 tot stand is gebracht.

8. A leest het cXML-antwoord en zendt dit terug aan het proces dat de aanvraag heeft geïnitieerd.
9. A sluit de HTTP-verbinding die in stap 1 tot stand is gebracht.

Het proces wordt vervolgens herhaald voor andere request/response-cycli.

Om de werkzaamheden in bovenstaande stappen te vereenvoudigen, zijn cXML-documenten onderverdeeld in twee afzonderlijke delen:

- Kop - bevat verificatie- en adresseringsgegevens.
- Request - of response-gegevens - bevatten een specifieke aanvraag of een specifiek antwoord, alsmede de gegevens die moeten worden doorgegeven.

Beide elementen worden in een overkoepelend envelopelement verzonden. In het volgende voorbeeld ziet u de structuur van een cXML-aanvraagdocument:

```
<cXML>
  <Header>
    Kopgegevens hier...
  </Header>
  <Request>
    Aanvraaggegevens hier...
  </Request>
</cXML>
```

In het volgende voorbeeld ziet u de structuur van een cXML-antwoorddocument:

```
<cXML>
  <Response>
    Response-gegevens hier...
  </Response>
</cXML>
```

De antwoordstructuur bevat geen kopelement. Dit is niet nodig omdat het antwoord altijd via dezelfde HTTP-verbinding wordt verzonden als de aanvraag.

XML-conventies

cXML maakt gebruik van elementen om afzonderlijke items, vaak eigenschappen in traditionele bedrijfsdocumenten, te beschrijven. Ook gegevens met duidelijke onderverdelingen en relaties tussen deze onderverdelingen, zoals een adres, worden met behulp van elementen beschreven.

In cXML wordt veel gebruikgemaakt van attributen.

In cXML wordt voor alle namen van elementen en attributen gebruikgemaakt van hele woorden die worden gescheiden met hoofdletters (niet met koppelstreepjes). Namen van elementen beginnen met een hoofdletter, namen van attributen beginnen met een kleine letter, bijvoorbeeld:

Elementen: Sender, Credential, Payment, ItemDetail
Attributen: version, payloadID, lineNumber, domain

cXML-envelop

Het envelopelement vormt de basis van de cXML-documentstructuur en bevat alle andere elementen. Het cXML-element is aanwezig in elke cXML-transactie.

In het volgende voorbeeld ziet u een volledig gespecificeerd cXML-element:

```
<cXML version="1.1.007" xml:lang="en-US"
  payloadID=1234567.4567.5678@test.ariba.com
  timestamp="1999-03-31T18:39:09-08:00">
```

cXML heeft de volgende attributen:

version (facultatief)	Geeft de versie van het cXML-protocol aan. Een XML-parser die een validatie uitvoert, kan het versieattribuut ook bepalen op basis van de DTD waarnaar wordt verwezen. Alle cXML-documenten dienen de versie echter expliciet te bevatten voor toepassingen die gebruikmaken van niet-validerende parsers.
xml:lang (facultatief)	De landinstelling die wordt gebruikt voor alle vrije tekst die in dit document wordt verzonden. De ontvanger moet antwoorden of informatie weergeven met dezelfde of een vergelijkbare landinstelling. Een client die xml:lang="en-UK" opgeeft in een aanvraag, kan bijvoorbeeld "en"-gegevens ontvangen.
payloadID	Een uniek nummer dat voor logdoeleinden wordt gebruikt om documenten aan te geven die mogelijk verloren zijn gegaan of waarbij problemen zijn opgetreden. Bij herhaalde pogingen dient dit nummer ongewijzigd te blijven. De aanbevolen implementatie is: datetime.process id.random number@hostname
timestamp	De datum en tijd waarop het bericht is verzonden, in ISO 8601-notatie. Bij herhaalde pogingen dient deze waarde ongewijzigd te blijven. De notatie is JJJJ-MM-DDTuu:mm:ss-uu:mm (bijvoorbeeld 2000-07-16T19:20:30+01:00).

Landinstelling opgegeven met xml:lang

Het attribuut `xml:lang` wordt ook weergegeven in de meeste vrije-tekstelementen (zoals `Description` en `Comments`). Hoewel de landinstelling volgens de XML-specificatie de standaardwaarde van een overkoepelend element kan krijgen, resulteren dergelijke standaardwaarden in inefficiënte query's van de documentstructuur. In cXML wordt geprobeerd om de identificatie van de landinstelling en de betreffende reeksen bijeen te houden.

`xml:lang`-attributen, die door het hele cXML-protocol heen worden weergegeven, hebben geen invloed op gegevens met een bepaalde notatie, zoals getallen, datums en tijdstippen. Zoals hierboven is beschreven voor het attribuut `timestamp`, worden dergelijke afzonderlijke waarden genoteerd op basis van het gegevenstype. Langere reeksen (en verwijzingen naar webpagina's) die niet bedoeld zijn voor automatische verwerking, kunnen een landspecifieke numerieke notatie of datumnotatie bevatten die overeenkomt met een nabijgelegen `xml:lang`-attribuut.

Tijdstippen en andere gegevenstypen

Het attribuut `timestamp` (en alle andere datums en tijden in cXML) dient de notatie te krijgen volgens de beperkte subset van ISO-8601 die is beschreven in de notitie van het World Wide Web Consortium (W3C) met de naam "Date and Time Formats" en die beschikbaar is op www.w3.org/TR/NOTE-datetime-970915.html.

Voor tijdregistratie zijn minimaal een volledige datum, uren, minuten en seconden vereist. Fracties van een seconde zijn facultatief. Voor dit protocol zijn tijdstippen uitgedrukt in plaatselijke tijd met een tijdzone-UTC-offset (Coordinated Universal Time, ook Greenwich Mean Time genoemd). De tijdzoneaanduiding "Z" is niet toegestaan.

2000-04-14T013:36:00-08:00 komt bijvoorbeeld overeen met 14 april 2000, 13:36 uur, U.S. Pacific Standard Time.

Meer informatie over de notatie van datum, tijd en andere gegevenstypen vindt u op:

- De XML Data Types Reference-site van Microsoft, msdn.microsoft.com/xml/reference/schema/datatypes.asp
- Het oorspronkelijke XML Data-voorstel aan het World Wide Web Consortium (W3C), www.w3c.org/TR/1998/NOTE-XML-data-0105

Inpaklagen

Het cXML-element vormt de inhoud van een normaal XML-document. Een document kan als volgt beginnen:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US"
  payloadID="0c300508b7863dccb_13550"
  timestamp="2000-01-09T01:36:05-08:00">
  ...
```

Dit document wordt gewoonlijk verzonden via HTTP met het MIME-mediumtype text/xml en een charset-parameter die overeenkomt met de codering in het document. Omdat HTTP volledig 8-bits is, kan elke codering die wordt ondersteund door de ontvangende parser, worden gebruikt zonder content-transfer-codering zoals base64 of quoted-printable. Alle XML-parsers ondersteunen de UTF-8-codering, die alle Unicode-tekens bevat. Toepassingen dienen daarom hiervan gebruik te maken bij het verzenden van cXML-documenten.

Opmerking: Volgens RFC 2376, "XML Media Types", overschrijft de MIME charset-parameter alle codering die is opgegeven in de XML-declaratie. Verder is de standaardcodering voor het mediumtype text/xml us-ascii, en niet UTF-8, zoals is aangegeven in sectie 4.3.3. van de XML-specificatie. Voor de duidelijkheid dient de XML-declaratie van cXML-documenten een expliciete codering te bevatten. In MIME-enveloppen dient een bijbehorende charset-parameter te worden gebruikt voor het mediumtype text/xml of application/xml.

Een HTTP-verzending van een cXML-document kan de volgende MIME- en HTTP-koppen bevatten:

```
POST /cXML HTTP/1.0
Content-type: text/xml; charset="UTF-8"
Content-length: 1862
Accept: text/html, image/gif, image/jpeg, *, q=.2, /*; q=.2
User-Agent: Java1.1
Host: localhost:8080
Connection: Keep-Alive

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

Verzending van bijlagen

Bij het verzenden van een OrderRequest die naar externe bestanden verwijst, kunnen de bestanden waarnaar wordt verwezen zich op een server bevinden die toegankelijk is voor de leverancier of kunnen deze bestanden met het cXML-document worden verzonden. Voor deze tweede optie is het gebruik van een meerdelige MIME-envelop vereist. Een van de cXML-vereisten voor deze envelop (naast de basiseisen die zijn beschreven in RFC 2046, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types") is het opnemen van content-ID-koppen in elk bijgevoegd bestand.

In het volgende voorbeeld ziet u de vereiste opzet van een cXML-document met een bijgevoegde JPEG-afbeelding (zonder bovenstaande HTTP-koppen):

```
POST /cXML HTTP/1.0
Content-type: multipart/mixed; boundary=unieke waarde

--unieke waarde
Content-type: text/xml; charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
...
--unieke waarde
Content-type: image/jpeg
Content-ID: <uniqueCID@cxml.org>
...
--unieke waarde--
```

Deze opzet is tevens alles wat een ontvangende MIME-parser moet kunnen verwerken. Toepassingen die gebruikmaken van het mediumtype dat is beschreven in RFC 2387, "The Mime Multipart/Related Content-type", zullen veel meer informatie ontvangen als de opzet wordt uitgebreid.

```
POST /cXML HTTP/1.0
Content-type: multipart/related; boundary=unieke waarde;
  type="text/xml"; start=<uniqueCIDmain@cxml.org>

--unieke waarde
Content-type: text/xml; charset="UTF-8"
Content-ID: <uniqueCIDmain@cxml.org>

<?xml version="1.0" encoding="UTF-8"?>
...
--unieke waarde
Content-type: image/jpeg
Content-ID: <uniqueCID@cxml.org>
...
--unieke waarde--
```

Ontvangende MIME-parsers die het mediumtype `multipart/related` niet kunnen verwerken, moeten de bovenstaande twee voorbeelden op dezelfde manier behandelen. Elk onderdeel van de MIME-verzending kan daarnaast een content-transfer-codering bezitten en gebruikmaken van deze codering. Deze toevoeging is niet noodzakelijk voor HTTP-verzending. De koppen Content-description en Content-disposition zijn facultatief binnen het cXML-protocol maar bieden nuttige informatie.

Zie "Attachment" op pagina 74 voor meer informatie over het koppelen van externe bestanden aan inkooporders.

Header

Het element Header bevat adresserings- en verificatiegegevens. Het Header-element blijft hetzelfde, ongeacht de specifieke Request of Response in de inhoud van het cXML-bericht. Toepassingen hebben de identiteit van de aanvrager nodig, maar hebben geen validatie nodig dat de ID-gegevens correct zijn.

In het volgende voorbeeld ziet u het Header-element:

```
<Header>
  <From>
    <Credential domain="AribaNetworkUserId">
      <Identity>admin@acme.com</Identity>
    </Credential>
  </From>
  <To>
    <Credential domain="DUNS">
      <Identity>012345678</Identity>
    </Credential>
  </To>
  <Sender>
    <Credential domain="AribaNetworkUserId">
      <Identity>sysadmin@ariba.com</Identity>
      <SharedSecret>abracadabra</SharedSecret>
    </Credential>
    <UserAgent>Ariba Network 1.1</UserAgent>
  </Sender>
</Header>
```

De From- en To-elementen zijn gelijk aan From en To in SMTP-mailberichten – het zijn de logische bron en bestemming van de berichten. Sender is de partij die de HTTP-verbinding opent en het cXML-document verzendt.

Sender bevat het Credential-element, waarmee de ontvangende partij de verzendende partij kan verifiëren. Zo is een krachtige verificatie mogelijk zonder infrastructuur waarin van begin tot einde gebruik wordt gemaakt van digitale certificaten en publieke sleutels.

Aanvankelijk zijn Sender en From hetzelfde. Als het cXML-document echter via de hubs van het e-commercenetwerk wordt verzonden, wordt het Sender-element gewijzigd in de huidige verzendende partij.

From

Dit element geeft de oorsprong van de cXML-aanvraag aan. Facultatief kan het meer dan één Credential-element bevatten, waardoor aanvragers zichzelf via meerdere identificatiemethoden bekend kunnen maken. Dit gebruik van meerdere referenties is vergelijkbaar met het verzenden van zowel een SMTP- als een X.400-adres in een e-mailbericht.

To

Dit element geeft de bestemming van de cXML-aanvraag aan. Net als het From-element, kan het meerdere referenties bevatten ter identificatie van het eindpunt.

Sender

Met dit element kan de ontvangende partij de partij die de HTTP-verbinding heeft geopend, identificeren en verifiëren. Dit element bevat een krachtiger verificatie-Credential dan de From- of To-elementen omdat de ontvangende partij moet verifiëren wie de werkzaamheden aanvraagt.

Credential

Dit element bevat de identificatie- en verificatiewaarden die worden gebruikt in cXML-berichten.

Credential heeft de volgende attributen:

domain	Geeft het type referentie aan. Documenten kunnen door dit attribuut verschillende soorten referenties voor verschillende verificatiedomeinen bevatten. In berichten die via het Ariba-netwerk worden verzonden, is het domein bijvoorbeeld meestal AribaNetworkUserId of DUNS.
type (facultatief)	Aanvragen aan of van een marketplace geven zowel de marketplace als het deelnemende bedrijf aan in de Credential-elementen From of To. In dit geval maakt de referentie voor de marketplace gebruik van het attribuut type, dat is ingesteld op de waarde "marketplace".

Credential bevat een Identity-element en een facultatief SharedSecret- of DigitalSignature-element. Het Identity-element geeft aan wie door Credential wordt vertegenwoordigd en de facultatieve verificatie-elementen verifiëren de identiteit van de partij.

Het SharedSecret-element wordt gebruikt wanneer Sender een combinatie van gebruikersnaam en wachtwoord bevat die door de aanvrager wordt herkend.

Het DigitalSignature-element kan worden gebruikt als de twee partijen overeenstemming hebben bereikt over een gemeenschappelijk certificaatindeling en certificaatuitgever. Het attribuut type in een DigitalSignature-element geeft aan welk type certificaat wordt gebruikt.

Opmerking: Gebruik geen verificatie-elementen in documenten die via one-way-communicatie worden verzonden. Bij dit transport worden documenten via browsers van gebruikers verzonden en kunnen gebruikers dus de documentbron zien (inclusief de Credential-elementen).

Request

Clients verzenden aanvragen voor bewerkingen. Maximaal één Request-element is toegestaan per cXML-envelopelement, hetgeen de serverimplementaties vereenvoudigt omdat bij het lezen van cXML-documenten geen demultiplexing hoeft te worden uitgevoerd. Het Request-element kan vrijwel elk type XML-gegevens bevatten.

Request heeft het volgende attribuut:

deploymentMode (facultatief)	Geeft aan of de aanvraag een testaanvraag of een productieaanvraag is. Toegestane waarden zijn "production" (standaardwaarde) of "test".
--	--

Response

Servers verzenden antwoorden om de resultaten van bewerkingen door te geven aan clients. Omdat de resultaten van bepaalde aanvragen mogelijk geen gegevens bevatten, kan het Response-element facultatief alleen een Status-element bevatten. Een Response-element kan tevens alle gegevens op toepassingsniveau bevatten. Voor de punchout-scenario's is dit een PunchOutSetupResponse-element.

Status

Dit element geeft het succes of de mislukking van een verzoekbewerking door.

Status heeft de volgende attributen:

code	De statuscode van de aanvraag. Deze komt overeen met het HTTP-statuscodemodel. Zo vertegenwoordigt 200 bijvoorbeeld een geslaagde aanvraag.
text	De tekst van het statusbericht. Deze tekst is een hulpmiddel voor gebruikers bij het lezen van logbestanden. De tekst bestaat uit Engelse standaardreeksen.
xml:lang (facultatief)	De taal van de gegevens in het Status-element. Optioneel voor compatibiliteit met 1.1. Is mogelijk vereist in toekomstige versies van cXML.

De attributen van het Status-element geven aan wat er met de aanvraag is gebeurd.

Het Status-element kan alle gegevens bevatten die de aanvrager nodig heeft. Voor de statuscode 200/OK zijn er mogelijk geen gegevens. Voor de statuscode 500/Internal Server Error verdient het sterk aanbeveling om de feitelijke XML-parserfout of toepassingsfout op te geven. Deze fout vergemakkelijkt het debuggen en interoperabiliteitstesten vanaf één zijde.

Servers mogen geen extra response-elementen toevoegen (bijvoorbeeld een PunchOutSetupResponse-element) als de statuscode niet in het 200-bereik ligt (bijvoorbeeld 200/OK).

De HTTP 1.1-specificatie bevat een groot aantal statuscodes die ongeschikt zijn voor cXML. Omdat cXML zich in de meeste gevallen in de laag boven HTTP bevindt, worden veel fouten (zoals 404/Not Found) verwerkt tijdens het transport. De meest waarschijnlijke statuscodes zijn 200/OK en 500/Internal Server Error. Validatiefouten bij het parseren van een Request-document resulteren normaal gesproken in een transportfout, zoals de HTTP-fout 400/Bad Request.

De volgende tabel bevat andere HTTP-codes die kunnen worden gebruikt.

cXML bevat weinig niet-HTTP-statuscodes:

- 550 – Kan volgende cXML-server niet bereiken om een transactie te voltooien waarvoor upstream-verbindingen nodig zijn. Deze code kan door een tussenhub worden teruggezonden wanneer een leverancierssite onbereikbaar is. (Als upstream-verbindingen zijn voltooid, moeten tussenhubs fouten rechtstreeks aan de client terugzenden.)
- 551 – Kan verzoek niet doorzenden wegens onjuiste configuratie van leverancier. De verificatie van een tussenhub bij een leverancier is bijvoorbeeld mislukt. Clients kunnen deze fout niet herstellen maar het is mogelijk dat deze fout is opgelost voordat de nieuwe poging van de client start.

- 560 – Tijdelijke serverfout. Een server kan buiten gebruik zijn wegens onderhoud. Clients kunnen het later opnieuw proberen.

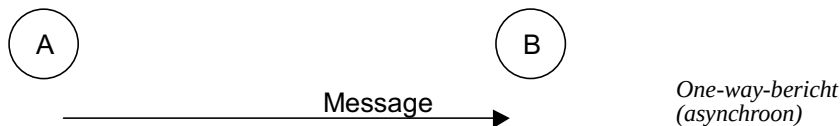
Status	Standaardtekst	Betekenis
200	OK	De server kon deze Request uitvoeren, maar de teruggezonden Response kan toepassingswaarschuwingen of -fouten bevatten.
201	Accepted	Mogelijk is niet de gehele verwerking voltooid. Zoals vermeld in "StatusUpdateRequest" op pagina 88 kan de client op een later tijdstip StatusUpdate-transacties verwachten als deze status wordt teruggezonden als antwoord op een OrderRequest.
204	No Content	Alle Request-gegevens zijn geldig en herkend. De server bezit geen Respons-gegevens van het gevraagde type. In een PunchOutOrderMessage geeft deze status aan dat de punchout-sessie is beëindigd zonder wijzigingen in het winkelwagentje (of in de clientaanvraag).
400	Bad Request	Aanvraag is correct geparseerd maar wordt niet door de server geaccepteerd.
401	Unauthorized	Referenties in Request (Sender-element) zijn niet herkend door de server.
402	Payment Required	Deze Request dient een volledig Payment-element te bevatten.
403	Forbidden	De gebruiker heeft onvoldoende bevoegdheden om deze Request uit te voeren.
406	Not Acceptable	Gelijk aan code 400: Request is niet geaccepteerd door de server, hoewel deze correct is geparseerd.
409	Conflict	Het bewerkingsverzoek (bijwerking) kon niet worden uitgevoerd wegens de huidige status van de server of de interne gegevens van de server. Een toekomstige, identieke Request kan mogelijk wel slagen, in het bijzonder wanneer de uitvoering van een andere bewerking is voltooid.
412	Precondition Failed	Er is niet voldaan aan een eerste vereiste van de Request (bijvoorbeeld een punchout-sessie die correct is voor een PunchOutSetupRequest edit). Deze status geeft meestal aan dat de client een deel van een eerdere verzending van een server heeft genegeerd (bijvoorbeeld het attribuut operationAllowed van een PunchOutOrderMessageHeader).
417	Expectation Failed	Request impliceerde een bronvoorwaarde waaraan niet werd voldaan. Een voorbeeld is een SupplierData-Request waarin informatie wordt gevraagd over een leverancier die niet bekend is bij de server. Deze status kan wijzen op verdwenen gegevens op de client of de server.
500	Internal Server Error	Server kon de Request niet voltooien.
501	Not Implemented	De server implementeert de specifieke Request niet, bijvoorbeeld een PunchOutSetupRequest, of de gevraagde bewerking wordt niet ondersteund. Deze status betekent meestal dat de client het profiel van de server heeft genegeerd.

Wanneer cXML-clients niet-herkende codes ontvangen, moeten de clients deze verwerken op basis van hun klasse. Oudere clients moeten daarom alle nieuwe 2xx-codes als 200, 4xx-codes als 400 en 5xx-codes als 500 behandelen. Hierdoor kunnen het cXML-protocol en serverspecifieke codes verder worden uitgebreid zonder verlies van interoperabiliteit.

One-way-model (asynchroon)

In tegenstelling tot request/response-transacties, zijn one-way-berichten niet beperkt tot het HTTP-transport.

One-way-berichten zijn bedoeld voor situaties waarin een HTTP-kanaal (een synchrone bewerking van het type request/response) niet geschikt is. De volgende afbeelding bevat een voorbeeld van communicatie tussen A en B met behulp van berichten in plaats van de request/response-transactie.



In dit geval is een mogelijk scenario:

1. A deelt een cXML-document in en codeert dit in een transport dat door B kan worden begrepen.
2. A verzendt het document met behulp van het bekende transport. A wacht niet actief op een antwoord van B (en kan dit ook niet doen).
3. B ontvangt het cXML-document en decodeert dit vanuit de transportstroom.
4. B verwerkt het document.

In het one-way-model hebben A en B *geen* expliciete request/response-cyclus. Tussen one-way-berichten kunnen bijvoorbeeld berichten van andere partijen arriveren en kan andere communicatie plaatsvinden.

Om een one-way-transactie volledig te specificeren, dient ook het transport dat voor het bericht wordt gebruikt, te worden gedocumenteerd. Voor de cXML-transacties die gebruikmaken van de one-way-methode, worden het transport en de codering gespecificeerd. Een veelvoorkomend voorbeeld van een transactie waarbij gebruik wordt gemaakt van one-way is PunchOutOrderMessage.

De structuur van one-way-berichten is vergelijkbaar met de structuur van het request/response-model:

```
<cXML>
  <Header>
    Kopgegevens hier...
  </Header>
  <Message>
    Message-gegevens hier...
  </Message>
</cXML>
```

Het Header-element wordt op dezelfde manier behandeld als bij het request/response-model. Ook het cXML-element is gelijk aan het hierboven beschreven element. De eenvoudigste manier om het verschil tussen een one-way-bericht en een request/response-bericht te zien, is aan de hand van de aanwezigheid van het Message-element (in plaats van het Request- of Response-element). In de volgende sectie wordt het Message-element uitgebreider besproken.

Message

Dit element bevat alle gegevens op inhoudsniveau in een cXML-bericht. Het kan een optioneel Status-element bevatten, dat gelijk is aan het Status-element in een Response-element - dit wordt gebruikt in berichten die logische antwoorden op aanvraagberichten vormen.

Message heeft de volgende attributen:

deploymentMode (facultatief)	Geeft aan of de aanvraag een testverzoek of een productieverzoek is. Toegestane waarden zijn "productie" (standaardwaarde) of "test".
inReplyTo (facultatief)	Geeft aan op welke Message deze Message een antwoord is. De inhoud van het attribuut inReplyTo is de payloadID van een Message die eerder is ontvangen. Dit kan worden gebruikt om een tweerichtingscommunicatie met veel berichten op te zetten.

Het attribuut inReplyTo kan ook verwijzen naar de payloadID van een eerder Request- of Response-document. Wanneer een request/response-transactie een communicatie initieert via meerdere one-way-interacties, kan het eerste bericht de payloadID bevatten van de meest recente relevante Request of Response die in de andere richting is verzonden. Zo kan een Message met een PunchOutOrderMessage bijvoorbeeld een inReplyTo-attribuut bevatten met de payloadID van de PunchOutSetupRequest waarmee de punchout-sessie is gestart. (De BuyerCookie in de punchout-documenten vervult een vergelijkbare functie als dit gebruik van het attribuut inReplyTo.)

Transportopties

Er zijn twee algemeen gebruikte transporten voor one-way-berichten: HTTP en URL-Form-Encoding. Dit zijn slechts twee van de goed gedefinieerde transporten die op het moment in gebruik zijn. In de toekomst worden mogelijk ook andere transporten ondersteund.

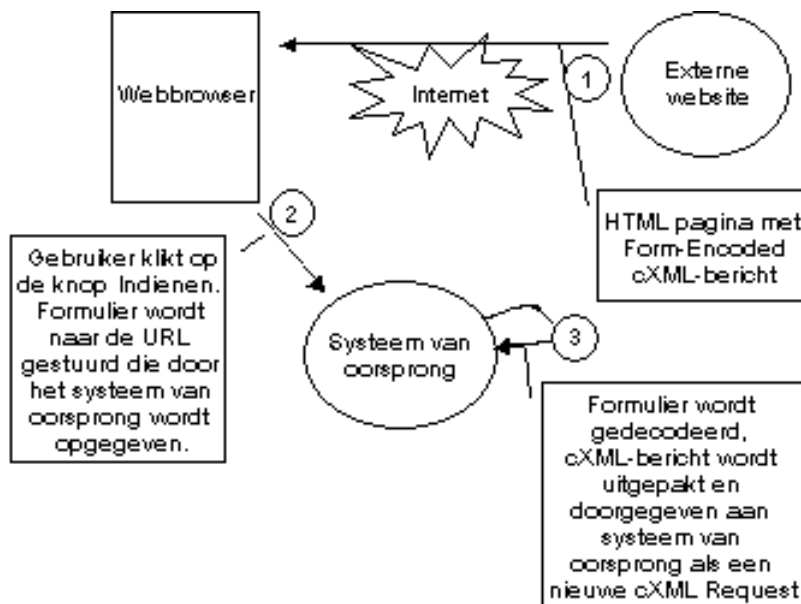
HTTP

HTTP wordt gebruikt voor one-way-communicatie, zodat inkooptoepassingen gegevens kunnen ophalen. Het transactietype dat gebruikmaakt van one-way-HTTP-communicatie is `GetPendingRequest`, dat wordt besproken op pagina 102.

URL-Form-Encoding

Dit transport kan het beste worden begrepen door te bekijken hoe de `PunchOutOrderMessage`-transactie wordt uitgevoerd. URL-Form-Encoding maakt integratie van een externe website en inkooptoepassingen mogelijk. Ook vervalt met URL-Form-Encoding de noodzaak van een direct via Internet toegankelijke listeningserver in het systeem van de inkoper.

Het cXML-bericht `PunchOutOrderMessage` wordt door de externe website niet rechtstreeks naar de inkooptoepassing verzonden, maar wordt als verborgen HTML-formulieveld gecodeerd en naar de URL in het `BrowserFormPost`-element van de `PunchOutSetupRequest` gestuurd. Als de gebruiker op Check Out klikt, stuurt de website de gegevens als HTML Form Submit. In het volgende diagram ziet u wat er gebeurt:



De semantiek van het inpakken en uitpakken wordt hieronder beschreven.

Formulieren inpakken

Het PunchOutOrderMessage-document wordt URL-gecodeerd (volgens de HTTP-specificatie) en wordt toegewezen aan een verborgen veld in het formulier met de naam cXML-urlencoded. De METHOD POST en een ACTION die bestaat uit de URL die is doorgegeven in het BrowserFormPost-element van de PunchOutSetupRequest, worden aan het HTML Form-element toegewezen. Voorbeeld:

```
<FORM METHOD=POST
  ACTION="http://workchairs.com:1616/punchoutexit">
  <INPUT TYPE=HIDDEN NAME="cXML-urlencoded"
    VALUE="URL-Encoded PunchOutOrderMessage document">
  <INPUT TYPE=SUBMIT VALUE="Proceed">
</FORM>
```

Aanvullende HTML-codes op de pagina kunnen bovenstaand fragment bevatten om de inhoud van het winkelwagentje in detail te beschrijven.

Opmerking: Wanneer het veld cXML-urlencoded door een webserver wordt verzonden, is dit veld nog niet URL-gecodeerd. De codering is alleen vereist wanneer het formulier wordt ingediend door een webbrowser (wanneer de gebruiker in bovenstaand voorbeeld op Check Out klikt). Webrowsers voldoen zelf aan deze eis. De webserver hoeft alleen de veldwaarde HTML-coderen, waarbij aanhalingstekens en andere speciale tekens worden voorafgegaan door een backslash (/), zodat het formulier correct aan de gebruiker wordt weergegeven.

De naam cXML-urlencoded is niet hoofdlettergevoelig.

Voor cXML-urlencoded-gegevens kan de ontvangende parser geen charset-parameter aannemen buiten de standaardwaarde voor het mediumtype, text/xml. In een HTTP POST zijn geen tekencoderingsgegevens opgenomen voor de verzonden gegevens. De ontvangende webserver kan de codering van de HTML-pagina met het verborgen veld niet bepalen. Voor cXML-documenten die op deze manier wordt doorgezonden, moet daarom de tekencodering us-ascii worden gebruikt. Alle tekens in het XML-brondocument (inclusief de tekens die "URI-gecodeerd" zijn als %XX") moeten daarom tot de tekenset "us-ascii" behoren. Andere Unicode-symbolen kunnen worden gecodeerd met behulp van tekeneenheden in dit brondocument.

Base64-codering

Het verborgen veld cXML-base64 vereenvoudigt de verwerking van internationale documenten. Dit veld moet in plaats van het verborgen veld cXML-urlencoded worden gebruikt in cXML-documenten met tekens die niet tot "us-ascii" behoren. De semantiek van dit alternatief is nagenoeg gelijk, maar het gehele document wordt tijdens het transport base64-gecodeerd en niet HTML-gecodeerd voor de browser of URL-gecodeerd voor de ontvangende webserver. Base64-codering is beschreven in RFC 2045, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

De naam cXML-base64 is niet hoofdlettergevoelig.

Bij base64-codering vanaf de zijde van de leverancier via de browser naar de ontvangende webserver bij de klant, blijft de oorspronkelijke tekencodering van een cXML-document behouden. Hoewel het ontvangen document geen charset-parameter bevat, kan het gedecodeerde document (nadat de overdrachts codering is verwijderd) als mediumtype application/xml worden behandeld. Met deze codering kan de ontvangende parser een eventueel encoding-attribuut honoreren dat in de XML-declaratie is opgegeven. Voor dit veld is de standaardtekencodering UTF-8 (zoals voor alle application/xml-documenten).

Een van deze verborgen velden (cXML-urlencoded of cXML-base64) moet zijn opgenomen in de gegevens die naar de inkooptoepassing worden verzonden. Hoewel de ontvangers eerst naar cXML-base64 moeten zoeken in de gegevens, is het onnodig om beide velden te verzenden.

Formulieren uitpakken en verwerken

De inkooptoepassing, die eerder de juiste URL heeft verschaft, ontvangt een HTML-formulier POST met de hierboven beschreven formuliergegevens. De verwerker van het formulier POST zoekt eerst naar de variabele cXML-base64, pakt de waarde uit en voert een base64-decodering van de inhoud uit. Als dit veld geen deel uitmaakt van de gegevens, zoekt de verwerker van het formulier POST naar de variabele cXML-urlencoded, pakt het URL-gecodeerde cXML-bericht uit en voert een URL-decodering uit. De gedecodeerde inhoud van het veld wordt vervolgens verwerkt alsof dit is ontvangen via een normale HTTP-request/response-cyclus.

Het opgenomen mediumtype van het document na decodering kan variëren, met verschillende mogelijke tekencoderingen:

- De variabele `cXML-urlencoded` is van het mediumtype `text/xml` zonder `charset`-attribuut. Deze variabele wordt daarom beperkt tot de tekencodering `us-ascii`. De ontvangende parser moet eventuele `encoding`-attributen in de XML-declaratie van het cXML-document negeren omdat de browser de codering mogelijk heeft gewijzigd.
- De variabele `cXML-base64` is van het mediumtype `application/xml` en kan dus elke tekencodering bezitten (aangegeven in het attribuut `encoding` van de opgenomen XML-declaratie, indien aanwezig).

Het belangrijkste verschil tussen deze transactie en een normale request-response-transactie is dat er geen antwoord kan worden gegenereerd omdat er geen HTTP-verbinding is die voor de verzending van het antwoord kan worden gebruikt.

Basiselementen

De volgende eenheden en elementen worden in de gehele cXML-specificatie gebruikt. De meeste definities zijn basistermen die worden gebruikt voor het beschrijven van zakelijke documenten op een hoger niveau. De gemeenschappelijke type-eenheden en elementen die worden gebruikt voor het beschrijven van objecten op een lager niveau, zijn hier gedefinieerd.

Type-eenheden

De meeste definities zijn afkomstig uit de notitie XML-Data die is ingediend bij het World Wide Web Consortium (W3C). Enkele eenheden op een hoger niveau die hier ook zijn gedefinieerd, zijn niet afkomstig uit XML-Data. Deze typen worden ook besproken in "cXML-envelop" op pagina 52.

isoLangCode

Een ISO-taalcode uit de ISO 639-standaard.

isoCountryCode

Een ISO-landcode uit de ISO 3166-standaard.

xmlLangCode

Een taalcode zoals gedefinieerd in de XML 1.0-specificatie (op www.w3.org/TR/1998/REC-xml-19980210.html). Over het algemeen bevat deze een ISO 639-taalcode en (optioneel) een ISO 3166-landcode, gescheiden door een koppelstreepje. In tegenstelling tot de volledige aanbeveling voor XML, mogen IANA- of eigen taalcodes niet worden gebruikt in cXML. IANA- en eigen subcodes zijn toegestaan, maar dienen na een geldige ISO 3166-landcode te worden geplaatst.

De aanbevolen notatie voor de cXML-taalcodes is xx[-YY[-zzz]*]?. Hierbij is xx een ISO 639-taalcode, YY een ISO 3166-landcode en zz een IANA of eigen subcode voor de betreffende taal. Nogmaals, het gebruik van de landcode verdient altijd aanbeveling. De conventie is dat de taalcode in kleine letters en de landcode in hoofdletters wordt geschreven. Voor een juiste afstemming van de codes is dit niet vereist.

unitOfMeasure

UnitOfMeasure geeft aan hoe het product is verpakt of wordt verzonden. Deze code moet overeenstemmen met UN/CEFACT Unit of Measure Common Codes. Zie www.unece.org/cefact voor een lijst met UN/CEFACT-codes.

URL

Een URL (Uniform Resource Locator) zoals gedefinieerd in de HTTP/1.1-standaard.

Basiselementen

Deze elementen, die in de gehele specificatie worden gebruikt, variëren van algemeen, zoals Name en Extrinsic, tot specifiek, zoals Money.

Profieltransactie

De ProfileRequest en ProfileResponse- documenten moeten worden ondersteund door cXML 1.1-serverimplementaties. Deze transactie kan worden gebruikt om de servermogelijkheden op te halen, zoals de ondersteunde cXML-versie, transacties en opties bij deze transacties.

Het antwoord dient alle aanvragen te bevatten die door een bepaalde website worden ondersteund – niet noodzakelijkerwijs alle aanvragen die door het bedrijf worden ondersteund. Leveranciers die OrderRequest-documenten kunnen ontvangen en verschillende berichten kunnen verzenden of request/response-transacties kunnen initiëren, beschrijven hun OrderRequest-ondersteuning in de profieltransactie.

De profieltransactie kan worden gebruikt om binnen het cXML-protocol een testsignaal naar een server te verzenden.

ProfileRequest

Dit element heeft geen inhoud. Het wordt eenvoudigweg met behulp van de Header naar de juiste cXML-server gerouteerd. De server antwoordt met één ProfileResponse, zoals hieronder beschreven. De enige dynamische onderdelen van dit antwoord zijn de attributen payloadId- en timestampvan het cXML-element zelf. In dit geval hoeft een leverancier geen antwoord met verschillende landinstellingen te geven.

Een voorbeeld van een Request van dit type is:

```
<Request>  
  <ProfileRequest />  
</Request>
```

ProfileResponse

Dit element bevat een lijst met ondersteunde transacties, hun locaties en de eventuele ondersteunde opties. Een mogelijk ProfileResponse zonder gedefinieerde opties is:

```
<ProfileResponse effectiveDate="2001-03-03T12:13:14-05:00">
  <Option name="Locale">1</Option>
  ...
  <Transaction requestName="PunchOutSetupRequest">
    <URL>http://www.workchairs.com/cXML/PunchOut.asp</URL>
    <Option name="operationAllowed">create inspect</Option>
    <Option name="dynamic pricing">0</Option>
    ...
  </Transaction>
  ...
</ProfileResponse>
```

Een meer waarschijnlijke ProfileResponse van een huidige leverancier zou er als volgt uitzien:

```
<ProfileResponse effectiveDate="2000-01-01T05:24:29-08:00">
  <Transaction requestName="OrderRequest">
    <URL>http://workchairs.com/cgi/orders.cgi</URL>
  </Transaction>
  <Transaction requestName="PunchOutSetupRequest">
    <URL>http://workchairs.com/cgi/PunchOut.cgi</URL>
  </Transaction>
</ProfileResponse>
```

ProfileResponse heeft het volgende attribuut:

effectiveDate	De datum en tijd waarop deze diensten beschikbaar zijn geworden. Datums mogen niet in de toekomst liggen.
----------------------	---

Option

Waarde voor een gedefinieerde optie (voor de gehele dienst of voor een specifieke transactie). Er zijn nog geen opties gedefinieerd.

Option heeft het volgende attribuut:

name	De naam van deze optie. Dit attribuut mag niet rechtstreeks zichtbaar zijn (omdat het profiel bedoeld is voor automatische verwerking).
-------------	---

Transaction

De omschrijving van een transactie die door deze dienst wordt ondersteund. De profieldefinitie geeft nu de locaties aan waarnaar specifieke aanvragen moeten worden verzonden. In toekomstige versies van cXML zullen Option-definities worden toegevoegd en zal de profielinformatie worden uitgebreid met meer informatie over ondersteunde aanvragen.

Het Transaction-element dient een URL-element te bevatten.

Transaction heeft het volgende attribuut:

requestName	Een specifieke aanvraag die deze server op de opgegeven URL accepteert. Mogelijke waarden:	
	ProfileRequest PunchOutSetupRequest GetPendingRequest SupplierListRequest SupplierDataRequest	OrderRequest StatusUpdateRequest SubscriptionListRequest SubscriptionContentRequest

Orderdefinities

De cXML-orderdocumenten zijn OrderRequest en een algemeen antwoord. OrderRequest is analoog aan een inkooporder. Het antwoord is een bevestiging dat de leverancier de inkooporder heeft ontvangen. Dit is geen toezegging dat de inkooporder wordt uitgevoerd, maar uitsluitend een bevestiging dat de order correct is ontvangen.

OrderRequest

In het volgende voorbeeld ziet u de structuur van het OrderRequest-element:

```
<OrderRequest>
  <OrderRequestHeader ... >
    ...
  </OrderRequestHeader>
  <ItemOut ... >
    ...
  </ItemOut>
  <ItemOut ... >
    ...
  </ItemOut>
</OrderRequest>
```

OrderRequestHeader

In het volgende voorbeeld ziet u alle details van een OrderRequestHeader:

```
<OrderRequestHeader orderID="DO1234"
  orderDate="1999-03-12T13:30:23+8.00"
  type="new"
  requisitionID="R1234">
  <Total>
    <Money currency="USD">12.34</Money>
  </Total>
  <ShipTo>
    <Address>
      <Name xml:lang="en">Acme Corporation</Name>
      <PostalAddress name="Headquarters">
        <DeliverTo>Joe Smith</DeliverTo>
        <DeliverTo>Mailstop M-543</DeliverTo>
        <Street>123 Anystreet</Street>
        <City>Sunnyvale</City>
        <State>CA</State>
        <PostalCode>90489</PostalCode>
        <Country isoCountryCode="US">USA</Country>
      </PostalAddress>
    </Address>
  </ShipTo>
  <BillTo>
    <Address>
      <Name xml:lang="en">Acme Corporation</Name>
      <PostalAddress name="Finance Building">
        <Street>124 Anystreet</Street>
        <City>Sunnyvale</City>
        <State>CA</State>
        <PostalCode>90489</PostalCode>
        <Country isoCountryCode="US">USA</Country>
      </PostalAddress>
    </Address>
  </BillTo>
  <Shipping>
    <Money currency="USD">12.34</Money>
    <Description xml:lang="en-US">FedEx 2-day</Description>
  </Shipping>
  <Tax>
    <Money currency="USD">12.34</Money>
    <Description xml:lang="en">CA State Tax</Description>
  </Tax>
  <Payment>
    <PCard number="1234567890123456" expiration="1999-03-12"/>
  </Payment>
  <Contact role="purchasingAgent">
    <Name xml:lang="en-US">Mr. Smart E. Pants</Name>
```

```
<Email>sepants@acme.com</Email>
<Phone name="Office">
  <TelephoneNumber>
    <CountryCode isoCountryCode="US">1</CountryCode>
    <AreaOrCityCode>800</AreaOrCityCode>
    <Number>555-1212</Number>
  </TelephoneNumber>
</Phone>
</Contact>
<Comments xml:lang="en-US">
  Elke gewenste tekst in correcte XML.
</Comments>
<Followup>
  <URL>http://acme.com/cgi/orders.cgi</URL>
</Followup>
</OrderRequestHeader>
```

OrderRequestHeader heeft de volgende attributen:

orderID	De ID voor deze order. Het huidige inkoopordernummer.
orderDate	De datum en tijd waarop deze order werd geplaatst, in ISO 8601-notatie.
type (facultatief)	Het type aanvraag: new (standaardwaarde), update of delete.
requisitionID (facultatief)	De aanvraag-ID van de inkoper voor deze gehele order. Deze kan gelijk zijn aan orderID of kan in het geheel niet zijn opgenomen. Deze mag niet worden toegevoegd als requisitionID is opgegeven in een ItemOut-element.
shipComplete (facultatief)	Geeft een voorkeur voor volledige verzendingen aan. De enige toegestane waarde is "yes". Standaard worden items verzonden wanneer deze beschikbaar zijn. Omdat orders items met verschillende ShipTo-elementen kunnen bevatten, dienen alleen groepen items met dezelfde verzendlocatie te worden vastgehouden tot deze zijn voltooid wanneer shipComplete="yes".

OrderRequestHeader en ItemOut (bij uitbreiding met ItemDetail) bevatten vergelijkbare gegevens. Waar OrderRequestHeader algemene facturerings- (BillTo) en betalingsgegevens (Payment) bevat, bevat ItemOut een omschrijving van de afzonderlijke items (in ItemID, ItemDetail en Distribution).

Gebruik de gegevens in OrderRequestHeader niet als standaardwaarde voor itemspecifieke elementen. Indien aanwezig, dienen ShipTo, Shipping, Contact en elke genoemde Extrinsic te worden opgenomen bij elke ItemOut of in de OrderRequestHeader. Comments- en Tax-elementen kunnen tegelijkertijd op beide niveaus worden opgenomen. De verschillende Comments-elementen dienen echter geen dubbele gegevens te bevatten en het Tax-element op kopsteksniveau bevat een totaal voor de order.

Total

Dit element bevat het totale geldbedrag van de order. Het is een container voor het Money-element.

ShipTo/BillTo

Deze elementen bevatten de adressen van de Ship To- en Bill To-eenheden in de OrderRequest.

Eén order dient aan één eenheid te worden gefactureerd. Het BillTo-element is daarom alleen opgenomen in de OrderRequestHeader. Items in een order kunnen naar verschillende locaties worden verzonden. Net als het Shipping-element (zie de volgende sectie), kan het ShipTo-element daarom in de OrderRequestHeader of in afzonderlijke ItemOut-elementen worden opgenomen.

Shipping

Dit element beschrijft hoe de items in de aanvraag worden verzonden en wat de kosten daarvan zijn. Als het Shipping-element in de OrderRequestHeader is opgenomen, mag dit niet in afzonderlijke ItemOut-elementen worden opgenomen. Als dit element niet in de OrderRequestHeader is opgenomen, dient het in de ItemOut-elementen te worden opgenomen.

Tax

Dit element bevat de belasting die met de order is verbonden. Dit element is aanwezig als de inkooporganisatie belasting berekent. Als Tax is opgenomen in de OrderRequestHeader, geeft Tax de totale belasting voor de order aan. Tax-elementen op itemniveau kunnen afzonderlijke belastingbedragen aangeven.

Payment

Dit element geeft de betalingsmethode voor de aangevraagde items aan. In bovenstaand voorbeeld bevat het Payment-element een PCard-element, waarmee een standaardinkoopkaart in het cXML-document wordt gecodeerd. In de toekomst zullen andere betalingsmethoden worden gedefinieerd en ondersteund.

Contact

Gegevens van een contactpersoon die de leverancier kan gebruiken voor vervolgacties na een order. Dit element geeft een persoon aan en bevat een lijst met wijzen waarop de persoon of eenheid kan worden bereikt. Het enige vereiste element is Name, de naam van de contactpersoon. Facultatieve en herhaalde mogelijkheden zijn PostalAddress (wordt niet aanbevolen voor directe correctie van orderproblemen), Email, Phone, Fax en URL.

Inkooporganisaties kunnen dit element gebruiken om de oorspronkelijke inkoper, de systeembeheerder van de inkooptoepassing of een andere contactpersoon aan te geven die de verantwoordelijkheid kan nemen voor het corrigeren van orderproblemen. Contact kan een andere waarde hebben dan de BillTo-en ShipTo-gegevens voor een order.

Contact heeft het volgende attribuut:

role (facultatief)	De positie van deze persoon in het inkoopproces. Kan de volgende waarden hebben: endUser, administrator, purchasingAgent, technicalSupport, customerService of sales.
------------------------------	--

Het is niet toegestaan dezelfde Contact role zowel op koptekst- als op itemniveau te gebruiken.

Wegens de ongelijksoortige inhoud van het Contact-element is er geen standaardwaarde. cXML-toepassingen behandelen een zonder role daarom als extra rol.

Comments

Willekeurige, door mensen leesbare gegevens die inkopers in inkooporders kunnen verzenden. Deze reeksgegevens zijn niet bedoeld voor de geautomatiseerde systemen op leverancierssites.

Het Comments-element kan een Attachmentelement voor opname van externe bestanden bevatten.

Attachment

Externe bestanden kunnen met behulp van Comments worden bijgevoegd om inkooporders uit te breiden. Het Attachment-element is opgenomen in Comments en bevat alleen een verwijzing naar het externe MIME-onderdeel van de bijlage. Alle bijlagen dienen in één meerdelige verzending met het OrderRequest-document te worden verzonden. Ook als dit niet mogelijk is, dient de contentID die in het Attachment-element wordt opgegeven, bruikbaar te zijn voor het ophalen van de bijlage.

Zie "Verzending van bijlagen" op pagina 55 voor gedetailleerde informatie over de verzending van bijgevoegde bestanden.

Attachment bevat één URL met "cid" als schema. Een bijgevoegd bestand kan als volgt in een cXML-document zijn opgenomen:

```
<Comments>
  <Attachment>
    <URL>cid: uniqueCID@cxml.org</URL>
  </Attachment>
  In het bijgevoegde bestand ziet u hoe het ontwerp
  eruit moet zien
</Comments>
```

Het Comments-element wordt op vele plaatsen in het cXML-protocol gebruikt maar dit element kan het Attachment-element alleen OrderRequest-documenten bevatten.

Followup

Geef de URL aan waarnaar toekomstige StatusUpdateRequest-documenten moeten worden verzonden. Deze locatie is de invoerlocatie voor latere documenten die naar het huidige OrderRequest document verwijzen.

Extrinsic

Dit element bevat automatisch leesbare informatie over de order die niet door het cXML-protocol is gedefinieerd. Het verschil met het Comments-element is dat dit laatste element informatie voor menselijk gebruik doorgeeft. Extrinsic-elementen bevatten gegevens die waarschijnlijk in latere documenten zullen worden opgenomen. Dit geldt niet voor het Comments-element. Op dit niveau vormt Extrinsic een uitbreiding van de omschrijving van alle items in de inkooporder. Bepaalde Extrinsic-gegevens kunnen ook de gehele inkooporder omschrijven, zonder de betekenis van de eventueel daarin opgenomen ItemOut-elementen te beïnvloeden.

Elke genoemde Extrinsic kan slechts eenmaal worden opgenomen in de lijsten die zijn verbonden met de OrderRequestHeader en met afzonderlijke ItemOut-elementen (in de daarin opgenomen ItemDetail-elementen). Dezelfde naam mag niet zowel in de OrderRequestHeader-lijst als in eventuele met de ItemOut-elementen verbonden lijsten zijn opgenomen. Als dezelfde Extrinsic-naam en -waarde in alle ItemOut-lijsten wordt herhaald, dient deze naar de OrderRequestHeader te worden verplaatst.

Het Extrinsic-element kan ook in de IndexItem-, PunchOutSetupRequest- en ContractItem-elementen worden opgenomen. Deze contexten worden later in dit document beschreven.

ItemOut

In het volgende voorbeeld ziet u een minimaal geldig ItemOut-element:

```
<ItemOut quantity="1">
  <ItemID>
    <SupplierPartID>5555</SupplierPartID>
  </ItemID>
</ItemOut>
```

ItemOut heeft de volgende attributen:

quantity	Het gewenste aantal items. Voor bepaalde maateenheden zijn breuken toegestaan. De waarde is mogelijk al door de leverancier gecontroleerd tijdens een punchout-sessie. Mag nooit negatief zijn.
lineNumber (facultatief)	Positie van dit item binnen een order. De waarde van dit rangtelwoord neemt eenmaal toe per ItemOut in een "nieuwe" OrderRequest. Klanten dienen dit attribuut altijd op te geven in een OrderRequest, hoewel het attribuut in andere ItemOut-contexten mogelijk geen nut heeft.
requisitionID (facultatief)	De aanvraag-ID van de inkoper voor dit regelitem. Mag niet worden toegevoegd als requisitionID is opgegeven in de OrderRequestHeader.
requestedDeliveryDate (facultatief)	De datum waarop levering van het item is aangevraagd. Hierdoor zijn leveringsdatums op itemniveau mogelijk in de OrderRequest. De datum dient in ISO 8601-notatie te zijn.

Het attribuut lineNumber voor items blijft gehandhaafd bij bijwerking van de order. Wanneer items uit een order worden verwijderd, blijft het lineNumber van de overgebleven items altijd ongewijzigd. Nieuwe items hebben hogere nummers dan items die eerder in de order zijn opgenomen. Wijzigingen in bestaande items (zoals een toename van de hoeveelheid) hebben geen invloed op het lineNumber van dat item.

In het volgende voorbeeld ziet u een ingewikkelder ItemOut-element:

```
<ItemOut quantity="2" lineNumber="1"
  equestedDeliveryDate="1999-03-12">
  <ItemID>
    <SupplierPartID>1233244</SupplierPartID>
    <SupplierPartAuxiliaryID>ABC</SupplierPartAuxiliaryID>
  </ItemID>
  <ItemDetail>
    <UnitPrice>
      <Money currency="USD">1.34</Money>
    </UnitPrice>
    <Description xml:lang="en">hello</Description>
    <UnitOfMeasure>EA</UnitOfMeasure>
  </ItemDetail>
</ItemOut>
```

```

        <Classification domain="SPSC">12345</Classification>
        <ManufacturerPartID>234</ManufacturerPartID>
        <ManufacturerName xml:lang="en">foobar</ManufacturerName>
        <URL>www.bar.com</URL>
    </ItemDetail>
    <ShipTo>
        <Address>
            <Name xml:lang="en">Acme Corporation</Name>
            <PostalAddress name="Headquarters">
                <Street>123 Anystreet</Street>
                <City>Sunnyvale</City>
                <State>CA</State>
                <PostalCode>90489</PostalCode>
                <Country isoCountryCode="US">USA</Country>
            </PostalAddress>
        </Address>
    </ShipTo>
    <Shipping>
        <Money currency="USD">1.34</Money>
        <Description xml:lang="en-US">FedEx 2-day</Description>
    </Shipping>
    <Tax>
        <Money currency="USD">1.34</Money>
        <Description xml:lang="en">foo</Description>
    </Tax>
    <Distribution>
        <Accounting name="DistributionCharge">
            <Segment type="G/L Account" id="23456"
                description="Entertainment"/>
            <Segment type="Cost Center" id="2323"
                description="Western Region Sales"/>
        </Accounting>
        <Charge>
            <Money currency="USD">.34</Money>
        </Charge>
    </Distribution>
    <Distribution>
        <Accounting name="DistributionCharge">
            <Segment type="G/L Account" id="456"
                description="Travel"/>
            <Segment type="Cost Center" id="23"
                description="Europe Implementation"/>
        </Accounting>
        <Charge>
            <Money currency="USD">1</Money>
        </Charge>
    </Distribution>
    <Comments xml:lang="en-US">
        Elke tekst die geldig is in XML.
    </Comments>
</ItemOut>

```

Met het ItemDetail-element kunnen extra gegevens naar leveranciers worden verzonden, naast de unieke ID voor het item via de ItemID.

De ShipTo-, Shipping-, Tax-, Contact-, Comments- en Extrinsic-elementen (sommige genest in ItemDetail) zijn gelijk aan de elementen die in de OrderRequestHeader kunnen worden opgenomen. Deze elementen kunnen gegevens per item, zoals verzending, verzendwijze en de daarmee verbonden kosten, bevatten. Gebruik deze elementen op OrderRequestHeader-niveau of op ItemOut-niveau, maar niet op beide niveaus tegelijk.

Distribution

Met Distribution worden de kosten van een item over meerder partijen verdeeld. Leveranciers geven het Distribution-element weer in facturen om het afstemmingsproces voor de inkoper te vergemakkelijken.

Accounting

Met het Accounting -element worden Segmenten gegroepeerd om aan te geven aan wie kosten in rekening worden gebracht.

Accounting heeft het volgende attribuut:

name	De naam van deze boekhoudcombinatie.
-------------	--------------------------------------

Segment heeft de volgende attributen:

type	Een identificatiennaam voor dit Segment in relatie tot de andere segmenten in het Accounting-element.
id	De unieke ID binnen dit Segment-type. Als het type "Cost Center" is, kan deze waarde de werkelijke rekeningcode zijn.

Charge

Dit element geeft het bedrag aan dat in rekening moet worden gebracht aan de eenheid die het Accounting-element vertegenwoordigt.

Response op een OrderRequest

Dit is het response-gedeelte van de synchrone request/response-transactie. In het volgende voorbeeld ziet u een Response op een OrderRequest-document:

```
<cXML version="1.1.007" payloadID="9949494" xml:lang="en"
  timestamp="1999-03-12T18:39:09-08:00">
  <Response>
    <Status code="200" text="OK"/>
  </Response>
</cXML>
```

Zoals hierboven weergegeven, is deze Response eenvoudig. In dit geval is er geen werkelijk element met de naam "OrderResponse" omdat alleen het Status-gedeelte van de Response aan de aanvrager hoeft te worden teruggezonden.

De Response laat de aanvrager weten dat de OrderRequest met succes is geparseerd en is uitgevoerd door het externe gedeelte van de HTTP-verbinding. Er worden geen bevestigingen op orderniveau doorgegeven, zoals de verzenddatum van items of de items die moeten worden nabesteld.

Punchout-transactie

De definities van punchout-berichten zijn request/response-berichten die binnen de Request- en Response-elementen worden verzonden. Voor ondersteuning van punchout dienen leveranciers alle volgende berichten te implementeren.

PunchOutSetupRequest

PunchOutSetupRequest en PunchOutSetupResponse zijn het request/response-paar dat wordt gebruikt om een punchout-sessie naar een extern systeem in te stellen. De client gebruikt deze om de inkooptoeepassing te identificeren, instellingsgegevens te verzenden en een antwoord te ontvangen dat aangeeft waar een HTML-bladersessie moet worden geïnitieerd op de externe website.

Het PunchOutSetupRequest-element wordt verzonden binnen het Request-element. In het volgende voorbeeld ziet u een PunchOutSetupRequest.

```
<PunchOutSetupRequest operation="create">
  <BuyerCookie>34234234ADFSDF234234</BuyerCookie>
  <Extrinsic name="department">Marketing</Extrinsic>
  <BrowserFormPost>
    <URL>http://orms.acme.com:1616/punchoutexit</URL>
```

```
</BrowserFormPost>
<SelectedItem>
  <ItemID>
    <SupplierPartID>54543</SupplierPartID>
  </ItemID>
</SelectedItem>
<SupplierSetup>
  <URL>http://workchairs.com/cxml</URL>
</SupplierSetup>
</PunchOutSetupRequest>
```

PunchOutSetupRequest heeft het volgende attribuut:

operation	Geeft het type PunchOutSetupRequest aan: "create", "inspect" of "edit".
------------------	---

Dit element bevat tevens de volgende elementen: BuyerCookie, Extrinsic, BrowserFormPost, Contact, ShipTo, SelectedItem, SupplierSetup en een ItemOut-lijst. Alleen het BuyerCookie-element is vereist. De structuur van de Extrinsic-, Contact- en ShipTo-elementen wordt uitgebreider besproken in "OrderRequestHeader" op pagina 71. Het ItemOut-element wordt besproken in "ItemOut" op pagina 76. In deze context (buiten een OrderRequest) voegen de Distribution- en Comments-elementen en de attributen lineNumber, requisitionID en requestedDeliveryDate van een ItemOut weinig toe en dienen deze niet te worden toegevoegd. Omdat punchout-sessies plaatsvinden vóór het bestellen, is deze informatie niet relevant binnen een PunchOutSetupRequest.

Een ItemOut-lijst beschrijft een bestaand winkelwagentje (items uit een eerdere punchout-sessie). De bewerking inspecteren initieert een alleen-lezen punchout-sessie (die zowel door de client als door de server wordt bekrachtigd) om details van de items in de lijst te bekijken. De actie bewerken start ook vanuit het vorige winkelwagentje (beschreven met behulp van de ItemOut-lijst) maar hierbij zijn wijzigingen toegestaan. Ondersteuning van bewerken impliceert ondersteuning van de bewerking inspecteren (zie "PunchOutOrderMessageHeader" op pagina 84 en "Winkelwagentjes legen" op pagina 84).

BuyerCookie

Dit element verzendt informatie die beperkt zichtbaar is voor de externe website maar die voor alle volgende punchout-bewerkingen moet worden teruggezonden aan het punt van oorsprong. Met dit element kan de inkooptoepassing meerdere openstaande punchout-aanvragen op elkaar afstemmen.

BrowserFormPost

Dit element vormt de bestemming voor de gegevens in PunchOutOrderMessage. Het bevat een URL-element waarvan het gebruik verder wordt uitgelegd in de definitie van PunchOutOrderMessage. Als geen gebruik wordt gemaakt van de methode URL-Form-Encoded, hoeft dit element niet te worden toegevoegd.

Extrinsic

Dit facultatieve element bevat eventuele extra gegevens die de aanvrager wil doorgeven aan de externe website. In dit voorbeeld wordt de afdeling doorgegeven van de gebruiker die de punchout-bewerking initieert. De inhoud van Extrinsic-elementen is niet gedefinieerd in de cXML-specificatie – elke aanvrager en externe website dient hierover tot overeenstemming te komen en deze te implementeren.

Extrinsic-elementen zijn bedoeld om extra automatisch leesbare informatie te bieden. Deze elementen vormen een uitbreiding van het cXML-protocol om functies te ondersteunen die niet voor alle implementaties vereist zijn. In deze context bieden de nieuwe gegevens een nadere omschrijving van de gebruiker die de punchout-aanvraag initieert.

Het Extrinsic-element kan ook in de OrderRequestHeader-, ItemDetail- en ContractItem-elementen worden opgenomen. Deze contexten worden later in dit document beschreven.

SelectedItem

Dit facultatieve element geeft de items aan die gebruikers in de punchout willen opnemen voor aankoop. Het element bevat één vereiste ItemID. Dit item brengt gebruikers van de plaatselijke catalogus naar de website van de leverancier.

Dit element is meestal aanwezig in maken-bewerkingen. Bij inkooptoepassingen waarbij gebruikers direct vanuit een leverancierslijst een punchout kunnen uitvoeren, moet SelectedItem niet worden gebruikt.

Bij bewerken en inspecteren dient SelectedItem alleen te worden opgenomen als de gebruiker ervoor heeft gekozen terug te keren naar de website van de leverancier tijdens het bekijken van nieuwe gegevens in de lokale catalogus in plaats van items in een bestaande aanvraag. In beide gevallen dient het winkelwagentje in de ItemOut-lijst te worden opgenomen.

Leveranciers kunnen hun catalogi zo opzetten dat SelectedItem leidt naar punchout op winkel-, middenpad- of productniveau. Hoe specifieker het item in de catalogus, des te minder gebruikers hoeven te zoeken op de website van de leverancier.

SupplierSetup

Dit facultatieve element geeft de URL aan waarnaar de PunchOutSetupRequest moet worden verzonden. Dit element is niet nodig als de punchout-URL van de leverancier bekend is bij de hub van het e-commercenetwerk.

PunchOutSetupResponse

Nadat de externe website een PunchOutSetupRequest heeft ontvangen, antwoordt de website met een PunchOutSetupResponse, zoals hieronder is weergegeven:

```
<PunchOutSetupResponse>
  <StartPage>
    <URL>
      http://premier.workchairs.com/store?23423SDFSDF23
    </URL>
  </StartPage>
</PunchOutSetupResponse>
```

StartPage

Dit element bevat een URL-element waarin de URL is opgenomen die moet worden doorgegeven aan de browser voor het initiëren van de punchout-bladersessie die is aangevraagd in PunchOutSetupRequest. Deze URL moet voldoende statusgegevens bevatten om met een sessiecontext op de externe website te kunnen verbinden, zoals de identiteit van de aanvrager en het juiste BuyerCookie-element.

Op dit punt bladert de gebruiker die de PunchOutSetupRequest heeft geïnitieerd, door de externe website en selecteert deze items die via een PunchOutOrderMessage aan de inkooptoepassing moeten worden teruggezonden.

PunchOutOrderMessage

Dit element zendt de context van het externe winkelwagentje terug naar het punt van oorsprong van een PunchOutSetupMessage. Dit element kan veel meer gegevens bevatten dan de andere berichten, omdat het de inhoud van elk mogelijk winkelwagentje op de externe website moet kunnen aangeven. Dit bericht voldoet niet volledig aan het request/response-model.

Bij de checkout van de gebruiker genereert de externe website een PunchOutOrderMessage. Dit bericht geeft de inhoud van het externe winkelwagentje door aan de inkooptoepassing, bijvoorbeeld:

```
<PunchOutOrderMessage>
  <BuyerCookie>34234234ADFSDF234234</BuyerCookie>
  <PunchOutOrderMessageHeader operationAllowed="create">
    <Total>
      <Money currency="USD">100.23</Money>
    </Total>
  </PunchOutOrderMessageHeader>
  <ItemIn quantity="1">
    <ItemID>
      <SupplierPartID>1234</SupplierPartID>
      <SupplierPartAuxiliaryID>
        aanvullende gegevens van dit item
      </SupplierPartAuxiliaryID>
    </ItemID>
    <ItemDetail>
      <UnitPrice>
        <Money currency="USD">10.23</Money>
      </UnitPrice>
      <Description xml:lang="en">
        Learn ASP in a Week!
      </Description>
      <UnitOfMeasure>EA</UnitOfMeasure>
      <Classification domain="SPSC">12345</Classification>
    </ItemDetail>
  </ItemIn>
</PunchOutOrderMessage>
```

Deze elementen worden in de volgende secties besproken.

BuyerCookie

Dit element is hetzelfde element dat is doorgegeven in de oorspronkelijke PunchOutSetupRequest. Het moet hier worden teruggezonden zodat de inkooptoepassing de PunchOutOrderMessage kan afstemmen op een eerdere PunchOutSetupRequest.

PunchOutOrderMessageHeader

Dit element bevat informatie over het gehele winkelwagentje dat wordt verzonden. Het enige vereiste element is `Total`, de algehele kosten van de items die aan de aanvraag worden toegevoegd. Extra toegestane elementen zijn `Shipping` en `Tax`, het bedrag en de omschrijving van verzendkosten of belastingen die op de externe website zijn berekend. Ook `ShipTo` is facultatief. Dit element bevat de verzendadresgegevens die de gebruiker op de externe site heeft geselecteerd of die zijn doorgegeven in de oorspronkelijke `PunchOutSetupRequest`. Alle geldbedragen bevinden zich in het `Money`-element, waarin valuta altijd in een standaardnotatie worden aangegeven.

`PunchOutOrderMessageHeader` heeft het volgende attribuut:

operationAllowed	Geeft aan welke <code>PunchOutSetupRequest</code> -bewerkingen toegestaan zijn: maken, inspecteren of bewerken.
-------------------------	---

Dit attribuut bepaalt of de gebruiker latere `PunchOutSetupRequest`-transacties kan initiëren met gegevens uit deze `PunchOutOrderMessage`. Als `operationAllowed="maken"`, kan alleen een latere `OrderRequest` deze items bevatten. In andere gevallen kan de inkooptoepassing het winkelwagentje later inspecteren of bewerken (waarbij volgende `PunchOutSetupRequest`-transacties worden geïnitieerd met de juiste bewerkingen en de `ItemID`-elementen die overeenkomen met de `ItemIn`-lijsten die zijn teruggezonden in deze `PunchOutOrderMessage`). Ondersteuning van bewerken betekent ondersteuning van inspecteren. De inkooptoepassing kan de items in een volgende `OrderRequest` altijd gebruiken.

Winkelwagentjes legen

`PunchOutOrderMessage` kan een lijst met items bevatten die overeenkomen met een winkelwagentje op de website van de leverancier. Dit element geeft altijd het einde van de interactieve punchout-sessie aan. De volgende alinea's bevatten de details van enkele gevallen waarbij `PunchOutOrderMessage` geen itemlijst bevat. Met deze berichten kunnen clients direct doorgaan wanneer de gebruiker de website van de leverancier verlaat.

Als de bewerking in de oorspronkelijke `PunchOutSetupRequest` inspecteren was, moet de itemlijst van de `PunchOutOrderMessage` genegeerd worden door de inkooptoepassing. In dat geval mag de site van de leverancier geen `ItemIn`-elementen terugzenden. Als een `PunchOutOrderMessage` geen `ItemIn`-elementen bevat en de bewerking maken was, mogen er geen items aan de aanvraag worden toegevoegd. De site van de leverancier of de gebruiker heeft de punchout-sessie geannuleerd zonder een winkelwagentje te maken. Als de bewerking bewerken was en de `PunchOutOrderMessage` geen `ItemIn`-elementen bevat, moeten bestaande items uit deze punchout-sessie uit de aanvraag in de inkooptoepassing worden verwijderd.

De statuscode "204/No Content" geeft het einde van een sessie zonder wijziging van het winkelwagentje aan. Ook nu dient de PunchOutOrderMessage (die altijd nodig is voor de BuyerCookie) geen ItemIn-elementen te bevatten. Deze code wordt op dezelfde manier verwerkt als de andere "empty"-gevallen die hierboven zijn beschreven, behalve wanneer de bewerking bewerken is. In dat geval heeft de gebruiker de sessie geannuleerd zonder wijzigingen uit te voeren en mogen er geen wijzigingen van de aanvraag in de inkooptoepassing worden uitgevoerd.

ItemIn

Dit element voegt een item uit een winkelwagentje toe aan een aanvraag in de inkooptoepassing. Dit element kan een aantal verschillende elementen bevatten, waarvan er slechts twee vereist zijn: ItemID en ItemDetail.

ItemIn heeft de volgende attributen:

quantity	Het aantal items dat de gebruiker op de externe website heeft geselecteerd. Omdat de site van de leverancier regels voor gedeeltelijke eenheden kan opleggen, staat het protocol breuken toe. Mag nooit negatief zijn.
lineNumber (facultatief)	De positie van dit item binnen een order. Omdat punchout-sessies normaal gesproken voorafgaand aan het bestellen plaatsvinden en de server de plaatsing van items binnen een order niet kan bepalen, is dit attribuut niet relevant in een PunchOutOrderMessage.

De optionele elementen zijn ShipTo, Shipping en Tax, dezelfde elementen als de elementen die hierboven zijn opgegeven in PunchOutOrderMessage.

Met uitzondering van de Distribution- en Comments-elementen en de attributen requisitionID- en requestedDeliveryDate die beschikbaar zijn in het ItemOut-element, komen de ItemIn- ItemOut-structuren volledig overeen. Het inkoopstelsel van oorsprong kan een directe conversie uitvoeren tussen ItemIn- en ItemOut-lijsten bij het initiëren van een inspect- of edit-bewerking. Leveranciers kunnen de een naar de ander converteren (waarbij de beschikbare opgenomen extensies in het ItemOut-element worden opgenomen) bij het uitvoeren van een edit-bewerking. Het inkoopstelsel van oorsprong kan de rechtstreekse conversie uitvoeren en aanvullende verzend- en distributiegegevens toevoegen bij het initiëren van een OrderRequest-transactie. ItemDetail-gegevens (met de mogelijke uitzondering van Extrinsic-elementen) binnen ItemIn-elementen mogen niet worden verwijderd bij de conversie van ItemIn naar ItemOut.

ItemID

Dit element vormt de unieke ID van het item voor de externe website. Het is het enige element dat dient te worden teruggezonden aan de externe website om het item dat wordt verzonden opnieuw te identificeren.

ItemID bevat twee elementen: SupplierPartID en SupplierPartAuxiliaryID. Alleen SupplierPartID is verplicht. SupplierPartAuxiliaryID is een hulpmiddel voor de externe website bij het transporteren van complexe configuratie- of vrachtbriefgegevens om het item opnieuw te identificeren wanneer het in de toekomst aan de externe website wordt gepresenteerd.

Als SupplierPartAuxiliaryID speciale tekens bevat (bijvoorbeeld extra XML-elementen die niet gedefinieerd zijn in het cXML-protocol), dienen deze op de juiste manier te worden voorafgegaan door een backslash (/). Door de noodzaak om SupplierPartAuxiliaryID-gegevens door te geven via toepassingen en terug naar de oorspronkelijke leverancier, is een interne subset met extra XML-elementen onvoldoende.

ItemDetail

Dit element bevat beschrijvende gegevens van het item die inkooptoepassingen aan gebruikers presenteren. de inhoud van een ItemDetail-element kan vrij complex zijn, maar de minimeisen zijn eenvoudig: UnitPrice, Description, UnitOfMeasure en Classification.

In de context van een ItemIn-element functioneren de Extrinsic-elementen die in een ItemDetail zijn opgenomen, op dezelfde manier als in een index (in het bijzonder een IndexItemAdd).

Description

Dit element beschrijft het item in tekstvorm. Omdat de tekst de grenzen van een korte tabel met regelitems (of een andere beperkte gebruikersinterface) kan overschrijden en er willekeurige afbrekingen kunnen optreden, bevat het Description-element een facultatief ShortName-element. Als de ShortName is opgegeven, moeten clients deze in plaats van een afbreking van de Description-text gebruiken in beperkte velden. Als er geen ShortName wordt gegeven, moeten clients doorgaan met afbreken van de Description-text.

Voorbeeld:

```
<Description xml:lang="en-US">
  <ShortName>Big Computer</ShortName>
  This wonder contains three really big disks, four CD-Rom drives, two Zip drives,
  an ethernet card or two, much more memory than you could ever use, four CPUs on two
  motherboards. We'll throw in two monitors, a keyboard and the cheapest mouse we can
  find lying around.
</Description>
```

kan worden weergegeven als "Big Computer" wanneer er weinig ruimte is en als "Big Computer: This wonder ... lying around." of als twee afzonderlijke, maar volledige velden waar ruimte is om meer tekst weer te geven.

Latere statuswijzigingen

Nadat de OrderRequest-transactie is voltooid, dienen leveranciers en tussenservers mogelijk aanvullende gegevens terug te zenden aan het inkoopstelsel. De transacties die in deze sectie worden beschreven, worden voor dat doel gebruikt. Deze transacties hebben een bepaalde semantiek en enkele elementen gemeen.

Net als het antwoord op een OrderRequest (zie "Response op een OrderRequest" op pagina 79), bevat geen van deze transacties een specifiek Response-element. In plaats daarvan bevat het teruggezonden document een bijna lege Response (met alleen een Status). Elk teruggezonden document heeft de volgende vorm:

```
<cXML version="1.1.007" payloadID="9949494@supplier.com"
  timestamp="2000-01-12T18:39:09-08:00" xml:lang="en-US">
  <Response>
    <Status code="200" text="OK"/>
  </Response>
</cXML>
```

De teruggezonden code is alleen "200" als de bewerking met succes is voltooid.

DocumentReference

Het DocumentReference-element bevat voldoende gegevens om het bijwerkverzoek met een bepaald document te verbinden. Het element herhaalt een verplicht attribuut van het eerdere document en voegt een facultatieve ID toe die door de leverancier is gegenereerd. Voorbeeld:

```
<DocumentReference
  payloadID="0c300508b7863dcclb_14999"
  orderID="DO4321" />
```

DocumentReference bevat geen elementen maar bezit de volgende attributen:

payloadID	Een uniek nummer dat wordt gebruikt voor de identificatie van documenten voor logdoeleinden. Deze waarde dient ongewijzigd te blijven in verband met nieuwe pogingen. De aanbevolen implementatie is: datetime.process id.random number@hostname Direct overgenomen uit het cXML-element van het OrderRequest-document.
orderID	De ID voor deze order. Het huidige inkoopordernummer. Direct overgenomen uit de OrderRequestHeader van het OrderRequest-document.

StatusUpdateRequest

Met deze transactie wordt een eerdere hub geïnformeerd over wijzigingen in de verwerkingsstatus van een order. Een wijziging is bijzonder belangrijk: als een tussenhub een OrderRequest met succes doorzendt, kan deze hub de oorspronkelijke afzender of een eerdere hub informeren over dit succes. Ook overgangen tussen verschillende wachtrijen en verwerkingsstappen bij een leverancier of hub kunnen van belang zijn voor de inkoper.

Met deze aanvraag wordt de verwerkingsstatus van één OrderRequest-document bijgewerkt. Voorbeeld:

```
<cXML version="1.1.007" xml:lang="en-US"
  payloadID="0c30050@supplier.com"
  timestamp="2000-01-08T23:00:06-08:00">
  <Header>
    Routing-, identificatie- en verificatiegegevens.
  </Header>
  <Request>
    <StatusUpdateRequest>
      <DocumentReference
        payloadID="0c300508b7863dcclb_14999"
```



```
        orderID="DO4321" />
        <Status code="200" text="OK" xml:lang="en-US">Forwarded
            to supplier</Status>
    </StatusUpdateRequest>
</Request>
</cXML>
```

Deze aanvraag bevat alleen een DocumentReference- en een Status-element. Beide zijn vereist. Via de Status kan een latere transportfout worden doorgegeven die bij een tussenhub is opgetreden. De semantiek van dit element is gelijk aan de Status die teruggezonden kan zijn in het aanvankelijke HTTP-antwoord op een OrderRequest-document.

De 200/OK code is vooral belangrijk wanneer documenten worden opgeslagen en doorgezonden. Deze code geeft aan dat een leverancier begonnen is met de verwerking van de OrderRequest of dat een hub het document heeft doorgezonden. De ontvanger krijgt geen StatusUpdateRequest-documenten meer nadat 200/OK is ontvangen.

Leveranciers en hubs die gebruikmaken van de StatusUpdate-transactie dienen code 201/Accepted terug te zenden wanneer een OrderRequest voor latere verwerking in de wachtrij is geplaatst. Nadat 200/OK is verzonden (in de directe Response op een OrderRequest of een latere StatusUpdateRequest), verstuurt de server geen verdere StatusUpdate-transacties voor die order. Fouten in de latere verwerking kunnen leiden tot uitzonderingen op deze regel.

Catalogusdefinities

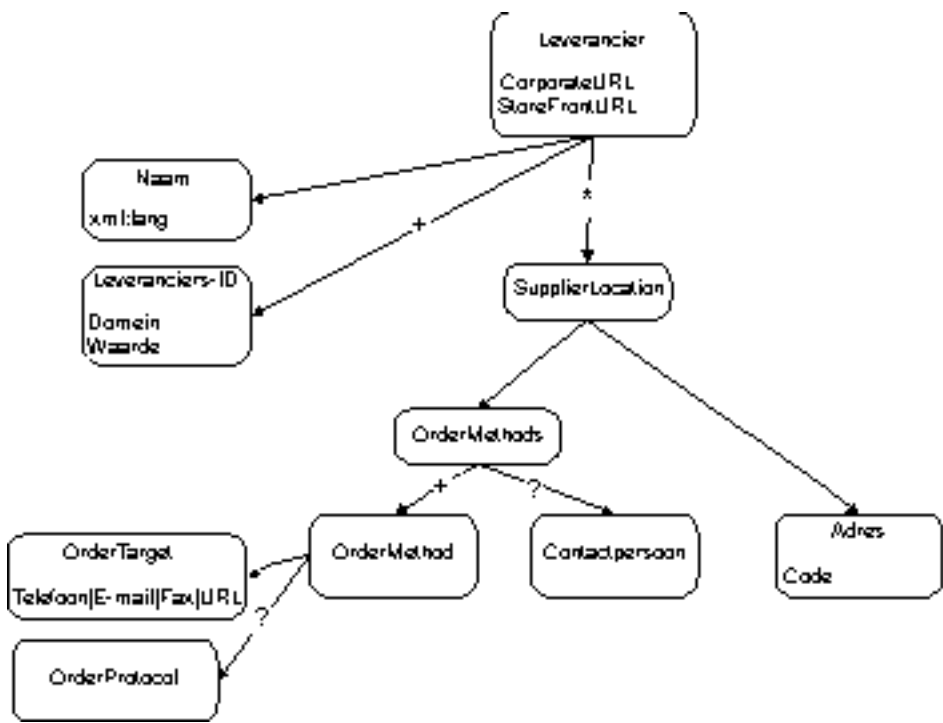
De cXML-catalogusdefinities bestaan uit drie hoofdelementen: Supplier, Index en Contract. Deze drie elementen beschrijven gegevens die bedoeld zijn voor blijvend of gecached gebruik in een hub of in het inkoopstelsel van een inkooporganisatie.

- Supplier – bevat basisgegevens van de leverancier, zoals adres, contactpersoon en bestelinformatie.
- Index – bevat gegevens van de goederen- en dienstenvoorraad van de leverancier, zoals omschrijving, onderdeelnummers en classificatiecodes.
- Contract – bevat gegevens van flexibele aspecten van de voorraad die overeen zijn gekomen tussen de inkoper en de leverancier, zoals prijs.

Bij Index wordt gebruikgemaakt van verschillende subelementen om regelitems in voorraden van leveranciers te beschrijven. Leveranciers kunnen prijsgegevens verzenden die in de systemen van inkopers worden opgeslagen of punchout-gegevens verzenden waarmee inkopers prijs- en andere gegevens kunnen opvragen bij externe websites.

Supplier

Het Supplier-element bevat een met name genoemde leverancier van goederen of diensten. Dit element moet een Name-element en een SupplierID-element bevatten. Daarnaast kan dit element optioneel adres- en ordergegevens van de leverancier bevatten:



Supplier heeft de volgende attributen:

corporateURL (facultatief)	URL van website van de leverancier.
storeFrontURL (facultatief)	URL van website voor winkelen of bladeren.

In het volgende voorbeeld ziet u een overzicht van het Supplier-element:

```

<Supplier>
  <SupplierID domain="InternalSupplierID">29</SupplierID>
  <SupplierID domain="DUNS">76554545</SupplierID>
  <SupplierLocation>
    <Address>
      <Name xml:lang="en-US">Main Office</Name>
      <PostalAddress>
        ...
      </PostalAddress>
      <Email>bobw@workchairs.com</Email>
      <Phone name="Office">
        ...
      </Phone>
      <Fax name="Order">
        ...
      </Fax>
      <URL>http://www.workchairs.com/Support.htm</URL>
    </Address>
    <OrderMethods>
      <OrderMethod>
        <OrderTarget>
          <URL>http://www.workchairs.com/cxmlorders</URL>
        </OrderTarget>
      </OrderMethod>
      <Contact>
        <Name xml:lang="en-US">Mr. Smart E. Pants</Name>
        <Email>sepants@workchairs.com</Email>
        <Phone name="Office">
          ...
        </Phone>
      </Contact>
    </OrderMethods>
  </SupplierLocation>
</Supplier>

```

SupplierLocation

Sommige leveranciers doen zaken vanaf meerdere locaties. Voor elke locatie kan een SupplierLocation-element worden gebruikt. In dit element kan tevens worden opgenomen hoe deze locatie zaken doet of op welke wijzen er orders worden geaccepteerd. Het SupplierLocation-element bevat een Address en een set OrderMethods.

OrderMethods en OrderMethod

Het OrderMethods-element is een verzameling van één of meer OrderMethod-elementen voor het gegeven SupplierLocation-element. De positie van OrderMethods in de lijst is van belang – het eerste element is de bestelmethode waaraan de voorkeur wordt gegeven, het tweede element is de volgende voorkeur, enzovoort, in afnemende volgorde van voorkeur.

OrderMethod bevat bestelgegevens in de vorm van een orderdoel (zoals telefoon, fax of URL) en een facultatief protocol om de verwachtingen bij het doel verder te verduidelijken, bijvoorbeeld "cxml" voor een URL-doel.

Index

Dit element is het basiselement voor het bijwerken van catalogi in de inkoopsystemen van inkooporganisaties.

Een Index-element is verbonden met één leverancier. Het Index-element kan een lijst met leveranciers-ID's bevatten, waarbij elke ID als synoniem met die leverancier wordt beschouwd.

De Index bevat een of meer IndexItem-elementen en een optionele set SearchGroup-elementen voor het definiëren van parameterzoekgegevens voor items. Het IndexItem-element bevat elementen waarmee wordt toegevoegd aan of verwijderd uit de gecacheerde catalogus van de inkooporganisatie. In het volgende voorbeeld ziet u een overzicht van het Index-element:

```
<Index>
  <SupplierID> ... </SupplierID>
  ...
  <IndexItem>
    <IndexItemAdd>
      <IndexItemDetail>
        ...
      </IndexItemDetail>
    </IndexItemAdd>
    ...
    <IndexItemDelete>
      ...
    </IndexItemDelete>
    ...
    <IndexItemPunchout>
      ...
    </IndexItemPunchout>
  </IndexItem>
</Index>
```

IndexItem, IndexItemAdd, IndexItemDelete en IndexItemPunchout

Het `IndexItem`-element bevat de lijst met items in een index. Het `IndexItem`-element bevat drie soorten elementen:

- `IndexItemAdd` – voegt een nieuw item in of werkt een bestaand item in de index bij. Dit element bevat het `ItemID`-element, het `ItemDetail`-element en het `IndexItemDetail`-element.
- `IndexItemDelete` – verwijdert een item uit de index. Dit element bevat het `ItemID`-element met de ID van het item.
- `IndexItemPunchout` – voegt een item in voor het initiëren van punchout naar de website van de leverancier. Dit element bevat het `PunchoutDetail`-element en het `ItemID`-element. `IndexItemPunchout` is gelijk aan het `IndexItemAdd`-element maar voor dit element zijn geen prijsgegevens vereist. Inkopers ontvangen real-time itemgegevens van de website van de leverancier.

ItemID

Het `ItemID`-element vormt de unieke ID van een item van de leverancier. Dit element bevat het `SupplierPartID`-element en het facultatieve `SupplierPartAuxiliaryID`-element.

SupplierPartAuxiliaryID

Als `SupplierPartID` geen unieke ID van het item is, dient de leverancier `SupplierPartAuxiliaryID` te gebruiken om een extra sleutel op te geven die een unieke ID van het onderdeel vormt in combinatie met `SupplierID` en `SupplierPartID`. Een leverancier kan bijvoorbeeld dezelfde `SupplierPartID` voor een item gebruiken maar verschillende prijzen in rekening brengen voor de eenheden "EA" en "BOX". In dat geval zijn "EA" en "BOX" mogelijke `SupplierPartAuxiliaryID`'s voor deze twee items.

`SupplierPartAuxiliaryID` kan tevens als leverancierscookie worden gebruikt. Hiermee kan de leverancier naar complexe configuratie- of onderdeelgegevens verwijzen. De cookie kan alle gegevens bevatten die de leverancier nodig heeft om te herleiden wat het betreffende item in het computersysteem is (een winkelwagentje of een cookie met gegevens die alleen betekenis hebben voor de leverancier). Zie "Cookies van inkoper en leverancier" op pagina 40 voor meer informatie.

ItemDetail

ItemDetail bevat gedetailleerde informatie over een item, of alle gegevens van een item die een gebruiker naast de basisgegevens in ItemID wil bekijken. Dit element moet een UnitPrice, een UnitOfMeasure, een of meer Description-elementen en een Classification bevatten en kan optioneel een ManufacturerPartID, een ManufacturerName, een URL en een onbeperkt aantal Extrinsic-elementen bevatten. Zie information "ItemDetail" op pagina 86 voor meer informatie.

In de context van een IndexItemAdd vormen de Extrinsic-elementen een uitbreiding van de informatie over een bepaald item. Deze uitbreidingen moeten niet in een OrderRequest aan een leverancier worden verzonden, omdat de leverancier dezelfde gegevens kan ophalen met behulp van de unieke ItemID.

IndexItemDetail

Het IndexItemDetail-element bevat elementen die specifiek voor de index zijn en die aanvullende aspecten van een item definiëren, zoals LeadTime, ExpirationDate, EffectiveDate, SearchGroupData of TerritoryAvailable.

PunchoutDetail

PunchoutDetail is gelijk aan ItemDetail maar voor PunchoutDetail zijn slechts een of meer Description-elementen en een Classification vereist. Dit element kan tevens URL-, ManufacturerName-, ManufacturerPartID-, ExpirationDate-, EffectiveDate-, SearchGroupData-, TerritoryAvailable- en Extrinsic-elementen bevatten. PunchoutDetail bevat geen prijs-, doorlooptijd- of maateenheidgegevens.

Contract

Een Contract-element vertegenwoordigt een contract tussen een leverancier en een inkoper voor goederen of diensten die in de index van de leverancier zijn omschreven. De leverancier kan dit element gebruiken om itemattributen, zoals de prijs, in de index te overschrijven met waarden die met de inkoper zijn overeengekomen. Dit element stelt leveranciers en inkopers tevens in staat om deze overschrijvingen te segmenteren op basis van een overeengekomen "segmentsleutel" die betekenis heeft in een inkooporganisatie, zoals de naam van een fabriek of een kostenplaats.

Contract heeft de volgende attributen:

effectiveDate	Ingangsdatum en -tijd van het contract, in ISO 8601-notatie.
expirationDate	Verloopdatum en -tijd van het contract, in ISO 8601-notatie.

Contract bevat een of meer ItemSegment-elementen, bijvoorbeeld:

```
<Contract effectiveDate="2000-01-03T18:39:09-08:00"
  expirationDate="2000-07-03T18:39:09-08:00">
  <SupplierID domain="InternalSupplierID">29</SupplierID>
  <ItemSegment segmentKey=Plant12>
    <ContractItem>
      <ItemID>
        <SupplierPartID>pn12345</SupplierPartID>
      </ItemID>
      <UnitPrice>
        <Money currency=USD>40.00</Money>
      </UnitPrice>
    </ContractItem>
    ...
  </ItemSegment>
</Contract>
```

ItemSegment

Het ItemSegment-element bevat een lijst met ContractItem-elementen voor een bepaald "segment", waarbij een segment een willekeurige opdeling van contractitems is op basis van een segmentsleutel die is overeengekomen tussen de leverancier en de inkoper.

ItemSegment heeft het volgende attribuut:

segmentKey (facultatief)	Overeengekomen reeks die wordt gebruikt voor het segmenteren van aangepaste prijzen.
------------------------------------	--

ContractItem

Het ContractItem-element is een bepaalde itemoverschrijving voor een indexitem. Dit element bevat een ItemID die de unieke ID van het indexitem vormt binnen het inkoopstelsel dat moet worden overschreven. Het ContractItem-element kan een onbepaald aantal Extrinsic-elementen bevatten met de overschreven waarde voor het met name genoemde indexitemattribuut.

Definities inschrijvingenbeheer

Tussenstations, zoals hubs van een e-commercenetwerk, kunnen de leveranciers en leverancierscatalogi beheren die door de inkoopsystemen van inkooporganisaties worden gebruikt. Deze tussenstations kunnen directe verbindingen tussen inkoopsystemen en leverancierssystemen tot stand brengen. Deze sectie bevat definities van elementen voor het beheer van leveranciersgegevens en de inhoud van catalogi. Deze definities bouwen voort op een groot aantal van de voorgaande definities van cXML-request/response, one-way-berichten en catalogusdefinities.

Leveranciersgegevens

De definities voor het beheer van leveranciersgegevens bestaan voornamelijk uit de elementen `SupplierListRequest`, `SupplierListResponse`, `SupplierDataRequest`, `SupplierDataResponse` en `SupplierChangeMessage`. Deze elementen worden hieronder beschreven en geïllustreerd aan de hand van voorbeelden waarin het tussenstation Ariba Network is.

SupplierListRequest

`SupplierListRequest` vraagt een lijst met leveranciers op waarmee de inkoper handelsrelaties onderhoudt.

```
<Request>
  <SupplierListRequest/>
</Request>
```

SupplierListResponse

`SupplierListRequest` geeft een lijst weer met leveranciers waarmee de inkoper handelsrelaties onderhoudt.

```
<Response>
  <Status code="200" text="OK"/>
  <SupplierListResponse>
    <Supplier corporateURL=http://www.workchairs.com
      storeFrontURL="http://www.workchairs.com">
      <Name xml:lang="en-US">Main Office</Name>
      <Comments xml:lang="en-US">this is a cool company</Comments>
      <SupplierID domain="DUNS">123456</SupplierID>
    </Supplier>
    <Supplier corporateURL=http://www.computersRus.com
      storeFrontURL="http://www.computersRus.com">
      <Name xml:lang="en-US">Computers R us</Name>
      <Comments xml:lang="en-US">another cool company</Comments>
      <SupplierID domain="DUNS">123456789</SupplierID>
    </Supplier>
  </SupplierListResponse>
</Response>
```


SupplierDataRequest

SupplierDataRequest vraagt gegevens van een leverancier op.

```
<Request>
  <SupplierDataRequest>
    <SupplierID domain="DUNS">123456789</SupplierID>
  </SupplierDataRequest>
</Request>
```

SupplierDataResponse

SupplierDataRequest bevat gegevens van een leverancier.

```
<Response>
  <Status code="200" text="OK"/>
  <SupplierDataResponse>
    <Supplier corporateURL=http://www.workchairs.com
      storeFrontURL="http://www.workchairs.com">
      <Name xml:lang="en-US">Main Office</Name>
      <Comments xml:lang="en-US">this is a cool company</Comments>
      <SupplierID domain="DUNS">123456</SupplierID>
      <SupplierLocation>
        <Address>
          <Name xml:lang="en-US">Main Office</Name>
          <PostalAddress>
            <DeliverTo>Bob A. Worker</DeliverTo>
            <Street>123 Front Street</Street>
            <City>Toosunny</City>
            <State>CA</State>
            <PostalCode>95000</PostalCode>
            <Country isoCountryCode="US">USA</Country>
          </PostalAddress>
          <Email>bobw@workchairs.com</Email>
          <Phone name="Office">
            <TelephoneNumber>
              <CountryCode
                isoCountryCode="US">1</CountryCode>
              <AreaOrCityCode>800</AreaOrCityCode>
              <Number>5551212</Number>
            </TelephoneNumber>
          </Phone>
          <Fax name="Order">
            <TelephoneNumber>
              <CountryCode
                isoCountryCode="US">1</CountryCode>
              <AreaOrCityCode>408</AreaOrCityCode>
              <Number>5551234</Number>
            </TelephoneNumber>
          </Fax>
          <URL>http://www.workchairs.com/Support.htm</URL>
        </Address>
        <OrderMethods>
```

```

        <OrderMethod>
          <OrderTarget>
            <URL>http://www.workchairs.com/cxmlorder</URL>
          </OrderTarget>
          <OrderProtocol>cXML</OrderProtocol>
        </OrderMethod>
      </OrderMethods>
    </SupplierLocation>
  </Supplier>
</SupplierDataResponse>
</Response>

```

SupplierChangeMessage

Dit element wordt gebruikt voor berichtgeving over wijzigingen in leveranciersgegevens.

```

<Message>
  <SupplierChangeMessage type="new">
    <Supplier corporateURL=http://www.workchairs.com
      storefrontURL="http://www.workchairs.com">
      <Name xml:lang="en-US">Main Office</Name>
      <Comments xml:lang="en-US">this is a cool company</Comments>
      <SupplierID domain="DUNS">123456</SupplierID>
      <SupplierLocation>
        <Address>
          <Name xml:lang="en-US">Main Office</Name>
          <PostalAddress>
            <DeliverTo>Bob A. Worker</DeliverTo>
            <Street>123 Front Street</Street>
            <City>Toosunny</City>
            <State>CA</State>
            <PostalCode>95000</PostalCode>
            <Country isoCountryCode="US">USA</Country>
          </PostalAddress>
          <Email>bobw@workchairs.com</Email>
          <Phone name="Office">
            <TelephoneNumber>
              <CountryCode
                isoCountryCode="US">1</CountryCode>
              <AreaOrCityCode>800</AreaOrCityCode>
              <Number>5551212</Number>
            </TelephoneNumber>
          </Phone>
          <Fax name="Order">
            <TelephoneNumber>
              <CountryCode
                isoCountryCode="US">1</CountryCode>
              <AreaOrCityCode>408</AreaOrCityCode>
              <Number>5551234</Number>
            </TelephoneNumber>
          </Fax>
        </Address>
      </SupplierLocation>
    </Supplier>
  </SupplierChangeMessage>
</Message>

```

```

        </TelephoneNumber>
      </Fax>
      <URL>http://www.workchairs.com/Support.htm</URL>
    </Address>
    <OrderMethods>
      <OrderMethod>
        <OrderTarget>
          <URL>http://www.workchairs.com/cxmlorder</URL>
        </OrderTarget>
        <OrderProtocol>cXML</OrderProtocol>
      </OrderMethod>
    </OrderMethods>
  </SupplierLocation>
</Supplier>
</SupplierChangeMessage>
</Message>

```

Catalogusinschrijvingen

De definities voor het beheer van catalogusinschrijvingen vindt u hieronder. Het tussenstation in de voorbeelden is Ariba Network.

Subscription

Dit element bevat metagegevens van een afzonderlijk cataloguselement. Subelementen zijn:

- InternalID – een unieke ID van het tussenstation.
- Name – de naam van het abonnement.
- ChangeTime – de datum en tijd waarop voor het laatst wijzigingen in het abonnement zijn aangebracht.
- SupplierID – de ID van de leverancier die de catalogus levert.
- Format – de indeling van de catalogus.
- Description – een omschrijving van de catalogus.

```

<Subscription>
  <InternalID>1234</InternalID>
  <Name xml:lang="en-US">Q2 Prices</Name>
  <Changetime>1999-03-12T18:39:09-08:00</Changetime>
  <SupplierID domain="DUNS">123456789</SupplierID>
  <Format version="2.1">CIF</Format>
  <Description xml:lang="en-US">The best prices for software</Description>
</Subscription>

```

SubscriptionListRequest

Dit element vraagt de huidige lijst met catalogusinschrijvingen van de inkoper op.

```
<Request>
  <SubscriptionListRequest/>
</Request>
```

SubscriptionListResponse

Dit element bevat de huidige lijst met catalogusabonnementen van de inkoper.

```
<Response>
  <Status code="200" text="OK"/>
  <SubscriptionListResponse>
    <Subscription>
      <InternalID>1234</InternalID>
      <Name xml:lang="en-US">Q2 Prices</Name>
      <Changetime>1999-03-12T18:39:09-08:00</Changetime>
      <SupplierID domain="DUNS">123456789</SupplierID>
      <Format version="2.1">CIF</Format>
      <Description xml:lang="en-US">The best prices for software</Description>
    </Subscription>
    <Subscription>
      <InternalID>1235</InternalID>
      <Name xml:lang="en-US">Q2 Software Prices</Name>
      <Changetime>1999-03-12T18:15:00-08:00</Changetime>
      <SupplierID domain="DUNS">555555555</SupplierID>
      <Format version="2.1">CIF</Format>
      <Description xml:lang="en-US">The best prices for software</Description>
    </Subscription>
  </SubscriptionListResponse>
</Response>
```

SubscriptionContentRequest

Dit element vraagt de inhoud op van een catalogus waarop de inkoper is ingeschreven. De aanvraag bevat de InternalID en SupplierID van de catalogus.

```
<Request>
  <SubscriptionContentRequest>
    <InternalID>1234</InternalID>
    <SupplierID domain="DUNS">123456789</SupplierID>
  </SubscriptionContentRequest>
</Request>
```

SubscriptionContentResponse

Dit element bevat de inhoud van een catalogus. De catalogusindeling kan CIF (Catalog Interchange Format) of cXML zijn. Als de catalogusindeling CIF is, wordt de catalogus gecodeerd in base64 en opgenomen als inhoud van een CIFContent-element. Als de indeling cXML is, worden de Index- en Contract-elementen reeks in dit element opgenomen.

```
<Response>
  <Status code="200" text="OK"/>
  <SubscriptionContentResponse>
    <Subscription>
      <InternalID>1234</InternalID>
      <Name xml:lang="en-US">Q2 Prices</Name>
      <Changetime>1999-03-12T18:39:09-08:00</Changetime>
      <SupplierID domain="DUNS">123456789</SupplierID>
      <Format version="3.0">CIF</Format>
      <Description xml:lang="en-US">The best prices for software
    </Description>
    </Subscription>
    <SubscriptionContent filename="foobar.cif">
      <CIFContent>
        <!-- base64 encoded data -->
        ABCDBBDBDBDBDB
      </CIFContent>
    </SubscriptionContent>
  </SubscriptionContentResponse>
</Response>
```

SubscriptionChangeMessage

Dit element geeft aan het inkoopstelsel van de inkooporganisatie door dat een catalogus waarvoor een abonnement bestaat, is gewijzigd.

```
<Message>
  <SubscriptionChangeMessage type="new">
    <Subscription>
      <InternalID>1234</InternalID>
      <Name xml:lang="en-US">Q2 Prices</Name>
      <Changetime>1999-03-12T18:39:09-08:00</Changetime>
      <SupplierID domain="DUNS">123456789</SupplierID>
      <Format version="2.1">CIF</Format>
    </Subscription>
  </SubscriptionChangeMessage>
</Message>
```

SubscriptionChangeMessage heeft het volgende attribuut:

type	Het type wijziging: new, delete of update.
------	--

Definities voor ophalen van berichten

Niet alle inkooporganisaties bezitten HTTP-ingangspunten voor de ontvangst van cXML-berichten die afkomstig zijn van buiten de brandschermen van het bedrijf. De cXML-specificatie voorziet in dergelijke omgevingen.

In deze sectie worden definities geïntroduceerd waarmee bronsystemen berichten in de wachtrij kunnen plaatsen wanneer doelen HTTP-posts niet rechtstreeks kunnen accepteren. De doelen kunnen de berichten vervolgens ophalen wanneer dat schikt.

GetPendingRequest

Dit element haalt een groep berichten op die wachten op de aanvrager. Het MessageType-element en de attributen lastReceivedTimestamp en maxMessages bepalen het type en het aantal opgehaalde berichten.

lastReceivedTimestamp (facultatief)	De tijdregistratie van het meest recente ontvangen bericht.
maxMessages (facultatief)	Maximum aantal berichten per antwoord dat door de aanvrager kan worden verwerkt.

Bij ontvangst van de aanvraag zendt de ontvanger de oudste berichten van de opgegeven typen terug met tijdstempels die gelijk aan het opgegeven tijdstempel zijn of die na het opgegeven tijdstempel liggen. Als meerdere berichten aan dit criterium voldoen, kunnen er meerdere berichten worden teruggezonden, afhankelijk van het attribuut maxMessages. Het wachtrijsysteem negeert alle wachtende berichten van de opgegeven typen met tijdstempels vóór het opgegeven tijdstempel.

```
<Request>
  <GetPendingRequest lastReceivedTimestamp="1999-03-12T18:39:09-08:00"
    maxMessages="5">
    <MessageType>SubscriptionChangedMessage</MessageType>
  </GetPendingRequest>
</Request>
```

GetPendingResponse

Dit element bevat een of meer berichten die wachten op de aanvrager.

```

<Response>
  <Status code="200" text="OK"/>
  <GetPendingResponse>
    <cXML version="1.1.007" xml:lang="en-US"
      payloadID="456778@ariba.com"
      timestamp="1999-03-12T18:39:09-08:00">
      <Header>
        <From>
          <Credential domain="AribaNetworkUserId">
            <Identity>admin@ariba.com</Identity>
          </Credential>
        </From>
        <To>
          <Credential domain="AribaNetworkUserId">
            <Identity>admin@acme.com</Identity>
          </Credential>
        </To>
        <Sender>
          <Credential domain="AribaNetworkUserId">
            <Identity>admin@ariba.com</Identity>
          </Credential>
          <UserAgent>Ariba.com</UserAgent>
        </Sender>
      </Header>
      <Message>
        <SubscriptionChangeMessage type="new">
          <Subscription>
            <InternalID>1234</InternalID>
            <Name xml:lang="en-US">Q2 Prices</Name>
            <Changetime>1999-03-12T18:39:09-08:00
            </Changetime>
            <SupplierID domain="DUNS">123456789
            </SupplierID>
            <Format version="2.1">CIF</Format>
          </Subscription>
        </SubscriptionChangeMessage>
      </Message>
    </cXML>
  </GetPendingResponse>
</Response>

```

Appendix B

Nieuwe functies in cXML 1.1

cXML 1.1 bevat de volgende categorieën nieuwe functies:

- Algemene wijzigingen in cXML
- Wijzigingen in Extrinsic-elementen
- Verbeteringen in punchout-transacties
- Nieuwe inkooporderfuncties
- Nieuwe statustransactie van inkooporders

Zie Appendix A, "Specificatie van cXML." voor een uitgebreide bespreking van elk element of attribuut dat hier wordt genoemd.

Algemene wijzigingen in cXML

De volgende wijzigingen zijn van invloed op grote gebieden van de cXML-taal. cXML is nu gemakkelijker internationaal te gebruiken en de verbeteringen zijn mede gericht op de compatibiliteit van cXML-versies. Daarnaast kunnen cXML-clients met een nieuwe profieltransactie de mogelijkheden van cXML-servers onderzoeken.

Verbeterde ondersteuning voor het gebruik van meerdere talen

Met het oog op consistentie en een betere ondersteuning van het gebruik van meerdere talen hebben de elementen cXML, Status en ManufacturerName nu een facultatief attribuut `xml:lang`.

Voorbeeld:

```
<Status
  xml:lang="en-US"
  code="200
  text="OK">
</Status>
```

Zie

"Status" op pagina 58
voor meer informatie.

Met dit attribuut wordt de taal aangegeven die cXML-clients bij reacties dienen te gebruiken en die op websites met punchout wordt weergegeven.

Gecentraliseerde DTD's

Voorheen was er geen centrale locatie gedefinieerd voor de DTD's van cXML, daarom konden cXML-parsers deze niet automatisch ophalen. Nu zijn de DTD's voor alle versies van cXML beschikbaar op consistente locaties op cxml.org.

Ga voor DTD's van cXML naar:

`http://xml.cXML.org/schemas/cXML/<version>/cXML.dtd`

waarbij `<version>` het volledige nummer is van de cXML-versie, zoals 1.1.007.

Deze mogelijkheid wordt het beste benut wanneer cXML-clients niet elke keer als cXML-documenten worden geparseerd, DTD's halen. In plaats daarvan dienen deze lokaal in het cachegeheugen te worden opgeslagen. Nadat een URL onder `//xml.cxml.org/schemas/cXML` is geplaatst, is het niet nodig deze locatie te wijzigen. DTD's worden nooit ter plaatse gewijzigd, maar er worden nieuwe branches toegevoegd.

Zie

"Validatie met DTD's" op pagina 8 voor meer informatie.

Nieuwe profieltransactie

Een nieuwe profieltransactie zorgt voor de overdracht van basisinformatie over cXML-servers. Deze transactie bestaat uit twee documenten: `ProfileRequest` en `ProfileResponse`. Met deze transactie worden servercapaciteiten opgehaald, waaronder de ondersteunde cXML-versie, ondersteunde transacties en de opties bij deze transacties.

Opmerking: Alle cXML 1.1-servers **moeten** deze transactie ondersteunen.

Clients kunnen de profieltransactie ook gebruiken om servers te 'pingen' en zo te controleren of deze beschikbaar zijn.

Zie

"Profieltransactie" op pagina 68 voor meer informatie.

ProfileRequest

Het document ProfileRequest bevat geen inhoud. Dit wordt eenvoudigweg gerouteerd naar de toepasselijke cXML-server met behulp van de referenties in de kop.

```
<cXML version="1.1.007" payloadID="9949494"
  xml:lang="en-US" timestamp="2000-03-12T18:39:09-08:00">
  <Header>
    Routing-, identificatie- en verificatiegegevens.
  </Header>
  <ProfileRequest />
</cXML>
```

De server reageert met een document ProfileResponse dat hierna wordt beschreven.

ProfileResponse

In het document ProfileResponse worden transacties weergegeven die worden ondersteund door de cXML-server, de locatie van deze transacties en eventuele benoemde opties met een stringwaarde.

```
<ProfileResponse effectiveDate="2000-01-01T05:24:29-08:00">
  <Transaction requestName="OrderRequest">
    <URL>http://workchairs.com/cgi/orders.cgi</URL>
  </Transaction>
  <Transaction requestName="PunchOutSetupRequest">
    <URL>http://workchairs.com/cgi/PunchOut.cgi</URL>
  </Transaction>
</ProfileResponse>
```

Nieuwe statuscodes

Zie

"Status" op pagina 58
voor meer informatie.

De specificatie van cXML 1.1 omvat nieuwe codes voor de status van transacties, waardoor een nauwkeuriger communicatie tussen client en server mogelijk is. Daarnaast bevat de specificatie betere beschrijvingen van bestaande statuscodes van HTTP en cXML.

Nieuw type attribuut voor leden van de marketplace

Het element `Credential` heeft een nieuw typeattribuut, waarmee wordt aangegeven of de zender of ontvanger lid is van de marketplace. Er kunnen verschillende marketplaces zijn en elke marketplace kan andere referentie-eisen stellen.

Zie

"Credential" op pagina 57 voor meer informatie.

Het nieuwe typeattribuut heeft één mogelijke waarde: `marketplace`. U kunt deze waarde gebruiken om referenties van marketplaceleden te onderscheiden van de referenties van gewone inkoop- of leveranciersbedrijven. `Credential`-elementen zonder een typeattribuut identificeren bedrijven die niet met een bepaalde marketplace zijn verbonden.

Bij aanvragen naar of vanuit een marketplace moeten zowel de marketplace als het bedrijf dat lid is van deze marketplace worden geïdentificeerd in de `Credential`-elementen `Aan` of `Van`.

Wijzigingen in Extrinsic-elementen

cXML 1.1 introduceert extra elementen en attributen voor gegevens die eerder in Extrinsic-elementen werden opgenomen. Met deze toevoegingen wordt informatie naar de basisspecificatie verplaatst die eerder met Extrinsic werd verzonden.

cXML 1.1 introduceert ook ondersteuning voor Extrinsic in de kop.

Nieuw element Contact

In de elementen `OrderRequestHeader`, `PunchOutSetupRequest` en `ItemOut` kunnen nu facultatieve `Contact`-elementen worden opgenomen, waarin een persoon of een groep personen wordt weergegeven met wie contact kan worden opgenomen voor aanvullende informatie.

Zie

"Contact" op pagina 74 voor meer informatie.

```
<ItemOut quantity="1">
  <ItemID>
    <SupplierPartID>5555</SupplierPartID>
  </ItemID>
  <ItemDetail>
    <UnitPrice>
      <Money currency="USD">134.00</Money>
    </UnitPrice>
    <Description xml:lang="en">
      <ShortName>Office Chair</ShortName>
      Black leather, with adjustable arms, adjustable height and back angle.
    </Description>
  </ItemDetail>
</ItemOut>
```

```

        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">12345</Classification>
    </ItemDetail>
    <Contact role="customerService">
        <Address>
            <Name xml:lang="en-US">Joe Bob Emmet</Name>
            <Email>joebob@workchairs.com</Email>
            <Phone name="Kantoor">
                <TelephoneNumber>
                    <CountryCode isoCountryCode="US">1</CountryCode>
                    <AreaOrCityCode>800</AreaOrCityCode>
                    <Number>5551212</Number>
                </TelephoneNumber>
            </Phone>
            <Fax name="Order">
                <TelephoneNumber>
                    <CountryCode isoCountryCode="US">1</CountryCode>
                    <AreaOrCityCode>408</AreaOrCityCode>
                    <Number>5551234</Number>
                </TelephoneNumber>
            </Fax>
        </Address>
    </Contact>
</ItemOut>
```

Het attribuut `role` geeft de positie of functie aan van de contactpersoon.
De mogelijkheden zijn: `endUser`, `administrator`, `purchasingAgent`, `technicalSupport`, `customerService` of `sales`.

Attribuut `RequisitionID` ondersteund

Het attribuut `requisitionID` wordt nu volledig ondersteund. Dit is een facultatief attribuut bij de elementen `OrderRequest` en `ItemOut` waarmee de aanvraag van een inkoper voor een regelitem wordt geïdentificeerd.

Zie

"OrderRequestHeader"
op pagina 71 voor meer
informatie.

```

<OrderRequest>
  <OrderRequestHeader
    orderID="DO1234"
    orderDate="2000-03-12T13:30:23+8.00"
    type="nieuw"
    requisitionID="R4321">
    <Total>
      <Money currency="USD">12.34</Money>
    </Total>
    <ShipTo>
      . . .
    </ShipTo>
  </OrderRequestHeader>
```

```
<ItemOut>
...
</ItemOut>
</OrderRequest>
```

U kunt requisitionID niet opnemen bij ItemOut wanneer u dit attribuut gebruikt bij OrderRequestHeader.

Overzicht van Extrinsic-informatie die is verplaatst

In de volgende tabel worden de wijzigingen weergegeven die zijn aangebracht in algemeen gebruikte Extrinsic-elementen.

Old Extrinsic	Bestaand of nieuw CXML-element of -attribuut.
Requested Ship Date	Het attribuut ItemOut requestedDeliveryDate
ship complete	Nieuw attribuut shipComplete
ReqNumber	Bestaand attribuut requisitionID
Requisition #	Bestaand attribuut requisitionID
Name	Nieuw element Contact
Phone	Nieuw element Contact
E-mailAddress	Nieuw element Contact
Buyer Name	Nieuw element Contact
Buyer Phone	Nieuw element Contact
OriginalRequester	Nieuw element Contact
Requester Phone Number	Nieuw element Contact
ETA	Bestaand attribuut requestedDeliveryDate

Extrinsics in de kop

Voorheen konden Extrinsic-elementen uitsluitend worden weergegeven op productregelniveau. Nu kunnen deze elementen overal binnen het document OrderRequest worden weergegeven.

Gebruik deze nieuwe mogelijkheid voor Extrinsic-gegevens die van toepassing zijn op de volledige inkooporder.

Extrinsic-elementen met dezelfde naam kunnen in een document OrderRequest niet tegelijk in de kop en op regelitemniveau voorkomen.

Verbeteringen in punchout-transacties

Punchout-transacties zijn verbeterd om het winkelen op middenpad- en productniveau beter te ondersteunen. cXML ondersteunt nu ook geannuleerde punchout-sessies.

Verbeterd PunchOutSetupRequest

Het document PunchOutSetupRequest dat door inkopers wordt verzonden, is gewijzigd om de flexibiliteit van punchout-transacties te vergroten. De URL die in de PunchOutSetupRequest werd aangegeven, is afgekeurd. cXML-servers negeren dit element voortaan.

De nieuwe methodologie maakt gebruik van de identiteit (uit de Credential) van de leverancier. Hubs van E-commercenetwerken ontvangen het document PunchOutSetupRequest, lezen de ID van de leverancier, zoeken de URL van de punchout-website op in de accountgegevens van de leverancier en verzenden PunchOutSetupRequest naar die URL. De URL van de punchout-website wordt gespecificeerd door de hub van het e-commercenetwerk, niet door de inkoper. Dit zorgt voor meer flexibiliteit.

Hubs van e-commercenetwerken kunnen door leveranciers worden gebruikt om de URL's van hun punchout-websites op te slaan.

Element SelectedItem

Als onderdeel van de uitbreiding van PunchOutSetupRequest biedt cXML nu een betere ondersteuning voor punchout op winkel-, middenpad- en productniveau. In het nieuwe, facultatieve element SelectedItem in dit document kunnen leveranciers punchout opgeven voor een volledige winkel of een subset van productaanbiedingen. Het element SelectedItem kan bij inkooptoepassingen worden opgenomen in PunchOutSetupRequest-documenten. Bij punchout-sites kan dit element worden gebruikt om te bepalen welke producten de gebruikers te zien krijgen. Wanneer het element SelectedItem niet wordt opgenomen, dienen leveranciers alle productaanbiedingen (winkelniveau) te presenteren.

Een SelectedItem bevat het element ItemOut, dat bijvoorbeeld een ItemID bevat:

```
<SelectedItem>
  <ItemID>
    <SupplierPartID>5555</SupplierPartID>
  </ItemID>
</SelectedItem>
```

Zie

"SelectedItem" op
pagina 81 voor meer
informatie.

Inkooptoeepassingen maken voor de inhoud van het element `SelectedItem` gebruik van de `ItemID` (`SupplierPartID` en `SupplierPartAuxiliaryID`) uit de indexcatalogus van de punchout. Wijzigingen in de catalogus zijn niet nodig.

Inkooptoeepassingen dienen in eerste instantie zowel het nieuwe element `SelectedItem` als de oude punchout-URL te verzenden in de `PunchOutSetupRequest`. De oude URL wordt door de hubs van e-commercenetwerken alleen gebruikt voor leveranciers die de bestemming van hun punchout-URL nog niet hebben opgeslagen.

Leeg document `PunchOutOrderMessage`

Bij cXML kunnen nu lege documenten `PunchOutOrderMessage` worden gebruikt waarmee gebruikers hun winkelsessie kunnen beëindigen zonder items te bestellen. Voorheen moest het teruggezonden document `PunchOutOrderMessage` ten minste één item bevatten.

Leveranciers kunnen een knop `Annuleren` implementeren waarmee een leeg document `PunchOutOrderMessage` wordt gegenereerd. Zowel in de punchout-site als in de inkooptoeepassing is dan bekend wanneer een gebruiker een winkelsessie heeft geannuleerd, zodat het winkelwagentje en items uit de aanvraag kunnen worden verwijderd en andere onderhoudstaken kunnen worden uitgevoerd.

Nieuw verborgen veld cXML-base64

Met het nieuwe verborgen veld cXML-base64 worden internationale documenten binnen de FORM POST ondersteund die worden geretourneerd vanuit punchout-sites. Bij cXML-documenten die symbolen van buiten "us-ascii" bevatten, moet dit veld worden gebruikt in plaats van het verborgen veld cXML-urlencoded. Dit alternatief heeft een vrijwel identieke semantiek, maar het gehele document is gedurende het transport base64-gecodeerd, en niet HTML-gecodeerd naar de browser of URL-gecodeerd naar de ontvangende webserver.

Zie

"URL-Form-Encoding" op pagina 63 voor meer informatie.

Met Base64-codering wordt de oorspronkelijke tekencodering van een cXML-document onderhouden. Hoewel geen parameter "charset" bij de verzonden informatie is opgenomen, kan het gecodeerde document (nadat de overdrachts codering is verwijderd) worden gehanteerd als het mediatype "application/xml". Hierdoor kan de ontvangende parser elk coderingsattribuut respecteren dat is gespecificeerd in de XML-definitie. De standaardtekencodering voor dit veld en voor alle application/xml-documenten is UTF-8.

Nieuwe inkooporderfuncties

cXML-inkooporderdocumenten zijn uitgebreid om aangevraagde functies te kunnen ondersteunen.

Nieuw attribuut lineNumber

Het attribuut lineNumber is een nieuw, optioneel attribuut bij het element ItemOut. Hiermee wordt de locatie van een item binnen de inkooporder van een inkoper aangegeven. Dit attribuut wordt gekozen als de order is geplaatst en is dus doorgaans niet relevant bij punchout-sessies.

Zie

"ItemOut" op pagina 76 voor meer informatie.

```
<ItemOut quantity="2" lineNumber="1"
  requestedDeliveryDate="2000-03-12">
  <ItemID>
    <SupplierPartID>11223344</SupplierPartID>
  </ItemID>
  <ItemDetail>
    . . .
  </ItemDetail>
</ItemOut>
```

De regelnummers van items moeten bij orderwijzigingen gelijk blijven om de wijziging te kunnen identificeren.

Bijlagen bij inkooporders

Inkopers lichten hun inkooporders vaak toe met memo's, tekeningen of faxen. In inkooptoepassingen kunnen nu bestanden van elk type aan cXML-inkooporders worden gehecht met behulp van MIME (Multipurpose Internet Mail Extensions).

cXML bevat alleen referenties naar externe MIME-delen die binnen één meerdelige MIME-envelop worden verzonden (bij het cXML-document, bij een e-mailbericht of gezamenlijk per fax).

Het nieuwe element Attachment bevat verwijzingen naar de bijlagen:

Zie

"Verzending van bijlagen" op pagina 55 voor meer informatie.

```
<Comments>
  <Attachment><URL>cid: uniqueCID@buyer.com</Attachment>
  Zie de afbeelding in de bijlage voor mijn opvatting van hoe dit eruit zou moeten zien
</Comments>
```

De bijlagen worden in de hubs van e-commercenetwerken ontvangen en kunnen worden doorgestuurd naar de leverancier of worden opgeslagen zodat deze op een later tijdstip on line kunnen worden opgehaald.

Bezoek de volgende websites voor meer informatie over de MIME-standaard:

www.hunnysoft.com/mime
www.rad.com/networks/1995/mime/mime.htm

Nieuw attribuut shipComplete

Het attribuut shipComplete is een nieuw, facultatief attribuut bij het element OrderRequestHeader waarmee de leverancier wordt geïnstrueerd een order alleen uit te voeren wanneer alle items beschikbaar zijn. Met dit attribuut worden gedeeltelijke leveringen voorkomen.

Zie

"OrderRequestHeader"
op pagina 71 voor meer
informatie.

```
<OrderRequest>
  <OrderRequestHeader
    orderID="DO1234"
    orderDate="2000-03-12T13:30:23+8.00"
    type="nieuw"
    requisitionID="R4321"
    shipComplete="yes">
  <Total>
    <Money currency="USD">12.34</Money>
  </Total>
  <ShipTo>
    . . .
  </ShipTo>
</OrderRequestHeader>
<ItemOut>
  . . .
</ItemOut>
</OrderRequest>
```

Nieuw element ShortName

ShortName is een nieuw, facultatief element binnen de elementen Item Description.

```
<Description xml:lang="en-US">
  <ShortName>Big Computer</ShortName>
  This wonder contains three really big disks, four CD-ROM drives, two Zip drives,
  an Ethernet card, much more memory than you could ever use, four CPUs on two
  motherboards. We'll throw in two monitors, a keyboard and the cheapest mouse we
  can find lying around.
</Description>
```

Zie

"ItemDetail" op pagina 86 voor meer informatie.

ShortName is een korte omschrijving voor een item die past binnen de lijst producten die gebruikers te zien krijgen. De aanbevolen lengte is 30 tekens, maar het maximum aantal is 50 tekens. Voorheen werden lange omschrijvingen willekeurig afgekort, omdat er maar één element Description was.

In inkooptoepassingen en andere cXML-clients dient in velden met beperkte ruimte ShortName te worden weergegeven in plaats van een afkorting van de Descriptiontekst. Wanneer geen ShortName wordt aangegeven, kunnen cXML-clients doorgaan met het afkorten van de Descriptiontekst.

Makers van catalogi dienen ShortName niet te gebruiken als duplicaat van de informatie in Description. In plaats daarvan dient ShortName te worden gebruikt om het product te noemen en Description om productdetails te omschrijven.

De catalogusindeling CIF 3.0 is eveneens uitgebreid ter ondersteuning van ShortName. De naam van het veld is Korte omschrijving.

Nieuwe statustransactie van inkooporders

cXML 1.1 introduceert een nieuwe transactie voor het verzenden van de status van inkooporders naar de hubs van e-commercenetwerken.

Deze transactie steunt op het nieuwe element OrderReference, waarmee de bijwerking van de status wordt verbonden met de OrderRequest die het laatst van de inkoper werd ontvangen.

Nieuw element OrderReference

Met het nieuwe element OrderReference wordt de bijwerking van een status verbonden met een bepaald OrderRequest-document. De vereiste attributen van de OrderRequestHeader en cXML-elementen uit het OrderRequest-document worden herhaald en er wordt een facultatieve identificatie toegevoegd die door de leverancier wordt gegenereerd.

Zie

"DocumentReference" op pagina 88 voor meer informatie.

```
<OrderReference
  payloadID="0c300508b7863dcclb_14999"
  timestamp="2000-01-08T14:36:05-07:00"
  orderID="DO4321"
  orderDate="2000-01-08T13:56:23-07:00"
  supplierOrderID="27-33-00-08-01"
/>
```

Bij de nieuwe transactie StatusUpdateRequest wordt van dit element gebruikgemaakt.

Nieuwe transactie StatusUpdateRequest

Partners bij orderverwerking (zoals fax- of EDI-serviceproviders) zenden de nieuwe transactie StatusUpdateRequest naar de hubs van e-commercenetwerken om de status van de inkooporder in te stellen. Dit beïnvloedt de statusindicator in de hub, die zichtbaar is voor zowel inkopers als leveranciers. Daarnaast kunnen leveranciers deze transactie verzenden om de inkoper in staat te stellen de status van de documentverwerking binnen de organisatie van de leverancier te zien.

Zie

"StatusUpdateRequest" op pagina 88 voor meer informatie.

Opmerking: Met deze transactie worden belanghebbende partijen geïnformeerd over wijzigingen in de leverings- en verwerkingsstatus van *inkooporderdocumenten*, niet over de verzendstatus van werkelijke items.

Eén wijziging is met name van belang. Wanneer een OrderRequest-document met succes vanuit een tussenliggende hub wordt doorgestuurd, kan de oorspronkelijke zender of een vorige hub op de hoogte worden gesteld van de geslaagde actie. De overgangen door verschillende wachtrijen en verwerkingsstappen bij een leverancier of een hub kunnen ook van belang zijn voor de inkoper.

Nieuw element Followup

Followup is een element binnen OrderRequestHeader waarmee de URL wordt opgegeven waarnaar toekomstige StatusUpdateRequest-documenten dienen te worden verzonden. Deze locatie is de invoerlocatie voor latere documenten die verwijzen naar het huidige OrderRequest-document.

Zie

"Followup" op pagina 75 voor meer informatie.

Index

A

Accounting, element 78
Attachment, element 75
attribuut operation 24
attribuut payloadID 24
attribuut timestamp 24

B

bepaling van het land, in cXML-kop 31
bijlagen bij inkooporders 47
BillTo, element 73
BrowserFormPost, element 81
BuyerCookie, element 80, 83

C

Charge, element 78
code, attribuut 59
codering, tekens 54
Comments, element 74
Contact, element 74
Contract, element 94
cookies, inkoper en leverancier 30, 40
corporateURL, attribuut 90
Credential, element 57
cXML, element 52
cxml.org, website 8
cXML-base64, verborgen veld 38, 65
cXML-urlencoded, verborgen veld 38, 64

D

datum- en tijdsnotatie 53
deploymentMode, attribuut 58, 62
Description, element 86
Distribution, element 78
DocumentReference, element 88
domain, attribuut 57
DTD's (Document Type Definitions) 8

E

EDI (X.12 Electronic Data Interchange) 4
editors voor XML 10
effectiveDate, attribuut 69, 94
Element Classification 22
Element Description 21
Element From 24
Element SelectedItem 27
Element Sender 24
Element SupplierID 21
Element SupplierPartAuxiliaryID
(Leverancierscookie) 30, 40
Element To 24
expirationDate, attribuut 94
Extrinsic, element 75, 81
Extrinsic-element 27, 41

F

Followup, element 75
form encoding 38
formuliercodering 64
From, To en Sender, elementen 57

G

GetPendingRequest, element 102
GetPendingResponse, element 103

H

Header, element 56
HTML form encoding 38
HTML-formuliercodering 64
hulpprogramma's voor gebruik bij XML 10
hulpprogramma's voor het werken met XML 10

I

id, attribuut 78
Index, element 92
IndexItemAdd, element 93
IndexItemDelete, element 93
IndexItemDetail, element 94
IndexItemPunchout, element 93
inkooporders 43–47
 bijlagen 47
inkoper en leverancier, cookies 30, 40
inReplyTo, attribuut 62
IsoCountryCode, element 66
IsoLanguageCode, element 66
ItemDetail, element 86, 94
ItemID, element 86
ItemIn, element 85
ItemOut, element 76
ItemSegment, element 95

L

lastReceivedTimestamp, attribuut 102
Launch Page 31
leverancier en inkoper, cookies 30, 40
lineNumber, attribuut 76, 85

M

Maateenheid 30, 67
maxMessages, attribuut 102
Message, element 62
MIME-bijlagen 47, 54

O

operation, attribuut 80
operationAllowed, attribuut 84
Order Receiver Page 39
orderDate, attribuut 72
orderID, attribuut 72, 88
OrderMethods, element 92
OrderRequest, element 70
OrderRequestHeader, element 71
orders boeken 20
orders noteren 19

P

payloadID, attribuut 52, 88
Payment, element 73
Profieltransactie 9
ProfileRequest, element 68
ProfileResponse, element 69
PunchoutDetail, element 94
PunchOutOrderMessage 29
PunchOutOrderMessage, element 83
PunchOutOrderMessageHeader, element 84
PunchOutSetupRequest 23
PunchOutSetupRequest, element 79
PunchOutSetupResponse 28
PunchOutSetupResponse, element 82

Q

quantity, attribuut 76, 85

R

Request, element 58
requestedDeliveryDate, attribuut 76
requestName, attribuut 70
requisitionID, attribuut 72, 76
Response, element 58
role, attribuut 74

S

- segmentKey, attribuut 95
- SelectedItem, element 81
- Sender Page 35
- Sender, To en From, elementen 57
- servers pingen met de profieltransactie 9
- shipComplete, attribuut 72
- Shipping, element 73
- ShipTo, element 73
- ShortName, element 86
- Start Page 35
- StartPage, element 82
- Status, element 58
- StatusUpdateRequest, element 88
- storeFrontURL, attribuut 90
- Subscription, element 99
- SubscriptionContentRequest, element 100
- SubscriptionContentResponse, element 101
- SubscriptionListRequest, element 100
- SubscriptionListResponse, element 100
- Supplier, element 90
- SupplierChangeMessage, element 98
- SupplierDataRequest, element 97
- SupplierDataResponse, element 97
- SupplierListRequest, element 96
- SupplierListResponse, element 96
- SupplierLocation, element 91
- SupplierPartAuxiliaryID, element
(leverancierscookie) 93
- SupplierSetup, element 82
- SupplierSetup-URL 27

T

- taal, in cXML-kop 31
- Tax, element 73
- tekencodering 54
- tijd- en datumnotatie 53
- timestamp, attribuut 52
- To, From en Sender, elementen 57
- Total, element 73
- Transaction, element 70
- type, attribuut 57

U

- URL, element 67
- URL-encoded 64

V

- valideren, cXML 8
- version, attribuut 52

X

- XML 15
- xml:lang 31
- xmllanguageCode, element 67



www.cxml.org