



# **Guide de l'utilisateur du langage cXML**

**VERSION 1.1**

**AVRIL 2000**

Ariba, Inc. (Ariba) vous accorde par la présente un droit et une licence d'utilisation perpétuelle, non-exclusive, gratuite et internationale de la spécification cXML (ci-après la « Spécification ») conformément aux copyrights Ariba indiqués dans la Spécification en matière d'utilisation, de publication, de modification et de distribution de la Spécification. Ariba consent en outre à vous octroyer gratuitement, en vertu des droits sur la propriété intellectuelle applicables à Ariba, une licence d'implémentation et d'utilisation des balises cXML et des modèles de structure inclus dans la Spécification à des fins de création de programmes informatiques conformes à de tels modèles. Votre engagement à ne pas vous prévaloir de droits sur la propriété intellectuelle à l'encontre d'Ariba et autres sociétés implémentant la Spécification constitue une condition préalable à l'octroi de ladite licence. Ariba se réserve expressément tous les autres droits auxquels elle peut prétendre tant au niveau de la forme que du contenu de cette Spécification. Ariba ne donne aucune garantie concernant cette Spécification, et ne garantit en particulier en aucune façon que cette Spécification ou ses implémentations ne violent pas les droits d'autres spécifications. Cette Spécification est fournie « en l'état » sans aucune garantie expresse ou tacite d'aucune sorte. Si vous publiez, copiez ou distribuez cette Spécification, vous devez impérativement lui joindre le présent copyright ; si vous modifiez cette Spécification, toutefois, vous n'avez pas l'obligation de faire figurer le terme « cXML » dans le nouveau nom de la spécification modifiée. Si vous faites part de certaines remarques ou suggestions à Ariba, et si Ariba en tient compte pour modifier la spécification cXML, Ariba reste propriétaire de la version cXML modifiée.

Les informations contenues dans ce document pourront être modifiées sans notification préalable.

---

# Table de matières

|   |          |
|---|----------|
| <b>Préface</b> .....                                    | <b>v</b> |
| Public concerné et conditions requises .....            | v        |
| Chapitres à lire en priorité .....                      | vi       |
| <br>  |          |
| <b>Chapitre 1</b>                                       |          |
| <b>Introduction au langage cXML</b> .....               | <b>1</b> |
| Possibilités du langage cXML .....                      | 1        |
| Catalogues .....  | 2        |
| Punchout .....  | 3        |
| Commandes d'achat .....                                 | 4        |
| Types d'applications qui utilisent le format cXML ..... | 5        |
| Applications de gestion des approvisionnements .....    | 5        |
| Plates-formes réseau commerciales .....                 | 5        |
| Catalogues punchout .....                               | 6        |
| Systèmes de réception des commandes .....               | 6        |
| Stratégie de distribution du contenu .....              | 6        |
| Validation par rapport aux DTD .....                    | 7        |
| Obtention de DTD cXML .....                             | 8        |
| Exécution d'une validation .....                        | 8        |
| Transaction Profile .....                               | 8        |
| Utilitaires XML .....                                   | 9        |

## Chapitre 2

|   |           |
|---|-----------|
| <b>Mise en œuvre d'un site punchout . . . . .</b>                 | <b>11</b> |
| Conditions requises par la mise en œuvre du punchout . . . . .    | 11        |
| Organisations acheteuses . . . . .                                | 11        |
| Fournisseurs . . . . .  | 13        |
| Séquence des événements punchout . . . . .                        | 16        |
| Étapes 1 et 2 : requête punchout . . . . .                        | 16        |
| Étape 3 : sélection de produits . . . . .                         | 17        |
| Étape 4 : passage au contrôle . . . . .                           | 18        |
| Étape 5 : transmission de la commande d'achat . . . . .           | 19        |
| Documents punchout . . . . .                                      | 21        |
| Catalogue-index punchout . . . . .                                | 21        |
| Document PunchOutSetupRequest . . . . .                           | 22        |
| Document PunchOutSetupResponse . . . . .                          | 27        |
| Document PunchOutOrderMessage . . . . .                           | 28        |
| Modification des pages Web . . . . .                              | 30        |
| Page de lancement . . . . .                                       | 31        |
| Page de démarrage . . . . .                                       | 34        |
| Page de l'expéditeur . . . . .                                    | 35        |
| Page du destinataire de la commande . . . . .                     | 38        |
| Suggestions sur la mise en œuvre d'un site Web punchout . . . . . | 39        |
| Instructions . . . . .  | 39        |
| Cookies d'acheteur et de fournisseur . . . . .                    | 40        |
| Personnalisation . . . . .  | 40        |

## Chapitre 3

|  |           |
|--|-----------|
| <b>Réception de commandes d'achat cXML . . . . .</b>   | <b>41</b> |
| Traitement des commandes d'achat . . . . .             | 41        |
| Réception de commandes d'achat . . . . .               | 42        |
| OrderRequest . . . . .                                 | 42        |
| OrderResponse . . . . .                                | 45        |
| Acceptation des pièces jointes aux commandes . . . . . | 45        |

---

|  |           |
|--|-----------|
| <b>Annexe A</b>                                |           |
| <b>Spécification du langage cXML</b> .....     | <b>47</b> |
| Spécification du protocole .....               | 48        |
| Modèle requête/réponse .....                   | 48        |
| Conventions XML .....                          | 49        |
| Enveloppe cXML .....                           | 50        |
| Couches imbriquées .....                       | 52        |
| Header .....                                   | 54        |
| Request .....                                  | 57        |
| Response .....                                 | 57        |
| Modèle unidirectionnel (asynchrone) .....      | 60        |
| Éléments de base .....                         | 65        |
| Entités types .....                            | 65        |
| Éléments de base .....                         | 66        |
| Transaction Profile .....                      | 67        |
| ProfileRequest .....                           | 67        |
| ProfileResponse .....                          | 67        |
| Définition des commandes .....                 | 69        |
| OrderRequest .....                             | 69        |
| Réponse à un document OrderRequest .....       | 78        |
| Transaction Punchout .....                     | 78        |
| PunchOutSetupRequest .....                     | 78        |
| PunchOutSetupResponse .....                    | 81        |
| PunchOutOrderMessage .....                     | 81        |
| Modifications de statut ultérieures .....      | 86        |
| DocumentReference .....                        | 86        |
| StatusUpdateRequest .....                      | 87        |
| Définitions de catalogue .....                 | 88        |
| Supplier .....                                 | 89        |
| Index .....                                    | 91        |
| Contract .....                                 | 93        |
| Définitions de gestion des abonnements .....   | 94        |
| Données fournisseur .....                      | 95        |
| Abonnements aux catalogues .....               | 98        |
| Définitions de récupération des messages ..... | 101       |
| GetPendingRequest .....                        | 101       |
| GetPendingResponse .....                       | 102       |

|  |            |
|--|------------|
| <b>Annexe B</b>  |            |
| <b>Nouveautés de la version 1.1 de cXML</b>                          | <b>103</b> |
| Modifications générales du langage cXML                              | 103        |
| Amélioration de la prise en charge multilingue                       | 104        |
| Centralisation des DTD   | 104        |
| Nouvelle transaction Profile   | 104        |
| Nouveaux codes de statut   | 105        |
| Nouvel attribut type pour les membres des espaces marchands          | 106        |
| Modifications apportées aux éléments Extrinsic                       | 106        |
| Nouvel élément Contact   | 106        |
| Prise en charge de l'attribut requisitionID                          | 107        |
| Récapitulatif des informations déplacées dans les éléments Extrinsic | 108        |
| Éléments Extrinsic de niveau en-tête                                 | 108        |
| Améliorations apportées aux transactions Punchout                    | 108        |
| Optimisation du document PunchOutSetupRequest                        | 109        |
| Élément SelectedItem   | 109        |
| Document PunchOutOrderMessage vide                                   | 110        |
| Nouveau champ masqué cXML-base64                                     | 110        |
| Nouvelles fonctionnalités des commandes d'achat                      | 110        |
| Nouvel attribut lineNumber   | 111        |
| Pièces jointes aux commandes d'achat                                 | 111        |
| Nouvel attribut shipComplete   | 112        |
| Nouvel élément ShortName   | 112        |
| Nouvelle transaction : statut des commandes d'achat                  | 113        |
| Nouvel élément OrderReference  | 113        |
| Nouvelle transaction StatusUpdateRequest                             | 114        |
| Nouvel élément Followup  | 114        |
| <br>   |            |
| <b>Index</b>   | <b>115</b> |

---

# Préface

Cet ouvrage se propose d'expliquer comment utiliser le langage cXML (commerce eXtensible Markup Language) dans le cadre du transfert de données associées au commerce électronique.

## Public concerné et conditions requises

---

Cet ouvrage s'adresse aux programmeurs concevant des applications compatibles cXML. Il est destiné plus particulièrement aux fournisseurs qui souhaitent mettre en œuvre un traitement punchout sur leurs sites Web de commerce électronique.

Le langage cXML est un langage ouvert et polyvalent qui assure les transactions nécessaires aux éléments suivants :

- Catalogues de produits électroniques
- Catalogues punchout cXML
- Applications de gestion des approvisionnements
- Communautés acheteuses

Le lecteur doit avoir une connaissance préalable des concepts sous-jacents au commerce électronique et du standard HTTP en matière de communication sur le Web.

Cet ouvrage n'a pas pour but de décrire l'utilisation de mises en œuvre spécifiques telles que des applications de gestion des approvisionnements ou des concentrateurs de commerce électronique en réseau.

---

## Chapitres à lire en priorité

---

- **Responsables des activités de commerce électronique** – Pour une description générale des possibilités du langage cXML, consultez le Chapitre 1, « Introduction au langage cXML ».
- **Programmeurs Web** – Tout programmeur Web mettant en place des sites de commerce électronique doit lire l'ensemble des chapitres.
- **Administrateurs de sites punchout** – Les ingénieurs Web déjà familiarisés avec les sites Web punchout peuvent consulter l'Annexe B, « Nouveautés de la version 1.1 du langage cXML ».



---

# Chapitre 1

## Introduction au langage cXML

Le présent chapitre constitue une introduction à la mise en œuvre du langage cXML (commerce eXtensible Markup Language) dans les transactions liées au commerce électronique.

Il offre une description du langage cXML et aborde plus précisément les questions suivantes :

- Possibilités du langage cXML
- Types d'applications qui utilisent le format cXML
- Stratégie de distribution du contenu
- Validation par rapport aux DTD
- Transaction Profile
- Utilitaires XML

### Possibilités du langage cXML

---

Le format cXML permet aux acheteurs, fournisseurs, syndicaux et intermédiaires de communiquer au moyen d'un langage unique, normalisé et ouvert.

Le succès d'un portail de commerce électronique de professionnel à professionnel (commerce électronique B2B) repose sur un protocole souple et adopté par le plus grand nombre d'intervenants possible. Le langage cXML est à la fois parfaitement défini, robuste, spécialement conçu pour le commerce électronique B2B et il a la faveur des acheteurs et fournisseurs de gros volumes : il offre donc incontestablement le meilleur accès possible à vos produits et à vos services.

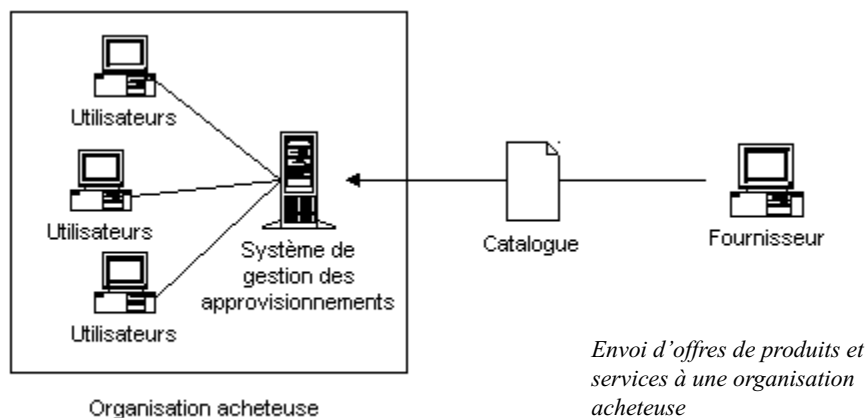
Les transactions cXML se composent de *documents*, en d'autres termes, de fichiers texte simples dont le format et le contenu sont précisément définis. La plupart des documents cXML s'apparentent aux documents papier généralement utilisés dans le commerce traditionnel.

Les paragraphes qui suivent décrivent les principaux types de documents cXML.

## Catalogues

Les catalogues sont des fichiers qui transmettent des offres de produits et de services (ou contenus) à des organisations acheteuses. Ils décrivent les produits et services que vous proposez et les tarifs que vous pratiquez ; ils constituent donc le principal moyen de communication entre vous et vos clients.

Vous créez des catalogues pour faire en sorte que les organisations qui ont recours à des applications de gestion des approvisionnements puissent accéder à vos offres de produits et de services et vous passer des commandes. Les applications d'approvisionnement lisent vos catalogues et les stockent dans leurs bases de données internes. Dès lors que vos catalogues ont été approuvés par une organisation acheteuse, votre offre de contenu est visible par les utilisateurs, qui peuvent alors sélectionner les articles qui les intéressent et les ajouter à des demandes d'achat.



Vous pouvez créer des catalogues pour tous les produits et services de votre choix, indépendamment de leurs méthodes de mesure, de tarification ou de livraison.

Il existe, pour chaque article référencé dans un catalogue, des informations de base obligatoires et des informations facultatives qui permettent d'appliquer des fonctions évoluées aux catalogues, telles les descriptions en plusieurs langues.

## Punchout

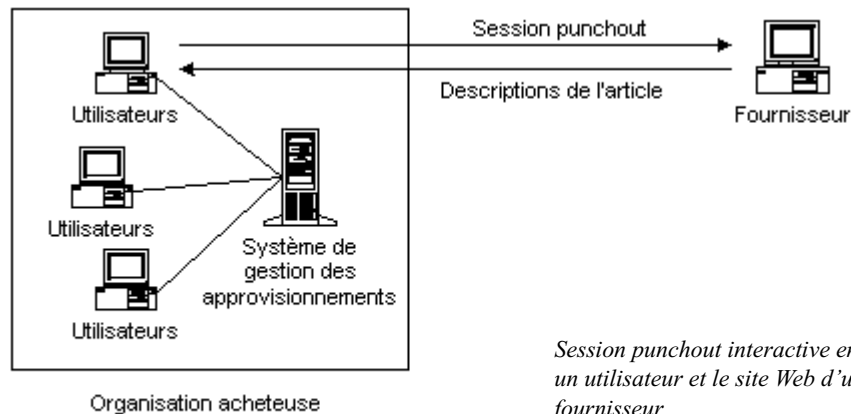
Le punchout constitue une alternative aux fichiers de catalogue statiques. Les sites punchout sont des catalogues interactifs exécutés sur votre site Web.

Si vous possédez un site Web de commerce électronique, vous pouvez le modifier en vue d'une prise en charge des fonctionnalités punchout. Sur Internet, les sites punchout communiquent avec les systèmes de gestion des approvisionnements via cXML.

Pour plus d'informations :

Chapitre 2, « Mise en œuvre d'un site punchout ».

Dans le cas des sites punchout, les applications de gestion des approvisionnements affichent un bouton au lieu de détails sur les produits ou sur les prix. Lorsque les utilisateurs cliquent sur ce bouton, leur navigateur Web affiche des pages de votre site Web local. Selon la façon dont vous mettez ces pages en œuvre, les utilisateurs peuvent consulter les options des produits, spécifier des configurations et sélectionner des méthodes de livraison. Une fois leurs sélections effectuées, les utilisateurs cliquent sur un bouton qui envoie les informations de commande à l'application de gestion des approvisionnements. Les produits entièrement configurés et leurs prix s'affichent alors sur les demandes d'achat des utilisateurs.



*Session punchout interactive entre un utilisateur et le site Web d'un fournisseur*

Votre site Web peut proposer des produits et des prix ayant fait l'objet d'un accord préalable.

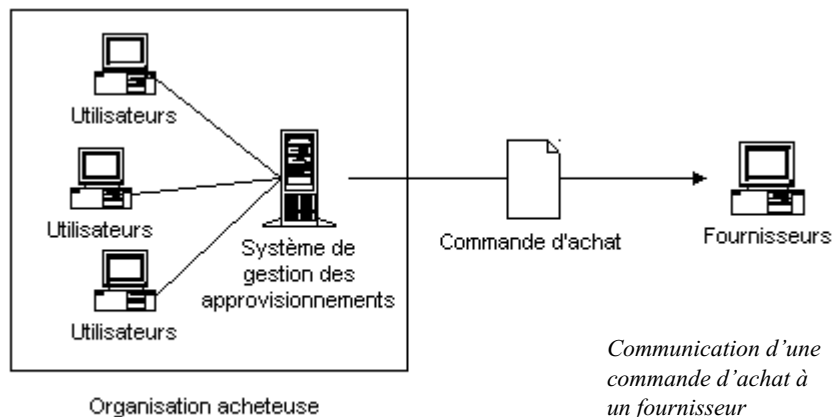
## Commandes d'achat

Les organisations acheteuses envoient des commandes d'achat aux fournisseurs pour leur demander l'exécution d'un contrat.

Pour plus d'informations :

Chapitre 3, « Réception de commandes d'achat cXML ».

Ces commandes peuvent être transmises aux fournisseurs par l'intermédiaire d'une plate-forme réseau de commerce électronique telle Ariba Network.



Le format cXML n'est pas le seul format possible pour les commandes d'achat : le courrier électronique, la télécopie et la transaction EDI (X.12 Electronic Data Interchange) sont également des options envisageables. C'est toutefois le format idéal pour les commandes d'achat compte tenu de sa souplesse, de son faible coût de mise en œuvre, de sa prise en charge inégalée d'une grande diversité de données et de pièces jointes et de sa possibilité d'intégration à des systèmes automatisés.

## Types d'applications qui utilisent le format cXML

---

Le format cXML peut être exploité par toutes les applications de commerce électronique. Les organisations acheteuses, les communautés d'achat verticales et horizontales, les fournisseurs et les éditeurs de logiciels y font aujourd'hui largement appel.

Les paragraphes qui suivent décrivent les principaux types d'applications qui ont actuellement recours à la spécification cXML.

### Applications de gestion des approvisionnements

Les applications de gestion des approvisionnements, telles Ariba ORMS (Operating Resource Management System) et Ariba IBX (Internet Business eXchange), font appel au format cXML pour les transactions externes.

Ariba ORMS est une application professionnelle. Elle est mise en œuvre dans les organisations de grande taille et accessible aux employés par le biais d'un intranet.

Ariba IBX est un service Internet qui permet la création de communautés acheteuses composées de plusieurs petites à moyennes entreprises.

Ces applications permettent à des communautés d'utilisateurs d'acheter des produits et des services contractuels auprès de fournisseurs agréés par leurs responsables des achats. Les demandes d'achat sont tout d'abord approuvées par les responsables au sein des communautés, et les commandes d'achat approuvées sont ensuite transmises aux fournisseurs par différents moyens de communication possibles, parmi lesquels les transactions cXML via Internet.

### Plates-formes réseau commerciales

Les plates-formes réseau commerciales, telles Ariba Network, sont des services Web permettant de relier des acheteurs à des fournisseurs.

Ces services Web remplissent une grande diversité de fonctions, parmi lesquelles la validation de catalogues et la gestion de fichiers, la publication de catalogues et d'abonnements, le routage automatique des commandes d'achat et l'historique des commandes d'achat.

La communication entre ces services Web, les applications des acheteurs et les applications des fournisseurs peut entièrement être réalisée par le biais de transactions cXML sur Internet.

## Catalogues punchout

Pour plus d'informations :

Chapitre 2, « Mise en œuvre d'un site punchout ».

Comme indiqué plus haut, les catalogues punchout sont des catalogues interactifs, disponibles sur les sites Web des fournisseurs. Ce sont des applications de serveur Web, écrites dans un langage de programmation tel ASP (Active Server Pages), JavaScript ou CGI, et qui gèrent les sessions punchout des acheteurs.

Les catalogues punchout acceptent les demandes punchout qui émanent des applications de gestion des approvisionnements, identifient l'organisation acheteuse et affichent les produits et prix voulus au format HTML. Les utilisateurs sélectionnent ensuite des articles, les configurent et choisissent des options, le cas échéant.

Au terme de la session punchout, le site punchout envoie des descriptions des sélections effectuées par les utilisateurs, au format cXML, aux applications de gestion des approvisionnements.

## Systèmes de réception des commandes

Pour plus d'informations :

Chapitre 3, « Réception de commandes d'achat cXML ».

Les systèmes de réception des commandes sont des applications qui acceptent et traitent les commandes d'achat envoyées par les organisations acheteuses sur les sites des fournisseurs. Tout système automatisé, tel qu'un système de gestion des stocks, un système d'exécution des commandes ou un système de traitement des commandes, peut faire office de système de réception des commandes.

L'extraction d'informations à partir de commandes d'achat cXML est très simple, et il est donc relativement aisé de créer les adaptateurs nécessaires à leur prise en charge par les systèmes de réception des commandes existants.

## Stratégie de distribution du contenu

---

Les applications de gestion des approvisionnements présentent des offres de produits et de services aux utilisateurs. Les fournisseurs tiennent généralement à contrôler la façon dont leurs produits et services sont présentés aux utilisateurs, car la présentation joue un rôle essentiel dans le processus de vente. Les organisations acheteuses souhaitent quant à elles que le contenu soit facilement accessible et consultable pour que le respect des contrats soit le plus fidèle possible.

Les organisations acheteuses et les fournisseurs ont le choix entre différentes méthodes de distribution des produits et services offerts. La méthode à employer est déterminée d'un commun accord entre une organisation acheteuse et un fournisseur, selon la nature des produits ou des services échangés.

Le tableau ci-après répertorie quelques exemples de catégories de produits et services communément échangés, en précisant leur méthode de distribution de prédilection.

| Marchandise   | Propriétés                                 | Méthode de distribution du contenu                                    |
|---|--|---|
| Fournitures de bureau, fournitures internes   | Contenu statique, prix stables             | Catalogues statiques  |
| Fournitures de laboratoire, fournitures de fabrication et d'entretien, composants électroniques | Standardisation impérative                 | Punchout vers un portail de marchandises vertical                     |
| Livres, produits chimiques  | Grand nombre d'articles                    | Punchout vers un site hébergé par un fournisseur                      |
| Ordinateurs, équipement réseau, périphériques   | Plusieurs configurations possibles         | Punchout vers l'outil de configuration d'un fournisseur               |
| Services, documents imprimés  | Contenu des attributs extrêmement variable | Punchout vers un formulaire électronique sur le site d'un fournisseur |

Les organisations acheteuses ont la possibilité d'opter pour un stockage local du contenu au sein de l'organisation même ou pour un accès à distance sur Internet, par punchout. Les catalogues cXML gèrent ces deux stratégies de stockage.

Comme l'indique le tableau ci-dessus, le punchout offre une structure souple à partir de laquelle les fournisseurs peuvent proposer un contenu personnalisé, en fonction de la marchandise ou du client. Cette stratégie de distribution du contenu a pour objectif de permettre aux acheteurs et aux fournisseurs d'échanger des données de catalogue à l'aide de la méthode la plus adéquate.

## Validation par rapport aux DTD

Le langage cXML étant un langage XML, il est explicitement défini par un ensemble de définitions de type de document (DTD). Ces DTD sont des fichiers texte qui décrivent précisément la syntaxe et l'ordre des éléments cXML. Les DTD permettent aux applications de valider le document cXML lu ou écrit.

La validation des documents cXML par les applications cXML n'est pas obligatoire, mais fortement recommandée.

## Obtention de DTD cXML

Des DTD correspondant aux différentes versions du langage cXML sont disponibles sur le site [cxml.org](http://cxml.org), à l'adresse :

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

où *<version>* représente le numéro de version cXML complet : 1.1.007, par exemple.

## Exécution d'une validation

Vos applications peuvent faire appel à ces DTD pour valider tous les documents cXML entrants et sortants. Des applications de validation XML sont accessibles sur le Web. Microsoft Internet Explorer 5 dispose d'une fonction de validation XML intégrée.

Pour sécuriser le traitement des transactions, validez tous les documents cXML reçus. Si vous détectez des erreurs, indiquez le code d'erreur correspondant pour que l'expéditeur puisse effectuer une nouvelle transmission.

Pour optimiser les performances, les clients cXML doivent éviter de récupérer des DTD chaque fois qu'ils analysent des documents cXML. Il est préférable qu'ils identifient la version cXML dans les en-têtes des documents et qu'ils aillent rechercher les DTD qui n'ont pas encore été stockées localement.

## Transaction Profile

---

La transaction Profile communique des informations de base sur les serveurs cXML. C'est la seule transaction qui doit impérativement être prise en charge par tous les serveurs cXML.

Cette transaction se compose de deux documents, ProfileRequest et ProfileResponse. Ces documents permettent d'identifier les caractéristiques du serveur, et notamment la version cXML prise en charge, les transactions gérées et les options possibles pour ces transactions.

**Remarque :** Tous les serveurs cXML 1.1 **doivent** gérer la transaction Profile.

La transaction Profile permet aux clients d'interroger les serveurs via une commande « ping » pour vérifier qu'ils sont disponibles.



### ***ProfileRequest***

Le document ProfileRequest est vide de tout contenu. Il est simplement acheminé jusqu'au serveur cXML spécifié.

### ***ProfileResponse***

Le serveur répond par le biais d'un document ProfileResponse, qui répertorie les transactions prises en charge par le serveur cXML, leur emplacement, ainsi que les éventuelles options nommées, représentées par une valeur de type chaîne.

## **Utilitaires XML**

---

Vous pouvez acheter ou vous procurer gratuitement sur le Web un certain nombre d'utilitaires d'édition et de validation de fichiers XML. Nous en avons recensé quelques-uns ci-après :

- **Internet Explorer 5** de Microsoft. Navigateur Web compatible XML capable de valider des fichiers XML à partir de DTD.

[www.microsoft.com/windows/ie/default.htm](http://www.microsoft.com/windows/ie/default.htm)

- **XML Notepad** de Microsoft. Éditeur XML simple.

[msdn.microsoft.com/xml/notepad/intro.asp](http://msdn.microsoft.com/xml/notepad/intro.asp)

- **XML Authority** d'Extensibility. Éditeur DTD XML, offrant des vues hiérarchiques et graphiques.

[www.extensibility.com](http://www.extensibility.com)

- **XML Spy** d'Icon Information Systems. Outil de gestion de DTD et de fichiers XML, présentant des vues grille, source et navigateur.

[www.icon-is.com](http://www.icon-is.com)

- **XMetaL** de Softquad Software. Outil de création XML personnalisable.

[www.softquad.com](http://www.softquad.com)

- **CLIP** de Techno2000 USA. Outil de création XML d'une grande facilité d'utilisation, proposant une assistance aux procédures d'édition.

[www.t2000-usa.com](http://www.t2000-usa.com)

- **XMLwriter** de Wattle Software. Outil de création XML graphique adapté à la gestion de projets XML.

[www.xmlwriter.net](http://www.xmlwriter.net)

Vous trouverez également d'autres outils XML sur les sites Web suivants :

[www.xmlsoftware.com](http://www.xmlsoftware.com)

[www.xml.com/pub/pt/Editors](http://www.xml.com/pub/pt/Editors)

---

# Chapitre 2

## Mise en œuvre d'un site punchout

Avec la technique de punchout, les utilisateurs des applications de gestion des approvisionnements ont la possibilité d'accéder directement à des contrats de fournisseurs placés sur votre site Web et relatifs à divers produits ou services. Ce système élimine de ce fait la nécessité d'envoyer des catalogues complets aux organisations acheteuses. Il vous suffit de leur faire parvenir des fichiers concis contenant une description de l'ensemble de votre magasin, des catégories de produits ou tout simplement des produits que vous proposez.

Ce chapitre décrit les modifications à apporter à un site Web pour que celui-ci prenne en charge punchout. Il aborde plus précisément les rubriques suivantes :

- Conditions requises par la mise en œuvre du punchout
- Séquence des événements punchout
- Documents punchout
- Modification des pages Web
- Suggestions sur la mise en œuvre d'un site Web punchout

### Conditions requises par la mise en œuvre du punchout

---

Avant qu'une organisation acheteuse ne configure son application de gestion des approvisionnements pour la prise en charge du punchout ou qu'un fournisseur ne décide de mettre en œuvre un site Web punchout, il est important que ces deux parties évaluent les avantages du punchout ainsi que les spécifications techniques requises par ce système.

### Organisations acheteuses

Moins d'une journée suffit pour mettre en service une application de gestion des approvisionnements compatible cXML et effectuer les tests avec le site Web punchout d'un fournisseur.

Du côté des organisations acheteuses, il y a peu d'obstacles techniques à l'intégration ; la décision d'utiliser ce système repose donc sur des considérations commerciales tenant compte des pratiques de la société et du type de marchandises achetées. (Consultez la section « Stratégie de distribution du contenu », page 6 pour obtenir une liste des marchandises convenant particulièrement à un traitement punchout.)

### **Considérations commerciales**

Toute organisation acheteuse doit se poser les questions suivantes :

- Les acheteurs et les approbateurs impliqués disposent-ils d'un accès à Internet ? Si tel n'est pas le cas, la mise en place d'un accès contrôlé à Internet est-elle envisageable ?
- L'organisation acheteuse souhaite-t-elle que ses fournisseurs se chargent de la création et de la mise à jour continue de leurs catalogues (notamment en ce qui concerne les prix) ?
- Les acheteurs ont-ils d'ores et déjà recours à Internet pour s'approvisionner en marchandises ? Si tel est le cas, les marchandises acquises par cette voie requièrent-elles la mise en place d'un outil de configuration côté fournisseur ou comportent-elles des attributs trop spécifiques pour s'adapter à un modèle à contenu statique ?
- L'organisation acheteuse a-t-elle recours à des syndicataires de catalogues (Aspect, TPN Register ou Harbinger, par exemple) ?
- L'organisation acheteuse utilise-t-elle Internet pour obtenir des services (conseil, travail temporaire ou maintenance, par exemple) ?
- L'organisation acheteuse sélectionne-t-elle d'ores et déjà ses fournisseurs via Internet ?

Si l'une des réponses à ces questions est affirmative, punchout constitue peut-être une solution adaptée à l'organisation acheteuse.

### **Considérations techniques**

L'organisation acheteuse doit disposer d'un système répondant aux spécifications techniques suivantes :

- Accès direct à Internet : les utilisateurs au sein de l'organisation acheteuse doivent disposer d'un accès direct à Internet. Le punchout s'appuie sur des sessions classiques de connexion par navigateur, où l'utilisateur interagit en direct avec le site Web de ses fournisseurs. Cette communication est établie via une infrastructure Internet standard, et non via l'application de gestion des approvisionnements.

- Connexion Internet fiable : l'accès à Internet doit être continuellement opérationnel et fiable. Si les utilisateurs ne peuvent pas s'approvisionner en raison de défaillances de la connexion, ils risquent d'effectuer des achats non conformes.
- Contrats avec des fournisseurs proposant des fonctions de punchout : les agents d'approvisionnement doivent préalablement avoir conclu des contrats avec des fournisseurs capables d'assurer le traitement du punchout. Les sites Web punchout ne sont accessibles qu'aux organisations acheteuses connues et authentifiées.

## Fournisseurs

Le terme *fournisseur* dans le contexte du punchout dépasse la définition traditionnelle du terme. Le protocole punchout a été conçu afin d'être un système souple, capable de transmettre des données sur pratiquement tout type de produits ou services proposés par la plupart des fournisseurs, distributeurs, syndicaux ou fabricants existants.

Notons par exemple les produits et services suivants :

- Ordinateurs provenant directement d'un fabricant ou d'un revendeur
- Produits chimiques ou réactifs fournis par un syndicaux
- Fournitures de bureau proposées par un distributeur
- Services sous contrat d'une agence de travail temporaire

Vous disposez peut-être déjà d'un site Web transactionnel pouvant héberger des contenus divers et recevoir des commandes d'achat. Si tel est le cas, il vous faut à présent prendre en compte les aspects commerciaux et techniques de votre société avant de décider de la mise en œuvre du punchout.

### Considérations commerciales

Tout fournisseur doit se poser les questions suivantes :

- Vendez-vous d'ores et déjà vos produits ou services via Internet ? Dans ce cas, proposez-vous des produits ou des services spécifiques à chaque client (soumis à une tarification contractuelle) sur votre site Web ?
- Vos produits et vos services appartiennent-ils à l'une des catégories punchout décrites dans le tableau de la section « Stratégie de distribution du contenu », page 6 ? Pour mémoire, ces catégories incluent :
  - les produits hautement configurables, tels que les ordinateurs ;
  - les lignes de produits vastes, telles que les livres ;
  - les produits dotés d'attributs très spécifiques, tels que les produits chimiques ;
  - les données normalisées, telles que les fournitures de fabrication et d'entretien.
- Préférez-vous recevoir une commande d'achat et/ou son paiement via votre site Web ?

Si l'une des réponses à ces questions est affirmative, le punchout constitue peut-être une solution adaptée à votre société.

### **Considérations techniques**

Le fournisseur doit disposer d'un système répondant aux spécifications techniques ci-dessous :

- Connexion Internet fiable : il est essentiel que l'infrastructure du serveur Web et la connexion Internet soient fiables. Si un utilisateur ne parvient pas à accéder à un contenu distant, il est probable qu'il s'adressera à un autre fournisseur.
- Administrateurs Web compétents : un site punchout et les applications de prise en charge requièrent une maintenance et une mise à jour régulières. Les besoins des utilisateurs et vos propres offres de produits évoluent dans le temps ; vous devez donc prévoir des ressources en personnel pour modifier l'infrastructure de punchout.
- Prise en charge des transactions élémentaires : il n'est pas obligatoire qu'un site punchout prenne en charge toutes les commandes cXML ; cependant, les opérations suivantes sont indispensables :

Transaction Profile  
PunchOutSetupRequest  
PunchOutSetupResponse  
PunchOutOrderMessage

### **Estimation de la charge de travail impliquée**

Le tableau ci-dessous contient une estimation de la charge de travail impliquée par l'intégration des fonctions de punchout cXML. Cette estimation a été effectuée par des fournisseurs :

| <b>Niveau de l'infrastructure existante</b>                   | <b>Estimation du temps requis</b>  |
|---|--|
| Site transactionnel doté d'une infrastructure XML             | 3 semaines pour une équipe interne du service informatique de la société<br>3 à 4 semaines pour une équipe externe en sous-traitance |
| Site transactionnel ne disposant pas d'une infrastructure XML | 4 semaines pour une équipe interne du service informatique de la société<br>4 à 5 semaines pour une équipe externe en sous-traitance |

## Présentation du langage XML

Avant de mettre en œuvre un site punchout, il est important de comprendre XML. Il s'agit d'un langage dont la fonction est de décrire d'autres langages. Les documents au format cXML sont créés à partir de DTD (définitions de type de document) pour XML. Ces DTD, qui jouent le rôle de modèles, permettent de définir la structure du contenu d'un document cXML (l'ordre et l'emboîtement corrects des éléments, par exemple) ainsi que le type de données des attributs.

### À propos du langage XML

Le langage XML (eXtensible Markup Language) est une norme de transfert de données entre applications Internet. Dans les documents XML, les données se présentent sous la forme de paires balise/valeur. La structure du langage XML s'apparente à celle du langage HTML (HyperText Markup Language), si ce n'est que l'extraction et l'utilisation des données s'effectuent plus facilement à partir des documents XML qu'avec les documents HTML.

Avec le développement croissant d'applications Internet, le langage XML est appelé à prendre de plus en plus d'importance.

Il est indispensable que vous ayez une certaine connaissance des opérations de création, d'analyse, d'interrogation, de réception et de transmission des données XML avec une source distante, que celle-ci soit émettrice ou destinataire.

Les analyseurs XML constituent l'outil de base pour le traitement des documents XML. Ces analyseurs sont proposés gratuitement par Microsoft et d'autres sociétés (l'analyseur XML est fourni en standard dans Microsoft Internet Explorer 5). Pour obtenir une liste d'outils XML, consultez la section « Utilitaires XML », page 9.

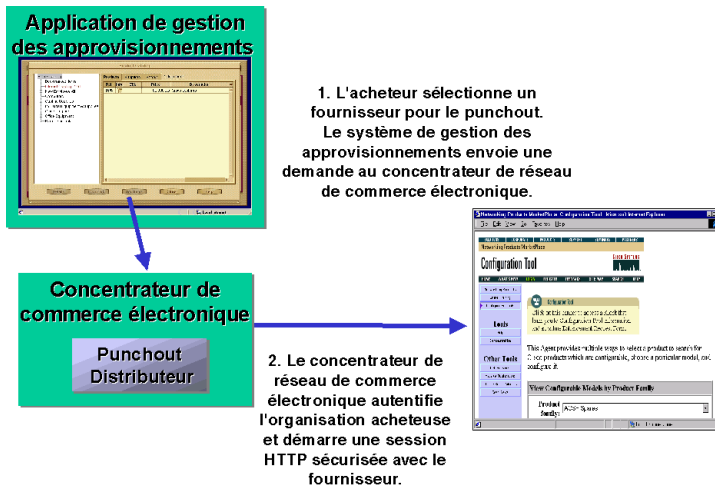
## Séquence des événements punchout

Une session punchout comprend plusieurs étapes.

### Étapes 1 et 2 : requête punchout

L'utilisateur se connecte à une application de gestion des approvisionnements et ouvre une nouvelle demande d'achat. Pour trouver les articles qui l'intéressent dans les catalogues à sa disposition, il effectue une recherche par marchandise, fournisseur ou description de produit. Lorsqu'il sélectionne un article punchout, l'application de gestion des approvisionnements ouvre une nouvelle fenêtre de navigation et connecte l'utilisateur à son compte sur le site Web du fournisseur.

La figure suivante illustre les étapes d'une requête punchout :



**Comment cela fonctionne-t-il ?** Lorsque l'utilisateur clique sur un article punchout, l'application de gestion des approvisionnements envoie un document PunchOutSetupRequest au format cXML à un concentrateur de commerce électronique en réseau. En tant que tierce partie approuvée, le concentrateur se charge d'accepter la demande, de vérifier l'identité de l'organisation acheteuse, puis de transmettre la requête à votre site Web punchout.

**Remarque :** Tous les documents cXML envoyés sur Internet peuvent être transmis via une connexion sécurisée HTTPS codée SSL (Secure Socket Layer) 3.0.



Cette requête vise à communiquer l'identité de l'acheteur à votre site Web, puis à lui indiquer l'opération à effectuer. Les opérations prises en charge comprennent notamment :

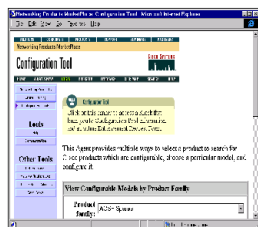
- **create** : établit une nouvelle session punchout.
- **edit** : ouvre une session punchout à des fins d'édition.
- **inspect** : ouvre une session punchout à des fins de vérification (aucune modification ne peut être apportée aux données).

Une fois la requête reçue par votre site Web, celui-ci renvoie une réponse PunchOutSetupResponse à l'application de gestion des approvisionnements. Cette réponse contient une URL qui indique à l'application l'adresse de connexion pour établir une session de navigation sur votre site Web.

L'application ouvre alors une nouvelle fenêtre de navigation, qui affiche une session connectée à un compte de votre site Web. Ce compte peut être spécifique à une région, à une société, à un service ou à un utilisateur.

### Étape 3 : sélection de produits

L'utilisateur sélectionne des articles dans le stock à l'aide de toutes les fonctions et de tous les services disponibles sur votre site Web.



3. L'acheteur utilise les sites des fournisseurs pour trouver et configurer les produits.

Selon le produit ou le client, le site peut proposer les fonctions suivantes :

- Outils de configuration permettant de constituer des produits personnalisés, tels que des ordinateurs ou des composés organiques.
- Moteurs de recherche pour trouver les produits désirés dans de grands catalogues.
- Affichage de données normalisées pour comparer des produits tels que des fournisseurs de fabrication et d'entretien en fonction de leur prix, de leurs caractéristiques ou de leur disponibilité.

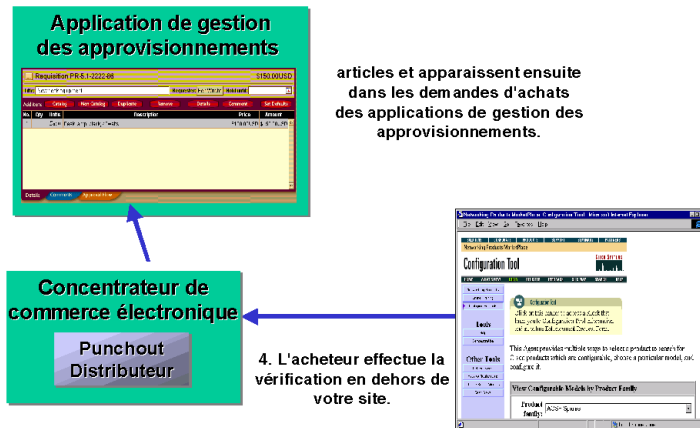
- Affichage d'attributs spécifiques à une marchandise particulière, tels que des documents imprimés, des produits chimiques, des réactifs ou des services.
- Calcul de prix, consultation des stocks et vérification de la disponibilité en temps réel.
- Calcul automatique des taxes et du fret en fonction de l'adresse de livraison, des dimensions des articles ou de la quantité sélectionnée. Ce calcul est inutile au cours de la session initiale de punchout.

**Comment cela fonctionne-t-il ?** Une fois que l'utilisateur a été dirigé sur votre site Web par l'application de gestion des approvisionnements, sa démarche d'achat est identique à celle de tout autre utilisateur accédant directement au site. Aussi n'est-il pas nécessaire de modifier les fonctions et services énumérés ci-dessus.

## Étape 4 : passage au contrôle

Le site Web calcule le montant total des articles sélectionnés par l'utilisateur, taxes, fret et réductions éventuelles compris. L'utilisateur clique alors sur le bouton « Contrôle » du site pour enregistrer le contenu de son caddie comme demande d'achat dans l'application de gestion des approvisionnements.

La figure ci-dessous illustre les différentes étapes de l'enregistrement de la commande :



**Comment cela fonctionne-t-il ?** Lorsqu'un utilisateur clique sur le bouton « Contrôle », votre site Web envoie un document cXML PunchOutOrderMessage à l'application de gestion des approvisionnements. Ce document contient des informations sur les produits sélectionnés et leur prix. Vous avez également la possibilité d'envoyer des cookies de fournisseur cachés, qui permettront ultérieurement d'associer les articles à la session d'achat concernée.

Cette étape s'apparente en essence à l'établissement d'un devis. En d'autres termes, vous n'avez pas encore reçu de commande d'achat et ne pouvez, de ce fait, enregistrer la commande.

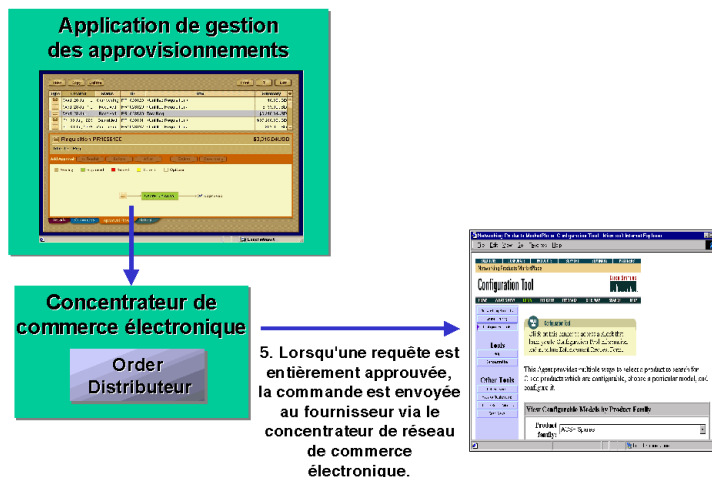
Par la suite, si l'utilisateur souhaite modifier sa demande d'achat, vous pouvez l'autoriser à effectuer un « punchout répété » pour accéder à votre site Web. L'application de gestion des approvisionnements se charge alors de renvoyer le contenu du caddie initial au site Web et l'utilisateur peut y apporter les modifications nécessaires. Une fois le passage au contrôle effectué, le site Web transfère de nouveau les articles à l'application.

Votre site Web constitue la source d'informations essentielle pour tous les articles punchout. La modification des quantités demandées ou l'ajout de nouveaux articles à une demande d'achat peut avoir des conséquences sur les taxes à payer ou sur les frais de livraison. Aussi est-il nécessaire que le site Web recalcule le montant total dû. Cela implique que toute modification du caddie initial doit être effectuée à partir du site et non de l'application de gestion des approvisionnements. C'est pourquoi il faut répéter le punchout. Cette étape correspond donc à une requête PunchOutSetupRequest comportant l'opération « edit ».

## Étape 5 : transmission de la commande d'achat

Lorsque le contenu du caddie est transféré du site Web à l'application de gestion des approvisionnements, le module de demande d'achat de l'application se charge de mettre en place la procédure d'approbation. Une fois la demande d'achat approuvée, l'application la convertit en commande et la renvoie au site Web pour traitement. Vous pouvez soit joindre des informations de paiement par carte d'achat avec la commande, soit facturer l'achat à la livraison.

La figure ci-dessous illustre la transmission d'une commande d'achat :



**Comment cela fonctionne-t-il ?** L'application de gestion des approvisionnements envoie toutes les commandes d'achat au concentrateur de commerce électronique, dans le format cXML. Le concentrateur les achemine alors vers votre site Web, selon le mode de routage des commandes que vous avez choisi. Lorsque vous accusez réception d'une commande, cela signifie en fait que vous l'avez enregistrée.

Pour les fournisseurs disposant de sites Web punchout, la méthode recommandée pour le routage des commandes est cXML. Voici pourquoi :

- Les commandes d'achat au format cXML vous permettent de récupérer les informations de cookie fournisseur. Étant donné que le cookie appartient à un type données banalisé (« any »), il n'est pas facile de le transmettre par d'autres méthodes de routage, telles que la télécopie, le courrier électronique ou l'EDI.
- Par définition, les fournisseurs disposant d'un site compatible punchout sont familiarisés avec le format cXML ; l'acceptation des commandes dans ce format ne requiert donc qu'un effort très limité de leur part.

Les commandes d'achat sont présentées en détail dans le Chapitre 3, « Réception de commandes d'achat cXML ».

## Documents punchout

Il existe quatre types de documents cXML :

- Catalogue-index punchout
- Document PunchOutSetupRequest
- Document PunchOutSetupResponse
- Document PunchOutOrderMessage

### Catalogue-index punchout

Les catalogues-index punchout sont des fichiers contenant une liste d'articles punchout et désignant le site Web punchout correspondant.

Le document ci-dessous constitue un exemple de catalogue-index punchout :

|   |  |
|---|--|
| <i>Type de document cXML et<br/>URL de la DTD</i>       | <pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd"&gt; &lt;Index&gt;   &lt;SupplierID domain="DUNS"&gt;83528721&lt;/SupplierID&gt;   &lt;IndexItem&gt;     &lt;IndexItemPunchout&gt;       &lt;ItemID&gt;         &lt;SupplierPartID&gt;5555&lt;/SupplierPartID&gt;       &lt;/ItemID&gt;       &lt;PunchoutDetail&gt;         &lt;Description xml:lang="en-US"&gt;Desk Chairs&lt;/Description&gt;         &lt;Description xml:lang="fr-FR"&gt;Chaises de bureau&lt;/Description&gt;         &lt;URL&gt;http://www.workchairs.com/punchout.asp&lt;/URL&gt;         &lt;Classification domain="UNSPSC"&gt;5136030000&lt;/Classification&gt;       &lt;/PunchoutDetail&gt;     &lt;/IndexItemPunchout&gt;   &lt;/IndexItem&gt; &lt;/Index&gt; </pre> |
| <i>Identificateur de l'article<br/>punchout</i>         | <pre>       &lt;SupplierPartID&gt;5555&lt;/SupplierPartID&gt;     &lt;/ItemID&gt;     &lt;PunchoutDetail&gt; </pre>  |
| <i>URL du site Web punchout<br/>(page de lancement)</i> | <pre>         &lt;Description xml:lang="fr-FR"&gt;Chaises de bureau&lt;/Description&gt;         &lt;URL&gt;http://www.workchairs.com/punchout.asp&lt;/URL&gt;         &lt;Classification domain="UNSPSC"&gt;5136030000&lt;/Classification&gt;       &lt;/PunchoutDetail&gt;     &lt;/IndexItemPunchout&gt;   &lt;/IndexItem&gt; &lt;/Index&gt; </pre>  |

SupplierID identifie l'organisation du fournisseur. Vous pouvez utiliser le domaine d'identification de votre choix ; nous vous conseillons toutefois les systèmes DUNS (Dun & Bradstreet Universal Naming System) et NetworkID. Pour plus d'informations sur les numéros DUNS, rendez-vous sur le site Web [www.dnb.com](http://www.dnb.com).

Description spécifie le texte affiché par l'application de gestion des approvisionnements dans les catalogues de produits. Cette description peut exister en plusieurs langues. Dans ce cas, l'application de gestion des approvisionnements sélectionne la description voulue d'après la langue de l'utilisateur.

Classification spécifique à l'acheteur le groupe de marchandises auquel un article appartient. Tous vos produits et services doivent être organisés selon les normes du système de classification de produits UNSPSC. Dans le cadre des catalogues-index punchout, l'élément Classification détermine l'emplacement d'un article punchout au sein des catalogues proposés à l'utilisateur. Pour obtenir la liste des codes UNSPSC, rendez-vous sur le site Web [www.unspsc.com](http://www.unspsc.com).

### ***Création et publication de catalogues-index***

Créez des catalogues-index, puis publiez-les sur un concentrateur de commerce électronique pour que vos clients puissent y accéder. Les gestionnaires de catalogues au sein des organisations acheteuses les téléchargent et les stockent dans leur application de gestion des approvisionnements.

Les utilisateurs accèdent ainsi à vos catalogues-index punchout et peuvent faire la différence avec les catalogues statiques traditionnels affichés simultanément.

### ***Portée des catalogues punchout***

Vous avez la possibilité de créer des catalogues de portée différente, au niveau magasin, rayon ou produit.

- Les catalogues au niveau magasin répertorient tous les produits et services proposés. L'utilisateur doit donc effectuer une recherche pour trouver l'article désiré.
- Les catalogues au niveau rayon répertorient des produits et des services apparentés.
- Les catalogues au niveau produit ne répertorient qu'un seul produit ou service. L'utilisateur n'a donc aucune recherche à effectuer.

Pour déterminer le champ d'activité de vos articles punchout, vous devez prendre en considération différents facteurs tels que le modèle commercial de votre société, la structure de vos offres de produits et de services ainsi que la structure de votre site Web punchout.

Plus un site Web dispose d'outils de recherche et de configuration, plus la portée de vos catalogues-index punchout peut être large.

## **Document PunchOutSetupRequest**

Pour lancer une session punchout, l'utilisateur doit sélectionner un article punchout. L'application de gestion des approvisionnements génère alors un document PunchOutSetupRequest et l'envoie à un concentrateur de commerce électronique, qui le fait suivre jusqu'au site Web punchout.

Voici un exemple de document PunchOutSetupRequest :

```

<?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>

```

*Demandeur (organisation acheteuse)*

*Destination (fournisseur)*

*Entité relais précédente (Ariba Network dans cet exemple)*

*Type de demande*

*Destination du document final PunchOutOrderMessage*

*Article sélectionné par l'utilisateur*

Les attributs payloadID et timestamp figurant au début de la demande permettent aux clients cXML d'effectuer le suivi des documents et de détecter d'éventuels doublons.

Les éléments From, To et Sender permettent aux systèmes de réception d'identifier les parties impliquées et de leur attribuer les droits d'accès appropriés. Les éléments From et To ne changent pas. Cependant, lors du transfert d'un document vers sa destination, certains nœuds intermédiaires (comme Ariba Network) modifient l'élément Sender.

### **Opérations de création, de modification et de vérification**

L'attribut operation spécifie le type de session lancé par l'utilisateur. Il existe trois types de sessions, identifiées par différents mots-clés : create (création), edit (modification) et inspect (vérification).

- create correspond à une session pendant laquelle un nouveau caddie est créé par l'utilisateur, ce qui entraîne une nouvelle demande d'achat.
- edit correspond à une session pendant laquelle un caddie d'achat existant est rouvert en vue d'être modifié. L'application de gestion des approvisionnements renvoie alors les données des articles figurant dans la demande PunchOutSetupRequest. Le site Web punchout peut utiliser ces données pour ré-instancier le caddie créé au cours de la session initiale.
- inspect correspond à une session pendant laquelle un caddie d'achat existant est rouvert à des fins d'affichage uniquement. Comme dans le cas d'une session edit, l'application de gestion des approvisionnements renvoie les données des articles figurant dans la demande PunchOutSetupRequest. Néanmoins, après qu'il l'a ré-instancié, le site Web punchout ne permet pas que le caddie soit modifié.

Le document suivant constitue un exemple de requête edit :

```
<?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="edit">
        <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>

```

Si l'utilisateur avait lancé une session edit en sélectionnant un article du catalogue, la demande PunchOutSetupRequest contiendrait un élément SelectedItem, comme dans le cas d'une session create.

### ***Authentification par un concentrateur de commerce électronique***

Tous les documents de type PunchOutSetupRequest sont acheminés via un concentrateur de commerce électronique qui se charge de les authentifier et de rechercher l'URL du site Web punchout. Les étapes suivies sont décrites ci-dessous :

1. Le concentrateur reçoit le document PunchOutSetupRequest de l'utilisateur.
2. Le concentrateur compare l'ID de l'acheteur (From et Shared Secret) aux données du compte de commerce électronique correspondant. Il identifie également le fournisseur demandé (To).
3. Le concentrateur recherche le code secret partagé dans le compte du fournisseur et l'insère dans le champ Shared Secret de l'élément Sender.

4. Le concentrateur recherche l'URL du site Web dans le compte du fournisseur, puis fait suivre le document PunchOutSetupRequest à cette adresse.
5. Le site Web du fournisseur reçoit le document cXML. Comme ce document contient le code secret partagé, le site sait qu'il a été authentifié.
6. Le site Web utilise les données de l'élément From pour identifier le demandeur au niveau de la société concernée (par exemple, acme.com).
7. Le fournisseur a la possibilité d'utiliser l'élément Contact ainsi que les données Extrinsic dans le corps de la demande pour identifier de façon univoque l'utilisateur demandeur (par exemple, John Smith du service Finance chez acme.com).

Les documents PunchOutSetupRequest et PunchOutSetupResponse passent par le concentrateur de commerce électronique à des fins d'authentification. Le document PunchOutOrderMessage, qui renvoie le contenu du caddie à l'application de gestion des approvisionnements, est directement transféré de votre site Web vers l'utilisateur en utilisant les protocoles standard HTTP ou HTTPS.

### ***URL de configuration du fournisseur et élément SelectedItem***

Dans les versions antérieures du langage cXML, l'élément SupplierSetup constituait la seule façon de spécifier l'URL du site Web punchout du fournisseur. Depuis la version 1.1 de cXML, le concentrateur de commerce électronique a déjà connaissance de cette URL.

En outre, depuis la version 1.1 de cXML, les applications de gestion des approvisionnements peuvent utiliser l'élément SelectedItem pour spécifier un traitement punchout de niveau magasin, rayon ou produit.

L'élément SupplierSetup a été annulé. Toutefois, le site Web punchout d'un fournisseur doit pouvoir gérer ces deux méthodes tant que tous les sites punchout ne reconnaissent pas l'élément SelectedItem.

### ***Données de contact pour les informations extrinsèques et l'identification des utilisateurs***

Un document `PunchoutSetupRequest` peut contenir des données détaillées sur les utilisateurs au niveau de l'élément `Contact`. Le site Web du fournisseur peut utiliser ces données pour authentifier et orienter les utilisateurs. Les données ci-dessous constituent des exemples de données d'identification :

- Nom d'utilisateur et rôle
- Adresse de messagerie

Les documents `PunchOutSetupRequest` peuvent également contenir des données *Extrinsic* permettant d'identifier les utilisateurs avec plus de précision. Ces données comprennent par exemple :

- Centre de coûts et sous-compte de l'utilisateur
- Région
- Superviseur
- Devise par défaut

Ce sont les organisations acheteuses qui se chargent de configurer leurs applications de gestion des approvisionnements sur l'insertion de données `Contact` et de données *Extrinsic*. Vous devez donc demander à vos clients le type de données qu'ils sont susceptibles de vous communiquer.

## **Document `PunchOutSetupResponse`**

Après réception d'un document `PunchOutSetupRequest`, le site Web du fournisseur renvoie une réponse `PunchOutSetupResponse`. Ce document `PunchOutSetupResponse` remplit deux fonctions :

- Il indique tout d'abord si la demande `PunchOutSetupRequest` a bien été enregistrée.
- Il fournit à l'application de gestion des approvisionnements une URL de renvoi vers la page de démarrage du site Web punchout.

En effet, ce document contient un élément `<URL>` qui spécifie l'adresse de la page de démarrage à remettre au navigateur Web de l'utilisateur lors d'une session de navigation interactive. Cette URL doit contenir suffisamment d'informations de statut pour permettre de retrouver le contexte d'une session sur un site Web. Ces informations comprennent notamment l'identité du demandeur et le contenu de l'élément `BuyerCookie`.

Le document suivant constitue un exemple de réponse PunchOutSetupResponse :

```
<?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>
```

## Document PunchOutOrderMessage

Après que l'utilisateur a sélectionné des articles sur le site Web du fournisseur, les a configurés et a cliqué sur le bouton « Contrôle », le site envoie un document PunchOutOrderMessage à l'application de gestion des approvisionnements concernée afin de lui communiquer le contenu du caddie. Ce document peut être beaucoup plus détaillé que la plupart des autres ; en effet, il doit être en mesure de représenter intégralement le contenu de n'importe quel caddie imaginable. Comme sa structure ne correspond pas exactement au paradigme requête/réponse utilisé jusqu'à présent, ce document fera l'objet d'une description détaillée.

Le document suivant constitue un exemple de message PunchOutOrderMessage :

```
<?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="edit">
      <Total>
        <Money currency="USD">763.20</Money>
      </Total>
    </PunchOutOrderMessageHeader>
    <ItemIn quantity="3">
      <ItemID>
        <SupplierPartID>5555</SupplierPartID>
        <SupplierPartAuxiliaryID>E000028901
        </SupplierPartAuxiliaryID>
      </ItemID>
      <ItemDetail>
        <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>

```

BuyerCookie permet à l'application de gestion des approvisionnements d'associer un document PunchOutOrderMessage à la requête PunchOutSetupRequest initiale. Aussi est-il nécessaire que votre site Web renvoie systématiquement cet élément chaque fois qu'il est présent. N'utilisez pas BuyerCookie dans le but d'effectuer le suivi de sessions punchout, car cet élément change à chaque nouvelle session (session de création, de vérification ou de modification).

SupplierPartAuxiliaryID joue le rôle de cookie de fournisseur. Ce champ permet au fournisseur de transmettre des données supplémentaires, telles que le numéro de devis ou un autre document cXML. L'application de gestion des approvisionnements vous renvoie ces données à chaque session PunchOutSetupRequest de type edit ou inspect ainsi que lors de la commande d'achat cXML résultante. Vous pouvez utiliser ce cookie pour associer les articles d'une demande d'achat aux articles du caddie correspondant sur votre site Web.

UnitOfMeasure décrit le mode de conditionnement ou le type de livraison du produit. Cet élément doit respecter les codes d'unité de mesure UN/CEFACT. Pour obtenir la liste des codes UN/CEFACT, rendez-vous sur le site [www.unece.org/cefact](http://www.unece.org/cefact).

Classification répertorie le code de marchandise UNSPSC (United Nations Standard Product and Service Code) de chaque article sélectionné. Les systèmes de suivi des organisations acheteuses et des fournisseurs utilisent ces codes dans le cadre de leur comptabilité et de la génération d'états. Pour obtenir la liste des codes UNSPSC, rendez-vous sur le site [www.unspsc.org](http://www.unspsc.org).

## Modification des pages Web

---

Afin de pouvoir recevoir ou envoyer les trois types de documents cXML punchout décrits, il vous faudra peut-être modifier ou créer les quatre pages mentionnées ci-dessous sur votre site Web :

- Page de lancement
- Page de démarrage
- Page de l'expéditeur
- Page du destinataire de la commande

Les exemples ci-après reposent sur l'utilisation de séquences de code ASP (Active Server Page) simples et de l'analyseur XML de Microsoft Internet Explorer 5. Toutefois, la mise en œuvre réelle varie selon l'environnement de développement de chaque fournisseur (CGI, JavaScript ou WebObjects, par exemple).

## Page de lancement

La page de lancement est celle à laquelle le concentrateur de commerce électronique envoie tous les documents PunchOutSetupRequest qu'il a authentifiés. Cette page lit le flux de données HTTP envoyé par le concentrateur et valide la requête cXML incorporée dans le flux en fonction de la DTD cXML. Pour ASP, cette opération est effectuée par des appels de méthode adressés à l'analyseur XML de Internet Explorer 5.

Une fois la validation effectuée, la page de lancement extrait les éléments du document afin d'effectuer les opérations suivantes :

1. Identifier l'utilisateur et déterminer la destination vers laquelle l'orienter.
2. Élaborer un document PunchOutSetupResponse et le renvoyer à l'expéditeur.

Une page de lancement doit enregistrer les données suivantes qui seront utilisées par la page de démarrage :

- Identité du demandeur (Sender).
- Langue de l'utilisateur (xml:lang) afin que le site du fournisseur affiche le contenu dans la langue requise.
- Type de requête (create, edit ou inspect).
- Données Extrinsic éventuelles, permettant une identification plus précise de l'utilisateur et de son emplacement géographique.

Le document ci-dessous constitue un exemple de page de lancement. Plutôt que d'utiliser un analyseur XML pour générer dynamiquement le document PunchOutSetupResponse, ce code s'appuie sur un modèle XML statique dans lequel les données des articles viennent s'insérer. **Ce code n'est présenté qu'à titre d'exemple.**

```
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 Créer les données requises dans le prologue.
%<
Randomize
randStr = Int( 100000001 * Rnd )
prologStr = genProlog( "1.0", randStr )
Response.ContentType = "text/xml"
Response.Charset = "UTF-8"
%>
<%
REM Reçoit la demande PunchOutSetup envoyée par le concentrateur de commerce
électronique.
REM Extrait l'URL ORMS et le cookie de l'acheteur, les joint à l'URL de la page de
démarrage,
REM puis renvoie une réponse au demandeur.
REM punchoutredirect.asp?bc=2133hfe&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 Cette commande lit la requête HTTP cXML entrante.
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 Mettre en forme le document cXML PunchOutSetupReponse
    <% 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%>

```

La page de lancement doit renvoyer l'URL d'une page de démarrage StartPage spécifique pour chaque session punchout. De plus, cette URL ne doit être valide que pendant un laps de temps limité. La désactivation de cette URL rend l'accès à la page de démarrage de votre site plus difficile pour tout utilisateur non autorisé.

Veillez à intégrer les fonctions nécessaires à toute session edit ou inspect ultérieure. En effet, un utilisateur ne peut pas modifier sa commande d'articles punchout (la quantité à commander, par exemple) depuis son application de gestion des approvisionnements. Il doit répéter le punchout à l'aide d'une session edit. De même, pour faciliter la tâche de l'utilisateur, les sessions inspect qui ont lieu après réception d'une commande d'achat doivent afficher le statut de cette commande.

## Page de démarrage

La page de démarrage connecte le demandeur à un compte sur le site Web du fournisseur. C'est à partir de cette page que l'utilisateur commence son activité d'achat. Cette page existe peut-être déjà sur votre site Web. Dans ce cas, modifiez-la pour qu'elle interroge le document PunchOutSetupRequest afin d'extraire le nom d'utilisateur et le mot de passe.

N'autorisez l'accès à la page de démarrage de votre site qu'aux utilisateurs authentifiés. Si vous attendez le passage au contrôle pour authentifier les utilisateurs, les données confidentielles sur les tarifs et les termes en vigueur ne seront pas protégées.

Si vous utilisez des cookies sur les navigateurs HTTP afin d'effectuer le suivi des préférences de chaque utilisateur, détruisez-les systématiquement après avoir envoyé le message PunchOutOrderMessage à l'acheteur. La destruction de ces cookies évite que des utilisateurs non autorisés accèdent à des fonctions réservées.

## Page de l'expéditeur

La page de l'expéditeur transmet le contenu du caddie sélectionné à l'utilisateur. Comme décrit plus haut, une fois qu'il a rempli son caddie, l'utilisateur clique sur le bouton « Contrôle ».

L'exemple ci-dessous illustre cette fonction à l'aide d'un code ASP simple. Plutôt que d'utiliser un analyseur XML pour générer dynamiquement le document PunchOutOrderMessage, ce code s'appuie sur un modèle XML statique dans lequel les données des articles viennent s'insérer. **Ce code n'est présenté qu'à titre d'exemple.**

Le texte ci-dessous représente une partie de la page des produits du site Web d'un fournisseur :

```
<!--#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>
```

La fonction AddBuyButton renvoie le document PunchOutOrderMessage à l'utilisateur.

La liste suivante correspond au texte du fichier à inclure (punchoutitem.inc) apparaissant ci-dessus :

```

<%
REM Ce code asp apparaît dans le fichier items.asp, qui spécifie les paramètres d'un
article,
REM structure un document au format cXML et permet à l'utilisateur de passer l'article au
contrôle.
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") %> " <%= 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
%>

```

La fonction AddBuyButton contient le code FORM POST, qui renvoie à l'utilisateur le document PunchOutOrderMessage codé sous forme d'URL.

### **Codage de formulaire en HTTP**

Pour envoyer un document `PunchOutOrderMessage`, vous devez utiliser un *codage de formulaire en HTML*, qui correspond à un modèle de transport différent du modèle de demande/réponse HTTP traditionnel. Ce transport permet une intégration plus aisée de votre site Web à l'application de gestion des approvisionnements. Il permet également aux organisations acheteuses de recevoir des données XML sans qu'elles disposent nécessairement d'un serveur Web protégé par pare-feu.

Plutôt que d'envoyer le document `PunchOutOrderMessage` directement à l'application de gestion des approvisionnements, le site Web en crée une représentation codée dans un champ de formulaire HTML masqué, puis l'envoie à l'URL spécifiée dans l'élément `BrowserFormPost` du document `PunchOutSetupRequest`. Ce champ de formulaire HTML masqué doit porter le nom « `xml-urlencoded` » ou « `xml-base64` » (ces noms ne font pas la distinction entre les majuscules et les minuscules).

Ce codage vous permet de créer une page Web de réservation contenant le document cXML. Lorsqu'un utilisateur clique sur le bouton « Contrôle », le site Web transmet les données (invisibles de l'utilisateur) à l'application de gestion des approvisionnements sous la forme d'un formulaire en HTML.

### **Annulation d'une session punchout**

Il peut être utile d'ajouter un bouton « Annuler » à vos pages afin que les utilisateurs puissent éventuellement annuler leur session punchout. Ce bouton renvoie un message `PunchOutOrderMessage` vide qui indique à l'application de gestion des approvisionnements qu'aucun article ne va lui être transmis et qu'elle doit supprimer les articles punchout de la demande existante. Ce bouton peut également servir à déclencher diverses tâches de maintenance sur le site Web, comme la suppression du caddie sélectionné et la fermeture de la session en cours.

### **Page du destinataire de la commande**

La page du destinataire de la commande accepte les commandes d'achat au format cXML envoyées par les organisations acheteuses. Elle peut être définie de façon similaire à la page de lancement décrite ci-dessus.

Pour plus d'information sur la réception des commandes d'achat, consultez le Chapitre 3, « Réception de commandes d'achat cXML ».

## Suggestions sur la mise en œuvre d'un site Web punchout

---

Si vous envisagez de mettre en œuvre un site Web punchout, examinez les suggestions ci-dessous.

### Instructions

Suivez les instructions ci-après lors du développement d'un site Web punchout :

- Examinez les spécifications cXML requises.
- Utilisez un analyseur XML et validez tous les documents en fonction de la DTD cXML.
- Utilisez la propriété `xml:lang=` pour identifier la langue de chaque utilisateur et lui fournir un contenu localisé.
- Utilisez la référence `From` pour identifier chaque organisation acheteuse.
- Lors de la réorientation, envoyez une URL temporaire spécifique à chaque session.
- Ne placez pas de cookies permanents sur les navigateurs.
- N'exigez pas de vos clients qu'ils transmettent des données Extrinsic excessives.
- Attribuez à chaque article une unité de mesure UNUOM (United Nations Units of Measure) et un code UNSPSC (United Nations Standard Product and Service Codes).
- Apportez une réelle valeur ajoutée à vos clients en affichant la disponibilité des produits, le statut des commandes et les promotions spéciales.
- L'étape de passage au contrôle doit être facile et intuitive. Dans la mesure du possible, faites en sorte que les utilisateurs n'aient à cliquer que sur trois boutons pour conclure un achat.
- Créez du code pour permettre des sessions ultérieures de modification et de vérification. En effet, les utilisateurs ne peuvent pas modifier les détails d'une commande punchout (tels que la quantité) depuis leur application de gestion des approvisionnements. Ils doivent répéter le punchout pour une session de modification.
- Dans l'intérêt des utilisateurs, faites en sorte que les sessions de vérification affichent le statut des commandes.
- Testez votre site Web punchout. Prévoyez une période de test avec les applications de gestion des approvisionnements de vos clients.
- Les transactions punchout génèrent des devis et non des commandes d'achat. Créez une page de réception des commandes d'achat au format cXML pour les enregistrer.

## Cookies d'acheteur et de fournisseur

Les cookies d'acheteur et de fournisseur permettent à ces deux parties de ré-instancier les données des articles utilisés et de les exploiter dans leurs systèmes de suivi.

- Renvoyez tout cookie d'acheteur (BuyerCookie) reçu sans le modifier.
- Exploitez les données des cookies de fournisseur (SupplierPartAuxiliaryID).

Un cookie d'acheteur remplit la fonction d'un numéro de demande d'achat ; il véhicule un statut qui permet au système de l'organisation acheteuse d'associer une demande d'achat au caddie correspondant.

Dans le même ordre d'idée, un cookie de fournisseur s'apparente à un numéro de devis ; il véhicule un statut qui permet au système du fournisseur d'associer un caddie à une demande d'achat et à une commande précises. Les applications de gestion des approvisionnements retransmettent les cookies de fournisseur au site Web concerné lors de sessions punchout edit et inspect ultérieures, ainsi que lors de la commande d'achat elle-même. Le site Web doit tirer pleinement profit de ces cookies afin d'éviter de renvoyer à l'acheteur des données visibles qui ne concernent que le fournisseur.

## Personnalisation

L'en-tête du document PunchOutSetupRequest identifie systématiquement l'organisation acheteuse, mais le corps de la demande peut contenir des informations supplémentaires de type Contact et Extrinsic (centre de coûts et emplacement de l'utilisateur, catégorie de produits, par exemple) permettant de déterminer l'URL dynamique à utiliser pour un acheteur précis.

Bien que toutes les organisations acheteuses ne transmettent pas forcément des données Extrinsic, ces données peuvent vous aider à personnaliser davantage votre magasin Web, sans vous limiter à un contexte par organisation. Il peut être utile, par exemple, de proposer un magasin Web différent pour chaque centre de coûts, chaque catégorie de produits ou chaque utilisateur au sein d'une organisation acheteuse.

De même, le stockage et l'affichage des devis antérieurs de chaque utilisateur pourraient s'avérer utiles. Les utilisateurs apprécieraient sans doute d'avoir accès à leurs devis, de pouvoir vérifier l'état de leurs commandes et de pouvoir créer des états sur leur activité antérieure. Afin d'éviter tout problème de sécurité, ne stockez l'historique des devis qu'au niveau de chaque utilisateur.

Avant toute mise en œuvre d'un site Web punchout hautement personnalisé et dynamique, il est important de réfléchir à la quantité de travail que cela implique. Il vous faut trouver un équilibre entre personnalisation et complexité. Un site Web complexe est plus long à mettre en place et à maintenir, mais il peut aussi présenter un plus grand intérêt pour les utilisateurs. Cela dit, il est recommandé de commencer par créer un site Web punchout relativement simple, puis d'en améliorer les fonctionnalités avec le temps.



---

# Chapitre 3

## Réception de commandes d'achat cXML

Ce chapitre explique comment configurer un site Web en vue de la réception de commandes d'achat au format cXML. Il détaille également la procédure d'envoi de messages indicateurs du statut des commandes d'achat à des organisations acheteuses ou à des espaces marchands.

### Traitement des commandes d'achat

---

Les applications de gestion des approvisionnements convertissent les demandes d'achat approuvées en une ou plusieurs commandes d'achat. Une commande d'achat est le document formel par lequel une organisation acheteuse demande à un fournisseur de remplir un contrat.

Le format cXML n'est qu'un des formats possibles pour la transmission des commandes d'achat ; le courrier électronique, la télécopie et les transactions EDI (X.12 Electronic Data Interchange) sont d'autres options envisageables. Le format cXML est toutefois celui qui convient le mieux aux commandes d'achat car il simplifie l'automatisation du traitement des commandes. Grâce à sa structure bien définie, il permet aux systèmes de traitement des commandes d'interpréter facilement les éléments constitutifs d'une commande d'achat. Les données correspondantes peuvent ainsi être acheminées jusqu'à vos services d'expédition, de facturation ou de vente avec une intervention manuelle réduite, voire sans intervention de votre part.

La méthode cXML de routage des commandes permet en outre la transmission de cookies fournisseur (SupplierPartAuxiliaryID) et de pièces jointes aux commandes d'achat.

Lors de la configuration de votre compte sur un concentrateur de réseau de commerce électronique, vous indiquez l'URL vers laquelle toutes les commandes d'achat cXML seront acheminées. Dès réception d'une commande d'achat, vous l'envoyez à votre système interne de gestion des commandes et l'exécutez selon la procédure habituelle. Votre site Web doit également retourner un accusé de réception de réponse à la commande au concentrateur de réseau de commerce électronique, pour indiquer à l'acheteur que la commande d'achat a bien été reçue et analysée.

La réception de commandes d'achat cXML n'exige pas de site Web compatible punchout ; punchout et réception de commandes cXML sont des fonctions bien distinctes. L'infrastructure et les applications nécessaires à la prise en charge des fonctionnalités punchout et à la réception des commandes d'achat cXML sont toutefois identiques.

## Réception de commandes d'achat

---

Deux types de documents cXML peuvent être utilisés pour la transmission des commandes d'achat. L'application de gestion des approvisionnements envoie un document OrderRequest, auquel vous répondez par un document OrderResponse. Ces documents transitent par le concentrateur de réseau de commerce électronique.

### OrderRequest

Le document OrderRequest s'apparente à une commande d'achat. Voici un exemple de document OrderRequest envoyé pour un article :

```
<?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="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, with 12.1" TFT XGA Display
        </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

Le document OrderResponse accuse réception de la commande d'achat et indique que son analyse est correcte. Il ne s'agit pas d'une garantie d'exécution de la commande, mais d'une simple confirmation de sa réception qui atteste de la validité du document cXML.

```
<?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>
```

## Acceptation des pièces jointes aux commandes

Les acheteurs précisent parfois certains points des commandes d'achat en leur associant des mémos, des dessins ou des télécopies. Ils peuvent joindre différents types de fichiers aux commandes d'achat cXML en faisant appel au standard MIME (Multipurpose Internet Mail Extensions).

Le document cXML contient seulement des références à des éléments MIME externes envoyés à l'intérieur d'une enveloppe MIME à plusieurs éléments (avec le document cXML, dans un message électronique ou par télécopie).

Le concentrateur de réseau de commerce électronique reçoit les pièces jointes et peut alors les envoyer au fournisseur ou les stocker en vue d'une récupération en ligne.

Pour plus d'informations sur les pièces jointes aux commandes d'achat, consultez la section « Transmission des pièces jointes », page 53.

Pour plus d'informations sur la norme MIME, rendez-vous sur les sites Web suivants :

[www.hunnysoft.com/mime](http://www.hunnysoft.com/mime)  
[www.rad.com/networks/1995/mime/mime.htm](http://www.rad.com/networks/1995/mime/mime.htm)



---

# Annexe A

## Spécification du langage cXML

Cette annexe décrit le protocole et les formats de données du langage cXML (commerce eXtensible Markup Language). Elle fournit toutes les informations nécessaires à la mise en œuvre des transactions prises en charge, aussi bien du point de vue du système client que de celui du système serveur. Les interactions liées au protocole, tout comme les documents commerciaux contenus dans les transactions, sont passés en revue de façon plus approfondie.

Des exemples concrets viennent en outre illustrer et clarifier l'utilisation du format cXML.

Cette annexe comprend les sections suivantes :

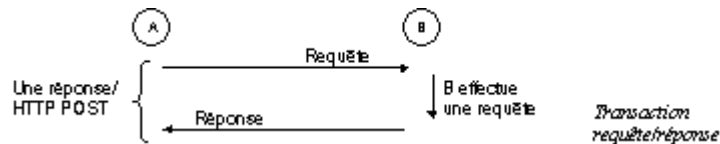
- Spécification du protocole
- Éléments de base
- Transaction Profile
- Définition des commandes
- Transaction Punchout
- Modifications de statut ultérieures
- Définitions de catalogue
- Définitions de gestion des abonnements
- Définitions de récupération des messages

## Spécification du protocole

Il existe deux modèles de communication possibles pour les transactions cXML : le modèle requête/réponse et le modèle unidirectionnel. Leur mise en œuvre est très simple, compte tenu de la description très stricte des opérations requises. Ces deux modèles s'avèrent nécessaires, car dans certains cas, seul l'un d'eux peut être appliqué.

### Modèle requête/réponse

Les transactions requête/réponse ne peuvent être exécutées que sur une connexion HTTP. La figure qui suit représente les étapes d'une interaction requête/réponse entre les partenaires A et B :



Cette transaction se déroule en différentes étapes :

1. A établit une connexion HTTP/1.x avec B sur une URL prédéterminée qui représente l'adresse de B.
2. A fait appel à une opération POST pour envoyer le document cXML sur la connexion HTTP.
3. A attend une réponse avant de revenir sur la connexion HTTP.
4. B est équipé d'un serveur compatible HTTP/1.x qui achemine la requête HTTP jusqu'à la ressource spécifiée par l'URL de l'étape 1. Cette ressource peut correspondre à n'importe quel emplacement valide connu du serveur HTTP de B : un programme CGI ou une page ASP, par exemple.
5. La ressource du partenaire B identifiée à l'étape 4 lit le contenu du document cXML et affecte la requête au gestionnaire correspondant.
6. Toujours du côté du partenaire B, le gestionnaire associé à la requête cXML effectue les tâches spécifiées par la requête et formate un document cXML en guise de réponse.
7. B envoie la réponse cXML à A via la connexion HTTP établie à l'étape 1.



8. A lit la réponse cXML et la renvoie au processus émetteur de la requête.
9. A referme la connexion HTTP établie à l'étape 1.

Cette procédure est répétée pour tous les autres cycles requête/réponse.

Pour simplifier les opérations décrites dans les étapes ci-dessus, les documents cXML se divisent en deux parties distinctes :

- En-tête – Contient des informations d'authentification et d'adressage.
- Données constitutives de la requête ou de la réponse – Contient une requête ou une réponse spécifique ainsi que les informations à transmettre.

Ces deux éléments sont transportés dans un élément enveloppe parent. L'exemple qui suit représente la structure d'un document de type requête cXML :

```
<cXML>
  <Header>
    Informations constitutives de l'en-tête...
  </Header>
  <Request>
    Informations constitutives de la requête...
  </Request>
</cXML>
```

L'exemple qui suit représente la structure d'un document de type réponse cXML :

```
<cXML>
  <Response>
    Informations constitutives de la réponse...
  </Response>
</cXML>
```

La structure de réponse n'utilise pas d'élément Header. Celui-ci est inutile parce que la réponse passe toujours par la même connexion HTTP que la requête.

## Conventions XML

Le langage cXML fait appel à des éléments pour décrire des unités discrètes, qui correspondent généralement aux propriétés des documents commerciaux traditionnels. Des éléments s'emploient aussi pour décrire les informations qui présentent des sous-divisions évidentes et les relations qui existent entre ces sous-divisions, telles les adresses.

Le langage cXML a largement recours aux attributs.

Dans la spécification cXML, tous les éléments et noms d'attributs se composent de mots entiers, ces mots étant séparés par des lettres majuscules (et jamais par des traits d'union). Les noms d'éléments commencent par une lettre majuscule tandis que les noms d'attributs commencent par une lettre minuscule, comme dans les exemples suivants :

Éléments : Sender, Credential, Payment, ItemDetail  
 Attributs : version, payloadID, lineNumber, domain

## Enveloppe cXML

L'élément enveloppe est la racine de la structure du document cXML ; il contient tous les autres éléments. L'élément cXML est présent dans chaque transaction cXML.

L'exemple qui suit représente un élément cXML entièrement défini :

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

L'élément cXML possède les attributs suivants :

|                                 |  |
|---------------------------------|--|
| <b>version</b><br>(facultatif)  | Précise la version du protocole cXML. Un programme d'analyse XML de validation est également en mesure de déterminer l'attribut version à partir de la DTD référencée. Tous les documents cXML doivent cependant inclure explicitement cet attribut, car certaines applications font appel à des analyseurs qui ne procèdent pas à une validation. |
| <b>xml:lang</b><br>(facultatif) | Langue utilisée pour l'ensemble du texte « brut » envoyé dans le document. Le destinataire doit répondre ou afficher les informations dans une langue identique ou similaire. Un client qui spécifie xml:lang="en-UK" dans une requête, par exemple, peut recevoir des données "en" en retour.   |
| <b>payloadID</b>                | Numéro unique indépendant de l'espace et de l'heure, utilisé à des fins de consignation pour identifier les documents perdus ou problématiques. Cette valeur ne doit pas être modifiée pour les nouvelles tentatives.<br><br>L'implémentation recommandée est :<br>dateheure.id processus.nombre aléatoire@nomhôte                                 |
| <b>timestamp</b>                | Date et heure d'envoi du message, au format ISO 8601. Cette valeur ne doit pas être modifiée pour les nouvelles tentatives.<br><br>Le format est AAAA-MM-JJThh:mm:ss-hh:mm (1997-07-16T19:20:30+01:00, par exemple).   |

### **Langue spécifiée par l'attribut `xml:lang`**

L'attribut `xml:lang` apparaît aussi avec la plupart des éléments de type texte « brut » (tels `Description` et `Comments`). Dans la spécification XML, la langue d'un élément peut, par défaut, prendre la valeur spécifiée pour tout élément parent, mais de telles valeurs par défaut ne permettent pas une exploitation correcte de l'arborescence des documents. Le langage cXML permet de faire en sorte que les identificateurs de langue accompagnent toujours les chaînes concernées.

Les attributs `xml:lang` qui apparaissent dans le protocole cXML n'ont aucune incidence sur les données mises en forme, telles les nombres, les dates et les heures. Comme indiqué ci-après pour l'attribut `timestamp`, de telles valeurs discrètes sont formatées en fonction de leur type de données. Les chaînes les plus longues (et les pages Web référencées) non destinées à un traitement informatique peuvent contenir un format de date ou un format numérique propre à la langue correspondant à un attribut `xml:lang` proche.

### **Heure et autres types de données**

L'attribut `timestamp` (et toutes les autres dates et heures des documents cXML) doit être mis en forme à l'aide du sous-jeu de caractères ISO 8601 restreint décrit dans la note « `Date and Time Formats` » du Word Wide Web Consortium (W3C), disponible sur le site [www.w3.org/TR/NOTE-datetime-970915.html](http://www.w3.org/TR/NOTE-datetime-970915.html).

Les horodatages exigent au minimum la date complète, accompagnée des heures, minutes et secondes. Les fractions de secondes sont facultatives. Les heures doivent impérativement être exprimées en heure locale, avec un décalage horaire par rapport à l'heure UTC (Coordinated Universal Time, ou Greenwich Mean Time). L'identificateur de fuseau horaire « Z » n'est pas autorisé.

La chaîne `2000-04-14T013:36:00-08:00`, par exemple, correspond au 14 avril 2000, 13:36, heure standard du Pacifique (U.S. Pacific Standard Time).

Parmi les autres références existantes sur les formats de date et heure et sur autres types de données utilisés par le langage cXML :

- Site de référence de Microsoft sur les types de données XML, [msdn.microsoft.com/xml/reference/schema/datatypes.asp](http://msdn.microsoft.com/xml/reference/schema/datatypes.asp)
- Proposition initiale sur les données XML soumise au Word Wide Web Consortium (W3C), à l'adresse [www.w3c.org/TR/1998/NOTE-XML-data-0105](http://www.w3c.org/TR/1998/NOTE-XML-data-0105)

## Couches imbriquées

L'élément cXML est la partie principale d'un document XML. Un document peut commencer ainsi :

```
<?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="0c300508b7863dcclb_13550"
  timestamp="2000-01-09T01:36:05-08:00">
  ...
```

Ce document est généralement transmis via HTTP avec un support MIME de type `text/xml` et un paramètre `charset` correspondant à l'encodage du document. Le protocole HTTP étant un protocole 8 bits, il est possible de recourir à n'importe quelle méthode d'encodage prise en charge par l'analyseur du destinataire sans faire appel à un codage de transfert pour le contenu tel « `base64` » ou « `quoted-printable` ». Tous les analyseurs XML gèrent l'encodage UTF-8, qui comprend tous les caractères Unicode. Les applications doivent par conséquent l'utiliser lors de la transmission de documents cXML.

**Remarque :** Conformément à la spécification RFC 2376 « XML Media Types », le paramètre MIME `charset` annule et remplace tout encodage spécifié dans la déclaration XML. Par ailleurs, le mode d'encodage par défaut pour le support de type `text/xml` est `us-ascii`, et pas UTF-8, comme indiqué dans la section 4.3.3 de la spécification XML. Pour plus de clarté, les documents cXML doivent inclure un encodage explicite dans la déclaration XML. Les enveloppes MIME doivent utiliser un paramètre `charset` correspondant pour les supports de type `text/xml` ou `application/xml`.

La transmission HTTP d'un document cXML peut inclure les en-têtes MIME et HTTP suivants :

```
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"?>
...
```

## Transmission des pièces jointes

Lors de l'envoi d'un document OrderRequest faisant référence à des fichiers externes, les fichiers référencés peuvent soit résider sur un serveur accessible au fournisseur, soit être transmis parallèlement au document cXML. Cette seconde option exige le recours à une enveloppe MIME à plusieurs éléments. L'une des exigences cXML requises pour cette enveloppe (outre les conditions de base décrites dans la spécification RFC 2046 « Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types ») est l'inclusion d'en-têtes Content-ID avec chaque fichier joint.

L'exemple qui suit représente la structure nécessaire pour un document cXML auquel a été jointe une image JPEG (sans les en-têtes HTTP indiqués ci-dessus) :

```
POST /cXML HTTP/1.0
Content-type: multipart/mixed; boundary=élément unique

--élément unique
Content-type: text/xml; charset="UTF-8"

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

Cette structure de base correspond aux informations minimales que l'analyseur MIME récepteur doit être en mesure de traiter. Les applications qui utilisent le type de support décrit dans la spécification RFC 2387 « The MIME Multipart/Related Content-type » obtiennent des informations beaucoup plus détaillées si la structure de base est améliorée :

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

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

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

Les analyseurs MIME récepteurs qui ne gèrent pas les supports de type `multipart/related` doivent traiter les deux exemples ci-dessus de façon identique. Chaque élément de la transmission MIME peut en outre utiliser un encodage de transfert pour le contenu. Cet encodage n'est pas nécessaire dans le cas d'une transmission HTTP. Les en-têtes `Content-description` et `Content-disposition` sont facultatifs dans le protocole cXML, bien qu'ils fournissent des renseignements utiles.

Pour plus d'informations sur le rattachement de fichiers externes à des commandes d'achat, consultez la section « Attachment », page 73.

## Header

L'élément `Header` contient des informations d'adressage et d'authentification. Cet élément est identique indépendamment de l'élément `Request` ou `Response` contenu dans la partie principale du message cXML. Les applications doivent connaître l'identité du demandeur ; la validation des informations indiquées pour l'identité n'est cependant pas indispensable.

Voici un exemple d'élément `Header` :

```
<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>
```

Les éléments `From` et `To` sont synonymes des éléments `From` et `To` qui figurent dans les messages SMTP ; ils représentent la source et la destination logiques des messages. L'élément `Sender` identifie le partenaire qui ouvre la connexion HTTP et envoie le document cXML.

L'élément `Sender` contient l'élément `Credential`, qui permet au destinataire d'authentifier l'expéditeur. Cette structure assure une authentification fiable sans exiger d'infrastructure de certificats numériques avec clés publiques entre les deux extrémités de la connexion. Seuls un nom d'utilisateur et un mot de passe doivent être émis par le destinataire pour autoriser l'expéditeur à exécuter des requêtes (`Requests`).

Au départ, les éléments `Sender` et `From` prennent la même valeur. Si le document cXML transite par différents concentrateurs de réseau de commerce électronique, toutefois, l'élément `Sender` est modifié en fonction des expéditeurs successifs.

### ***From***

Cet élément identifie l'émetteur de la requête cXML. Il peut éventuellement contenir plusieurs éléments `Credential`, qui permettent aux demandeurs de s'identifier au moyen de différentes méthodes d'identification. Ce recours à des éléments `Credential` multiples équivaut à l'envoi simultané d'adresses SMTP et X.400 dans un message électronique.

### ***To***

Cet élément identifie la destination de la requête cXML. Tout comme l'élément `From`, il peut contenir plusieurs éléments `Credential` qui facilitent l'identification de la cible.

### ***Sender***

Cet élément permet au destinataire d'identifier et d'authentifier le partenaire qui a ouvert la connexion HTTP. Il contient des éléments `Credential` qui procèdent à une authentification plus stricte que celle des éléments `From` ou `To`, car le récepteur doit authentifier le demandeur avant d'exécuter les opérations.

### ***Credential***

Cet élément contient les valeurs d'identification et d'authentification utilisées dans les messages cXML.

L'élément Credential possède les attributs suivants :

|                             |  |
|-----------------------------|--|
| <b>domain</b>               | Spécifie le type de référence d'identification. Cet attribut permet aux documents de contenir plusieurs types de références d'identification pour plusieurs domaines d'authentification.<br><br>Dans le cas des messages envoyés sur Ariba Network, par exemple, le domaine est généralement AribaNetworkUserld ou DUNS. |
| <b>type</b><br>(facultatif) | Les requêtes à destination ou en provenance d'un espace marchand identifient à la fois l'espace marchand et la société membre dans les éléments From ou To Credential. Dans ce cas, la référence d'identification (Credential) de l'espace marchand utilise l'attribut type, qui prend la valeur « marketplace ».        |

L'élément Credential contient un élément Identity et éventuellement un élément SharedSecret ou DigitalSignature. L'élément Identity indique l'identité du partenaire représenté par l'élément Credential, tandis que les éléments d'authentification facultatifs vérifient l'identité de ce partenaire.

L'élément SharedSecret s'emploie lorsque l'élément Sender prend pour valeur une combinaison nom d'utilisateur/mot de passe reconnue par le demandeur.

L'élément DigitalSignature peut s'employer si les deux partenaires se mettent d'accord sur un organisme et un format de certification communs. L'attribut type d'un élément DigitalSignature spécifie le type de certificat utilisé.

**Remarque :** N'insérez pas d'éléments d'authentification dans les documents faisant l'objet d'une communication unidirectionnelle. Ce protocole de transport achemine en effet les informations jusqu'aux navigateurs des utilisateurs, ce qui permettrait à ces derniers d'afficher la source du document (éléments Credential inclus).



## Request

Les clients envoient des requêtes en vue de l'exécution d'opérations. Un seul élément Request est autorisé pour chaque élément enveloppe cXML, ce qui simplifie les implémentations serveur, puisqu'aucun démultiplexage ne s'avère nécessaire lors de la lecture des documents cXML. L'élément Request peut inclure pratiquement tout type de données XML.

L'élément Request possède l'attribut suivant :

|                                       |  |
|---------------------------------------|--|
| <b>deploymentMode</b><br>(facultatif) | Indique si la requête est de type test ou production. Les valeurs possibles sont « production » (valeur par défaut) et « test ». |
|---------------------------------------|--|

## Response

Les serveurs envoient des réponses aux clients pour leur indiquer les résultats des opérations. Il se peut que le résultat de certaines requêtes ne renferme aucune donnée, et que l'élément Response ne contienne par conséquent aucune information exception faite d'un élément Status. Un élément Response peut également contenir des données en rapport avec les applications (autrement dit, un élément PunchOutSetupResponse dans les scénarios punchout).

### Status

Cet élément indique le succès ou l'échec d'une requête.

L'élément Status possède les attributs suivants :

|                                 |   |
|---------------------------------|---|
| <b>code</b>                     | Code du statut de la requête. Il obéit au modèle de codes de statut HTTP. Le code 200 représente une requête réussie, par exemple.  |
| <b>text</b>                     | Texte du message de statut. Ce texte facilite la lisibilité des journaux et se compose de chaînes au format canonique, en anglais.  |
| <b>xml:lang</b><br>(facultatif) | Langue des données contenues dans l'élément Status. Cet attribut est facultatif pour compatibilité avec la version cXML 1.1. Il risque de devenir obligatoire dans les versions futures de la spécification cXML. |

Les attributs de l'élément Status précisent le traitement subi par la requête.

L'élément Status peut inclure une grande diversité de données nécessaires au demandeur. Dans le cas d'un code de statut 200/OK, la présence d'informations n'est pas obligatoire. Avec un code de statut 500/Internal Server Error, toutefois, il est fortement recommandé d'explicitier l'erreur d'application ou l'erreur d'analyse XML survenue. Cette erreur facilite le débogage unilatéral et les tests d'interopérabilité.

Les serveurs ne doivent pas inclure d'éléments de réponse supplémentaires (élément PunchOutSetupResponse, par exemple), sauf si le code de statut se situe dans la plage 200 (200/OK, par exemple).

La spécification HTTP 1.1 comprend plusieurs codes de statut qui ne conviennent pas au langage cXML. Le langage cXML se situant généralement dans des couches de niveau supérieur à celles du protocole HTTP, bon nombre d'erreurs (telles 404/Not Found) sont traitées par la couche de transport. Les codes de statut 200/OK et 500/Internal Server Error le sont dans la majorité des cas. Les erreurs de validation lors de l'analyse d'un document Request se traduisent généralement par une erreur de transport, telle une erreur HTTP 400/Bad Request.

Le tableau qui suit répertorie d'autres codes HTTP possibles.

Le langage cXML comprend quelques rares codes de statut non HTTP :

- 550 – Impossible d'atteindre le serveur cXML suivant pour effectuer une transaction exigeant des connexions en amont. Un concentrateur intermédiaire peut retourner ce code lorsque le site d'un fournisseur n'est pas accessible (en cas d'interruption des connexions en amont, les concentrateurs intermédiaires doivent retourner directement les erreurs au client).
- 551 – Impossible d'acheminer une requête à cause d'une mauvaise configuration du fournisseur. Un concentrateur intermédiaire n'est pas parvenu à s'authentifier auprès d'un fournisseur, par exemple. Les clients ne sont pas en mesure de rectifier cette erreur, mais elle doit toutefois être résolue avant toute nouvelle tentative de leur part.
- 560 – Erreur temporaire du serveur. Un serveur est peut-être arrêté pour des raisons de maintenance, par exemple. Le client doit effectuer une nouvelle tentative ultérieurement.

| Statut | Texte canonique     | Signification   |
|--------|---------------------|---|
| 200    | OK                  | Le serveur a été en mesure d'exécuter cette requête (Request), malgré les erreurs ou avertissements d'application contenus dans la réponse (Response) retournée.  |
| 201    | Accepted            | Certaines opérations ne sont peut-être pas terminées.<br>Comme indiqué au paragraphe « StatusUpdateRequest », page 87, le client doit s'attendre à des transactions StatusUpdate ultérieures si ce statut est retourné en réponse à une requête de type OrderRequest.   |
| 204    | No Content          | Toutes les informations de la requête (Request) étaient valides et ont été reconnues. Le serveur ne possède aucune donnée Response correspondant au type demandé.<br><br>Dans un document PunchOutOrderMessage, ce statut indique que la session punchout a pris fin sans modification du caddie (ou de la demande du client).                                      |
| 400    | Bad Request         | Requête inacceptable pour le serveur, malgré une analyse correcte.  |
| 401    | Unauthorized        | Les références d'identification indiquées dans l'élément Request (élément Sender) n'ont pas été reconnues par le serveur.   |
| 402    | Payment Required    | L'élément Request doit inclure un élément Payment complet.  |
| 403    | Forbidden           | L'utilisateur ne dispose pas de privilèges suffisants pour exécuter cet élément Request.  |
| 406    | Not Acceptable      | Alias pour le code 400 : l'élément Request n'est pas acceptable pour le serveur, malgré une analyse correcte.   |
| 409    | Conflict            | L'état actuel du serveur ou ses données internes ont empêché l'exécution de l'opération requise (mise à jour). Un élément Request identique peut aboutir ultérieurement, notamment après l'exécution d'une autre opération.   |
| 412    | Precondition Failed | Une condition préalable à l'élément Request (une session punchout nécessaire à une opération PunchOutSetupRequest, par exemple) n'a pas été remplie. Ce statut indique généralement que le client a ignoré une portion d'une transmission précédente en provenance d'un serveur (l'attribut operationAllowed d'un élément PunchOutOrderMessageHeader, par exemple). |

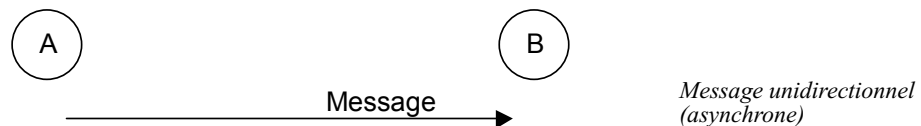
| Statut | Texte canonique       | Signification  |
|--------|-----------------------|--|
| 417    | Expectation Failed    | L'élément Request impliquait une condition de ressource qui n'a pas été remplie : une requête SupplierDataRequest demandant des informations relatives à un fournisseur inconnu du serveur, par exemple. Ce statut peut signaler la perte de certaines informations au niveau du client ou du serveur. |
| 500    | Internal Server Error | Le serveur n'a pas été en mesure d'exécuter cette requête (Request).   |
| 501    | Not Implemented       | Le serveur ne gère pas l'élément Request spécifié. Il se peut que l'élément PunchOutSetupRequest ou que l'opération demandée ne soient pas pris en charge, par exemple. Ce statut indique généralement que le client a ignoré le profil du serveur.  |

Lors de la réception de codes non reconnus, les clients cXML doivent les traiter en fonction de leur catégorie. Ainsi, les clients les plus anciens doivent considérer tous les nouveaux codes *2xx* comme des codes 200, les codes *4xx* comme des codes 400 et les codes *5xx* comme des codes 500. Ce mode de traitement permet à la fois les évolutions ultérieures du protocole cXML et des codes propres au serveur sans perte d'interopérabilité.

## Modèle unidirectionnel (asynchrone)

Contrairement aux transactions de type requête/réponse, les messages asynchrones ne sont pas limités au protocole de transport HTTP.

Les messages unilatéraux conviennent dans les cas où un canal HTTP (opération de type requête/réponse synchrone) ne peut être utilisé. La figure ci-après donne un exemple de communication entre A et B au moyen de messages au lieu d'une transaction requête/réponse.



L'un des scénarios possibles dans ce cas serait le suivant :

1. A formate et code un document cXML avec un protocole de transport reconnu par B.
2. A envoie le document à l'aide de ce protocole de transport. A n'attend pas (et ne peut pas attendre) de façon active de réponse de la part de B.

3. B reçoit le document cXML et le décode.

4. B traite le document.

Dans le modèle asynchrone, il n'existe *pas* de cycle requête/réponse explicite entre A et B. Des messages en provenance d'autres partenaires et d'autres conversations peuvent venir s'intercaler entre les messages asynchrones, par exemple.

Pour qu'une transaction asynchrone soit entièrement définie, le protocole de transport utilisé pour le message doit également être précisé. Pour les transactions cXML qui font appel au modèle asynchrone, le protocole de transport et le codage sont spécifiés. La transaction PunchOutOrderMessage est un exemple type de message asynchrone.

Les messages asynchrones possèdent une structure similaire à celle du modèle requête/réponse :

```
<cXML>
  <Header>
    Informations constitutives de l'en-tête...
  </Header>
  <Message>
    Informations constitutives du message...
  </Message>
</cXML>
```

L'élément Header est traité exactement de la même façon que dans le cas requête/réponse. L'élément cXML est également identique à celui décrit plus haut. La différence la plus notable entre un message asynchrone et un message requête/réponse réside dans la présence d'un élément Message (au lieu d'un élément Request ou Response). Les sections qui suivent détaillent plus précisément l'élément Message.

## Message

Cet élément transporte toutes les informations qui figurent dans la partie principale d'un message cXML. Il peut comporter un élément Status facultatif, identique à celui qui se trouve dans un élément Response – il peut s'employer dans les messages qui constituent des réponses logiques à des requêtes, par exemple.

L'élément Message possède les attributs suivants :

|                                       |  |
|---------------------------------------|--|
| <b>deploymentMode</b><br>(facultatif) | Indique si la requête est de type test ou production. Les valeurs possibles sont « production » (valeur par défaut) et « test ».   |
| <b>inReplyTo</b><br>(facultatif)      | Spécifie le Message auquel ce Message répond. Le contenu de l'attribut inReplyTo correspond à l'attribut payloadID d'un Message reçu antérieurement. Cette structure permet de construire une conversation bilatérale avec plusieurs messages. |

L'attribut inReplyTo peut aussi faire référence à l'attribut payloadID d'un document Request ou Response antérieur. Lorsqu'une transaction requête/réponse instaure une « conversation » au moyen de plusieurs interactions unidirectionnelles, le premier message peut contenir l'attribut payloadID du document Request ou Response le plus récent et le plus approprié envoyé dans l'autre direction. Un Message qui contient un élément PunchOutOrderMessage, par exemple, peut inclure un attribut inReplyTo dans lequel figure l'attribut payloadID de l'élément PunchOutSetupRequest qui a ouvert la session punchout (l'élément BuyerCookie intégré dans les documents punchout remplit une fonction similaire à celle de l'attribut inReplyTo).

### **Options de transport**

Il existe deux principaux protocoles de transport pour les messages asynchrones : HTTP et codage de formulaire par URL. Ce sont seulement les deux protocoles les mieux définis à ce jour ; d'autres seront sans doute pris en charge sous peu.

#### **Protocole HTTP**

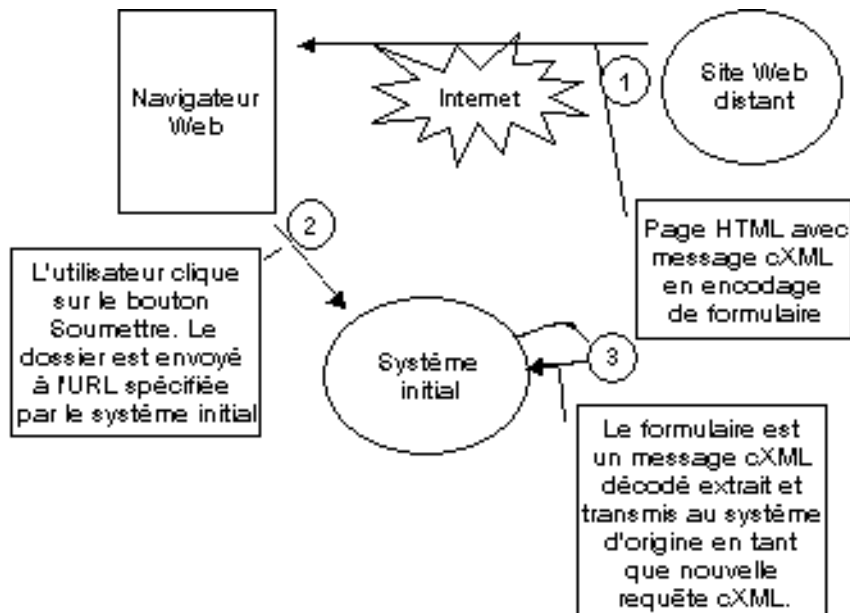
Le protocole HTTP s'emploie dans les communications asynchrones pour permettre aux applications de gestion des approvisionnements de se procurer des informations. Le seul type de transaction à faire appel à une communication HTTP asynchrone est la transaction GetPendingRequest, présentée à la page 101.

#### **Protocole de codage de formulaire par URL**

L'analyse de l'exécution de la transaction PunchOutOrderMessage permet de mieux comprendre les modalités de fonctionnement de ce protocole de transport. Le protocole de codage de formulaire par URL permet l'intégration entre un site Web distant et des applications de gestion des approvisionnements. Il permet en outre d'éviter de faire appel, sur le système de l'acheteur, à un serveur « à l'écoute » qui est directement accessible via Internet.

Le message cXML PunchOutOrderMessage n'est pas directement envoyé à l'application de gestion des approvisionnements par le site Web distant, mais codé sous forme de champ HTML Form masqué et acheminé jusqu'à l'URL spécifiée dans l'élément BrowserFormPost de la requête PunchOutSetupRequest. Lorsque l'utilisateur clique sur

Check Out, le site Web envoie les données à l'application de gestion des approvisionnements sous forme d'élément HTML Form Submit. Le diagramme qui suit illustre ce processus :



La syntaxe de compactage et de décompactage est détaillée ci-après.

### Compactage du formulaire

Le document PunchOutOrderMessage est encodé sous forme d'URL (conformément à la spécification HTTP) et affecté à un champ masqué sur le formulaire nommé cXML-urlencoded. L'élément HTML Form reçoit une méthode de transmission (METHOD of POST) et une ACTION définie par l'URL passée dans l'élément BrowserFormPost de la requête PunchOutSetupRequest. Exemple :

```
<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>
```

D'autres balises HTML de la page peuvent renfermer le fragment de code ci-dessus pour décrire plus précisément le contenu du panier d'achat.

**Remarque :** Lorsque les serveurs Web envoient le champ `cXML-urlencoded`, il n'est pas encore encodé sous forme d'URL. Cet encodage n'est exigé que lorsque le formulaire est adressé par des navigateurs Web (lorsque les utilisateurs cliquent sur Check Out dans l'exemple ci-dessus). Les navigateurs Web eux-mêmes répondent à cette exigence. Le serveur Web doit encoder en HTML la valeur du champ exclusivement, en ignorant les guillemets et autres caractères spéciaux pour que le formulaire s'affiche correctement pour l'utilisateur.

Le nom `cXML-urlencoded` ne fait pas la distinction entre majuscules et minuscules.

Pour les données `cXML-urlencoded`, l'analyseur récepteur n'est pas en mesure d'identifier un paramètre `charset` autre que le paramètre par défaut pour un support de type `text/xml`. Aucune information d'encodage des caractères n'est transportée dans la transaction HTTP POST pour les données transmises. Le serveur Web récepteur est incapable de déterminer l'encodage de la page HTML qui contient le champ masqué. Le document `cXML` acheminé de cette façon doit par conséquent faire appel à un codage des caractères `us-ascii`. Tous les caractères (y compris ceux qui sont « encodés URL » sous la forme « %XX ») présents dans le document XML doivent appartenir au jeu de caractères « `us-ascii` ». D'autres symboles Unicode peuvent être encodés à l'aide d'entités de type caractère dans ce document source.

### Encodage Base64

Le champ masqué `cXML-base64` facilite le traitement des documents internationaux. Les documents `cXML` contenant des symboles extérieurs au jeu « `us-ascii` » doivent faire appel à ce champ plutôt qu'au champ masqué `cXML-urlencoded`. Cette alternative présente une syntaxe pratiquement identique, si ce n'est que le document entier est codé en base64 pendant le transport au lieu d'être codé en HTML jusqu'au navigateur, ou codé sous forme d'URL jusqu'au serveur Web récepteur. L'encodage en base64 est décrit dans la spécification RFC 2045 « Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies ».

Le nom `cXML-base64` ne fait pas la distinction entre majuscules et minuscules.

L'encodage d'un document `cXML` en base64 à partir du site du fournisseur, via le navigateur, et jusqu'au serveur Web récepteur du site client permet de préserver le codage original des caractères. Bien qu'aucun paramètre `charset` n'accompagne les informations transmises, le document décodé (après suppression du codage de transfert) peut être traité comme un support de type `application/xml`. Grâce à cet encodage, l'analyseur récepteur est en mesure d'honorer tout attribut `encoding` spécifié dans la déclaration XML. Pour ce champ (comme pour tous les documents `application/xml`), l'encodage des caractères par défaut est UTF-8.



L'un de ces champs masqués (cXML-urlencoded ou cXML-base64) doit apparaître dans les données envoyées à l'application de gestion des approvisionnements. Bien que les destinataires doivent rechercher en priorité le champ cXML-base64 dans les données, l'envoi des deux champs est inutile.

### Décompactage et traitement d'un formulaire

L'application de gestion des approvisionnements, qui a préalablement fourni l'URL voulue, reçoit une transaction HTML Form POST contenant les données de type Form décrites plus haut. Le processeur de transactions Form POST doit tout d'abord rechercher la variable cXML-base64, extraire la valeur et décoder son contenu en base64. Si ce champ n'existe pas dans les données, le processeur de transactions Form POST recherche la variable cXML-urlencoded, extrait le message cXML encodé URL et le décode sous forme d'URL. Le contenu décodé du champ est ensuite traité de la même façon que s'il avait été reçu par le biais d'un cycle requête/réponse normal.

Le type de support du document après le décodage varie, et il existe différentes possibilités d'encodage des caractères :

- La variable cXML-urlencoded est de type text/xml sans attribut charset. Elle est donc limitée à un codage des caractères us-ascii. L'analyseur récepteur doit ignorer tout attribut encoding dans la déclaration XML du document cXML car le navigateur a peut-être modifié le codage.
- La variable cXML-base64 est de type application/xml et peut donc faire l'objet de n'importe quel encodage (indiqué par l'attribut encoding de la déclaration XML existante, le cas échéant).

La principale différence entre cette transaction et une transaction requête/réponse normale réside dans le fait qu'aucune réponse ne peut être générée, puisqu'il n'existe aucune connexion HTTP par l'intermédiaire de laquelle l'envoyer.

## Éléments de base

Les entités et éléments qui suivent apparaissent dans l'ensemble de la spécification cXML. La plupart des définitions constituent le vocabulaire de base grâce auquel les documents commerciaux de niveau supérieur sont décrits. Les Entités types et les éléments communs représentant des objets de niveau inférieur sont également définis ici.

### Entités types

La plupart de ces définitions sont tirées de la note sur les données XML soumise au World Wide Web Consortium (W3C). Certaines des entités types de niveau supérieur définies ici ne sont toutefois pas extraites de la note sur les données XML. Elles sont également présentées à la section « Enveloppe cXML », page 50.

***isoLangCode***

Code de langue ISO conforme à la norme ISO 639.

***isoCountryCode***

Code de pays ISO conforme à la norme ISO 3166.

***xmlLangCode***

Code de langue défini par la spécification XML 1.0 (à l'adresse [www.w3.org/TR/1998/REC-xml-19980210.html](http://www.w3.org/TR/1998/REC-xml-19980210.html)). En règle générale, il inclut un code de langue ISO 639 et (facultativement) un code de pays ISO 3166 séparés par un trait d'union. Contrairement à la recommandation XML, les codes IANA ou les codes de langue privés ne doivent pas être employés dans les documents cXML. Les sous-codes IANA et privés sont autorisés, mais doivent apparaître à la suite d'un code de pays ISO 3166 valide.

Le format de code de langue cXML recommandé est `xx[-YY[-zzz]*]?` où `xx` représente un code de langue ISO 639, `YY` un code de pays ISO 3166 et `zzz` un sous-code IANA ou privé pour la langue en question. Là encore, l'utilisation du code pays est toujours recommandée. Par convention, le code de la langue apparaît en minuscules et le code du pays en majuscules. Cette convention n'a cependant rien d'obligatoire pour l'identification correcte des codes.

***unitOfMeasure***

L'élément `UnitOfMeasure` décrit le mode de conditionnement ou de livraison du produit. Il doit être conforme aux codes d'unités de mesure UN/CEFACT. Vous trouverez une liste des codes UN/CEFACT sur le site [www.unece.org/cefact](http://www.unece.org/cefact).

***URL***

URL (Uniform Resource Locator), conforme à la définition du standard HTTP/1.1.

**Éléments de base**

Ces éléments, présents dans l'ensemble de la spécification, peuvent aussi bien être des éléments génériques tels `Name` et `Extrinsic` que des éléments plus spécifiques tels `Money`.

---

## Transaction Profile

---

Les documents ProfileRequest et ProfileResponse doivent être pris en charge par les implémentations de serveur cXML 1.1. La transaction Profile permet d'identifier les caractéristiques du serveur, notamment la version cXML prise en charge, les transactions et les options possibles pour ces transactions.

La réponse doit répertorier tous les documents Request gérés sur un site Web spécifique, et pas nécessairement tous ceux pris en charge par la société. Les fournisseurs aptes à recevoir des documents OrderRequest et à envoyer différents messages ou à prendre l'initiative de transactions requête/réponse décrivent les éléments OrderRequest qu'ils prennent en charge dans la transaction Profile.

La transaction Profile permet d'interroger un serveur via une commande « ping » au sein du protocole cXML.

### ProfileRequest

Cet élément est vide de tout contenu. Il est seulement acheminé jusqu'au serveur cXML voulu via l'élément Header. Le serveur répond par un élément ProfileResponse unique, comme indiqué ci-après. Les seules portions dynamiques de cette réponse sont les attributs payloadId et timestamp de l'élément cXML lui-même. Dans ce cas particulier, le fournisseur n'a pas l'obligation de fournir des réponses dans plusieurs langues.

Voici un exemple de ce type d'élément Request :

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

### ProfileResponse

Cet élément répertorie les transactions prises en charge, leur emplacement et toutes les options gérées. Voici un exemple d'élément ProfileResponse possible tant qu'aucune option n'est encore définie :

```
<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>

```

Et voici un exemple d'élément ProfileResponse plus probable de la part d'un fournisseur existant :

```

<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>

```

L'élément ProfileResponse possède l'attribut suivant :

|                      |  |
|----------------------|--|
| <b>effectiveDate</b> | Date et heure auxquelles ces services sont devenus accessibles. Les dates ne doivent pas être futures. |
|----------------------|--|

### **Option**

Valeur d'une option définie (pour le service dans son ensemble ou pour une transaction spécifique). Aucune option n'est encore définie.

L'élément Option possède l'attribut suivant :

|             |  |
|-------------|--|
| <b>name</b> | Nom de cette option. Cet attribut ne doit pas être affiché directement (le profil est destiné à un usage interne). |
|-------------|--|

### **Transaction**

Description d'une transaction prise en charge par ce service. La définition Profile actuelle indique les adresses auxquelles envoyer des requêtes spécifiques. De nouvelles définitions Option et de plus amples informations sur les requêtes prises en charge apparaîtront dans les informations de profil des versions futures du langage cXML.

L'élément Transaction doit contenir un élément URL.

L'élément Transaction possède l'attribut suivant :

|                    |   |  |
|--------------------|---|--|
| <b>requestName</b> | Requête spécifique acceptée par ce serveur à l'URL indiquée.<br>Les valeurs possibles sont :              |  |
|                    | ProfileRequest<br>PunchOutSetupRequest<br>GetPendingRequest<br>SupplierListRequest<br>SupplierDataRequest | OrderRequest<br>StatusUpdateRequest<br>SubscriptionListRequest<br>SubscriptionContentRequest |

## Définition des commandes

Les documents de commande cXML se composent d'un élément OrderRequest et d'une réponse générique. L'élément OrderRequest s'apparente à une commande d'achat. La réponse est un accusé de réception de la commande d'achat par le fournisseur. Il ne s'agit en aucun cas d'une garantie d'exécution de la commande, mais d'une simple confirmation de sa réception.

### OrderRequest

L'exemple qui suit représente la structure de l'élément OrderRequest :

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

### OrderRequestHeader

L'exemple qui suit détaille plus précisément un élément 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">

```

```

        Informations au format XML.
    </Comments>
    <Followup>
        <URL>http://acme.com/cgi/orders.cgi</URL>
    </Followup>
</OrderRequestHeader>

```

L'élément OrderRequestHeader possède les attributs suivants :

|                                      |   |
|--------------------------------------|---|
| <b>orderID</b>                       | Identificateur de cette commande. Analogue au numéro d'une commande d'achat.  |
| <b>orderDate</b>                     | Date et heure auxquelles cette commande a été passée, au format ISO 8601.   |
| <b>type</b><br>(facultatif)          | Type de la requête : new (valeur par défaut), update ou delete.   |
| <b>requisitionID</b><br>(facultatif) | Identificateur de la demande d'achat émise par l'acheteur pour la commande dans son intégralité. Il peut prendre la même valeur que l'attribut orderID ou ne pas apparaître du tout. Il ne doit pas être mentionné si l'attribut requisitionID est déjà spécifié dans un élément ItemOut.   |
| <b>shipComplete</b><br>(facultatif)  | Préférence interdisant les livraisons partielles. La seule valeur autorisée est « yes ». Par défaut, les articles sont livrés lorsqu'ils sont disponibles.<br><br>Les commandes peuvent inclure des articles dont les éléments ShipTo sont variables, et, par conséquent, seuls les groupes d'articles possédant des adresses d'expédition communes doivent être retenus lorsque l'attribut shipComplete prend la valeur « yes ». |

Les éléments OrderRequestHeader et ItemOut (développé à l'aide de l'élément ItemDetail) contiennent des informations similaires. Tandis que OrderRequestHeader fournit des informations générales sur la facturation (BillTo) et le paiement (Payment), ItemOut décrit au contraire des articles spécifiques (dans les éléments ItemID, ItemDetail et Distribution).

N'utilisez pas les informations indiquées dans l'élément OrderRequestHeader comme valeur par défaut pour les éléments propres aux articles. Lorsqu'ils sont présents, les attributs ShipTo, Shipping, Contact et Extrinsic doivent apparaître soit avec chaque élément ItemOut soit dans l'élément OrderRequestHeader. Les éléments Comments et Tax peuvent apparaître simultanément sur les deux niveaux. Les différents éléments Comments ne doivent toutefois pas dupliquer des informations et l'élément Tax de niveau en-tête contient un total pour la commande.

### Total

Cet élément indique le montant monétaire total de la commande. Il sert de conteneur à l'élément Money.

## ShipTo/BillTo

Ces éléments contiennent les adresses des entités « Envoyer à » et « Facturation » qui figurent sur le document OrderRequest.

Une commande doit être facturée à une entité unique. Par conséquent, l'élément BillTo apparaît seulement dans la partie OrderRequestHeader. Les articles répertoriés dans une commande peuvent être livrés à des adresses différentes. Tout comme l'élément Shipping (voir section suivante), l'élément ShipTo peut donc apparaître soit dans l'élément OrderRequestHeader soit dans des éléments ItemOut individuels.

## Shipping

Cet élément précise le mode et le coût de livraison des articles répertoriés dans la commande. Si l'élément Shipping est présent dans l'élément OrderRequestHeader, il ne doit pas apparaître dans les éléments ItemOut individuels. S'il est absent de l'élément OrderRequestHeader, il doit par contre figurer dans les éléments ItemOut.

## Tax

Cet élément indique la taxe associée à la commande. Il est présent si l'organisation acheteuse procède au calcul des taxes. Lorsqu'il apparaît dans l'élément OrderRequestHeader, l'élément Tax indique la taxe totale d'une commande. Lorsqu'ils figurent au niveau des articles, les élémentsTax peuvent définir le montant des taxes applicables à des articles spécifiques.

## Payment

Cet élément décrit l'instrument servant au paiement des articles commandés. Dans l'exemple ci-dessus, l'élément Payment contient un élément PCard, qui code une carte d'achat standard dans le document cXML. Dans les versions futures, d'autres instruments de paiement seront définis et pris en charge.

## Contact

Informations qui permettent au fournisseur d'assurer le suivi d'une commande. Cet élément identifie une personne et répertorie un certain nombre de moyens permettant d'entrer en contact avec cette personne ou entité. Le seul élément obligatoire est le nom (Name) du contact. Les autres options possibles et récurrentes sont les éléments PostalAddress (déconseillé pour la correction immédiate des problèmes de commande), Email, Phone, Fax et URL.



Les organisations acheteuses peuvent choisir de recourir à cet élément pour identifier le demandeur initial, l'administrateur système de l'application de gestion des approvisionnements ou tout autre contact responsable de la correction des problèmes liés aux commandes. L'élément Contact peut différer des éléments BillTo et ShipTo d'une commande.

L'élément Contact possède l'attribut suivant :

|                             |  |
|-----------------------------|--|
| <b>role</b><br>(facultatif) | Fonction de cette personne dans le cadre du processus de gestion des approvisionnements. Les valeurs possibles sont endUser, administrator, purchasingAgent, technicalSupport, customerService et sales. |
|-----------------------------|--|

Un même role ne peut apparaître à la fois au niveau de l'en-tête et au niveau des articles.

Il n'existe aucun rôle par défaut, compte tenu de la diversité des valeurs possibles pour l'élément Contact. Les applications cXML assimilent donc un Contact sans attribut role à un rôle supplémentaire.

### Comments

Informations arbitraires lisibles à « l'œil nu » que les acheteurs peuvent envoyer dans les commandes d'achat. Ces données de type chaîne de caractères ne sont pas destinées à être traitées par les systèmes automatisés des sites des fournisseurs.

L'élément Comments peut contenir un élément Attachment qui permet de joindre des fichiers externes.

### Attachment

L'élément Comments permet de joindre des fichiers externes qui complètent les commandes d'achat. L'élément Attachment apparaît à l'intérieur de l'élément Comments, et il peut exclusivement contenir une référence à la partie MIME externe de la pièce jointe. Toutes les pièces jointes doivent être envoyées par le biais d'une transmission à éléments multiples unique, parallèlement au document OrderRequest. Même si c'est impossible, la valeur contentID indiquée par l'élément Attachment doit permettre la récupération de la pièce jointe.

Pour plus de précisions sur le transfert des fichiers joints, consultez la section « Transmission des pièces jointes », page 53.

Un élément Attachment contient une URL unique suivie de la notation « cid: ». Voici un exemple de fichier joint à un document cXML :

```
<Comments>
  <Attachment>
    <URL>cid: uniqueCID@cxml.org</URL>
  </Attachment>
  Veuillez consulter l'image en pièce jointe pour avoir un aperçu
  du résultat final
</Comments>
```

L'élément Comments apparaît à différents emplacements dans le protocole cXML, mais ne peut contenir d'élément Attachment que dans les documents OrderRequest.

### Followup

Indique l'URL à laquelle les documents StatusUpdateRequest futurs devront être adressés. Cet emplacement est l'emplacement cible pour tous les documents ultérieurs faisant référence au document OrderRequest en cours.

### Extrinsic

Cet élément contient des informations lisibles par machine en rapport avec la commande, mais non définies par le protocole cXML. L'élément Comments, en revanche, transmet des informations à « usage humain ». Les éléments Extrinsic contiennent des données susceptibles d'apparaître dans des documents ultérieurs, contrairement à l'élément Comments. À ce niveau, l'élément Extrinsic développe la description de tous les articles qui figurent dans la commande d'achat. Certains éléments Extrinsic peuvent également décrire la commande d'achat dans son ensemble sans aucune incidence sur la signification des éléments ItemOut présents.

Chaque élément Extrinsic nommé ne peut apparaître qu'une seule fois dans les listes associées à l'élément OrderRequestHeader et à des éléments ItemOut individuels (placés à l'intérieur des éléments ItemDetail présents). Un même nom ne doit jamais apparaître à la fois dans la liste OrderRequestHeader et dans une liste associée aux éléments ItemOut. Si un nom et une valeur Extrinsic identiques sont répétés dans toutes les listes ItemOut, il est préférable de les placer dans l'élément OrderRequestHeader.

L'élément Extrinsic peut également apparaître dans les éléments IndexItem, PunchOutSetupRequest et ContractItem. Ces contextes sont décrits un peu plus loin dans ce document.

## ItemOut

L'exemple qui suit représente un élément ItemOut valide réduit au strict minimum.

```
<ItemOut quantity="2">
  <ItemID>
    <SupplierPartID>5555</SupplierPartID>
  </ItemID>
</ItemOut>
```

L'élément ItemOut possède les attributs suivants :

|  |  |
|--|--|
| <b>quantity</b>                              | Nombre d'articles désiré. Les fractions sont autorisées pour certaines unités de mesure. La valeur peut déjà avoir été contrôlée par le fournisseur lors d'une session punchout. Elle ne doit jamais être négative.  |
| <b>lineNumber</b><br>(facultatif)            | Position de cet article dans une commande. Cette valeur ordinale augmente d'un point par ItemOut dans un document OrderRequest « nouveau ». Les clients doivent toujours spécifier cet attribut dans un document OrderRequest, bien qu'il puisse n'avoir aucune utilité dans d'autres contextes ItemOut. |
| <b>requisitionID</b><br>(facultatif)         | Identificateur de la demande émise par l'acheteur pour cet article. Il ne doit pas être inclus si l'attribut requisitionID est spécifié dans l'élément OrderRequestHeader.   |
| <b>requestedDeliveryDate</b><br>(facultatif) | Date de livraison demandée pour l'article, qui autorise des dates de livraison au niveau des articles dans le document OrderRequest. Elle doit être conforme au format ISO 8601.   |

L'attribut lineNumber affecté à un article ne change pas lors des différentes mises à jour de la commande. La suppression d'articles d'une commande ne modifie jamais l'attribut lineNumber des articles restants. Les numéros attribués aux nouveaux articles sont toujours supérieurs à ceux qui figurent déjà dans la commande. La modification d'un article existant (une augmentation de quantité, par exemple) n'a aucune incidence sur l'attribut lineNumber de cet article.

L'exemple qui suit représente un élément ItemOut un peu plus complexe.

```
<ItemOut quantity="2" lineNumber="1"
  requestedDeliveryDate="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>
    <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">
    Informations au format XML.
  </Comments>
</ItemOut>

```

L'élément `ItemDetail` permet d'envoyer des informations supplémentaires aux fournisseurs, en plus de l'identificateur unique de l'article représenté par l'élément `ItemID`.

Les éléments `ShipTo`, `Shipping`, `Tax`, `Contact`, `Comments` et `Extrinsic` (dont certains sont imbriqués dans l'élément `ItemDetail`) sont identiques à ceux qui peuvent apparaître dans l'élément `OrderRequestHeader`. Ces éléments permettent de représenter des données propres à chaque article, telles que le conditionnement, le type de conditionnement et le coût lié. Utilisez ces éléments au niveau `OrderRequestHeader` ou au niveau `ItemOut`, mais jamais aux deux.

### Distribution

L'élément `Distribution` divise le coût d'un article entre plusieurs partenaires. Les fournisseurs font ressortir l'élément `Distribution` sur les factures pour faciliter la procédure de rapprochement du côté de l'acheteur.

### Accounting

L'élément `Accounting` regroupe des éléments `Segments` qui permettent d'identifier les destinataires des factures.

L'élément `Accounting` possède l'attribut suivant :

|             |                                     |
|-------------|-------------------------------------|
| <b>name</b> | Nom de cette combinaison comptable. |
|-------------|-------------------------------------|

L'élément `Segment` possède les attributs suivants :

|             |   |
|-------------|---|
| <b>type</b> | Nom qui identifie ce Segment par rapport aux autres dans l'élément <code>Accounting</code> .  |
| <b>id</b>   | Identificateur unique au sein de ce type de Segment. Cette valeur peut correspondre au code du compte en cas de segment de type « <code>Cost Center</code> ». |

### Charge

Cet élément représente le montant à facturer à l'entité représentée par l'élément `Accounting`.

## Réponse à un document OrderRequest

Il s'agit de la partie réponse d'une transaction requête/réponse. L'exemple qui suit représente la réponse (élément Response) à un document OrderRequest :

```
<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>
```

Comme indiqué ci-dessus, cet élément Response est direct. Dans ce cas, il n'existe aucun élément nommé « OrderResponse », car la seule information qui doit être renvoyée au demandeur est la partie Status de l'élément Response.

L'élément Response indique au demandeur que son document OrderRequest a correctement été analysé et traité par la partie distante de la connexion HTTP. Il ne communique pas d'accusé de réception en rapport avec la commande (articles qui peuvent être livrés et articles qui doivent être commandés, par exemple).

## Transaction Punchout

---

Les définitions de message punchout sont des messages requête/réponse transportés par les éléments Request et Response. Les fonctionnalités punchout ne sont accessibles que si tous les messages qui suivent sont mis en œuvre par les fournisseurs.

### PunchOutSetupRequest

Les éléments PunchOutSetupRequest et PunchOutSetupResponse constituent la paire requête/réponse utilisée pour la configuration d'une session punchout sur un système distant. Ils permettent au client d'identifier l'application de gestion des approvisionnements, d'envoyer des informations de configuration et de recevoir une réponse indiquant la destination à laquelle se rendre pour ouvrir une session de navigation HTML sur le site Web distant.

Un élément PunchOutSetupRequest est transporté par l'élément Request. L'exemple qui suit représente un élément 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>

```

L'élément PunchOutSetupRequest possède l'attribut suivant :

|                  |  |
|------------------|--|
| <b>operation</b> | Spécifie le type d'élément PunchOutSetupRequest : « create », « inspect » ou « edit ». |
|------------------|--|

Cet élément contient aussi les éléments suivants : BuyerCookie, Extrinsic, BrowserFormPost, Contact, ShipTo, SelectedItem, SupplierSetup et une liste ItemOut. Seul l'élément BuyerCookie est obligatoire. La structure des éléments Extrinsic, Contact et ShipTo est présentée plus précisément au paragraphe « OrderRequestHeader », page 69. L'élément ItemOut est détaillé à la section « ItemOut », page 75. Dans ce contexte (à l'extérieur d'un document OrderRequest), les éléments Distribution et Comments et les attributs lineNumber, requisitionID et requestedDeliveryDate d'un élément ItemOut sont d'une utilité relative, voire nulle, et leur présence n'est donc pas indispensable. Les sessions punchout intervenant avant la commande, ces informations n'ont aucune raison d'être dans un document PunchOutSetupRequest.

Une liste ItemOut décrit un caddie existant (articles d'une session punchout antérieure). L'opération inspect établit une session punchout en lecture seule (appliquée à la fois par le client et par le serveur) pour afficher des détails sur les articles répertoriés. L'opération edit démarre aussi à partir du caddie précédent (décrit à l'aide de la liste ItemOut), mais autorise les modifications. La prise en charge de l'opération edit suppose la prise en charge de l'opération inspect (consultez les sections « PunchOutOrderMessageHeader », page 82 et « Caddies vides », page 83).

### **BuyerCookie**

Cet élément transmet des informations qui sont invisibles au site Web distant, mais qui doivent être renvoyées à l'émetteur pour toutes les opérations punchout suivantes. Il permet à l'application de gestion des approvisionnements de répondre à plusieurs requêtes punchout en attente.

### ***BrowserFormPost***

Cet élément indique la destination des données qui figurent dans le document `PunchOutOrderMessage`. Il contient un élément URL dont l'usage sera ultérieurement détaillé dans la définition du document `PunchOutOrderMessage`. Si la méthode d'encodage de formulaire par URL n'est pas utilisée, la présence de cet élément est inutile.

### ***Extrinsic***

Cet élément facultatif contient toutes les données supplémentaires que le demandeur souhaite transmettre au site Web externe. Cet exemple transmet le service auquel appartient l'utilisateur à l'origine de l'opération punchout. La spécification cXML ne définit pas le contenu des éléments `Extrinsic` – il appartient à chaque demandeur et à chaque site Web distant de convenir de ce contenu et de sa mise en œuvre.

Les éléments `Extrinsic` ont pour but de fournir des informations supplémentaires lisibles par machine. Ils optimisent le protocole cXML en assurant la prise en charge de fonctionnalités qui ne sont pas exigées par toutes les implémentations. Dans ce contexte, les nouvelles données fournissent plus de précisions sur l'utilisateur qui est à l'origine de la requête punchout.

L'élément `Extrinsic` peut également apparaître dans les éléments `OrderRequestHeader`, `ItemDetail` et `ContractItem`. Ces contextes sont décrits plus précisément dans d'autres sections de ce document.

### ***SelectedItem***

Cet élément facultatif indique les articles que les utilisateurs souhaitent acheter par punchout. Il contient un élément `ItemID` unique et obligatoire. Cet élément conduit les utilisateurs de leur catalogue local jusqu'au site Web du fournisseur.

Cet élément est généralement présent dans les opérations `create`. Les applications de gestion des approvisionnements qui permettent aux utilisateurs d'exécuter des opérations de punchout directement à partir de listes fournisseurs doivent omettre l'élément `SelectedItem`.

Pour les opérations `edit` et `inspect`, l'élément `SelectedItem` ne doit apparaître que si l'utilisateur a choisi de retourner sur le site Web du fournisseur alors qu'il visualisait les nouvelles informations présentes dans le catalogue local plutôt que les articles contenus dans une demande existante. Dans les deux cas, le caddie en cours doit figurer dans la liste `ItemOut`.

Les fournisseurs peuvent créer leurs catalogues de façon à ce que l'élément `SelectedItem` conduise à un punchout de niveau magasin, rayon ou produit. Plus l'article est précis dans le catalogue, moins les utilisateurs doivent effectuer de recherches sur le site Web du fournisseur.



## ***SupplierSetup***

Cet élément facultatif spécifie l'URL à laquelle adresser le document PunchOutSetupRequest. Cet élément est inutile si le concentrateur de réseau de commerce électronique connaît l'URL punchout du fournisseur.

## **PunchOutSetupResponse**

Dès réception d'un document PunchOutSetupRequest, le site Web distant répond par un document PunchOutSetupResponse, comme indiqué ci-après :

```
<PunchOutSetupResponse>
  <StartPage>
    <URL>
      http://premier.workchairs.com/store?23423SDFSD23
    </URL>
  </StartPage>
</PunchOutSetupResponse>
```

## ***StartPage***

Cet élément contient un élément URL qui indique l'URL à transmettre au navigateur pour ouvrir la session de navigation punchout demandée dans le document PunchOutSetupRequest. Cette URL doit renfermer suffisamment d'informations d'état (identité du demandeur et élément BuyerCookie approprié, par exemple) pour établir une liaison avec un contexte de session sur le site Web distant.

À ce stade, l'utilisateur qui est à l'origine du document PunchOutSetupRequest explore le site Web externe et sélectionne les articles à retransmettre à l'application de gestion des approvisionnements par l'intermédiaire d'un élément PunchOutOrderMessage.

## **PunchOutOrderMessage**

Cet élément envoie le contenu du panier d'achat distant à l'émetteur d'un document PunchOutSetupMessage. Il peut contenir un nombre d'informations bien plus élevé que les autres messages car il doit être en mesure d'exprimer entièrement le contenu de tout panier d'achat concevable sur le site Web externe. Ce message n'obéit pas strictement au modèle requête/réponse.

Le site Web distant génère un document PunchOutOrderMessage lorsque l'utilisateur fait son choix. Ce message communique le contenu du panier d'achat distant à l'application de gestion des approvisionnements, comme dans l'exemple :

```
<PunchOutOrderMessage>
  <BuyerCookie>34234234ADF5DF234234</BuyerCookie>
  <PunchOutOrderMessageHeader operationAllowed="create">
    <Total>
      <Money currency="USD">100.23</Money>
    </Total>
  </PunchOutOrderMessageHeader>
  <ItemIn quantity="1">
    <ItemID>
      <SupplierPartID>1234</SupplierPartID>
      <SupplierPartAuxiliaryID>
        informations supplémentaires sur cet article
      </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>
```

Ces éléments sont détaillés dans les sections qui suivent.

### ***BuyerCookie***

Cet élément est identique à celui qui a été passé dans le document PunchOutSetupRequest original. Il doit être renvoyé ici pour permettre à l'application de gestion des approvisionnements de faire correspondre le document PunchOutOrderMessage à un document PunchOutSetupRequest antérieur.

### ***PunchOutOrderMessageHeader***

Cet élément fournit des informations relatives au contenu intégral du panier d'achat qui fait l'objet du transfert. Seul l'élément Total, qui représente le coût total des articles ajoutés à la demande, est obligatoire. D'autres éléments autorisés sont Shipping et Tax ; ils représentent le montant et la description des frais d'expédition ou des taxes calculés sur le site Web distant. L'élément ShipTo est également facultatif et indique les informations d'adresse « Envoyer à » que l'utilisateur a sélectionnées sur le site distant ou qui ont été transmises dans le document PunchOutSetupRequest original. Tous les montants monétaires apparaissent dans un élément Money qui précise toujours la devise dans un format standard.

L'élément `PunchOutOrderMessageHeader` possède l'attribut suivant :

|                         |  |
|-------------------------|--|
| <b>operationAllowed</b> | Spécifie les opérations <code>PunchOutSetupRequest</code> autorisées : <code>create</code> , <code>inspect</code> ou <code>edit</code> . |
|-------------------------|--|

Cet attribut détermine si l'utilisateur peut effectuer des transactions `PunchOutSetupRequest` ultérieures contenant des données tirées de ce document `PunchOutOrderMessage`. Avec la valeur `operationAllowed="create"`, seul un document `OrderRequest` ultérieur peut contenir ces articles. Dans le cas contraire, l'application de gestion des approvisionnements peut inspecter ou modifier le caddie ultérieurement (en déclenchant des transactions `PunchOutSetupRequest` avec les opérations voulues et les éléments `ItemID` correspondant à la liste `ItemIn` renvoyée dans ce document `PunchOutOrderMessage`). La prise en charge de l'opération `edit` suppose la prise en charge de l'opération `inspect`. L'application de gestion des approvisionnements peut toujours utiliser les articles dans un document `OrderRequest` ultérieur.

### ***Caddies vides***

Le document `PunchOutOrderMessage` peut contenir une liste d'articles correspondant à un caddie sur le site Web du fournisseur. Il indique toujours la fin d'une session `punchout` interactive. Les paragraphes qui suivent donnent quelques exemples de cas où le document `PunchOutOrderMessage` ne comporte pas de liste d'articles. Ces messages permettent aux clients de reprendre immédiatement les opérations dès que l'utilisateur quitte le site Web du fournisseur.

Si l'opération effectuée dans le document `PunchOutSetupRequest` original était une opération `inspect`, la liste d'articles du document `PunchOutOrderMessage` doit être ignorée par l'application de gestion des approvisionnements. Le site du fournisseur ne doit pas retourner d'éléments `ItemIn` dans ce cas. Si un document `PunchOutOrderMessage` ne contient aucun élément `ItemIn` et si l'opération était `create`, aucun article ne doit être ajouté à la demande. Le site fournisseur ou l'utilisateur a annulé la session `punchout` sans créer de caddie. Si l'opération était `edit` et si le document `PunchOutOrderMessage` ne contient pas d'éléments `ItemIn`, les articles existants de cette session `punchout` doivent être supprimés de la demande dans l'application de gestion des approvisionnements.

Le code de statut « 204/No Content » indique la fin d'une session sans modification du caddie. Là encore, le document `PunchOutOrderMessage` (qui est toujours exigé pour l'élément `BuyerCookie`) ne doit pas contenir d'éléments `ItemIn`. Ce code est traité de la même façon que dans les autres cas de caddies « vides » détaillés ci-dessus, sauf s'il s'agissait d'une opération `edit`. Dans ce cas, l'utilisateur a annulé la session sans apporter de modification et la demande ne doit, par conséquent, pas être modifiée dans l'application de gestion des approvisionnements.

## **ItemIn**

Cet élément ajoute un article qui se trouve dans un panier d'achat à une demande d'achat dans l'application de gestion des approvisionnements. Il peut contenir différents éléments, dont deux seulement sont obligatoires : ItemID et ItemDetail.

L'élément ItemIn possède les attributs suivants :

|                                   |  |
|-----------------------------------|--|
| <b>quantity</b>                   | Nombre d'articles sélectionnés par l'utilisateur sur le site Web distant. Le site fournisseur pouvant accepter les unités partielles, le protocole autorise les quantités fractionnées. Cette valeur ne doit jamais être négative.   |
| <b>lineNumber</b><br>(facultatif) | Position de cet article dans une commande. Étant donné que les sessions punchout surviennent généralement avant la commande et que le serveur n'est jamais en mesure de contrôler la position des articles dans une commande, cet attribut est inutile dans le cas d'un document PunchOutOrderMessage. |

Les éléments facultatifs sont ShipTo, Shipping et Tax ; ce sont les mêmes éléments que ceux indiqués dans le document PunchOutOrderMessage ci-dessus.

Exception faite des éléments Distribution et Comments et des attributs requisitionID et requestedDeliveryDate accessibles dans l'élément ItemOut, les structures ItemIn et ItemOut sont mises en correspondance une à une. Le système acheteur initial peut effectuer une conversion directe entre les listes ItemIn et ItemOut lors du déclenchement d'une opération inspect ou edit. Les fournisseurs peuvent procéder à une conversion de l'une à l'autre (en ignorant les extensions répertoriées dans l'élément ItemOut) lors de l'exécution d'une opération edit. Le système acheteur initial peut effectuer la conversion directe et ajouter des informations d'expédition et de distribution et des commentaires complémentaires lors de l'établissement d'une transactionOrderRequest. Les données ItemDetail (à l'exception des éléments Extrinsic) contenues dans les éléments ItemIn ne doivent pas être supprimées lors de la conversion d'éléments ItemIn en éléments ItemOut.

## **ItemID**

Cet élément identifie l'article de façon unique sur le site Web distant. C'est le seul élément qui doit impérativement être renvoyé sur le site Web distant en vue d'identifier à nouveau l'article transféré.

L'élément ItemID en contient deux autres : SupplierPartID et SupplierPartAuxiliaryID. Seul l'élément SupplierPartID est obligatoire. L'élément SupplierPartAuxiliaryID permet la configuration complexe du transport sur le site Web distant ou la définition d'informations de nomenclature, ce qui facilite la réidentification de l'article lorsqu'il est présenté au site Web distant par la suite.

Si l'élément `SupplierPartAuxiliaryID` contient des caractères spéciaux (des éléments XML supplémentaires qui ne sont pas définis dans le protocole cXML, par exemple), ils doivent être traités correctement. Comme les informations `SupplierPartAuxiliaryID` doivent transiter par les applications et être retransmises au fournisseur initial, un sous-jeu interne contenant des éléments XML supplémentaires est insuffisant.

### ItemDetail

Cet élément fournit des informations descriptives sur l'article, qui sont présentées aux utilisateurs par les applications de gestion des approvisionnements. Le contenu d'un élément `ItemDetail` peut être très complexe, mais les exigences minimales sont simples : `UnitPrice`, `Description`, `UnitOfMeasure` et `Classification`.

Dans le contexte d'un élément `ItemIn`, les éléments `Extrinsic` contenus dans un élément `ItemDetail` fonctionnent de la même façon que ceux qui figurent dans un élément `Index` (et plus précisément un élément `IndexItemAdd`).

### Description

Cet élément décrit l'article sous forme de texte. Ce texte risquant de dépasser les limites d'une table courte d'articles (ou toute autre interface utilisateur restreinte) et des troncatures aléatoires étant susceptibles de se produire, l'élément `Description` renferme un élément `ShortName` facultatif. Lorsqu'il est défini, les clients doivent présenter l'élément `ShortName` plutôt que le texte `Description` tronqué dans les champs dont l'espace est limité. Les clients doivent poursuivre la troncature du texte `Description` en l'absence d'élément `ShortName`.

Exemple :

```
<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>
```

peut apparaître sous la forme « Big Computer » lorsque l'espace est limité, et sous la forme « Big Computer: This wonder ... lying around. » (ou sous forme de deux champs distincts mais complets) lorsque l'espace est suffisant pour afficher davantage de texte.

---

## Modifications de statut ultérieures

---

Au terme de la transaction OrderRequest, les fournisseurs et les serveurs intermédiaires peuvent être amenés à retransmettre des informations complémentaires au système acheteur. Les transactions décrites dans cette section s'utilisent dans cette optique. Ces transactions présentent un certain nombre de similitudes au niveau de la syntaxe et des éléments.

Tout comme la réponse à un document OrderRequest (détaillée à la section « Réponse à un document OrderRequest », page 78), aucune de ces transactions n'inclut d'élément Response spécifique. Le document renvoyé contient en effet un élément Response pratiquement vide (seul un élément Status est présent). Chaque document renvoyé se présente sous la forme :

```
<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>
```

Le code renvoyé est « 200 » seulement si l'opération a réussi.

## DocumentReference

L'élément DocumentReference contient les informations qui permettent d'associer la demande de mise à jour à un document spécifique. Il répète l'attribut obligatoire du document antérieur et ajoute un identificateur facultatif généré par le fournisseur. Exemple :

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

L'élément `DocumentReference` ne renferme aucun autre élément, mais possède les attributs suivants :

|                  |  |
|------------------|--|
| <b>payloadID</b> | <p>Numéro unique indépendant de l'espace et de l'heure, utilisé à des fins de consignation pour identifier des documents. Cette valeur ne doit pas être modifiée pour les nouvelles tentatives.</p> <p>L'implémentation recommandée est :</p> <p>dateheure.id processus.nombre aléatoire@nomhôte</p> <p>Il est directement tiré de l'élément cXML du document <code>OrderRequest</code>.</p> |
| <b>orderID</b>   | <p>Identificateur de cette commande. Indique le numéro de la commande d'achat à ce jour.</p> <p>Il est directement tiré de l'élément <code>OrderRequestHeader</code> du document <code>OrderRequest</code>.</p>  |

## StatusUpdateRequest

Cette transaction signale à un nœud antérieur les modifications intervenues dans le statut de traitement d'une commande. Une modification mérite plus particulièrement d'être soulignée : lorsqu'un concentrateur intermédiaire parvient à retransmettre un document `OrderRequest`, il peut faire part de ce succès à l'expéditeur original ou à un concentrateur situé en amont. Le passage par différentes files d'attente et procédures au niveau du fournisseur ou du concentrateur peut également revêtir une importance particulière pour l'acheteur.

Cette requête procède à la mise à jour du statut d'un document `OrderRequest` unique.  
Exemple :

```
<cXML version="1.1.007" xml:lang="en-US"
  payloadID="0c30050@supplier.com"
  timestamp="2000-01-08T23:00:06-08:00">
  <Header>
    Informations de routage, d'identification et d'authentification.
  </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>
```

Cette requête contient seulement un élément `DocumentReference` et un élément `Status`. Ils sont tous deux obligatoires. L'élément `Status` peut communiquer une erreur de transport ultérieure rencontrée par un concentrateur intermédiaire. La syntaxe de cet élément est identique à celle d'un élément `Status` qui peut avoir été retourné dans la réponse HTTP initiale à un document `OrderRequest`.

Le code `200/OK` est essentiel lorsque des documents sont stockés et acheminés. Ce code indique qu'un fournisseur a commencé le traitement du document `OrderRequest` ou qu'un concentrateur a transmis le document. Le destinataire sait qu'il ne doit plus s'attendre à recevoir d'autres documents `StatusUpdateRequest` après réception du code `200/OK`.

Les fournisseurs et concentrateurs qui font appel à la transaction `StatusUpdate` doivent retourner le code `201/Accepted` lorsqu'un document `OrderRequest` est mis en file d'attente en vue d'un traitement ultérieur. Après l'envoi du code `200/OK` (dans la réponse `Response` immédiate à un document `OrderRequest` ou à un document `StatusUpdateRequest` ultérieur), le serveur ne doit pas envoyer d'autres transactions `StatusUpdate` pour cette commande. Les erreurs qui surviennent plus tard dans le traitement peuvent entraîner des exceptions à cette règle.

## Définitions de catalogue

---

Les définitions de catalogue cXML se composent de trois éléments principaux : `Supplier`, `Index` et `Contract`. Ils décrivent tous trois les données utilisées de façon permanente ou mises en cache au sein du système de gestion des approvisionnements d'une organisation acheteuse ou d'un concentrateur.

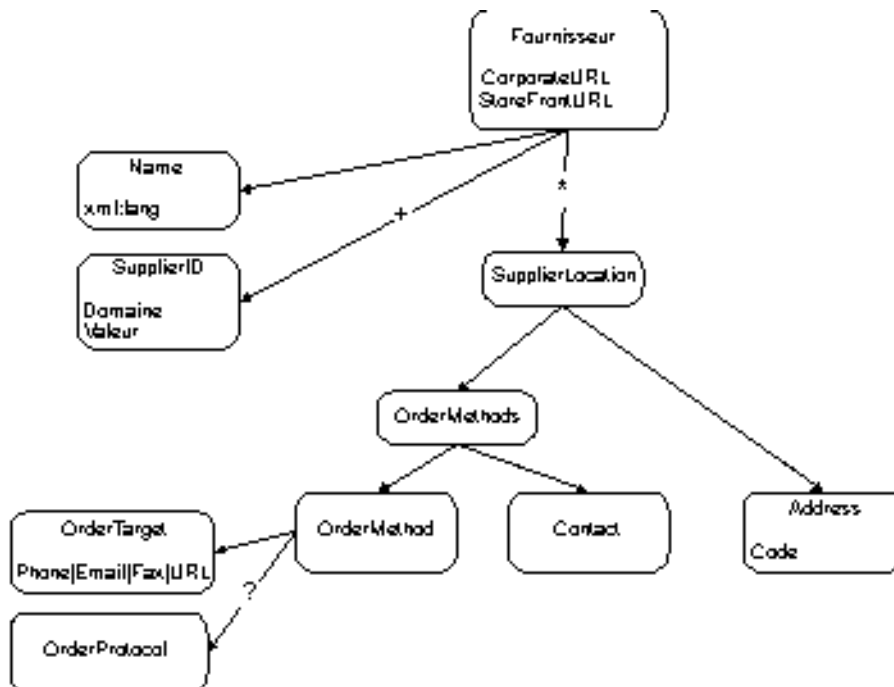
- `Supplier` – Contient des données de base sur le fournisseur telles adresse, contact et informations de commande.
- `Index` – Fournit des détails sur le stock de produits et de services du fournisseur : description, références et codes de classification, par exemple.
- `Contract` – Fournit des indications sur les aspects modulables du stock qui ont fait l'objet d'une négociation entre l'acheteur et le vendeur (le prix, par exemple).

Notez que l'élément `Index` fait appel à plusieurs sous-éléments pour décrire les articles répertoriés dans les stocks des fournisseurs. Les fournisseurs peuvent envoyer les informations tarifaires en vue de leur mise en cache dans les systèmes des acheteurs, ou des informations punchout pour permettre aux acheteurs de communiquer par punchout avec des sites Web distants pour rechercher des prix et autres informations.



## Supplier

L'élément Supplier intègre un fournisseur de produits ou services nommément désigné. Il doit posséder un élément Name et un élément SupplierID. Il peut en outre fournir des informations d'adresse et de commande supplémentaires pour le fournisseur :



L'élément Supplier possède les attributs suivants :

|                                      |  |
|--------------------------------------|--|
| <b>corporateURL</b><br>(facultatif)  | URL du site Web du fournisseur.                          |
| <b>storeFrontURL</b><br>(facultatif) | URL du site Web permettant des achats ou une navigation. |

L'exemple qui suit représente la structure de l'élément `Supplier` :

```

<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***

Certains fournisseurs gèrent leurs activités à partir de plusieurs sites. Un élément `SupplierLocation` peut être employé pour chacun de ces sites. Cet élément indique aussi les types d'activité exécutés sur ce site ou les modalités d'acceptation des commandes. Un élément `SupplierLocation` contient un élément `Address` et un ensemble d'éléments `OrderMethods`.

## OrderMethods et OrderMethod

L'élément `OrderMethods` regroupe un ou plusieurs éléments `OrderMethod` pour l'élément `SupplierLocation` en question. La position des éléments `OrderMethods` dans la liste est significative – le premier élément désigne la méthode de commande de prédilection, le deuxième la priorité suivante, et ainsi de suite par ordre de préférence décroissant.

L'élément `OrderMethod` synthétise les informations de commande sous forme d'une cible de commande (téléphone, télécopie ou URL, par exemple) et d'un protocole facultatif qui précise les types de commande attendus au niveau de la cible (« cxml » dans le cas d'une cible URL, par exemple).

## Index

Cet élément est un élément racine pour la mise à jour des catalogues au sein des systèmes de gestion des approvisionnements des organisations acheteuses.

Un élément `Index` est associé à un fournisseur unique. L'élément `Index` autorise une liste d'ID fournisseur, dans laquelle chaque ID est considéré comme le synonyme de ce fournisseur.

L'élément `Index` contient un ou plusieurs éléments `IndexItem` ainsi qu'un ensemble facultatif d'éléments `SearchGroup` qui permettent de définir des données de recherche paramétrique pour certains articles. L'élément `IndexItem` contient des éléments qui opèrent des ajouts ou des suppressions dans le catalogue mis en cache par l'organisation acheteuse. L'exemple qui suit représente la structure d'un élément `Index` :

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

### ***IndexItem, IndexItemAdd, IndexItemDelete et IndexItemPunchout***

L'élément `IndexItem` sert de conteneur à la liste d'articles qui figurent dans un index. Il contient trois types d'éléments :

- `IndexItemAdd` – Insère un nouvel article ou met à jour un article existant dans l'index. Il contient un élément `ItemID`, un élément `ItemDetail` et un élément `IndexItemDetail`.
- `IndexItemDelete` – Supprime un article de l'index. Il contient un élément `ItemID` qui identifie l'article.
- `IndexItemPunchout` – Insère un article qui permet de déclencher un punchout avec le site Web du fournisseur. Il contient un élément `PunchoutDetail` et un élément `ItemID`. Il s'apparente à un élément `IndexItemAdd` si ce n'est qu'il n'exige pas d'informations de tarification. Les acheteurs se procurent des informations détaillées sur les articles en temps réel à partir du site Web du fournisseur.

#### **ItemID**

L'élément `ItemID` identifie de façon unique les articles d'un fournisseur. Il contient un élément `SupplierPartID` et un élément `SupplierPartAuxiliaryID` facultatif.

#### **SupplierPartAuxiliaryID**

Si l'élément `SupplierPartID` n'identifie pas l'article de façon unique, le fournisseur doit faire appel à l'élément `SupplierPartAuxiliaryID` pour spécifier une clé « auxiliaire » qui identifie la pièce de façon unique lorsqu'elle est associée aux éléments `SupplierID` et `SupplierPartID`. Un fournisseur peut utiliser le même élément `SupplierPartID` pour un article, par exemple, mais proposer un prix différent pour les unités « EA » et « BOX ». Dans ce cas, un élément `SupplierPartAuxiliaryID` possible pour les deux articles pourrait être « EA » et « BOX ».

L'élément `SupplierPartAuxiliaryID` peut également faire office de cookie fournisseur, pour permettre à ce dernier de faire référence à une configuration complexe ou à des données très spécifiques. Il peut renfermer toutes les données grâce auxquelles le fournisseur a la possibilité de reconstruire l'article en question dans son système informatique (un panier ou un cookie de données qui n'ont de signification que pour le fournisseur). Pour plus d'informations, consultez la section « Cookies d'acheteur et de fournisseur », page 40.

#### **ItemDetail**

L'élément `ItemDetail` fournit des informations détaillées sur un article, ou toutes les données auxquelles un utilisateur peut souhaiter accéder en plus des informations de base représentées dans l'élément `ItemID`. Il doit contenir un élément `UnitPrice`, un élément `UnitOfMeasure`, un ou plusieurs éléments `Description` et un élément `Classification`. Il peut éventuellement renfermer un élément `ManufacturerPartID`, un élément `ManufacturerName`, un élément `URL` et plusieurs éléments `Extrinsic`. Pour plus d'informations, consultez la section « `ItemDetail` », page 85.

Dans le contexte d'un élément `IndexItemAdd`, les éléments `Extrinsic` complètent les informations relatives à un article spécifique. Il est inutile de transmettre ces compléments au fournisseur dans un document `OrderRequest`, car le fournisseur peut retrouver ces mêmes informations au moyen de l'élément `ItemID` unique.

### IndexItemDetail

L'élément `IndexItemDetail` contient des éléments propres aux index qui, tels les éléments `LeadTime`, `ExpirationDate`, `EffectiveDate`, `SearchGroupData` ou `TerritoryAvailable`, définissent d'autres aspects d'un article.

### PunchoutDetail

L'élément `PunchoutDetail` s'apparente à l'élément `ItemDetail`, si ce n'est qu'il exige seulement un ou plusieurs éléments `Description` et un élément `Classification`. Il peut aussi renfermer des éléments `URL`, `ManufacturerName`, `ManufacturerPartID`, `ExpirationDate`, `EffectiveDate`, `SearchGroupData`, `TerritoryAvailable` et `Extrinsic`. Il ne fournit pas d'informations sur les tarifs, les délais d'exécution ni les unités de mesure.

## Contract

Un élément `Contract` représente un contrat passé entre un fournisseur et un acheteur et portant sur des biens ou des services répertoriés dans l'index du fournisseur. Il permet au fournisseur de « recouvrir » certains attributs d'articles (tels le prix) de l'index par des valeurs négociées avec l'acheteur. Il donne en outre la possibilité aux fournisseurs et aux acheteurs de segmenter ces couches à partir d'une « clé de segment » prédéfinie, prenant une signification particulière au sein d'une organisation (nom d'une division ou d'un centre de coût, par exemple).

L'élément `Contract` possède les attributs suivants :

|                       |   |
|-----------------------|---|
| <b>effectiveDate</b>  | Date et heure d'entrée en vigueur du contrat, au format ISO 8601. |
| <b>expirationDate</b> | Date et heure d'expiration du contrat, au format ISO 8601.        |

L'élément `Contract` contient un ou plusieurs éléments `ItemSegment`, comme dans l'exemple :

```
<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>
    </ContractItem>
  </ItemSegment>
</Contract>
```

```

        <UnitPrice>
            <Money currency=USD>40.00</Money>
        </UnitPrice>
    </ContractItem>
    ...
</ItemSegment>
</Contract>

```

### ***ItemSegment***

L'élément `ItemSegment` sert de conteneur à une liste d'éléments `ContractItem` relatifs à un « segment » donné, un segment représentant ici un partitionnement arbitraire d'articles contractuels à partir d'une clé de segment ayant fait l'objet d'un accord préalable entre le fournisseur et l'acheteur.

L'élément `ItemSegment` possède l'attribut suivant :

|                                   |   |
|-----------------------------------|---|
| <b>segmentKey</b><br>(facultatif) | Chaîne acceptée utilisée pour segmenter des prix personnalisés. |
|-----------------------------------|---|

### ***ContractItem***

Un tel élément vient se superposer à un article spécifique dans un index. Il contient un élément `ItemID` qui identifie l'article d'index de façon unique dans le système de gestion des approvisionnements à recouvrir. Il peut renfermer différents éléments `Extrinsic` indiquant la valeur par laquelle remplacer l'attribut désigné dans l'index.

## **Définitions de gestion des abonnements**

---

Les intermédiaires, tels les concentrateurs de réseau de commerce électronique, peuvent gérer les fournisseurs et les catalogues fournisseur utilisés par les systèmes de gestion des approvisionnements des organisations acheteuses. Ces intermédiaires sont en mesure d'assurer des liaisons directes entre les systèmes de gestion des approvisionnements et les systèmes fournisseurs. Cette section définit les éléments qui permettent de gérer les données fournisseur et les contenus des catalogues. Ces définitions reposent sur plusieurs des définitions antérieures indiquées pour les requêtes/réponses cXML, les messages asynchrones et les définitions de catalogue.

## Données fournisseur

Les définitions applicables à la gestion des données fournisseur sont principalement constituées par les éléments `SupplierListRequest`, `SupplierListResponse`, `SupplierDataRequest`, `SupplierDataResponse` et `SupplierChangeMessage`. Ces éléments sont décrits ci-après à l'aide d'exemples dans lesquels Ariba Network fait office d'intermédiaire.

### *SupplierListRequest*

L'élément `SupplierListRequest` demande la liste des fournisseurs avec lesquels l'acheteur a établi des relations commerciales.

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

### *SupplierListResponse*

L'élément `SupplierListResponse` dresse la liste des fournisseurs avec lesquels l'acheteur a établi des relations commerciales.

```
<Response>
  <Status code="200" text="OK"/>
  <SupplierListResponse>
    <Supplier corporateURL=http://www.workchairs.com
      storeFrontURL="http://www.workchairs.com">
      <Name xml:lang="en-US">Workchairs, Inc.</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*

L'élément `SupplierDataRequest` demande des informations sur un fournisseur.

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

## SupplierDataResponse

L'élément SupplierDataResponse fournit des informations sur un fournisseur.

```

<Response>
  <Status code="200" text="OK"/>
  <SupplierDataResponse>
    <Supplier corporateURL=http://www.workchairs.com
      storeFrontURL="http://www.workchairs.com">
      <Name xml:lang="en-US">Workchairs, Inc.</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>
    </SupplierDataResponse>
  </Response>

```



```

    </SupplierLocation>
  </Supplier>
</SupplierDataResponse>
</Response>

```

### **SupplierChangeMessage**

Cet élément signale les modifications apportées aux données d'un fournisseur.

```

<Message>
  <SupplierChangeMessage type="new">
    <Supplier corporateURL=http://www.workchairs.com
      storeFrontURL="http://www.workchairs.com">
      <Name xml:lang="en-US">Workchairs, Inc.</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>
</SupplierChangeMessage>
</Message>

```

## Abonnements aux catalogues

Les définitions applicables à la gestion des abonnements aux catalogues sont décrites ci-après. Dans les exemples donnés, Ariba Network fait office d'intermédiaire.

### *Subscription*

Cet élément renferme les métadonnées relatives à un abonnement unique à un catalogue. Il contient les sous-éléments suivants :

- InternalID – ID unique propre à l'intermédiaire
- Name – nom de l'abonnement
- ChangeTime – date et heure de la dernière modification de l'abonnement
- SupplierID – ID du fournisseur du catalogue
- Format – format du catalogue
- Description – description du catalogue

```

<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*

Cet élément demande la liste des catalogues auxquels l'acheteur est actuellement abonné.

```

<Request>
  <SubscriptionListRequest/>
</Request>

```

### ***SubscriptionListResponse***

Cet élément dresse la liste des catalogues auxquels l'acheteur est actuellement abonné.

```
<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***

Cet élément demande le contenu d'un catalogue auquel l'acheteur est abonné.  
La demande précise les éléments InternalID et SupplierID associés au catalogue.

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

### ***SubscriptionContentResponse***

Cet élément répertorie le contenu d'un catalogue. Le catalogue peut être au format CIF (Catalog Interchange Format) ou cXML. Dans le premier cas, il est codé en base64 et inséré dans un élément CIFContent. Dans le deuxième cas, les éléments Index et Contract sont directement mentionnés.

```

<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***

Cet élément signale au système de gestion des approvisionnements d'une organisation acheteuse que l'un des catalogues auxquels elle est abonnée a été modifié.

```

<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>

```

L'élément SubscriptionChangeMessage possède l'attribut suivant :

|             |  |
|-------------|--|
| <b>type</b> | Type de la modification : new, delete ou update. |
|-------------|--|

## Définitions de récupération des messages

Certaines organisations acheteuses ne disposent pas de points d'entrée HTTP qui leur permettent de recevoir des messages cXML provenant de sites extérieurs à leur pare-feu d'entreprise. La spécification cXML gère ces environnements.

Cette section répertorie les définitions qui permettent aux systèmes source de placer des messages en file d'attente lorsque des cibles ne sont pas en mesure d'accepter directement des transactions HTTP. Les cibles viennent récupérer les messages à leur convenance.

### GetPendingRequest

Cet élément récupère un ensemble de messages en attente pour le demandeur. L'élément `MessageType` et les attributs `lastReceivedTimestamp` et `maxMessages` contrôlent le type et le nombre des messages récupérés.

|  |   |
|--|---|
| <b>lastReceivedTimestamp</b><br>(facultatif) | Horodatage du dernier message reçu.   |
| <b>maxMessages</b><br>(facultatif)           | Nombre maximal de messages par réponse que le demandeur est en mesure de traiter. |

Dès réception de la requête, le récepteur renvoie les messages les plus anciens qui répondent aux types indiqués et dont la date est égale ou ultérieure à l'horodatage spécifié. Si plusieurs messages répondent à ce critère, différents messages peuvent être renvoyés, en fonction de l'attribut `maxMessages`. Le système de gestion des files d'attente ignore tous les messages existants qui répondent au type indiqué mais dont la date est ultérieure à l'horodatage spécifié.

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

## GetPendingResponse

Cet élément contient un ou plusieurs messages mis en attente à l'intention du demandeur.

```
<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>
```

---

# Annexe B

## Nouveautés de la version 1.1 de cXML

La version 1.1 du langage cXML a fait l'objet d'un certain nombre de nouveautés que l'on peut ranger dans les catégories suivantes :

- Modifications générales du langage cXML
- Modifications apportées aux éléments Extrinsic
- Améliorations apportées aux transactions Punchout
- Nouvelles fonctionnalités des commandes d'achat
- Nouvelle transaction : statut des commandes d'achat

Pour une présentation plus approfondie des éléments ou attributs mentionnés ici, consultez l'Annexe A, « Spécification du langage cXML ».

### Modifications générales du langage cXML

---

Les modifications ci-après s'appliquent au langage cXML dans son ensemble. Ces améliorations simplifient l'utilisation du format cXML au niveau international et résolvent les questions de compatibilité entre les versions de cXML. Une nouvelle transaction Profile permet en outre aux clients cXML d'identifier les possibilités des serveurs cXML.

## Amélioration de la prise en charge multilingue

Pour améliorer la cohérence et la prise en charge multilingue, les éléments cXML, Status et ManufacturerName possèdent désormais un attribut `xml:lang` facultatif.

Pour plus d'informations :

« Status », page 57.

Exemple :

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

Cet attribut spécifie la langue que les clients cXML doivent employer dans les réponses, et dans laquelle les sites Web compatibles punchout doivent communiquer les informations aux utilisateurs.

## Centralisation des DTD

Il n'existait auparavant aucun emplacement central officiel pour les DTD cXML, ce qui interdisait leur récupération automatique par les programmes d'analyse cXML. Désormais, des DTD correspondant aux différentes versions du langage cXML sont accessibles à des adresses bien déterminées sur le site [xml.org](http://xml.org).

Pour vous procurer des DTD cXML, rendez-vous sur le site :

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

Pour plus d'informations :

« Validation par rapport aux DTD », page 7.

où `<version>` représente le numéro de version cXML complet : 1.1.007, par exemple.

Pour optimiser les performances, les clients cXML doivent éviter de récupérer des DTD chaque fois qu'ils analysent des documents cXML. Il est préférable qu'ils les enregistrent en mémoire cache locale. Dès lors qu'une URL a été créée sous [//xml.cxml.org/schemas/cXML](http://xml.cxml.org/schemas/cXML), il n'y a pas lieu d'en modifier le contenu. Les DTD ne sont jamais modifiées sur place ; de nouvelles branches sont en effet ajoutées au fur et à mesure des besoins.

## Nouvelle transaction Profile

Une nouvelle transaction Profile communique des informations de base sur les serveurs cXML. Cette transaction se compose de deux nouveaux documents, ProfileRequest et ProfileResponse. Elle permet d'identifier les caractéristiques du serveur, et notamment la version cXML prise en charge, les transactions gérées, et les options possibles pour ces transactions.



Pour plus  
d'informations :

« Transaction  
Profile », page 67

**Remarque :** Tous les serveurs cXML 1.1 **doivent** gérer cette transaction.

La transaction Profile permet aussi aux clients d'interroger les serveurs via une commande « ping » pour vérifier qu'ils sont disponibles.

### **ProfileRequest**

Le document ProfileRequest est vide de tout contenu. Il est simplement acheminé jusqu'au serveur cXML spécifié au moyen des références Header.

```
<cXML version="1.1.007" payloadID="9949494"
  xml:lang="en-US" timestamp="2000-03-12T18:39:09-08:00">
  <Header>
    Informations de routage, d'identification et d'authentification.
  </Header>
  <ProfileRequest />
</cXML>
```

Le serveur répond par un document ProfileResponse, décrit ci-après.

### **ProfileResponse**

Le document ProfileResponse répertorie les transactions prises en charge par le serveur cXML, leur emplacement, ainsi que les éventuelles options nommées, représentées par une valeur de type chaîne.

```
<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>
```

## **Nouveaux codes de statut**

Pour plus  
d'informations :

« Status », page 57

La spécification cXML 1.1 comprend de nouveaux codes de statut de transaction, pour davantage de précision dans les échanges client/serveur. Les descriptions des codes de statut HTTP et cXML existants ont en outre été améliorées.

## Nouvel attribut type pour les membres des espaces marchands

L'élément `Credential` (références d'identification) possède un nouvel attribut `type`, qui indique si l'expéditeur ou le destinataire est membre d'un espace marchand. Il peut exister plusieurs espaces marchands, et chacun d'eux peut présenter des exigences différentes en matière de références d'identification.

Pour plus d'informations :

« `Credential` », page 55

La seule valeur possible pour le nouvel attribut `type` est `marketplace`. Cet attribut vous permet de différencier les références d'identification des membres d'espaces marchands de celles d'organisations acheteuses ou de fournisseurs traditionnels. Les éléments `Credential` exempts d'attribut `type` identifient les sociétés qui ne sont associées à aucun espace marchand.

Les requêtes à destination ou en provenance d'un espace marchand doivent identifier à la fois l'espace marchand et la société membre dans les éléments `To` ou `From Credential`.

## Modifications apportées aux éléments Extrinsic

La version cXML 1.1 introduit des éléments et des attributs supplémentaires pour les données précédemment contenues dans les éléments `Extrinsic`. Ces nouveautés font passer dans la spécification de base des informations qui étaient jusqu'alors envoyées à l'aide de l'élément `Extrinsic`.

La version cXML 1.1 assure en outre la prise en charge de l'élément `Extrinsic` au niveau de l'en-tête.

## Nouvel élément Contact

Les éléments `OrderRequestHeader`, `PunchOutSetupRequest` et `ItemOut` peuvent désormais renfermer des éléments `Contact` facultatifs, qui indiquent la personne ou le groupe à contacter pour obtenir des informations supplémentaires.

```
<ItemOut quantity="2">
  <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>
    <UnitOfMeasure>EA</UnitOfMeasure>
    <Classification domain="UNSPSC">12345</Classification>
  </ItemDetail>
</ItemOut>
```

Pour plus d'informations :

« `Contact` », page 72

```

<Contact role="customerService">
  <Address>
    <Name xml:lang="en-US">Joe Bob Emmet</Name>
    <Email>joebob@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>
</Contact>
</ItemOut>

```

L'attribut `role` précise la fonction ou le rôle de la personne qui fait office de contact. Les rôles possibles sont `endUser`, `administrator`, `purchasingAgent`, `technicalSupport`, `customerService` et `sales`.

## Prise en charge de l'attribut `requisitionID`

L'attribut `requisitionID` est désormais parfaitement pris en charge. Il s'agit d'un attribut facultatif qui peut accompagner les éléments `OrderRequest` et `ItemOut` et qui identifie la demande d'achat émise par un acheteur pour un poste donné.

Pour plus  
d'informations :

« `OrderRequestHeader` »,  
page 69

```

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

```

N'insérez pas l'attribut `requisitionID` au niveau de l'élément `ItemOut` si vous l'utilisez au niveau de l'élément `OrderRequestHeader`.

## Récapitulatif des informations déplacées dans les éléments Extrinsic

Le tableau ci-dessous répertorie les modifications apportées aux éléments Extrinsic les plus fréquents.

| Ancien élément Extrinsic | Élément ou attribut cXML existant ou nouveau |
|--------------------------|--|
| Requested Ship Date      | Attribut ItemOut requestedDeliveryDate       |
| ship complete            | Nouvel attribut shipComplete                 |
| ReqNumber                | Attribut requisitionID existant              |
| Requisition #            | Attribut requisitionID existant              |
| Name                     | Nouvel élément Contact                       |
| Phone                    | Nouvel élément Contact                       |
| E-mailAddress            | Nouvel élément Contact                       |
| Buyer Name               | Nouvel élément Contact                       |
| Buyer Phone              | Nouvel élément Contact                       |
| OriginalRequester        | Nouvel élément Contact                       |
| Requester Phone Number   | Nouvel élément Contact                       |
| ETA                      | Attribut requestedDeliveryDate existant      |

### Éléments Extrinsic de niveau en-tête

Jusqu'à présent, les éléments Extrinsic des commandes d'achat pouvaient exclusivement figurer au niveau du poste d'un produit. Ils peuvent désormais apparaître à n'importe quel emplacement du document OrderRequest.

Recourez à cette nouvelle option pour les données Extrinsic qui s'appliquent à la commande d'achat dans son intégralité.

Des éléments Extrinsic de même nom ne doivent pas apparaître à la fois au niveau de l'en-tête et au niveau des articles dans un document OrderRequest.

## Améliorations apportées aux transactions Punchout

Les transactions Punchout ont été améliorées dans l'optique d'une meilleure prise en charge des achats au niveau rayon et au niveau produit. Désormais, le format cXML gère aussi les sessions punchout annulées.

## Optimisation du document PunchOutSetupRequest

Le document PunchOutSetupRequest envoyé par les acheteurs a été modifié pour améliorer la souplesse des transactions punchout. L'URL spécifiée dans le document PunchOutSetupRequest a été supprimée ; les serveurs cXML ignoreront cet élément à l'avenir.

La nouvelle méthodologie utilise l'identité (tirée de l'élément Credential) du fournisseur. Les concentrateurs de réseau de commerce électronique reçoivent le document PunchOutSetupRequest, lisent l'ID du fournisseur, retrouvent l'URL du site Web punchout à partir des informations de compte du fournisseur et envoient le document PunchOutSetupRequest à cette URL. C'est le concentrateur, et non pas l'acheteur, qui spécifie l'URL du site Web punchout, d'où une plus grande souplesse.

Les concentrateurs de réseau de commerce électronique permettent aux fournisseurs de stocker les URL de leurs sites Web punchout.

## Élément SelectedItem

Grâce à l'amélioration apportée au document PunchOutSetupRequest, le format cXML assure aussi, désormais, une meilleure prise en charge des fonctions punchout au niveau magasin, rayon et produit. Un nouvel élément SelectedItem facultatif ajouté à ce document permet aux fournisseurs de spécifier un punchout pour un magasin entier ou pour un sous-ensemble d'offres de produits. Les applications de gestion des approvisionnements peuvent inclure l'élément SelectedItem dans des documents PunchOutSetupRequest et les sites punchout peuvent y faire appel pour déterminer les produits à proposer aux utilisateurs. En l'absence d'élément SelectedItem, les fournisseurs doivent présenter la totalité de leurs offres de produits (niveau magasin).

Un élément SelectedItem contient un élément ItemOut, qui renferme lui-même un élément ItemID, comme dans l'exemple :

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

Pour le contenu de l'élément SelectedItem, les applications de gestion des approvisionnements utilisent l'élément ItemID (SupplierPartID et SupplierPartAuxiliaryID) du catalogue-index punchout. Aucune modification de catalogue n'est nécessaire.

Les applications de gestion des approvisionnements doivent initialement envoyer à la fois le nouvel élément SelectedItem et l'ancienne URL punchout dans le document PunchOutSetupRequest. Les concentrateurs de réseau de commerce électronique n'utilisent l'ancienne URL que pour les fournisseurs qui n'ont pas encore enregistré leurs destinations d'URL punchout.

Pour plus  
d'informations :

« SelectedItem »,  
page 80

## Document PunchOutOrderMessage vide

Le protocole cXML autorise désormais les documents PunchOutOrderMessage vides, qui permettent aux utilisateurs d'interrompre des sessions d'achat punchout sans sélectionner aucun article. Jusqu'alors, le document PunchOutOrderMessage renvoyé devait contenir un article au minimum.

Les fournisseurs peuvent définir un bouton « Annuler » qui génère un document PunchOutOrderMessage vide. Le site punchout et l'application de gestion des approvisionnements sont alors en mesure de détecter l'annulation d'une session d'achat par un utilisateur, ce qui leur permet de supprimer le caddie, d'annuler les articles précédemment sélectionnés dans une demande d'achat et d'exécuter différentes tâches de maintenance.

## Nouveau champ masqué cXML-base64

Le nouveau champ masqué cXML-base64 gère les documents internationaux contenus dans le message FORM POST renvoyé par les sites punchout. Les documents cXML renfermant des symboles extérieurs au jeu « us-ascii » doivent faire appel à ce champ plutôt qu'au champ masqué cXML-urlencoded. Cette alternative présente une syntaxe pratiquement identique, si ce n'est que le document entier est codé en base64 pendant le transport au lieu d'être codé en HTML jusqu'au navigateur, ou codé sous forme d'URL jusqu'au serveur Web récepteur.

Pour plus d'informations :

« Protocole de codage de formulaire par URL », page 62

Le codage en base64 préserve le codage original des caractères d'un document cXML. Bien qu'aucun paramètre « charset » n'accompagne les informations transmises, le document décodé (après suppression du codage de transfert) peut être traité comme un support de type « application/xml ». Ce traitement permet au programme d'analyse récepteur d'honorer un éventuel attribut « encoding » spécifié dans la déclaration XML. Pour ce champ, comme pour tous les documents « application/xml », le mode de codage par défaut des caractères est UTF-8.

## Nouvelles fonctionnalités des commandes d'achat

---

Les commandes d'achat cXML ont été optimisées et gèrent un certain nombre de nouvelles fonctions.

## Nouvel attribut lineNumber

lineNumber est un nouvel attribut facultatif pour l'élément ItemOut. Il précise la position d'un article dans la commande d'achat émise par un acheteur. Il est choisi au moment où la commande est passée (et ne s'applique donc généralement pas aux sessions punchout).

Pour plus d'informations :

« ItemOut », page 75

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

Les numéros de ligne associés aux articles doivent rester constants dans les commandes modifiées, afin d'identifier la modification.

## Pièces jointes aux commandes d'achat

Les acheteurs précisent parfois certains points des commandes d'achat en leur associant des mémos, des dessins ou des télécopies. Les applications de gestion des approvisionnements peuvent désormais joindre différents types de fichiers aux commandes d'achat cXML en faisant appel à la norme MIME (Multipurpose Internet Mail Extensions).

Le document cXML contient seulement des références à des éléments MIME externes envoyés à l'intérieur d'une enveloppe MIME à plusieurs éléments (avec le document cXML, dans un message électronique ou par télécopie).

Un nouvel élément Attachment contient les références aux pièces jointes :

Pour plus d'informations :

« Transmission des pièces jointes », page 53

```
<Comments>
  <Attachment><URL>cid: uniqueCID@buyer.com</Attachment>
  Veuillez consulter l'image en pièce jointe pour avoir un aperçu du résultat final
</Comments>
```

Les concentrateurs de réseau de commerce électronique reçoivent les pièces jointes et peuvent alors les envoyer au fournisseur ou les stocker en vue d'une récupération en ligne.

Pour plus d'informations sur la norme MIME, rendez-vous sur les sites Web suivants :

[www.hunnysoft.com/mime](http://www.hunnysoft.com/mime)  
[www.rad.com/networks/1995/mime/mime.htm](http://www.rad.com/networks/1995/mime/mime.htm)

## Nouvel attribut shipComplete

shipComplete est un nouvel attribut facultatif pour l'élément OrderRequestHeader qui demande aux fournisseurs d'exécuter une commande seulement lorsque tous les articles sont disponibles. Cet attribut empêche les livraisons partielles.

Pour plus  
d'informations :

« OrderRequestHeader »,  
page 69

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

## Nouvel élément ShortName

ShortName est un nouvel élément facultatif inséré dans les éléments Description des articles.

```
<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>
```

Pour plus  
d'informations :

« ItemDetail »,  
page 85

ShortName est un nom succinct (30 caractères recommandés, 50 caractères au maximum) pour l'article, qui décrit les listes de produits présentées aux utilisateurs. Auparavant, comme il n'existait qu'un élément Description, les descriptions longues étaient tronquées de façon aléatoire.



Dans les champs dont l'espace est limité, les applications de gestion des approvisionnements et autres clients cXML doivent afficher l'élément ShortName au lieu du texte Description tronqué. En l'absence de valeur pour l'élément ShortName, les clients cXML peuvent continuer de tronquer le texte Description.

Les créateurs de catalogues ne doivent pas se servir de l'élément ShortName comme d'un double des informations contenues dans l'élément Description. Ils doivent faire appel à l'élément ShortName pour désigner le produit, et à l'élément Description pour le décrire de façon plus détaillée.

Le format de catalogue CIF 3.0 gère aussi l'élément ShortName. Le nom du champ CIF est Short Name.

## Nouvelle transaction : statut des commandes d'achat

La version cXML 1.1 comprend une nouvelle transaction qui permet d'envoyer le statut des commandes d'achat aux concentrateurs de réseau de commerce électronique.

Cette transaction repose sur le nouvel élément OrderReference, qui établit le lien entre la mise à jour du statut et le dernier document OrderRequest reçu de l'acheteur.

### Nouvel élément OrderReference

Le nouvel élément OrderReference associe une mise à jour de statut à un document OrderRequest spécifique. Il répète les attributs obligatoires des éléments OrderRequestHeader et les éléments cXML du document OrderRequest, auxquels il ajoute un identificateur facultatif généré par le fournisseur.

Pour plus  
d'informations :

« DocumentReference »,  
page 86

```
<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"
/>
```

La nouvelle transaction StatusUpdateRequest utilise cet élément.

## Nouvelle transaction StatusUpdateRequest

Les entreprises partenaires pour le traitement des commandes (fournisseurs de services de télécopie ou de services EDI) envoient la nouvelle transaction StatusUpdateRequest aux concentrateurs de réseau de commerce électronique pour définir le statut de la commande d'achat. L'indicateur de statut de commande au niveau du concentrateur, visible aussi bien aux acheteurs qu'aux fournisseurs, est alors modifié. Les fournisseurs peuvent aussi envoyer cette transaction pour permettre aux acheteurs de connaître le statut de traitement du document au sein de leur organisation.

Pour plus d'informations :

« StatusUpdateRequest », page 87

**Remarque :** Cette transaction renseigne les partenaires en présence sur les changements de statut de livraison et de traitement des documents de *commande d'achat*, et non sur le statut de l'expédition des articles proprement dits.

Une modification mérite plus particulièrement d'être soulignée : lorsqu'un concentrateur intermédiaire parvient à retransmettre un document OrderRequest, il peut faire part de ce succès à l'expéditeur original ou à un concentrateur situé en amont. Le passage par différentes files d'attente et procédures au niveau du fournisseur ou du concentrateur peut également revêtir une importance particulière pour l'acheteur.

## Nouvel élément Followup

L'élément Followup est incorporé à l'élément OrderRequestHeader et indique l'URL à laquelle les documents StatusUpdateRequest futurs devront être adressés. Cet emplacement constitue l'emplacement cible pour tous les documents ultérieurs faisant référence au document OrderRequest en cours.

Pour plus d'informations :

« Followup », page 74

---

# Index

## A

Accounting, élément 77  
acheteur, cookie 29, 40  
Attachment, élément 73

## B

BillTo, élément 72  
BrowserFormPost, élément 80  
BuyerCookie, élément 79, 82

## C

caractères, encodage 52  
champ masqué cXML-base64 38  
champ masqué cXML-urlencodé 38  
Charge, élément 77  
Classification, élément 22  
codage de formulaire 38  
codage de formulaire en HTML 38  
code, attribut 57  
commandes d'achat 41–45  
    pièces jointes 45  
Comments, élément 73  
Contact, élément 72  
Contract, élément 93  
cookies, acheteur et fournisseur 29, 40  
corporateURL, attribut 89  
Credential, élément 55  
cXML, élément 50  
cxml.org, site Web 8  
cXML-base64, champ masqué 64  
cXML-urlencodé, champ masqué 63

## D

date et heure, format 51  
deploymentMode, attribut 57, 62  
Description, élément 21, 85  
Distribution, élément 77  
DocumentReference, élément 86  
domain, attribut 56  
DTD (définitions de type de document) 7

## E

EDI (X.12 Electronic Data Interchange) 4  
éditeurs de fichiers XML 9  
effectiveDate, attribut 68, 93  
encodage des caractères 52  
encodage des formulaires 63  
encodage des formulaires HTML 63  
enregistrement des commandes 20  
établissement de devis 19  
expirationDate, attribut 93  
Extrinsic, élément 27, 40, 74, 80

## F

Followup, élément 74  
format de date et heure 51  
fournisseur, cookie 29, 40  
From, élément 24  
From, To et Sender, éléments 55

## G

GetPendingRequest, élément 101  
GetPendingResponse, élément 102

**H**

Header, élément 54

**I**

id, attribut 77  
Index, élément 91  
IndexItemAdd, élément 92  
IndexItemDelete, élément 92  
IndexItemDetail, élément 93  
IndexItemPunchout, élément 92  
inReplyTo, attribut 62  
interrogation des serveurs à l'aide de la transaction Profile 8  
IsoCountryCode, élément 66  
IsoLanguageCode, élément 66  
ItemDetail, élément 85, 92  
ItemID, élément 84  
ItemIn, élément 84  
ItemOut, élément 75  
ItemSegment, élément 94

**L**

langue, au niveau de l'en-tête cXML 31  
lastReceivedTimestamp, attribut 101  
lineNumber, attribut 75, 84

**M**

maxMessages, attribut 101  
Message, élément 61  
MIME, pièces jointes 45, 52

**O**

operation, attribut 24, 79  
operationAllowed, attribut 83  
orderDate, attribut 71  
orderID, attribut 71, 87  
OrderMethods, élément 91  
OrderRequest, élément 69  
OrderRequestHeader, élément 69  
outils d'édition de fichiers XML 9

**P**

page de démarrage 34  
page de l'expéditeur 35  
page de lancement 31  
page du destinataire de la commande 38  
payloadID, attribut 24, 50, 87  
Payment, élément 72  
pièces jointes aux commandes d'achat 45  
Profile, transaction 8  
ProfileRequest, élément 67  
ProfileResponse, élément 67  
PunchoutDetail, élément 93  
PunchOutOrderMessage, élément 28, 81  
PunchOutOrderMessageHeader, élément 82  
PunchOutSetupRequest, document 22  
PunchOutSetupRequest, élément 78  
PunchOutSetupResponse, document 27  
PunchOutSetupResponse, élément 81

**Q**

quantity, attribut 75, 84

**R**

Request, élément 57  
requestedDeliveryDate, attribut 75  
requestName, attribut 69  
requisitionID, attribut 71, 75  
Response, élément 57  
role, attribut 73

**S**

segmentKey, attribut 94  
SelectedItem, élément 26, 80  
sélection de la langue, au niveau de l'en-tête cXML 31  
Sender, élément 24  
Sender, To et From, éléments 55  
shipComplete, attribut 71  
Shipping, élément 72  
ShipTo, élément 72  
ShortName, élément 85  
StartPage, élément 81

Status, élément 57  
StatusUpdateRequest, élément 87  
storeFrontURL, attribut 89  
Subscription, élément 98  
SubscriptionContentRequest, élément 99  
SubscriptionContentResponse, élément 100  
SubscriptionListRequest, élément 98  
SubscriptionListResponse, élément 99  
Supplier, élément 89  
SupplierChangeMessage, élément 97  
SupplierDataRequest, élément 95  
SupplierDataResponse, élément 96  
SupplierID, élément 21  
SupplierListRequest, élément 95  
SupplierListResponse, élément 95  
SupplierLocation, élément 90  
SupplierPartAuxiliaryID, élément  
(cookie de fournisseur) 30, 40 92  
SupplierSetup, élément 81

## T

Tax, élément 72  
timestamp, attribut 24, 50  
To, élément 24  
To, From et Sender, éléments 55  
Total, élément 71  
Transaction, élément 68  
type, attribut 56

## U

unité de mesure 30, 66  
URL SupplierSetup 26  
URL, élément 66  
URL, encodage 63  
utilitaires pour fichiers XML 9

## V

validation de documents cXML 7  
version, attribut 50

## X

XML 15  
xml:lang 31  
xmllanguageCode, élément 66