Guia do Usuário cXML

VERSÃO 1.1

ABRIL, 2000

Pelo presente, a Ariba, Inc. (Ariba) concede-lhe o direito e a licença ampla, vitalícia, irrestrita e sem custos de direitos autorais de utilizar a especificação cXML (a "Especificação"), de acordo com os direitos autorais de utilização, publicação, modificação e distribuição da Especificação. A Ariba concorda em conceder-lhe uma licença sem custos de direitos autorais, de acordo com os direitos de propriedade intelectual aplicáveis da Ariba, de implementação e utilização das tags e diretrizes de planejamento incluídas na Especificação, com o objetivo de criar programas de computador relacionados com essas diretrizes. Uma condição dessa licença será concordar em não declarar direitos de propriedade intelectual contra a Ariba e outras empresas para sua implementação da Especificação. A Ariba expressamente se reserva todos os direitos que porventura estejam descritos no material e nos tópicos dessa Especificação. A Ariba nega expressamente toda e qualquer garantia referente a essa Especificação, incluindo as garantias de que essa Especificação ou suas implementações não viola os direitos de terceiros. A Especificação é fornecida "como está", sem nenhuma garantia expressa ou implícita. Em caso de publicação, cópia ou distribuição dessa Especificação, a notificação de direitos autorais deve ser inserida; entretanto, se for modificá-la, o nome da especificação modificada não deverá incluir o termo "cXML" . Se você enviar comentários ou sugestões à Ariba e ela modificar a cXML com base em suas informações, a propriedade da versão cXML modificada será da Ariba.

As informações neste documento estão sujeitas a alterações sem aviso prévio.

Sumário

Prefácio v
Usuários e pré-requisitos v
Capítulos a serem lidos v

Capítulo 1
Introdução a cXML 1
Recursos de cXML 1
Catálogos 2
Punchout 3
Ordens de compra 4
Tipos de aplicativo que usam cXML 5
Aplicativos de compra 5
Plataformas de rede de comércio 5
Catálogos punchout 6
Sistemas de pedido e recebimento 6
Estratégia de entrega de conteúdo 6
Validação comparando com DTDs 7
Obtendo DTDs de cXML 8
Efetuando a validação 8
Transação de perfil 8
Utilitários XML 9

Capítulo 2
Implementando um site punchout 11
Exigências de punchout 11
Organizações de compra 11
Fornecedores 13
Seqüência de eventos punchout 15
Etapas 1 & 2: Solicitação de punchout 15
Etapa 3: Seleção de produtos 17
Etapa 4: Registro de saída 18
Etapa 5: Transmissão de ordem de compra 19
Documentos punchout 20
Catálogo de índice punchout 20
PunchOutSetupRequest 22
PunchOutSetupResponse 26
PunchOutOrderMessage 27
Modificações nas páginas da Web 29
Página de início 29
Página inicial 33
Página do remetente 34
Página do receptor do pedido 37
Sugestões de sites punchout da Web 37
Implementação de diretrizes 37
Cookies do comprador e do fornecedor 38
Personalização 39

Capítulo 3
Recebendo ordens de compra cXML 41
Processo de ordem de compra 41
Recebendo ordens de compra 42
OrderRequest 42
OrderResponse 45
Aceitando anexos de ordem 45

Apêndice A
Especificação da linguagem cXML 47
Especificação de protocolo 48
Modelo de solicitação/resposta 48
Convenções XML 50
Envelope cXML 50
Camadas encobertas 52
Header 54
Request 57
Response 57
Modelo unidirecional (assíncrono) 60
Elementos básicos 65
Tipos de entidade 65
Elementos de base 66
Transação Perfil 67
ProfileRequest 67
ProfileResponse 67
Definições de pedido 69
OrderRequest 69
Resposta a uma OrderRequest 78
Transação punchout 78
PunchOutSetupRequest 78
PunchOutSetupResponse 81
PunchOutOrderMessage 82
Alterações de status posteriores 86
DocumentReference 86
StatusUpdateRequest 87
Definições de catálogo 88
Supplier 89
Index 91
Contract 93
Definições de gerenciamento de assinaturas 95
Dados sobre o fornecedor 95
Assinaturas de catálogos 98
Definições de recuperação de mensagens 102
GetPendingRequest 102
GetPendingResponse 103

Apêndice B
Novos recursos em cXML 1.1 105
Alterações gerais em cXML 105
Suporte aperfeiçoado a vários idiomas 105
DTDs centralizadas 106
Nova transação de perfil 106
Novos códigos de status 107
Novo tipo de atributo para membros do mercado 108
Alterações em elementos Extrinsic. 108
Novo elemento Contact 108
Atributo requisitionID suportado. 109
Resumo das informações de elementos Extrinsic que foram movidas 110
Elementos Extrinsic no nível de cabeçalho 110
Melhorias na transação punchout 111
PunchOutSetupRequest melhorado 111
Elemento SelectedItem 111
PunchOutOrderMessage vazio 112
cXML-base64: um novo campo oculto 112
Novos recursos de ordem de compra 113
Novo atributo lineNumber. 113
Anexos de ordem de compra 113
Novo atributo shipComplete 114
Novo elemento ShortName 114
Nova transação de status de ordem de compra 115
Novo elemento OrderReference 115
Nova transação StatusUpdateRequest 116
Novo elemento Followup 116

Índice 117

Prefácio

Este documento descreve como usar a cXML (commerce eXtensible Markup Language) para transmissão de dados relacionados com comércio eletrônico.

Usuários e pré-requisitos

Este documento destina-se a programadores de aplicativos cXML-. É dirigido a fornecedores que estejam modificando sites de e-commerce na Web para punchout.

A cXML é uma linguagem aberta e versátil que atende aos requisitos das seguintes transações:

- Catálogos de produtos eletrônicos
- Catálogos punchout cXML
- Aplicativos de compra
- Comunidades de compra

Os leitores devem ter conhecimento funcional dos conceitos de e-commerce e dos padrões de comunicação HTTP da Web.

Este documento não descreve como usar aplicativos de compra específicos ou hubs de rede de e-commerce.

Capítulos a serem lidos

- **Gerentes comerciais de e-commerce**- Para ter uma visão geral dos recursos cXML, consulte o Capítulo 1, "Introdução a cXML."
- **Programadores da Web**- Programadores da Web que estejam implementando sites de e-commerce devem ler todos os capítulos.
- **Administradores de site punchout**- Engenheiros da Web com experiência em sites punchout devem ler o Apêndice B, "Novos recursos em cXML 1.1."

Capítulo 1
Introdução a cXML

Este capítulo apresenta a cXML (commerce eXtensible Markup Language) usada para transações de comércio eletrônico.

Este capítulo oferece uma visão geral da linguagem cXML. Os seguintes tópicos são abordados:

- Recursos de cXML
- Tipos de aplicativo que usam cXML
- Estratégia de entrega de conteúdo
- Validação comparando com DTDs
- Transação de perfil
- Utilitários XML

Recursos de cXML

A linguagem cXML possibilita que compradores, fornecedores, incorporadores e intermediários comuniquem-se usando uma única linguagem aberta e padrão.

Os portais bem-sucedidos de comércio eletrônico business-to-business (e-commerce B2B) dependem de um protocolo flexível amplamente usado. A cXML é a solução para o fornecimento do mais amplo acesso a produtos e serviços, porque é uma linguagem bem-definida e robusta, criada especificamente para o e-commerce B2B, além de ser a melhor opção para um número significativo de compradores e fornecedores.

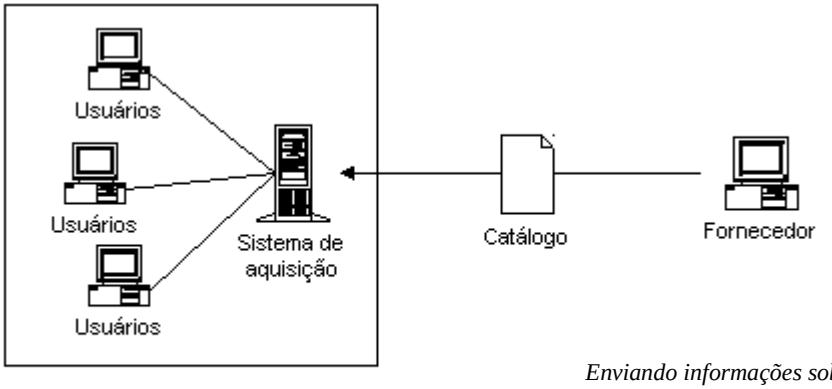
As transações cXML consistem em *documentos*, que são arquivos de texto simples, com formatos e conteúdos bem-definidos. Os tipos de documento cXML, em sua maioria, são análogos aos documentos impressos tradicionalmente usados nas empresas.

Nas seções subseqüentes são descritos os principais tipos de documento cXML.

Catálogos

Catálogos são arquivos que transmitem informações sobre produtos e serviços a organizações de compra. Os catálogos descrevem os produtos e serviços oferecidos e os respectivos preços e são o principal canal de comunicação entre você e seus clientes.

Eles são criados para que as organizações que usam aplicativos de compra possam ver e adquirir os produtos e serviços oferecidos por sua empresa. Esses aplicativos analisam e armazenam os catálogos em seus bancos de dados. Depois que os catálogos são aprovados pelas organizações de compra, seu conteúdo torna-se disponível aos usuários, que podem escolher os itens desejados e acrescentá-los às requisições de compra.



Enviando informações sobre produtos e serviços a

É possível criar catálogos para quaisquer produtos ou serviços, independentemente da forma como são avaliados e entregues e de como seus preços são determinados.

Para cada item em um catálogo, há informações básicas e opcionais que possibilitam uma apresentação mais aprimorada, como, por exemplo, descrições em vários idiomas.

Punchout

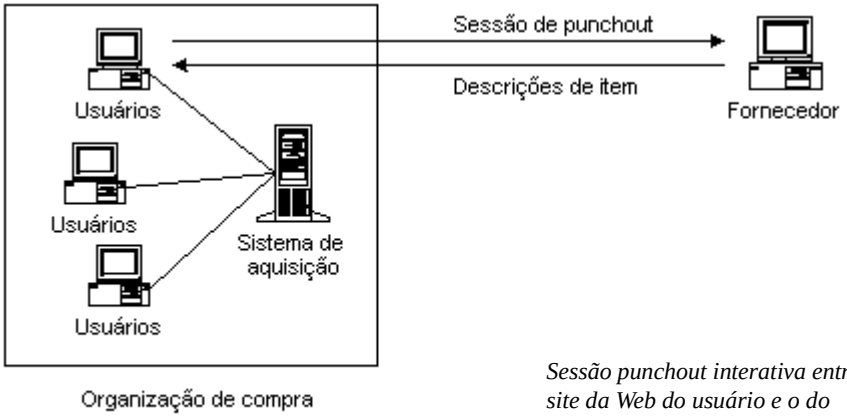
O punchout oferece a opção de arquivos de catálogo estáticos. Os sites punchout consistem em catálogos animados e interativos executados no site da Web.

Se sua empresa tiver um site de e-commerce na Web, é possível modificá-lo para que suporte punchout. Esse tipo de site comunica-se com os sistemas de compra na Internet usando a linguagem cXML.

Para obter mais informações, consulte o:

Capítulo 2, "Implementando um site punchout."

Nos sites punchout, os aplicativos de compra exibem um botão em vez de detalhes sobre o produto ou preço. Quando os usuários clicam nesse botão, o browser exibe páginas do site da Web local. Dependendo de como as páginas foram implementadas, os usuários podem pesquisar opções de produto, especificar configurações e selecionar o método de entrega. No momento em que os usuários forem selecionar os itens desejados, deverão clicar em um botão que enviará as informações sobre o pedido ao aplicativo de compra. Os produtos totalmente configurados e respectivos preços serão exibidos nas requisições de compra dos usuários.



Sessão punchout interativa entre o site da Web do usuário e o do

O site da Web pode propor um contrato de produtos e preços previamente combinados.

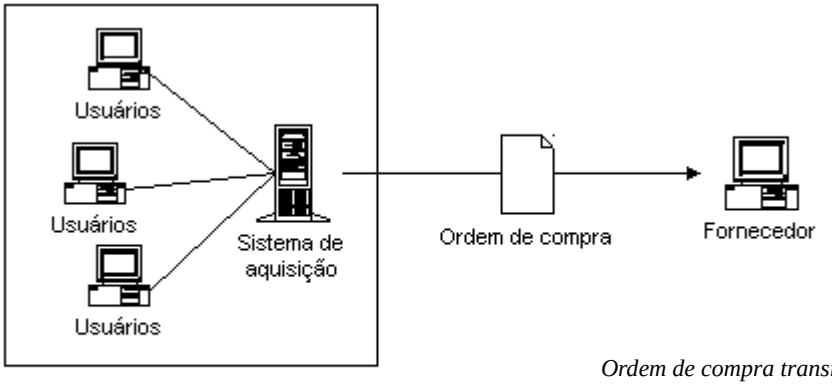
Ordens de compra

As organizações de compra enviam ordens de compra aos fornecedores para solicitar o preenchimento de um contrato.

Para obter mais informações, consulte o:

Capítulo 3, "Recebendo ordens de compra cXML."

Essas ordens de compra podem ser encaminhadas aos fornecedores por meio de uma plataforma de rede de e-commerce, como o Ariba Network.



Ordem de compra transmitida ao fornecedor

A linguagem cXML é apenas um entre vários formatos possíveis para ordens de compra. Outros formatos são correio eletrônico, fax e EDI (X.12 Electronic Data Interchange). No entanto, a cXML é o formato ideal para ordens de compra, pois sua implementação é flexível e econômica, suporta a mais ampla estrutura de dados e anexos e pode ser usada por sistemas automatizados.

Tipos de aplicativo que usam cXML

A cXML pode ser usada em quaisquer aplicativos de e-commerce. Atualmente, é usada por organizações de compra, comunidades de compra horizontais e verticais, fornecedores de produtos e fornecedores de aplicativos.

As subseções a seguir descrevem os principais tipos de aplicativo que usam cXML atualmente.

Aplicativos de compra

Aplicativos de compra, como o Ariba ORMS (Operating Resource Management System) e Ariba IBX (Internet Business eXchange), usam cXML para transações externas.

O Ariba ORMS é um aplicativo empresarial adotado por grandes organizações para ser usado pelos funcionários em uma intranet.

O Ariba IBX é um serviço de Internet que possibilita a criação de comunidades de compra que compreendem várias empresas de pequeno e médio porte.

Esses aplicativos permitem a comunidades de usuários adquirir serviços e produtos de contrato de fornecedores aprovados por seus gerentes de compra. As solicitações de compra são primeiramente aprovadas pelos gerentes nessas comunidades e as ordens de compra aprovadas são enviadas aos fornecedores por meio de vários canais possíveis, incluindo cXML, na Internet.

Plataformas de rede de comércio

Plataformas de rede de comércio, como o Ariba Network, são serviços com base na Web para fazer a conexão entre compradores e fornecedores.

Esses serviços fornecem recursos como validação de catálogo e gerenciamento de arquivos, publicação e assinatura de catálogo, encaminhamento automático de ordens de compra e histórico de ordens de compra.

A comunicação entre esses serviços da Web e os aplicativos dos compradores e fornecedores pode ser feita exclusivamente por meio da cXML na Internet.

Catálogos punchout

Conforme descrito anteriormente, os catálogos punchout são interativos e estão disponíveis nos sites dos fornecedores na Web. Esses catálogos consistem em aplicativos de servidor da Web, escritos em linguagem de programação, como ASP (Active Server Pages), JavaScript ou CGI, que gerenciam sessões punchout dos compradores.

Para obter mais informações, consulte o:

Capítulo 2, "Implementando um site punchout."

Os catálogos punchout aceitam solicitações punchout de aplicativos de compra, identificam a organização de compra e exibem os produtos e preços apropriados em formato HTML. Os usuários selecionam e configuram os itens desejados e, se necessário, fazem as opções.

Ao fim da sessão punchout, o site punchout envia as informações fornecidas pelos usuários, em formato cXML, aos aplicativos de compra.

Sistemas de pedido e recebimento

Os sistemas de pedido e recebimento consistem em aplicativos existentes nos sites dos fornecedores que aceitam e processam as ordens de compra enviadas pelas organizações de compra. Esses sistemas podem ser qualquer sistema automatizado, como os de gerenciamento de inventário, de preenchimento ou de processamento de pedidos.

Para obter mais informações, consulte o:

Capítulo 3, "Recebendo ordens de compra cXML."

Como a extração de informações de ordens de compra em cXML é simples, é relativamente fácil criar adaptadores que possibilitam que os sistemas de pedido e recebimento existentes as aceitem.

Estratégia de entrega de conteúdo

Os aplicativos de compra apresentam informações sobre produtos e serviços aos usuários. Os fornecedores desejam controlar a forma como seus clientes vêem os produtos e serviços pois a apresentação é fundamental no processo de venda. As organizações de compra querem tornar fáceis o acesso e a busca de conteúdo, com o objetivo de garantir um alto nível de concordância na contratação.

As organizações de compra e os fornecedores têm à disposição vários métodos de entrega dos produtos e serviços. O método a ser usado depende do acordo entre a organização de compra e o fornecedor e dos produtos e serviços envolvidos.

A tabela a seguir relaciona categorias, exemplificando produtos e serviços geralmente adquiridos e seus métodos de entrega preferidos.

Mercadoria	Características	Método de entrega
Material de escritório, Equipamentos internos	Conteúdo estático, preço estável	Catálogos estáticos
Equipamentos de laboratório, Manutenção, reparo e operações (MRO, Maintenance, Repair, and Operations), Peças eletrônicas	Exigem normalização para serem vantajosos	Punchout para um portal de mercadorias vertical
Livros, Produtos químicos	Grande número de linhas do item	Punchout para o site do fornecedor
Computadores, Equipamentos de rede, Periféricos	Várias configurações possíveis	Punchout para uma ferramenta de configuração do fornecedor
Serviços, Material impresso	O conteúdo tem muitos atributos diferentes	Punchout para um formulário eletrônico no site do fornecedor

As organizações de compra podem tanto armazenar o conteúdo de forma local na organização quanto acessá-lo remotamente na Internet, por meio de punchout. Os catálogos cXML suportam ambas as estratégias de armazenamento.

De acordo com a tabela apresentada anteriormente, o punchout oferece uma estrutura flexível, por meio da qual os fornecedores, dependendo da mercadoria e do cliente, podem gerar conteúdos personalizados. O objetivo dessa estratégia de conteúdo é permitir que compradores e fornecedores façam intercâmbio de dados do catálogo usando o método mais apropriado.

Validação comparando com DTDs

Como a cXML é uma linguagem XML, um conjunto de definições de tipo de documento (DTDs, Document Type Definitions) determina cuidadosamente seu significado. Essas DTDs consistem em arquivos de texto que descrevem a sintaxe e a ordem precisa dos elementos cXML e possibilitam aos aplicativos validar a cXML lida ou escrita por eles.

Embora recomendáveis, os aplicativos cXML não são necessários à validação de documentos cXML.

Obtendo DTDs de cXML

As DTDs para todas a versões de cXML estão disponíveis em localizações fixas, em cxml.org:

http://xml.cXML.org/schemas/cXML/<*versão*>/cXML.dtd

onde <*versão*> compreende todos os números de versão cXML, como 1.1.007.

Efetuando a validação

Seus aplicativos podem usar as DTDs para validar todos os documentos cXML enviados e recebidos. Os aplicativos de validação XML estão disponíveis na Web. O Microsoft Internet Explorer 5 dispõe de uma capacidade embutida de validação XML.

Para operar a maioria das transações robustas, valide todos os documentos cXML recebidos. Se detectar erros, informe o respectivo código de erro de forma que o remetente possa retransmitir.

Para obter o melhor desempenho, os clientes cXML não devem recuperar DTDs toda vez que analisarem documentos cXML. Em lugar disso, devem verificar a versão cXML no cabeçalho do documento e recuperar DTDs que ainda não foram armazenadas localmente.

Transação de perfil

A Transação de perfil transmite informações básicas sobre servidores cXML e é a única transação suportada por todos os servidores cXML.

Essa transação consiste em dois documentos: ProfileRequest e ProfileResponse. Juntos, esses documentos recuperam capacidades do servidor, incluindo versão cXML e transações suportadas e opções para essas transações.

Nota: Todos os servidores cXML 1.1**devem** suportar a Transação de perfil.

Os clientes podem usar a Transação de perfil para "fazer ping" dos servidores e, dessa forma, verificar se estão disponíveis.

ProfileRequest

Não há conteúdo nesse documento ProfileRequest e ele simplesmente efetua o envio ao servidor cXML.

ProfileResponse

O servidor responde com um documento chamado ProfileResponse, que relaciona as transações suportadas pelo servidor cXML, a localização dessas transações e quaisquer opções designadas por um valor de caracteres.

Utilitários XML

Os utilitários destinados à edição e à validação de arquivos XML estão disponíveis na Web e podem ser adquiridos gratuitamente ou comprados. A lista a seguir descreve alguns desses utilitários:

- **Internet Explorer 5** da Microsoft. Browser da Web que funciona com a linguagem XML e pode validar arquivos XML, comparando-os com as DTDs.

  www.microsoft.com/windows/ie/default.htm

- **XML Notepad** da Microsoft. Um editor de XML simples.

  msdn.microsoft.com/xml/notepad/intro.asp

- **XML Authority** da Extensibility. Um editor de XML DTD com base em Java e visualização hierárquica e gráfica.

  www.extensibility.com

- **XML Spy** da Icon Information Systems. Ferramenta para manutenção de DTDs e arquivos XML, com visualização de grade, fonte e browser.

  www.icon-is.com

- **XMetaL** da Softquad Software. Uma ferramenta de criação XML personalizada.

  www.softquad.com

- **CLIP** da Techno2000 USA. Ferramenta de criação XML fácil de ser usada, com edição orientada.

  www.t2000-usa.com

- **XMLwriter** da Wattle Software. Ferramenta de criação XML gráfica projetada para gerenciar projetos XML.

  www.xmlwriter.net

Além desses endereços, os seguintes sites da Web relacionam outras ferramentas XML:

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

Capítulo 2
Implementando um site punchout

O punchout possibilita aos usuários de aplicativos de compra acessar contratos de fornecedores para produtos ou serviços que residem no site da Web. Dessa forma, não há necessidade de enviar catálogos inteiros a organizações de compra. Em vez disso, são enviados apenas arquivos pequenos contendo uma descrição sobre a empresa, linhas de produtos ou produtos.

Este capítulo mostra como modificar um site da Web para suportar punchout. Os seguintes tópicos são abordados:

- Exigências de punchout
- Seqüência de eventos punchout
- Documentos punchout
- Modificações nas páginas da Web
- Sugestões de sites punchout da Web

Exigências de punchout

Antes de as organizações de compra configurarem os aplicativos de compra para punchout ou de os fornecedores implementarem os sites punchout da Web, ambos devem avaliar os benefícios e as exigências do punchout.

Organizações de compra

A configuração e o teste dos aplicativos de compra compatíveis com cXML com um fornecedor que trabalhe com punchout podem ser executados em menos de um dia.

Como as barreiras à integração técnica são pequenas no caso das organizações de compra, a decisão de usar o punchout deve ser com base nas práticas de negócios e nos tipos de mercadoria comprados. (Consulte a seção "Estratégia de entrega de conteúdo" na página 6, que apresenta uma lista de mercadorias que se adaptam bem ao punchout.)

Questões comerciais

As organizações de compra devem levar em consideração as seguintes questões:

- Os solicitantes e os aprovadores têm acesso à Internet? Se não, um acesso regulamentado à Internet seria autorizado?
- A organização de compra deseja que os fornecedores criem e mantenham o conteúdo dos catálogos (incluindo a determinação de preços)?
- Atualmente, os solicitantes adquirem produtos pela Internet? Em caso afirmativo, esses produtos requerem uma ferramenta de configuração por parte do fornecedor ou contêm atributos exclusivos que não se adaptam a um modelo de conteúdo estático?
- A organização de compra usa outros incorporadores de conteúdo para catálogos (por exemplo, Aspect, TPN Register ou Harbinger)?
- A organização de compra atualmente obtém serviços de compra (por exemplo, consultores, serviços temporários ou manutenção) pela Internet?
- A organização de compra atualmente faz seleção de fornecedores on-line?

Se a resposta a uma das perguntas acima for sim, talvez o punchout seja apropriado à organização de compra.

Questões técnicas

As organizações de compra devem atender aos seguintes requisitos técnicos:

- Acesso direto à Internet: Os usuários nas organizações de compra devem ter acesso direto à Internet. O punchout depende de sessões regulares de pesquisa na Web em que o usuário interaja com sites de fornecedores animados. Essa comunicação ocorre por meio de uma rede e infra-estrutura de Internet regular e não por meio do aplicativo de compra.
- Conexão confiável com a Internet: O acesso à Internet deve ser sempre operacional e confiável. Se os usuários não puderem adquirir produtos em decorrência de interrupções na Internet, é provável que efetuem compras que não sejam confiáveis.
- Contratos com fornecedores punchout: Os agentes de compra devem ter contratos estabelecidos com fornecedores que trabalhem com punchout. Os sites punchout da Web permitem acesso apenas a organizações de compra conhecidas e autenticadas.

Fornecedores

O termo *fornecedor*, no contexto de punchout, possui um significado mais abrangente que o tradicional. O protocolo punchout foi projetado como uma estrutura flexível capaz de transmitir dados sobre quase todo tipo de produto ou serviço de qualquer tipo de fornecedor, distribuidor, incorporador ou fabricante.

Exemplos de produtos e serviços incluem:

- Computadores comprados diretamente de fabricantes ou revendedores
- Produtos químicos e reagentes de um incorporador
- Materiais de escritório de um distribuidor
- Serviços contratados de uma agência de serviços temporários

Pode ser que você já tenha um site da Web de transações para apresentar conteúdos e receber ordens de compra. Supondo que tenha, precisa então analisar suas práticas de negócios e recursos técnicos ao decidir se deve ou não implementar o punchout.

Questões comerciais

Os fornecedores devem levar em consideração as seguintes questões:

- Atualmente, seus produtos ou serviços são vendidos pela Internet? Em caso afirmativo, você oferece conteúdos específicos aos clientes (determinação de preço do contrato) no site da Web?
- Esses produtos e serviços pertencem a uma das categorias de punchout descritas no gráfico da seção "Estratégia de entrega de conteúdo" na página 6? Recapitulando, essas categorias incluem:

  Produtos altamente configuráveis (por exemplo, computadores)
  Grande número de linhas do item (por exemplo, livros)
  Atributos de produto exclusivos (por exemplo, produtos químicos)
  Dados normalizados (por exemplo, suprimentos MRO)

- Você prefere receber ordens de compra e/ou pagamentos pelo site da Web?

Se a resposta a qualquer uma das perguntas anteriores for sim, talvez o punchout seja apropriado à sua organização.

Questões técnicas

Os fornecedores devem atender aos seguintes requisitos técnicos:

- Conexão confiável com a Internet: A infra-estrutura do servidor da Web e a conexão com a Internet devem ser extremamente confiáveis. Se os usuários não conseguirem acessar o conteúdo remoto, provavelmente procurarão outro fornecedor.
- Administradores de sites da Web competentes: O site punchout da Web e os aplicativos de suporte requerem manutenção e modificação periódicas. As necessidades dos usuários e as ofertas de produtos serão alteradas, portanto você necessita de pessoal para modificar a infra-estrutura do punchout.
- Suporte para transações básicas: Os sites punchout da Web não precisam suportar todas as funcionalidades cXML, exceto as seguintes transações:

  Transação de perfil
  PunchOutSetupRequest
  PunchOutSetupResponse
  PunchOutOrderMessage

Estimativa de trabalho

A tabela a seguir relaciona estimativas de trabalho requerido para a integração de punchout cXML, com base em estimativas dos fornecedores:

Nível da infra-estrutura preexistente	Tempo estimado para conclusão
Site de transações com infra-estrutura XML	Três semanas com a equipe de tecnologia da informação (TI) interna Três a quatro semanas com os contratantes
Site de transações sem infra-estrutura XML	Quatro semanas com a equipe de TI interna Quatro a cinco semanas com os contratantes

Compreendendo o que é XML

O primeiro passo para poder trabalhar com punchout é compreender o que é XML. XML é uma linguagem para descrição de linguagens. Os documentos cXML são criados com base nas definições de tipo de documento (DTDs, Document Type Definitions) XML. Da mesma forma que os gabaritos, as DTDs podem ser usadas para definir modelos de conteúdo em um documento cXML (por exemplo, o pedido válido e o estabelecimento de elementos) e os tipos de dados dos atributos.

Sobre XML

A linguagem XML (eXtensible Markup Language) é um padrão na transferência de dados entre aplicativos de Internet. Os documentos XML contêm dados em forma de pares de tags/valores. A XML tem estrutura semelhante à HTML (Hypertext Markup Language), mas os aplicativos de Internet podem extrair dados de documentos XML e usá-los mais facilmente que de documentos HTML simples.

À medida que os aplicativos de Internet forem difundidos, a XML será predominante.

Para implementar um site punchout da Web, é fundamental compreender como os dados XML são criados, analisados, consultados, recebidos de origens remotas e transmitidos a elas.

As ferramentas básicas para processar documentos XML são os analisadores XML. Esses analisadores são oferecidos gratuitamente pela Microsoft e outras empresas (por exemplo, um analisador XML é padrão no Microsoft Internet Explorer 5). Para obter uma lista de ferramentas XML, consulte "Utilitários XML" na página 9.

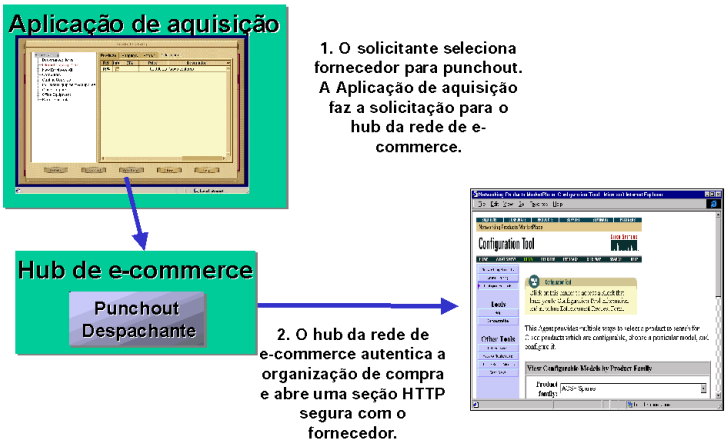
Seqüência de eventos punchout

Uma sessão punchout compreende várias etapas distintas.

Etapas 1 & 2: Solicitação de punchout

Os usuários acessam um aplicativo de compra e abrem novas requisições de compra. Para achar os itens desejados, pesquisam os catálogos locais por descrição da mercadoria, do fornecedor ou do produto. Quando um item punchout é selecionado, o aplicativo de compra abre uma nova janela de pesquisa que permite aos usuários acessar as próprias contas no site da Web.

A figura a seguir exemplifica as etapas de solicitação de punchout:



Como isso funciona? Quando um usuário clica em um item punchout, o aplicativo de compra envia um documento cXML PunchOutSetupRequest a um hub de rede de e-commerce. Atuando como um intermediário confiável, o hub aceita a solicitação, verifica a organização de compra e transfere a solicitação para o site punchout da Web.

> **Nota:** Todos os documentos cXML enviados pela Internet podem trafegar através de conexões HTTPS seguras criptografadas usando SSL (Secure Socket Layer) 3.0.

O propósito dessa solicitação é notificar ao a identidade do comprador e comunicar a operação a ser executada. Dentre as operações possíveis, incluem-se as seguintes:

- **Criar**: Inicia uma nova sessão punchout
- **Editar**: Abre novamente uma sessão punchout para edição
- **Inspecionar**: Abre novamente uma sessão punchout para inspeção (nenhuma alteração pode ser feita nos dados)

Depois de receber uma solicitação, o site da Web devolve um PunchOutSetupResponse contendo um URL que informa ao aplicativo de compra o local em que a sessão de pesquisa no site deve ser iniciada.

O aplicativo de compra abre uma nova janela de pesquisa, que exibe uma sessão registrada em uma conta no site. Essa conta pode ser específica de uma região, empresa, departamento ou usuário.

Etapa 3: Seleção de produtos

Os usuários selecionam itens do inventário usando todos os recursos e serviços oferecidos pelo site.

3. O solicitante usa o local do fornecedor para localizar e configurar os produtos.

Dependendo do produto ou do cliente, esses recursos podem incluir o seguinte:

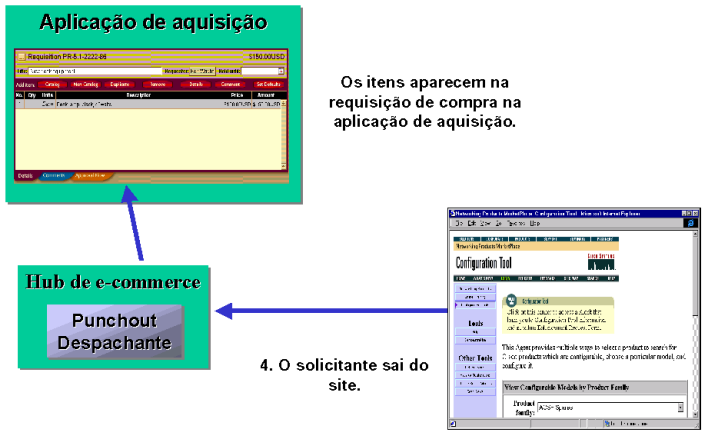
- Ferramentas de configuração para criar produtos personalizados (por exemplo, computadores, compostos orgânicos ou produtos personalizados).
- Mecanismos de pesquisa para localizar produtos desejados em catálogos grandes.
- Exibições de dados normalizados para comparação de produtos feita com base em preço, recursos ou disponibilidade (por exemplo, produtos MRO).
- Exibições de atributos exclusivos a uma determinada mercadoria (por exemplo, material impresso, produtos químicos e reagentes ou serviços).
- Determinação de preços em tempo real, inventário e verificação de disponibilidade.
- Cálculos automáticos de impostos e frete com base no local de entrega, tamanho ou quantidade de itens (não é necessário efetuar esses cálculos durante a sessão punchout).

Como isso funciona? Depois que o aplicativo de compra direciona os usuários para o site da Web, o processo de compra é igual a quando eles acessam o site diretamente. Sendo assim, nenhum dos recursos ou serviços anteriormente relacionados necessita de modificação.

Etapa 4: Registro de saída

O site calcula o custo total dos itens selecionados pelo usuário, incluindo impostos, frete e descontos específicos ao cliente. Em seguida, os usuários clicam no botão "Registrar saída" do site para enviar o conteúdo do carrinho de compras para as requisições no aplicativo de compra.

A figura a seguir exemplifica as etapas do registro de saída:



Como isso funciona? Quando os usuários clicam no botão "Registrar saída", o site da Web envia ao aplicativo de compra um PunchOutOrderMessage cXML contendo detalhes sobre o produto e os preços. É possível também enviar cookies ocultos de fornecedores, que posteriormente podem associar itens a uma sessão de compras específica.

Na verdade, você fez uma cotação para os itens solicitados — mas ainda não recebeu uma ordem de compra, portanto ainda não pode registrar o pedido.

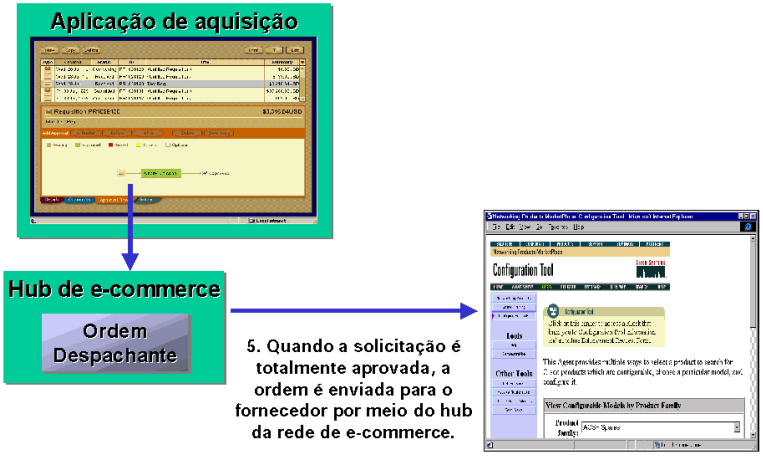
Você pode permitir que os usuários abram novamente a sessão punchout se, posteriormente, precisarem editar um dos itens em uma requisição de compra. O aplicativo de compra devolve ao site o conteúdo original do carrinho de compras e os usuários efetuam as alterações desejadas. No momento em que o registro de saída é efetuado, o site devolve os itens à requisição de compra.

O site é a fonte de informações para todos os itens punchout. Alterações na quantidade ou adição de novos itens à requisição de compra podem alterar as despesas com impostos ou transporte, o que exigirá que o site efetue um novo cálculo. Portanto, quaisquer alterações nos itens originais precisam ser feitas no site, e não na requisição de compra, daí a necessidade de reabrir a sessão punchout. Na verdade, esse novo punchout é simplesmente um PunchOutSetupRequest que tem uma "edição" como sua operação.

Etapa 5: Transmissão de ordem de compra

Depois que o conteúdo do carrinho de compras é transmitido do site para a requisição de compra do usuário, o processo de aprovação do aplicativo de compra tem início. Quando a requisição de compra é aprovada, o aplicativo de compra a converte em ordem de compra e envia de volta ao site para ser preenchida. Dados sobre o cartão de compras podem ser transmitidos juntamente com o pedido ou o pedido pode ser faturado no momento da entrega.

A figura a seguir exemplifica a transmissão da ordem de compra:



Como isso funciona? O aplicativo de compra envia todas as ordens de compra em formato cXML ao hub de e-commerce. Em seguida, o hub encaminha essas ordens a sua empresa, usando o método de encaminhamento de pedidos de sua preferência. Quando você confirma o recebimento da ordem de compra, na verdade estáregistrando o pedido.

Para os fornecedores que trabalhem com punchout, o método de encaminhamento de pedidos deve ser cXML pelos seguintes motivos:

- As ordens de compra cXML permitem que as informações sobre cookies de fornecedores embutidos sejam retransmitidas. Como os dados do cookie do fornecedor são de "qualquer" tipo, o cookie não é mapeado facilmente para outros métodos de encaminhamento de pedidos, como fax, correio eletrônico ou EDI.
- Os fornecedores que trabalham com punchout conhecem a linguagem cXML, por isso o fato de aceitar as ordens de compra cXML é apenas um esforço adicional.

As ordens de compra são discutidas em detalhes no Capítulo 3, "Recebendo ordens de compra cXML."

Documentos punchout

Há quatro tipos de documento cXML:

- Catálogo de índice punchout
- PunchOutSetupRequest
- PunchOutSetupResponse
- PunchOutOrderMessage

Catálogo de índice punchout

Os catálogos de índice punchout são arquivos que relacionam os itens punchout e apontam para o site punchout.

O seguinte exemplo apresenta um catálogo de índice punchout:

Tipo de documento cXML e URL de DTD

Seu identificador para o item punchout

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<Index>
    <SupplierID domain="DUNS">83528721</SupplierID>
    <IndexItem>
        <IndexItemPunchout>
            <ItemID>
                <SupplierPartID>5555</SupplierPartID>
            </ItemID>
             <PunchoutDetail>
                <Description xml:lang="en-US">Desk Chairs</Description>
```

URL do site punchout da Web (página de início)

```
                <Description xml:lang="fr-FR">Chaises de Bureau</Description>
                <URL>http://www.workchairs.com/punchout.asp</URL>
                <Classification domain="UNSPSC">5136030000</Classification>
            </PunchoutDetail>
        </IndexItemPunchout>
    </IndexItem>
</Index>
```

SupplierID identifica a organização do fornecedor. É possível usar qualquer domínio de identificação, mas os recomendados são os domínios DUNS (Dun & Bradstreet Universal Naming System) e NetworkID. Para obter mais informações sobre números DUNS, consulte www.dnb.com.

Description especifica o texto que o aplicativo de compra exibe nos catálogos de produtos. Essa descrição pode ser oferecida em vários idiomas e o aplicativo de compra exibirá o idioma apropriado para a localidade do usuário.

Classification especifica ao comprador o grupo de mercadorias da linha do item. Todos os seus produtos e serviços devem ser mapeados e padronizados no esquema UNSPSC. Nos catálogos de índice punchout, o elemento Classification determina a localização do item punchout nos catálogos exibidos aos usuários. Para obter uma lista de códigos UNSPSC, consulte www.unspsc.com.

Criando e publicando catálogos de índice

Crie e publique esses catálogos para os clientes em um hub de e-commerce. O gerenciador de catálogo das organizações de compra descarrega e armazena os catálogos para usá-los nos aplicativos de compra.

Os usuários visualizam o conteúdo dos catálogos de índice punchout ao lado dos itens de catálogos regulares e estáticos.

Granularidade do item punchout

É possível criar catálogos de nível loja, seção ou produtos.

- Catálogos de nível loja relacionam todos os produtos e serviços. Os usuários devem pesquisar para localizar o item desejado.
- Catálogos de nível seção apresentam produtos e serviços relacionados entre si.
- Catálogos de nível produtos relacionam apenas um produto ou serviço. Os usuários não precisam executar nenhuma pesquisa.

Para determinar a extensão dos itens punchout, considere o tipo de negócio de sua empresa, as características que compõem os produtos e serviços oferecidos e a estrutura do site punchout.

Quanto maior o número de ferramentas de pesquisa e configuração no site da Web, maior o número de itens punchout nos catálogos de índice.

PunchOutSetupRequest

Para iniciar uma sessão punchout, o usuário deve selecionar o item punchout. O aplicativo de compra gera um documento PunchOutSetupRequest e o envia a um hub de e-commerce, que o encaminha ao site punchout.

Veja a seguir um exemplo de documento 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>                                          Originador (organização de compra)
            <Credential domain="DUNS">
                <Identity>65652314</Identity>
            </Credential>
        </From>
        <To>                                            Destinatário (fornecedor)
            <Credential domain="DUNS">
                <Identity>83528721</Identity>
            </Credential>
        </To>
        <Sender>                                        Entidade de retransmissão anterior (Ariba Network, neste caso)
            <Credential domain="AribaNetworkUserId">
                <Identity>sysadmin@ariba.com</Identity>
                <SharedSecret>abracadabra</SharedSecret>
            </Credential>
            <UserAgent>Ariba ORMS 6.1</UserAgent>
        </Sender>
    </Header>
    <Request>
        <PunchOutSetupRequest operation="create">   Tipo de solicitação
            <BuyerCookie>1CX3L4843PPZO</BuyerCookie>
            <Extrinsic name="CostCenter">610</Extrinsic>
            <Extrinsic name="User">john_smith</Extrinsic>
            <BrowserFormPost>                       Destino para PunchOutOrderMessage final
                <URL>https://aribaorms:26000/punchout.asp</URL>
            </BrowserFormPost>
            <SupplierSetup>
```

```
                    <URL>http://www.workchairs.com/punchout.asp</URL>
                </SupplierSetup>
Item selecionado pelo usuário
                <SelectedItem>
                    <ItemOut quantity="1">
                        <ItemID>
                            <SupplierPartID>5555</SupplierPartID>
                        </ItemID>
                    </ItemOut>
                </SelectedItem>
            </PunchOutSetupRequest>
        </Request>
    </cXML>
```


Os atributos payloadID e timestamp na parte inicial são usados pelos clientes cXML para rastrear documentos e detectar documentos duplicados.

Os elementos From, To e Sender permitem aos sistemas de recebimento identificar e autorizar participantes. Os elementos From e To em um documento não são alterados. Entretanto, no momento em que o documento está sendo enviado a seu destino, os nós intermediários (como, por exemplo, Ariba Network) alteram o elemento Sender.

Criar, editar e inspecionar operações

O atributo operation especifica o tipo de sessão iniciada pelo comprador. É possível criar, editar ou inspecionar.

- As sessões create geram novos carrinhos de compras, que correspondem a novas requisições de compra.
- As sessões edit reabrem carrinhos de compra criados anteriormente para que o usuário possa efetuar modificações. O aplicativo de compra envia dados da linha do item como parte do PunchOutSetupRequest. O site punchout da Web pode usar esses dados para instanciar novamente o carrinho de compras criado durante a sessão original.
- As sessões inspect reabrem carrinhos de compra criados anteriormente apenas para visualização. Da mesma forma que na operação edit, o aplicativo de compra envia dados da linha do item como parte do PunchOutSetupRequest. Entretanto, depois de reinstanciar o carrinho de compras, o site punchout da Web não permite que seu conteúdo seja modificado.

O seguinte exemplo relaciona uma solicitação 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>
```

Se o usuário iniciasse a sessão edit selecionando um item do catálogo, o PunchOutSetupRequest conteria um elemento SelectedItem, como em uma sessão create.

Autenticação efetuada por um hub de e-commerce

Todos os documentos PunchOutSetupRequest são encaminhados por meio de um hub de e-commerce para autenticação e consulta do URL do site punchout. Eis as etapas:

1. O hub recebe o documento PunchOutSetupRequest enviado pelo usuário.
2. O hub compara o código do comprador (From e SharedSecret) com sua conta de e-commerce. Além disso, identifica o fornecedor solicitado (To).
3. O hub procura a chave compartilhada da conta e a insere (Shared Secret) no elemento Sender.
4. O hub localiza o URL do site punchout em sua conta e envia-lhe o documento PunchOutSetupRequest.
5. O site recebe o documento cXML e reconhece que é um documento autenticado porque contém uma chave compartilhada própria.
6. O site da Web usa as informações no elemento From para identificar o solicitante no nível da empresa (por exemplo, acme.com).
7. É possível usar os dados Contact e extrínsecos no corpo da solicitação para identificar exclusivamente o usuário (por exemplo, John Smith, do Financeiro, em acme.com).

Os documentos PunchOutSetupRequest e PunchOutSetupResponse são transmitidos pelo hub de e-commerce para autenticação. O documento PunchOutOrderMessage (retornando o conteúdo da cesta de compras ao aplicativo de compra) trafega diretamente entre o site da Web e o usuário usando HTTP-padrão ou HTTPS.

URL de configuração do fornecedor e SelectedItem

Em versões anteriores do cXML, o elemento SupplierSetup era o único meio de especificar o URL de um site punchout da Web. Da versão cXML 1.1 em diante, o hub de e-commerce passou a reconhecer o URL dos sites punchout.

Além disso, a partir da versão cXML 1.1, os aplicativos de compra passaram a usar o elemento SelectedItem para especificar se o punchout é do nível loja, seção ou produtos.

O elemento SupplierSetup foi desaprovado. Entretanto, o site punchout deve usar os dois métodos até que todos os sites punchout reconheçam o elemento SelectedItem.

Dados de contato para dados extrínsecos e identificação de usuário

O documento PunchoutSetupRequest pode conter as seguintes informações detalhadas sobre o usuário no elemento Contact, que podem ser usadas pelo site da Web para autenticar e direcionar usuários:

- Nome de usuário e função
- Endereço de correio eletrônico

Além disso, o documento PunchOutSetupRequest pode conter os seguintes dados *extrínsecos*, que podem ser usados para identificar usuários posteriormente:

- Centro de custos do usuário e subconta
- Região
- Supervisor
- Moeda-padrão

As organizações de compra configuram os aplicativos de compra para inserir dados Contact e extrínsecos. Pergunte aos clientes que tipo de dados desejam receber.

PunchOutSetupResponse

Depois de receber um documento PunchOutSetupRequest, o site da Web envia um documento PunchOutSetupResponse. O documento PunchOutSetupResponse executa duas funções:

- Indica se o PunchOutSetupRequest foi executado com sucesso.
- Oferece ao aplicativo de compra um URL redirecionado à Página inicial.

Contém um elemento <URL> que especifica o URL da Página inicial para transmiti-lo ao browser da Web do usuário e possibilitar uma sessão de pesquisa interativa. Esse URL deve conter informações suficientes sobre o estado para que seja realizada a conexão com um contexto de sessão no site, como, por exemplo, a identidade do solicitante e conteúdo do elemento BuyerCookie.

O exemplo a seguir apresenta um documento 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>
```

PunchOutOrderMessage

Depois que o usuário seleciona e configura itens no site da Web e clica no botão "Registrar saída", é enviado um documento PunchOutOrderMessage para informar ao aplicativo de compra do comprador o conteúdo da cesta de compras. A quantidade de dados que esse documento pode conter é muito maior, se comparada a outros documentos, porque ele precisa estar apto a exibir totalmente o conteúdo de qualquer cesta de compras imaginável. Esse documento não segue estritamente o paradigma Request/Response; seu uso será explicado em detalhes posteriormente.

O exemplo a seguir apresenta um documento 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 possibilita que o aplicativo de compra associe um determinado documento PunchOutOrderMessage ao documento PunchOutSetupRequest que o originou. Portanto, o site deverá retornar esse elemento sempre que ele aparecer. Não use o BuyerCookie para rastrear sessões punchout, porque ele é alterado em cada sessão, da criação à inspeção e edição.

SupplierPartAuxiliaryID atua como um cookie de fornecedor. Esse campo permite transmitir dados adicionais, como, por exemplo, número da cotação ou outro documento cXML. Esse cookie é retransmitido pelo aplicativo de compra em qualquer sessão edit ou inspect PunchOutSetupRequest subseqüente e na ordem de compra cXML resultante. É possível usar o cookie do fornecedor para associar itens em uma requisição de compra com os itens correspondentes em um carrinho de compras no site da Web.

UnitOfMeasure descreve como o produto é embalado ou entregue. Essas características devem estar de acordo com os Códigos Comuns de Unidade de Medida (Unit of Measure Common Codes) UN/CEFACT. Para obter uma lista de códigos UN/CEFACT, consulte o site www.unece.org/cefact.

Classification relaciona o código de mercadoria UNSPSC (United Nations Standard Product and Service Code) referente a cada item selecionado. Esses códigos são usados por sistemas auxiliares tanto na organização de compra quanto na organização do fornecedor para contabilidade e geração de relatórios. Para obter uma lista de códigos UNSPSC, consulte o site www.unspsc.org.

Modificações nas páginas da Web

Para receber ou enviar os três documentos punchout cXML, convém que você modifique ou crie quatro páginas no site da Web:

- Página de início
- Página inicial
- Página do remetente
- Página do receptor do pedido

Para ilustrar como essas páginas podem ser implementadas, serão usados exemplos simples de códigos ASP (Active Server Page) e o Microsoft Internet Explorer 5 XML Parser. A implementação efetiva dessas páginas variará, dependendo do ambiente de desenvolvimento do fornecedor (por exemplo, CGI, JavaScript ou WebObjects).

Página de início

A Página de início recebe do hub de e-commerce todos os documentos PunchOutSetupRequest autenticados. Essa página lê o fluxo HTTP enviado do hub e valida a solicitação cXML embutida nesse fluxo, comparando-a com a DTD cXML (no caso da ASP, usa chamadas de método ao Internet Explorer 5 XML parser).

Após a validação, a Página de início extrai elementos do documento, com o objetivo de:

1. Identificar o usuário e determinar para onde deve redirecioná-lo.
2. Compor um documento PunchOutSetupResponse e retorná-lo ao remetente.

A Página de início deve armazenar os seguintes dados para que sejam usados pela Página inicial:

- Identidade do solicitante (Sender).
- Identidade do idioma do usuário (xml:lang), para que possa oferecer conteúdo localizado.
- Tipo de solicitação (create, edit ou inspect).
- Qualquer dado extrínseco que posteriormente identifique o usuário e sua localidade.

A seguir, veja um exemplo de Página de início. Em vez de usar um analisador XML para gerar dinamicamente o documento PunchOutSetupResponse, esse código usa um gabarito XML estático no qual são inseridos dados do item da linha. **Esse código serve apenas como exemplo.**

```
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 Crie dados necessário na introdução.
%<
Randomize
randStr = Int( 100000001 * Rnd )
prologStr = genProlog( "1.0", randStr )
Response.ContentType = "text/xml"
Response.Charset = "UTF-8"
%>
```

```
<%
REM Recebe a solicitação PunchOutSetup do hub de e-commerce.
REM Anexa o ORMSURL e o buyercookie ao URL da Página inicial,
REM e reenvia a resposta ao solicitante.
REM punchoutredirect.asp?bc=2133hfefe&url="http://workchairs/com/..&redirect="
      Dim ret
      Dim punch
      Dim statusText
      Dim statusCode
      Dim cookie
      Dim url
      Dim xmlstr
      Dim fromUser
      Dim toUser
      cookie = ""
      url = ""
      xmlstr = ""
      dir = ""
      path = Request.ServerVariables("PATH_INFO")
      dir = Left(path, InstrRev(path, "/"))
      if IsEmpty(dir) then
            dir = "/"
      end if

REM Esse comando lê a solicitação cXML HTTP
      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 Formata o PunchOutSetupReponse cXML
<% 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;amp;url="<%= url%>"&amp;amp;from=<%= fromUser%>&amp;amp;to=<%=
toUser%>&amp;amp;redirect=<%= StartPage%></URL>
                </StartPage>
            </PunchOutSetupResponse>
        </Response>
</cXML>
<%end if%>
```

A Página de início deve retornar um URL StartPage exclusivo para essa sessão punchout. Além disso, esse URL deve ser válido por um período limitado. Se for desativado, ficará mais difícil para usuários não autorizados acessar a Página inicial.

Lembre-se de implementar a funcionalidade para sessões edit e inspect subseqüentes. Os usuários não podem alterar detalhes de pedidos em itens punchout (como quantidade) nos aplicativos de compra. Eles devem abrir novamente a sessão punchout com uma sessão edit. Para que os usuários beneficiem-se ao máximo, sessões inspect que ocorrerem depois que o pedido for recebido deverão exibir o seu status.

Página inicial

APágina inicial registra o solicitante em uma conta no site. É da Página inicial que os usuários começam o processo de compra. Essa página provavelmente já existe no site, por isso modifique-a para consultar o nome de usuário e a senha do documento PunchOutSetupRequest.

Permita que somente usuários autorizados acessem a Página inicial. Caso a autenticação desses usuários ocorra no momento do registro de saída, os preços ou termos confidenciais não estarão protegidos.

Se estiver usando cookies de pesquisa HTTP para rastrear as preferências do usuário, elimine-os depois de enviar o documento PunchOutOrderMessage aos compradores. A eliminação desses cookies impede que sejam oferecidos recursos privilegiados a usuários não autorizados.

Página do remetente

APágina do remetente envia o conteúdo do carrinho de compras ao usuário. Conforme descrito anteriormente, depois que os usuários enchem seus carrinhos de compras, devem clicar no botão "Registrar saída".

A seguir, veja um exemplo de implementação ASP básica desse recurso. Em vez de usar um analisador XML para gerar dinamicamente o documento PunchOutOrderMessage, esse código usa um gabarito XML estático, no qual dados sobre a linha do item são inseridos. **Esse código serve apenas como exemplo.**

Veja a seguir uma parte da página de produtos do site da Web do fornecedor:

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

A função AddBuyButton reenvia o documento PunchOutOrderMessage ao usuário.

Na seguinte lista encontra-se o arquivo (punchoutitem.inc) citado anteriormente:

```
<%
REM Essa áspide é incluída em items.asp, que especifica os parâmetros e formatos do
item
REM um documento cXML e permite que o usuário continue com o registro de saída do
item.
function CreateCXML(toUser, fromUser, buyerCookie, unitPrice, supPartId, desc)
%>
<?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?>
```

```
<!DOCTYPE cXML SYSTEM
&quot;http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd&quot;>
<cXML version=&quot;1.1.007&quot; payloadID=&quot;<%= Now &"@"&
Request.ServerVariables("LOCAL_ADDR")%>&quot; timestamp=&quot;<%= Now
%>&quot;>
    <Header>
        <From>
            <Credential domain=&quot;ariba.com&quot;>
                <Identity><%= toUser%></Identity>
            </Credential>
        </From>
        <To>
            <Credential domain=&quot;ariba.com&quot;>
                <Identity><%= fromUser%></Identity>
            </Credential>
        </To>
        <Sender>
            <Credential domain=&quot;ariba.com&quot;>
                <Identity><%= toUser%></Identity>
            </Credential>
            <UserAgent>PunchoutSite</UserAgent>
        </Sender>
    </Header>
    <Message>
        <PunchOutOrderMessage>
            <BuyerCookie><%= buyerCookie%></BuyerCookie>
            <PunchOutOrderMessageHeader
            operationAllowed=&quot;edit&quot;>
                <Total>
                    <Money currency=&quot;USD&quot;><%=
                    unitPrice%></Money>
                </Total>
            </PunchOutOrderMessageHeader>
            <ItemIn quantity=&quot;1&quot;>
                <ItemID>
                    <SupplierPartID><%= supPartId%></SupplierPartID>
                    <SupplierPartAuxiliaryID><%= supPartAuxId%>
                    </SupplierPartAuxiliaryID>
                </ItemID>
                <ItemDetail>
                    <UnitPrice>
                        <Money currency=&quot;USD&quot;><%= unitPrice%>
                        </Money>
                    </UnitPrice>
                    <Description xml:lang=&quot;en&quot;><%= desc%>
                    </Description>
                    <UnitOfMeasure>EA</UnitOfMeasure>
                    <Classification
                        domain=&quot;SupplierPartID&quot;><%= 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
%>
```

A função AddBuyButton contém o FORM POST que reenvia ao usuário o URL PunchOutOrderMessage codificado.

Formulário HTTP codificado

Para enviar um documento PunchOutOrderMessage, use o *formulário HTML codificado*, um modelo de transmissão diferente do tradicional modelo HTTP solicitação/resposta. Essa transmissão diferente facilita a integração entre o site e o aplicativo de compra e possibilita às organizações de compra receber dados XML sem a necessidade de haver um servidor da Web disponível em um firewall.

Em vez de enviar um PunchOutOrderMessage diretamente ao aplicativo de compra, o site o codifica como um campo de Formulário HTML oculto e envia ao URL especificado no elemento BrowserFormPost do PunchOutSetupRequest. O campo Formulário HTML oculto deve ser designado por cxml-urlencoded ou cxml-base64 (esses nomes não diferenciam maiúsculas de minúsculas).

Essa codificação possibilita que você crie uma página da Web de registro de saída que contenha o documento cXML. Quando os usuários clicarem no botão "Registrar saída", o site apresentará os dados (ocultos para os usuários) ao aplicativo de compra como um Formulário HTML de submissão.

Cancelando uma sessão punchout

Convém acrescentar um botão "Cancelar" nas páginas de forma que os usuários possam cancelar sessões punchout. O botão "Cancelar" envia um documento PunchOutOrderMessage vazio que informa ao aplicativo de compra que nenhum item será retornado e que os itens da requisição do punchout de saída devem ser excluídos. Esse procedimento também pode ser usado para executar qualquer manutenção necessária no site, como, por exemplo, esvaziar o carrinho de compras e fechar a sessão do usuário.

Página do receptor do pedido

APágina do receptor do pedido aceita as ordens de compra cXML enviadas pelas organizações de compra. Essa operação pode ser semelhante à da Página de início discutida anteriormente.

Para obter informações sobre o recebimento de ordens de compra, consulte o Capítulo 3, "Recebendo ordens de compra cXML."

Sugestões de sites punchout da Web

Quando estiver planejando a implementação do site punchout da Web, leve em consideração as seguintes sugestões:

Implementação de diretrizes

Siga essas diretrizes no momento em que estiver desenvolvendo o site punchout:

- Estude a especificação cXML.
- Use um analisador XML e compare os documentos com a DTD cXML.
- Use a propriedade xml:lang= para identificar o idioma dos usuários, a fim de oferecer conteúdo localizado.
- Use a credencial From para identificar organizações de compra.
- Envie um URL exclusivo e temporário à sessão que está sendo redirecionada.

- Não repita cookies de pesquisa.
- Não sobrecarregue os clientes exigindo dados que não sejam essenciais.
- Para cada linha do item, use UNUOM (United Nations Units of Measure) e UNSPSC (United Nations Standard Product and Service Codes).
- Ofereça benefícios reais a seus clientes. Exponha os produtos disponíveis e exiba o status do pedido e promoções especiais.
- O registro de saída deve ser fácil e intuitivo. O ideal é que os usuários possam clicar apenas em três botões para efetuar a compra.
- Efetue a codificação para sessões de edição e inspeção subseqüentes. Os usuários não podem alterar detalhes do pedido de itens punchout (como quantidade) em seus aplicativos de compra. Eles devem abrir novamente a sessão punchout e editá-la.
- Para que os usuários obtenham os melhores benefícios, a inspeção das sessões deve exibir o status do pedido.
- Teste o site punchout. Reserve um tempo para efetuar esse teste com os aplicativos de compra dos clientes.
- As transações punchout geram cotações e não ordens de compra. Implemente uma página cXML de compra/pedido/recebimento para a aceitação de pedidos.

Cookies do comprador e do fornecedor

Os cookies docomprador e do fornecedor habilitam tanto os compradores quanto os fornecedores a reinstanciar os próprios dados de linhas do item para o seu sistema auxiliar.

- Retorne o cookie do comprador (BuyerCookie) recebido. Não o altere.
- Use o cookie do fornecedor (SupplierPartAuxiliaryID).

O cookie do comprador, análogo ao número da requisição de compra, transmite o estado que possibilita ao sistema da organização de compra manter a conexão entre a requisição e a cesta de compras.

Da mesma forma, o cookie do fornecedor, análogo ao número de cotação, transmite o estado que possibilita ao sistema manter a conexão entre a cesta de compras e a requisição e a ordem de compra do comprador. Os aplicativos de compra transmitem o cookie do fornecedor novamente em sessões punchout edit ou inspect subseqüentes e na ordem de compra resultante. O site deve aproveitar o cookie do fornecedor para eliminar a necessidade de transmitir ao comprador dados específicos do fornecedor que podem ser visualizados nesse processo.

Personalização

O cabeçalho do documento PunchOutSetupRequest sempre identifica a organização de compra, mas a solicitação também deve conter dados Contact e Extrinsic (como, por exemplo, centro de custos do usuário, localização do usuário ou categoria de produtos) que possam ser usados para determinar o URL dinâmico para o usuário.

Embora nem todas as organizações de compra enviem esses dados extrínsecos, eles possibilitam que você personalize sua loja da Web transformando-a em muito mais que uma organização simples. Por exemplo, você poderia oferecer uma loja da Web distinta para cada centro de custos dentro da organização de compra (ou a cada categoria de produto ou usuário).

Poderia ainda armazenar e exibir as cotações anteriores do usuário. Dessa forma, possibilitaria que os usuários reutilizassem as cotações, verificassem o status dos pedidos e criassem relatórios de atividades anteriores. Para evitar problemas de segurança, armazene o histórico de cotações somente no nível do usuário.

Um fator relevante durante o planejamento são os esforços necessários para a implementação de um site punchout altamente dinâmico e personalizado. É preciso equilibrar personalização e complexidade — a implementação e a manutenção de um site complexo tomam tempo, mas pode oferecer maior valor aos usuários. A princípio, é recomendável um site punchout simples, que pode ser aperfeiçoado com o tempo.

Capítulo 3
Recebendo ordens de compra cXML

Este capítulo descreve como configurar um site da Web para receber ordens de compra no formato cXML. Além disso, descreve como enviar mensagens de status de ordens de compra a organizações de compra ou mercados.

Processo de ordem de compra

Os aplicativos de compra convertem requisições de compra aprovadas em uma ou mais ordens de compra. Uma ordem de compra consiste em uma solicitação formal de uma organização de compra a um fornecedor para que cumpra um contrato.

cXML é apenas um formato para transmissão de ordens de compra. Outros formatos comuns são o correio eletrônico, fax e intercâmbio eletrônico de dados (EDI, X.12 Electronic Data Interchange). cXML é o melhor formato para ordens de compra, por possibilitar a automatização fácil do processamento de pedidos. Além disso, a estrutura bem-definida desse formato permite que sistemas de processamento de pedidos interpretem com facilidade os elementos contidos em uma ordem de compra. Com pouca ou nenhuma intervenção humana, os dados adequados nas ordens de compra podem ser encaminhados aos departamentos de entrega, faturamento e vendas conforme necessário.

O método de encaminhamento de pedidos cXML possibilita a transmissão de cookies de fornecedores (SupplierPartAuxiliaryID) e anexos de ordens de compra.

Quando você configurar uma conta em um hub de rede de e-commerce, especifique o URL para o qual todas as ordens de compra serão enviadas. Depois de receber a ordem de compra, você deverá enviá-la ao sistema de gerenciamento de pedidos interno e preenchê-la normalmente. O site da Web também deverá retornar a confirmação de Resposta do pedido ao hub de rede de e-commerce, que informará ao comprador que você recebeu e analisou a ordem de compra.

Não há necessidade de um site punchout para receber ordens de compra cXML; punchout e recebimento de ordens cXML são dois recursos distintos. No entanto, a infra-estrutura e os aplicativos necessários para suportar o punchout são os mesmos usados para receber ordens de compra cXML.

Recebendo ordens de compra

Dois tipos de documento cXML são usados na transmissão de ordens de compra. O aplicativo de compra envia um OrderRequest e você responde, enviando um OrderResponse. Esses documentos passam pelo hub de rede de e-commerce.

OrderRequest

O documento OrderRequest é análogo a uma ordem de compra. Os exemplos seguintes mostram um OrderRequest para um item:

```
<?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&amp;quot; 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

O documento OrderResponse confirma que a ordem de compra foi recebida e analisada corretamente. A execução da ordem de compra não é obrigatória; esse documento confirma que você a recebeu e que se trata de um documento cXML válido.

```
<?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>
```


Aceitando anexos de ordem

Os compradores muitas vezes precisam tornar as ordens de compra mais claras usando memorandos, desenhos ou fax. Eles podem anexar arquivos a qualquer tipo de ordem de compra cXML com o MIME (Multipurpose Internet Mail Extensions).

A cXML contém apenas referências a seções MIME externas enviadas em um envelope MIME contínuo (com o documento cXML, enviado por correio eletrônico ou fax).

O hub de rede de e-commerce recebe os anexos e pode enviá-los ao fornecedor ou armazená-los para recuperação on-line.

Para obter mais informações sobre anexos de ordens de compra, consulte "Transmissão de anexos" na página 53.

Para obter mais informações sobre o padrão MIME, consulte os seguintes sites da Web:

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

Apêndice A
Especificação da linguagem cXML

Este apêndice descreve o protocolo e os formatos de dados da linguagem cXML (commerce eXtensible Markup Language). Contém todas as informações necessárias para a implementação de qualquer uma das transações suportadas, tanto sob a perspectiva do sistema cliente quanto do sistema servidor. As interações de protocolo e os documentos comerciais contidos nas transações são discutidos em detalhes.

Além disso, exemplos de implementações efetivas ilustram e tornam mais claro o uso da cXML.

Este apêndice contém as seguintes seções:

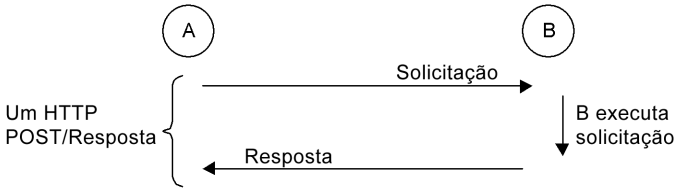
- Especificação de protocolo
- Elementos básicos
- Transação Perfil
- Definições de pedido
- Transação punchout
- Alterações de status posteriores
- Definições de catálogo
- Definições de gerenciamento de assinaturas
- Definições de recuperação de mensagens

Especificação de protocolo

Há dois modelos de comunicação para transações cXML: Solicitação/resposta e unidirecional. Esses modelos possibilitam uma implementação básica porque as operações exigidas são rigorosamente descritas. Os dois modelos são necessários pois, em algumas situações, um dos modelos não é adequado.

Modelo de solicitação/resposta

Transações de solicitação/resposta podem ser executadas somente em uma conexão HTTP. A figura a seguir apresenta as etapas em uma interação de solicitação/resposta entre as partes A e B:



Transação de solicitação/resposta A

Essa transação compreende as seguintes etapas:

1. A inicia uma conexão HTTP/1.x com B em um URL predeterminado que representa o endereço de B.
2. A usa uma operação POST para enviar o documento cXML por meio da conexão HTTP.
3. A espera uma resposta para retornar por meio da conexão HTTP.
4. B tem um servidor compatível com HTTP/1.x que despacha a solicitação HTTP ao recurso especificado pelo URL usado na etapa 1. Esse recurso pode ser qualquer localização válida conhecida pelo servidor HTTP de B. Por exemplo, um programa CGI ou uma página ASP.
5. O recurso de B identificado na etapa 4 lê o conteúdo do documento cXML e mapeia a solicitação para o manipulador apropriado a ela.
6. O manipulador de B para a solicitação cXML executa o trabalho especificado pela solicitação e formata um documento cXML como uma resposta.

7. B envia a resposta cXML para A por meio da conexão HTTP estabelecida na etapa 1.

8. A lê a resposta cXML e a retorna ao processo que iniciou a solicitação.

9. A fecha a conexão HTTP estabelecida na etapa 1.

Em seguida, esse processo é repetido em ciclos de solicitação/resposta posteriores.

Para simplificar o trabalho nas etapas descritas anteriormente, os documentos cXML são divididos em duas partes distintas:

- Cabeçalho: Contém informações de autenticação e endereçamento.
- Dados de solicitação ou resposta: Contêm uma solicitação ou resposta específica e as informações a serem transmitidas.

Os dois elementos são transportados em um elemento envelope-pai. O exemplo a seguir mostra a estrutura de um documento de solicitação cXML:

```
<cXML>
    <Header>
        Header information here…
    </Header>
    <Request>
        Request information here…
    </Request>
</cXML>
```

O exemplo a seguir mostra a estrutura de um documento de resposta cXML:

```
<cXML>
    <Response>
        Response information here…
    </Response>
</cXML>
```

A estrutura do documento de resposta não usa um elemento Header. Esse elemento não é necessário porque o documento de resposta sempre trafega na mesma conexão HTTP que o documento de solicitação.

Convenções XML

A cXML usa elementos para descrever itens discretos, que, em geral, são propriedades em documentos comerciais tradicionais. Informações com subdivisões óbvias e relações entre essas subdivisões, como endereços, também são descritas com o uso de elementos.

A cXML usa muito os atributos.

Em cXML, todos os nomes de elementos e atributos usam palavras inteiras com letras maiúsculas (sem hifens) separando as palavras. Os nomes dos elementos começam com letra maiúscula; os nomes dos atributos, com letra minúscula. Por exemplo:

Elementos: Sender, Credential, Payment, ItemDetail
Atributos: version, payloadID, lineNumber, domain

Envelope cXML

O elemento envelope é a raiz da estrutura do documento cXML e contém todos os outros elementos. O elemento cXML está presente em todas as transações cXML.

O exemplo a seguir mostra um elemento cXML inteiramente especificado:

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

cXML tem os seguintes atributos:

version (opcional)	Especifica a versão do protocolo cXML. Um analisador XML de validação também consegue determinar o atributo version, usando a DTD de referência. Entretanto, todos os documentos cXML devem incluir a versão explicitamente para ajudar os aplicativos que usam analisadores não-validação.
xml:lang (opcional)	A localidade usada para todos os textos livres enviados nesse documento. O receptor deve responder ou exibir as informações na mesma localidade ou em localidade semelhante. Por exemplo, um cliente que especificar xml:lang="en-UK" em uma solicitação deve receber o dado "en" de volta.
payloadID	Um número exclusivo referente a espaço e horário, usado para registrar propósitos para identificar documentos que possam ter sido perdidos ou ter apresentado problemas. Esse valor não deve ser alterado para tentativas de recuperação. A implementação recomendada é a seguinte: datetime.process id.random number@hostname
timestamp	Data e horário de envio da mensagem, no formato ISO 8601. Esse valor não deve ser alterado para tentativas de resposta. O formato é YYYY-MM-DDThh:mm:ss-hh:mm (por exemplo, 1997-07-16T19:20:30+01:00).

Localidade especificada pelo atributo xm1:lang

O atributo xml:lang também é exibido na maioria dos elementos de texto livre (como Description e Comments). Embora a especificação XML permita que a localidade referente a um elemento seja padrão para o que foi especificado a qualquer elemento-pai, esses padrões resultam em consultas ineficientes à árvore de documentos. A cXML tenta manter os identificadores de localidade junto com os caracteres afetados.

Os atributos xml:lang exibidos em todo o protocolo cXML não têm efeito sobre dados formatados, como números, datas e horários. Conforme descrito abaixo, para o atributo timestamp, esses valores discretos são formatados de acordo com o tipo de dados correspondente. Caracteres maiores (e páginas da Web consultadas) não destinados ao processamento da máquina devem conter um formato numérico ou de data específico à localidade que corresponda ao máximo com o atributo xml:lang.

Horário e outros tipos de dados

O atributo timestamp (e todos os outros horários e datas em cXML) deve ser formatado no subconjunto restrito da ISO 8601 descrito na nota do World Wide Web Consortium (W3C) intitulada "Date and Time Formats", disponível no endereço www.w3.org/TR/NOTE-datetime-970915.html.

Registros horários exigem no mínimo uma data completa, mais horas, minutos e segundos. A notação de frações de segundo é opcional. Esse protocolo exige horários expressos em horário local, com diferença de fuso horário em relação à hora universal (UTC, Universal Time Coordinated, também conhecida como Hora de Greenwich). O designador "Z" de fuso horário não é permitido.

Por exemplo, 2000-04-14T013:36:00-08:00 corresponde a 14 de abril de 2000, 13:36, Hora do Pacífico dos EUA padrão.

Veja a seguir outras referências a data, horário e a outros formatos de tipo de dados usados pela cXML:

- Site XML Data Types Reference (Referência a Tipos de Dados XML) da Microsoft, msdn.microsoft.com/xml/reference/schema/datatypes.asp
- A proposta original da XML Data ao Word Wide Web Consortium (W3C), www.w3c.org/TR/1998/NOTE-XML-data-0105

Camadas encobertas

O elemento cXML é o corpo de um documento XML normal. Um documento deve começar da seguinte forma:

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

Esse documento normalmente é transmitido por HTTP, com um tipo de mídia MIME text/xml e um parâmetro charset correspondente à codificação no documento. Como o HTTP é um sistema limpo de oito bits, qualquer codificação suportada pelo analisador destinatário pode ser usada sem uma codificação de transferência de conteúdo, como base 64 ou imprimível entre aspas. Todos os analisadores XML suportam a codificação UTF-8, que inclui todos os caracteres Unicode. Portanto, os aplicativos devem usá-la para transmitir documentos cXML.

Nota: De acordo com o "XML Media Types" RFC 2376, o parâmetro MIME charset sobrepõe qualquer codificação especificada na declaração XML. Ademais, a codificação padrão para o tipo de mídia text/xml é us-ascii, e não UTF-8, como mencionado na Seção 4.3.3 da Especificação XML. Para esclarecer, os documentos cXML devem incluir uma codificação explícita na declaração XML. Os envelopes MIME devem usar um parâmetro de correspondência charset para o tipo de mídia text/xml ou application/xml.

A transmissão HTTP de um documento cXML pode incluir os seguintes cabeçalhos MIME e HTTP.

```
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"?>
…
```

Transmissão de anexos

Quando um documento OrderRequest que consulta arquivos externos está sendo enviado, os arquivos consultados podem residir em um servidor que possa ser acessado pelo fornecedor ou podem ser transmitidos com o documento cXML. A segunda opção requer o uso de um envelope MIME contínuo. Uma exigência cXML com relação a esse envelope (além das descritas em "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046) é a inclusão de cabeçalhos Content-ID (com código de conteúdo) em todos os arquivos anexados.

O exemplo a seguir mostra a estrutura exigida de um documento cXML com uma imagem JPEG anexada (sem os cabeçalhos HTTP apresentados anteriormente):

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

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

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

Esse esboço também é tudo o que um analisador MIME receptor deve estar apto a processar. Os aplicativos que usam o tipo de mídia descrito em "The MIME Multipart/Related Content-type", RFC 2387, obterão muito mais informações se a estrutura for otimizada:

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

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

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

Os analisadores MIME receptores que não entendem o tipo de mídia multipart/related devem lidar com os dois exemplos anteriores de maneira idêntica. Cada parte da transmissão MIME pode ter, adicionalmente, uma codificação de transferência de conteúdo e usar essa codificação. Esse processo não é necessário à transmissão HTTP. Os cabeçalhos Content-description e Content-disposition são opcionais no protocolo cXML, embora forneçam uma documentação útil.

Para obter mais informações sobre como anexar arquivos externos a ordens de compra, consulte "Attachment" na página 73.

Header

O elemento Header contém informações de endereçamento e autenticação. O elemento Header é o mesmo, independentemente do Request ou Response específico no corpo da mensagem cXML. Os aplicativos necessitam da identidade do solicitante, mas não da confirmação de que a informação fornecida para essa identidade é correta.

O exemplo a seguir mostra o elemento 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>
```

Os elementos From e To são sinônimos de From e To em mensagens de correio SMTP; eles são a origem e o destino lógicos das mensagens. Sender é a parte que abre a conexão HTTP e envia o documento cXML.

Sender contém o elemento Credential, que permite à parte receptora autenticar a parte emissora, possibilitando uma autenticação robusta, sem a necessidade de uma infra-estrutura de certificação digital terminal a terminal de chave pública. Apenas o nome de usuário e a senha precisam ser divulgados pela parte receptora para que a parte emissora possa executar Requests.

Inicialmente, Sender e From são idênticos. Entretanto, se o documento cXML trafegar pelo hub de e-commerce, o elemento Sender será alterado, para indicar a parte emissora atual.

From

Esse elemento identifica o originador da solicitação cXML. Opcionalmente, pode conter mais de um elemento Credential, o que possibilita a auto-identificação dos solicitantes por meio de vários métodos de identificação. O uso de múltiplas credenciais é análogo ao envio dos endereços SMTP e X.400 em uma mensagem de correio eletrônico.

To

Esse elemento identifica o destino da solicitação cXML. Da mesma forma que o elemento From, esse elemento pode conter mais de um elemento Credential para ajudar a identificar o destino.

Sender

Esse elemento permite à parte receptora identificar e autenticar a parte que abriu a conexão HTTP. O Sender contém um elemento Credential cuja autenticação é mais robusta que a de outros nos elementos From ou To, porque a parte receptora precisa autenticar o solicitante a fim de executar a solicitação.

Credential

Esse elemento contém valores de identificação e autenticação usados nas mensagens cXML.

Credential tem os seguintes atributos:

domain	Especifica o tipo de credencial. Esse atributo permite que os documentos contenham vários tipos de credenciais para vários domínios de autenticação. No caso de mensagens enviadas no Ariba Network, por exemplo, o domínio em geral é AribaNetworkUserId ou DUNS.
type (opcional)	Solicitações para ou de um mercado identificam tanto o mercado quanto o membro da empresa nos elementos Credential From ou To. Nesse caso, a credencial para o mercado usa o atributo type, que é acrescentado ao valor "marketplace".

Credential contém o elemento Identity e, opcionalmente, o elemento SharedSecret ou DigitalSignature. O elemento Identity estabelece quem é representado pelo Credential, enquanto os elementos opcionais de autenticação confirmam a identidade da parte.

O elemento SharedSecret é usado quando o Sender tem um nome de usuário e senha reconhecidos pelo solicitante.

O elemento DigitalSignature pode ser empregado se as duas partes concordarem em usar um formato de certificado e autoridade comuns. O atributo type em um elemento DigitalSignature especifica o tipo de certificado que está sendo usado.

> **Nota:** Não empregue elementos de autenticação em documentos com transmissão unidirecional. Esse transporte é encaminhado pelos browsers dos usuários, possibilitando que os usuários vejam a origem do documento (incluindo os elementos Credential).

Request

Os clientes enviam solicitações para operações. Para cada elemento envelope cXML é permitido um elemento Request, o que simplifica as implementações do servidor, pois não há necessidade de desfazer a multiplexação no momento da leitura de documentos cXML. O elemento Request pode conter praticamente qualquer tipo de dado XML.

Request tem os seguintes atributos:

deploymentMode (opcional)	Indica se a solicitação é uma solicitação de teste ou de produção. Os valores permitidos são "production" (padrão) ou "test".

Response

Os servidores enviam respostas para informar aos clientes os resultados das operações. Como o resultado de algumas solicitações podem não conter dados, o elemento Response pode conter, opcionalmente, somente um elemento Status. O elemento Response pode conter também quaisquer dados no nível de aplicativo. No contexto de punchout, isso significa um elemento PunchOutSetupResponse.

Status

Esse elemento é responsável pelo sucesso ou falha de uma operação de solicitação.

Status tem os seguintes atributos:

code	Código de status da solicitação. Segue o modelo de código de status HTTP. Por exemplo, 200 representa uma solicitação bem-sucedida.
text	Texto da mensagem de status. Esse texto ajuda os usuários na leitura de logs e isso consiste em caracteres canônicos em inglês.
xml:lang (opcional)	Linguagem dos dados no elemento Status. Opcional em caso de compatibilidade com cXML 1.1. Pode ser exigido em versões cXML posteriores.

Os atributos do elemento Status indicam o que ocorreu com a solicitação.

Os dados no elemento Status podem ser quaisquer dados necessários ao solicitante. No caso do código de status 200/OK, não deve haver nenhum dado. Entretanto, para o código de status 500/Internal Server Error, é fundamental que o erro do analisador XML ou do aplicativo seja apresentado. Esse erro permite efetuar parcialmente uma depuração e um teste de interoperacionalidade superiores.

Os servidores não devem incluir elementos adicionais de resposta (por exemplo, um elemento PunchOutSetupResponse), a menos que o código de status estiver na faixa de 200 (por exemplo, 200/OK).

A especificação HTTP 1.1 inclui muitos códigos de status inadequados para cXML. Como, na maioria dos casos, a cXML sobrepõe-se em camadas ao HTTP, muitos erros (como, por exemplo, 404/Not Found) serão tratados pelo transporte. Os códigos de status 200/OK e500/Internal Server Error são mais prováveis. Erros de validação na análise de um documento Request normalmente resultam em um erro de transporte, como, por exemplo, um erro HTTP 400/Bad Request.

A tabela a seguir apresenta outros códigos HTTP que podem ser usados.

A cXML compreende pouquíssimos códigos de status não-HTTP:

- 550: Incapacidade de alcançar o servidor cXML para concluir uma transação que exija conexões opostas ao fluxo. Um hub intermediário pode retornar esse código quando um site de fornecedor está inacessível. (Se as conexões opostas ao fluxo forem concluídas, hubs intermediários retornarão os erros diretamente ao cliente.)
- 551: Incapacidade de encaminhar a solicitação em decorrência de configuração incorreta do fornecedor. Por exemplo, um hub intermediário não consegue autentificar-se para um fornecedor. Embora os clientes não possam retificá-lo, esse erro pode ser corrigido antes que o cliente faça uma nova tentativa.
- 560: Erro temporário no servidor. Por exemplo, um servidor pode estar inativo por falta de manutenção. O cliente deve tentar de novo posteriormente.

Status	Texto canônico	Significado
200	OK	O servidor estava apto a executar essa solicitação, embora a resposta retornada pudesse conter erros ou avisos do aplicativo.
201	Accepted	Algum processamento pode não ter sido concluído. De acordo com o que foi descrito em "StatusUpdateRequest" na página 87, o cliente deve esperar transações StatusUpdate posteriores caso esse status seja retornado em resposta a um OrderRequest.
204	No Content	Todas as informações de Request eram validadas e reconhecidas. O servidor não tem dados de Response do tipo solicitado. Em um PunchOutOrderMessage, esse status indica que a sessão punchout foi finalizada sem alterações no carrinho de compras (ou na requisição do cliente).
400	Bad Request	Solicitação inaceitável para o servidor, embora tenha sido analisada corretamente.

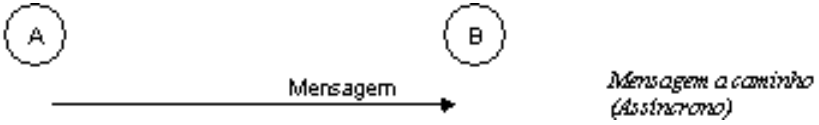
Status	Texto canônico	Significado
401	Unauthorized	As credenciais fornecidas em Request (o elemento Sender) não foram reconhecidas pelo servidor.
402	Payment Required	O documento Request deve conter um elemento Payment completo.
403	Forbidden	O usuário não tem privilégios suficientes para executar o Request.
406	Not Acceptable	Um nome alternativo para o código 400: Solicitação não aceita pelo servidor, embora tenha sido analisada corretamente.
409	Conflict	O estado atual do servidor ou seus dados internos impediram a solicitação da operação (atualização). Um documento Request idêntico pode vir posteriormente, especialmente depois de uma nova operação ter sido executada.
412	Precondition Failed	Uma precondição de Request (por exemplo, uma sessão punchout para um PunchOutSetupRequest edit) não foi atendida. O status em geral significa que o cliente ignorou parte de uma transmissão anterior de um servidor (por exemplo, o atributo operationAllowed de PunchOutOrderMessageHeader).
417	Expectation Failed	Request indicou uma condição de recurso que não foi atendida. Um exemplo pode ser a solicitação de informações por parte de SupplierDataRequest sobre um fornecedor não conhecido pelo servidor. Esse status pode significar perda de informações por parte do cliente ou do servidor.
500	Internal Server Error	O servidor não estava apto a concluir Request.
501	Not Implemented	O servidor não implementa esse Request em particular. Por exemplo, PunchOutSetupRequest ou a operação solicitada pode não ser suportada. O status em geral significa que o cliente ignorou o perfil do servidor.

Ao receber códigos irreconhecíveis, os clientes cXML devem tratá-los de acordo com a respectiva classe. Entretanto, clientes mais antigos devem tratar todos os novos códigos 2*xx* como códigos 200, códigos 4*xx* como 400 e códigos 5*xx* como 500. Dessa forma, permitirão expansões posteriores do protocolo cXML e de códigos específicos do servidor, sem perda de interoperacionalidade.

Modelo unidirecional (assíncrono)

Diferentemente das transações de solicitação/resposta, mensagens unidirecionais não estão restritas ao transporte HTTP.

Mensagens unidirecionais destinam-se a situações em que um canal HTTP (uma operação síncrona de solicitação/resposta) não é apropriado. A figura a seguir mostra um exemplo de como A e B podem se comunicar por mensagens, em vez de por uma transação de solicitação/resposta.



Nesse caso, uma situação possível seria:

1. A formata e codifica um documento cXML em um transporte entendido por B.
2. A envia o documento usando o transporte conhecido. A não espera (e não pode esperar) ativamente uma resposta de B.
3. B recebe o documento cXML e decodifica-o fora do fluxo de transporte.
4. B processa o documento.

Em um modelo unidirecional, A e B *não* têm um ciclo explícito de solicitação/resposta. Por exemplo, entre as mensagens unidirecionais, podem chegar mensagens de outras partes e ocorrer comunicações paralelas.

Para especificar uma transação totalmente unidirecional, o transporte de mensagens usado deve também ser documentado. No caso de transações cXML que usam o modelo unidirecional, o transporte e a codificação são especificados. Um exemplo básico de transação que usa o modelo unidirecional é o documento PunchOutOrderMessage.

A estrutura das mensagens unidirecionais é similar à do modelo solicitação/resposta:

```
<cXML>
    <Header>
        Header information here…
    </Header>
    <Message>
        Message information here…
    </Message>
</cXML>
```

A forma pela qual o elemento Header é tratado é exatamente igual à forma como é tratado em solicitação/resposta. O elemento cXML é também idêntico àquele descrito anteriormente. A maneira mais fácil de diferenciar uma mensagem unidirecional de uma mensagem de solicitação/resposta é observar a existência de um elemento Message (em vez de um elemento Request ou Response). A seção a seguir discute o elemento Message mais detalhadamente.

Message

Esse elemento transporta todas as informações estruturais em uma mensagem cXML. Pode conter um elemento Status opcional, idêntico ao encontrado no elemento Response, que pode ser usado em mensagens que fossem respostas lógicas a mensagens de solicitação.

Message tem os seguintes atributos:

deploymentMode (opcional)	Indica se a solicitação é uma solicitação de teste ou de produção. Os valores permitidos são "production" (padrão) ou "test".
inReplyTo (opcional)	Especifica a que mensagem essa mensagem deve responder. O conteúdo do atributo inReplyTo pode ser o payloadID de uma mensagem recebida mais recentemente. Isso seria usado para gerar uma conversação bidirecional em que houvesse muitas mensagens.

O atributo inReplyTo também pode mencionar o payloadID de um documeto Request ou Response mais recente. Quando uma transação de solicitação/resposta inicia uma "conversação" por meio de múltiplas interações unidirecionais, a primeira mensagem pode conter o atributo payloadID da solicitação ou resposta mais recente e relevante que seguir a direção oposta. Por exemplo, uma mensagem contendo um PunchOutOrderMessage deve incluir um atributo inReplyTo que, por sua vez, contivesse o atributo payloadID do PunchOutSetupRequest que iniciasse a sessão punchout. (O elemento BuyerCookie incluído nos documentos punchout desempenham uma função similar à desempenhada pelo atributo inReplyTo nesse caso.)

Opções de transporte

Há dois transportes comumente usados em mensagens unidirecionais: HTTP e codificação de formulário URL. Ambos são apenas dois dos atuais transportes bem-definidos; outros podem vir a ser suportados.

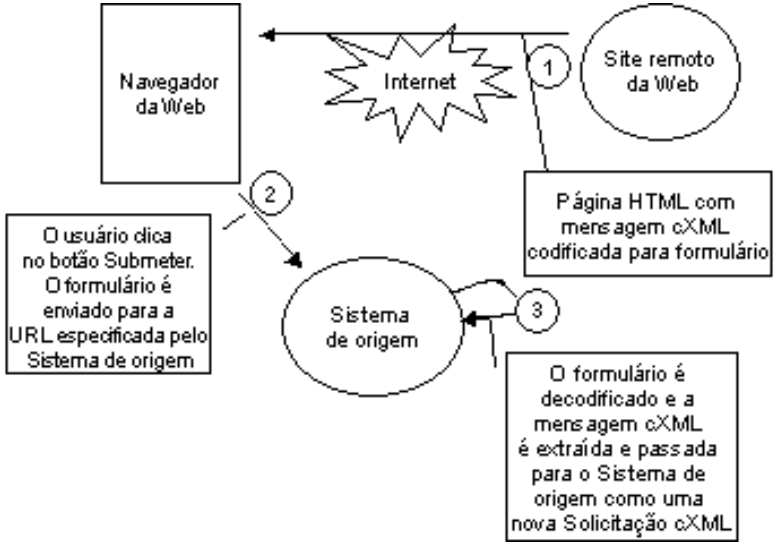
HTTP

HTTP é usado em comunicações unidirecionais com o objetivo de possibilitar a extração de informações por parte dos aplicativos de compra. O único tipo de transação que usa comunicações HTTP unidirecionais é o GetPendingRequest, o qual é discutido na página 102.

Codificação de formulário URL

Esse transporte é mais bem compreendido quando se analisa a forma pela qual é executada a transação PunchOutOrderMessage. A codificação de formulário URL possibilita a integração entre um site remoto da Web e aplicativos de compra. Além disso, é uma forma de evitar a necessidade de um servidor de escuta no sistema do comprador diretamente acessível por meio da Internet.

A mensagem cXML PunchOutOrderMessage não é enviada diretamente à aplicativo de compra pelo site remoto da Web, mas é codificado como um Formulário HTML oculto e transmitido ao URL especificado no elemento BrowserFormPost de PunchOutSetupRequest. Quando o usuário clica em Registrar saída, o site da Web envia os dados ao aplicativo de compra como Formulário HTML de Submissão.
O diagrama a seguir ilustra o que ocorre nesse processo:



A semântica de empacotamento e desempacotamento é descrita a seguir.

Empacotamento de formulário

O documento PunchOutOrderMessage, um URL-Encoded (de acordo com a especificação HTTP), é atribuído a um campo oculto em Form named cXML-urlencoded. O elemento HTML Form é atribuído a um MÉTODO DE POSTAGEM (METHOD of POST) e a uma AÇÃO (ACTION) que consistem no URL transmitido no elemento BrowserFormPost do PunchOutSetupRequest. Por exemplo:

```
<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>
```

Outras tags de HTML na página devem conter o fragmento recém-apresentado para descrever detalhadamente o conteúdo da cesta de compras.

> **Nota:** Quando os servidores da Web enviam o campo cXML-urlencoded, ainda não é um URL codificado. Essa codificação é exigida somente quando o formulário é submetido por navegadores da Web (quando os usuários clicam em Registrar saída, no exemplo apresentado anteriormente). Os próprios navegadores da Web atendem a essa exigência. O servidor da Web deve codificar em HTML apenas o valor do campo, aspas de escape e outros caracteres especiais, de forma que o formulário seja exibido apropriadamente ao usuário.

O nome cXML-urlencoded é sensível a maiúsculas e minúsculas.

No caso de dados cXML-urlencoded, o analisador gramatical receptor não consegue adotar um outro parâmetro charset além do padrão para mídias do tipo text/xml. Nenhuma informação de codificação de caracteres relacionada com os dados enviados é transportada em HTTP POST. O servidor da Web receptor não consegue determinar a codificação da página HTML que contém o campo oculto. O documento cXML enviado dessa maneira deve, portanto, usar a codificação de caracteres us-ascii. Qualquer caractere (incluindo aqueles "URI codificados" como "%XX") encontrado em um documento XML de origem deve pertencer ao conjunto "us-ascii". Outros símbolos Unicode podem ser codificados usando-se entidades de caractere nesse documento de origem.

Codificação base 64

O campo oculto cXML-base64 facilita a manipulação de documentos internacionais. Os documentos cXML que contiverem símbolos que não pertençam ao "us-ascii" devem usar esse campo, em vez de o campo oculto cXML-urlencoded. Essa alternativa tem uma semântica quase idêntica, mas o documento completo usa a codificação base 64 ao longo do transporte, e não uma codificação HTML para o navegador ou uma codificação URL para o servidor da Web. A codificação base 64 é descrita em "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045.

O nome cXML-base64 é sensível a maiúsculas e minúsculas.

A codificação base 64 — do site do fornecedor, passando pelo navegador até chegar ao servidor da Web receptor e então ao cliente — mantém a codificação de caracteres original de um documento cXML. Embora nenhum parâmetro charset chegue junto com a informação enviada, o documento decodificado (depois que a codificação transferida é removida) pode ser tratado como uma mídia do tipo application/xml. Essa codificação possibilita ao analisador receptor aceitar qualquer atributo encoding especificado na declaração XML. Para esse campo (da mesma forma que para qualquer documento application/xml), a codificação-padrão de caracteres é UTF-8.

Ambos esses campos ocultos (cXML-urlencoded ou cXML-base64) devem aparecer nos dados enviados ao aplicativo de compra. Embora os receptores devam primeiramente procurar cXML-base64 nos dados, é dispendioso enviar ambos os campos.

Desempacotamento e processamento de formulário

O aplicativo de compra, que fornece previamente o URL apropriado, recebe um HTML Form POST contendo os dados de formulário, como foi descrito anteriormente. O processador Form POST primeiramente procurará a variável cXML-base64, extrairá o valor e procederá à decodificação base 64 do conteúdo. Se esse campo não aparecer nos dados, o processador Form POST procurará a variável cXML-urlencoded, extrairá a mensagem cXML de codificação URL e procederá à decodificação URL. O conteúdo decodificado do campo é então processado, como se houvesse sido recebido por meio de um ciclo normal HTTP de solicitação/resposta.

O tipo de mídia implícito do documento depois da decodificação varia e há a possibilidade de diferentes codificações de caracteres:

- A variável cXML-urlencoded é do tipo de mídia text/xml, sem nenhum atributo charset. Isso, portanto, está restrito à codificação de caracteres us-ascii. O analisador receptor deve ignorar qualquer atributo encoding na declaração XML do documento cXML porque é possível que o navegador tenha alterado a codificação.
- A variável cXML-base64 é do tipo de mídia application/xml e, portanto, deve ter alguma codificação de caracteres (se tiver, será indicada pelo atributo encoding da declaração XML).

A principal diferença entre essa transação e uma transação de solicitação/resposta normal é que não há nenhuma resposta que possa ser gerada, porque não há nenhuma conexão HTTP por meio da qual a resposta possa ser enviada.

Elementos básicos

Os elementos e entidades a seguir são usados na especificação cXML. A maioria das definições apresentadas aqui usa um vocabulário básico por meio do qual os documentos comerciais de mais alta ordem são descritos. Os tipos de entidade comuns e os elementos comuns que representam objetos de baixo nível são definidos aqui.

Tipos de entidade

A maioria dessas definições provém da submissão de notas XML-Data ao Word Wide Web Consortium (W3C). Algumas entidades de nível mais alto, que também são definidas aqui, não são provenientes do XML-Data. Esse tipo de entidade também é discutido em "Envelope cXML" na página 50.

isoLangCode

Código ISOde Idioma, proveniente da ISO 639 padrão.

isoCountryCode

Código ISO de País, proveniente da ISO 3166 padrão.

xmlLangCode

É um código de idioma, como é definido pela XML 1.0 Specification (emwww.w3.org/TR/1998/REC-xml-19980210.html). Nos casos mais comuns, isso inclui um Código ISO 639 de Idioma e (opcionalmente) um Código ISO 3166 de País separado por um hífen. Diferentemente de toda a recomendação XML, IANA ou códigos de idioma privados não devem ser usados em cXML. IANA e subcódigos privados são permitidos, embora devam vir depois de um Código ISO 3166 de País.

O formato de código de idioma recomendado é xx[-YY[-zzz]*]?, no qual xx é um Código ISO 639 de Idioma, YY é um Código ISO de País e zzz é um subcódigo IANA ou privado para o idioma em questão. Uma vez mais, o uso do código de país é sempre recomendado. Por convenção, o código de idioma é escrito em letras minúsculas e o de país, em letras maiúsculas. Isso não exigido para uma combinação correta dos códigos.

unitOfMeasure

UnitOfMeasure descreve como o produto é embalado ou entregue. Isso deve estar de acordo com o UN/CEFACT Unit of Measure Common Codes (Códigos Comuns de Unidades de Medida UN/CEFACT. Para obter uma lista de códigos UN/CEFACT, consulte www.unece.org/cefact.

URL

Um URL (Uniform Resource Locator), como definido por HTTP/1.1 padrão.

Elementos de base

Esses elementos, usados ao longo da especificação, variam de genéricos, como Name e Extrinsic, a específicos, como Money.

Transação Perfil

Os documentos ProfileRequest e ProfileResponse devem ser suportados por implementações cXML 1.1 do servidor. Essa transação pode ser usada para recuperar capacidades do servidor, incluindo versões cXML que são suportadas, transações e opções para essas transações.

A resposta deve relacionar todas as solicitações suportadas por determinado site da Web, mas não necessariamente todos aqueles suportados pela empresa. Os fornecedores que podem receber documentos OrderRequest e enviar várias mensagens ou iniciar transações de solicitação/resposta descrevem o respectivo suporte OrderRequest na transação de perfil.

A transação de perfil pode ser usada para "alternar" um servidor dentro de um protocolo cXML.

ProfileRequest

Esse elemento não tem nenhum conteúdo. É simplesmente encaminhado ao servidor apropriado, usando Header. O servidor responde com um único ProfileResponse, como descrito anteriormente. As únicas partes dinâmicas dessa resposta são os atributos payloadId e timestamp do próprio elemento cXML. Nesse caso em particular, não se exige que o forneça respostas em múltiplas localidades.

Um exemplo de Request desse tipo é:

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

ProfileResponse

Esse elemento contém uma lista de transações suportadas, a respectiva localização e quaisquer opções suportadas. Embora nenhuma opção tenha sido definida até o momento, a seguinte possivelmente é um ProfileResponse:

```
<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>
```

Um ProfileResponse mais provável, proveniente do fornecedor, pode ser:

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

ProfileResponse tem os seguintes atributos:

effectiveDate	A data e o horário desses serviços tornaram-se disponíveis. As datas não devem estar no futuro.

Option

Valor para uma opção definida (tanto para o serviço geral quanto para uma transação específica). Nenhuma opção foi definida até o momento.

Option tem os seguintes atributos:

name	Nome dessa opção. Esse atributo não deve ser visto diretamente (porque o perfil é destinado à consumação da máquina).

Transaction

Descrição da transação suportada por esse serviço. A descrição atual de perfil indica as localizações para as quais as solicitações devem ser enviadas. Versões cXML posteriores vão acrescentar definições Option e estender informações de perfil a fim de incluir mais informações sobre solicitações suportadas.

O elemento Transaction deve conter um elemento URL.

Transaction tem os seguintes atributos:

requestName	Uma solicitação específica que esse servidor aceita no URL fornecido. Os valores podem ser: ProfileRequest PunchOutSetupRequest GetPendingRequest SupplierListRequest SupplierDataRequest OrderRequest StatusUpdateRequest SubscriptionListRequest SubscriptionContentRequest

Definições de pedido

Os documentos cXML de processamento de pedidos são OrderRequest e uma resposta genérica. OrderRequest é análogo a uma ordem de compra. A resposta é a confirmação de que o fornecedor recebeu a ordem de compra. Não é obrigatório executar a ordem de compra, mas o é a confirmação de que foi corretamente recebida.

OrderRequest

O exemplo a seguir ilustra a estrutura do elemento OrderRequest:

```
<OrderRequest>
    <OrderRequestHeader … >
        …
    </OrderRequestHeader>
    <ItemOut … >
        …
    </ItemOut>
    <ItemOut … >
        …
    </ItemOut>
</OrderRequest>
```

OrderRequestHeader

O exemplo a seguir mostra detalhadamente um 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">
    Anything well formed in XML can go here.
```

```
        </Comments>
        <Followup>
            <URL>http://acme.com/cgi/orders.cgi</URL>
        </Followup>
    </OrderRequestHeader>
```

OrderRequestHeader tem os seguintes atributos:

orderID	Identificador desse pedido. É análogo ao número de ordem de compra.
orderDate	A data e o horário em que esse pedido foi feito, no formato ISO 8601.
type (opcional)	Tipo da solicitação: new (padrão), update ou delete.
requisitionID (opcional)	Identificador da requisição do comprador para esse pedido todo. Deve ser igual a orderID e não deve ser incluído de modo algum. Não deve ser incluído, se requisitionID for especificado em qualquer elemento ItemOut.
shipComplete (opcional)	É preferível a entregas parciais. O único valor válido é "yes". Por padrão, os itens são entregues quando disponíveis. Talvez os pedidos contenham itens com diferentes elementos ShipTo, então devem ser retidos apenas grupos de itens cujas localizações de entrega forem os mesmos devem ser mantidos até que shipComplete="yes".

OrderRequestHeader e ItemOut (quando estendidos com ItemDetail) contêm informações semelhantes. Enquanto OrderRequestHeader contém o faturamento geral (BillTo) e informações sobre pagamento (Payment), ItemOut descreve cada item (em ItemID, ItemDetail e Distribution).

Não use a informação em OrderRequestHeader como é o padrão para elementos de item específico. Se presentes, ShipTo, Shipping, Contact e todos os denominados Extrinsic devem aparecer com todos os ItemOut ou em OrderRequestHeader. Os elementos Comments e Tax podem aparecer simultaneamente em ambos os níveis. Mas os elementos Comments diferentes não devem repetir informações e o elemento Tax de nível de cabeçalho deve conter o total do pedido.

Total

Esse elemento contém a quantia total do pedido. É o recipiente do elemento Money.

ShipTo/BillTo

Esseselementos contêm os endereços de remessa das entidades Ship To e Bill To em OrderRequest.

Um pedido deve ser faturado em uma única entidade. Entretanto, o elemento BillTo aparece somente em OrderRequestHeader. Os itens de um pedido podem ser enviados a múltiplas localizações. Da mesma forma que o elemento Shipping (consulte a próxima seção), o elemento ShipTo pode, portanto, aparecer tanto em OrderRequestHeader quanto em elementos ItemOut individuais.

Shipping

Esse elemento descreve como entregar os itens solicitados e calcular o custo dessa remessa. Se o elemento Shipping estiver presente em OrderRequestHeader, deverá aparecer nos elementos ItemOut individuais. Caso não estejam presentes em OrderRequestHeader, deverão aparecer nos elementos ItemOut.

Tax

Esse elemento contém o imposto referente ao pedido. Está incluído no cálculo de impostos da organização de compra. Quando aparecer em OrderRequestHeader, Tax descreverá o total de impostos do pedido. Os elementos Tax no nível item podem descrever quantias unitárias de impostos.

Payment

Esse elemento descreve o instrumento pagamento usado para efetuar o pagamento dos itens solicitados. No exemplo apresentado anteriormente, o elemento Payment contém um elemento PCard, o qual codifica um cartão de compras padrão no documento cXML. No futuro, outros instrumentos de pagamento serão definidos e suportados.

Contact

Informações sobre contato que o fornecedor pode usar para dar prosseguimento a um pedido. Esse elemento identifica uma pessoa e fornece uma lista de caminhos para encontrar essa pessoa ou entidade. O único elemento exigido é Name, referente ao nome do contato. Dentre as possibilidades opcionais e repetitivas encontram-se PostalAddress (não recomendada para correção imediata de problemas no pedido), Email, Phone, Fax e URL.

As organizações de compra devem escolher esse elemento para identificar o solicitante original, o administrador do sistema do aplicativo de compra ou algum outro contato que possa se responsabilizar pela correção de problemas no pedido. Contact pode diferir tanto das informações BillTo quanto ShipTo de um pedido.

Contact tem os seguintes atributos:

role (opcional)	Cargo dessa pessoa no processo de compra. Pode pegar os valores endUser, administrator, purchasingAgent, technicalSupport, customerService ou sales.

Um mesmo Contact role não deve aparecer no cabeçalho e nos níveis de item.

Não há uma função-padrão, em decorrência da disparidade de conteúdo do elemento Contact. Portanto, os aplicativos cXML tratam um Contact sem um atributo role como uma função adicional.

Comments

Informações arbitrárias sobre compradores que possam ser lidas por humanos podem ser enviadas com as ordens de compra. Esses dados de caracteres não se destinam a sistemas automatizados em sites de fornecedores.

O elemento Comments pode conter um elemento Attachment para incluir arquivos externos.

Attachment

Comments pode anexar arquivos externos para aumentar as ordens de compra. O elemento Attachment aparece em Comments e contém apenas uma referência à parte MIME externa do anexo. Todos os anexos devem ser enviados em uma única transmissão contínua com o documento OrderRequest. Mesmo se isso não for possível, o contentID fornecido pelo elemento Attachment deve ser utilizável para recuperar o anexo.

Para obter detalhes sobre a transferência de arquivos anexados, consulte "Transmissão de anexos" na página 53.

Attachment contém um único URL com o esquema "cid:". Um arquivo anexado a um documento cXML deve aparecer como:

```
<Comments>
    <Attachment>
        <URL>cid: uniqueCID@cxml.org</URL>
    </Attachment>
    Please see attached image for my idea of what this
    should look like
    </Comments>
```

O elemento Comments aparece em vários lugares dentro do protocolo cXML, mas pode conter o elemento Attachment somente dentro de documentos OrderRequest.

Followup

Especifica o URL para o qual documentos StatusUpdateRequest futuros devem ser enviados. Essa localização é a entrada de qualquer documento posterior que se refira ao atual documento OrderRequest.

Extrinsic

Esse elemento contém informações sobre o pedido que podem ser lidas por máquina, mas não podem ser definidas pelo protocolo cXML. Diferentemente, o elemento Comments envia informações utilizáveis por humanos. Os elementos Extrinsic contêm dados que têm maior probabilidade de aparecer em documentos posteriores; o elemento Comments não. Nesse nível, Extrinsic estende a descrição de todos os itens de uma ordem de compra. Algumas informações Extrinsic devem também descrever a ordem de compra geral, sem afetar o significado de nenhum ItemOut.

Cada uma das designações Extrinsic pode aparecer somente uma vez nas listas associadas com OrderRequestHeader e elementos ItemOut individuais (dentro dos elementos ItemDetail). Nomes idênticos não devem aparecer na lista OrderRequestHeader e em nenhuma lista associada com os elementos ItemOut. Se o mesmo nome e valor Extrinsic for repetido em todas as listas ItemOut, deverá ser movido ao OrderRequestHeader.

O elemento Extrinsic também pode aparecer nos elementos IndexItem, PunchOutSetupRequest e ContractItem. Esses contextos são descritos posteriormente neste documento.

ItemOut

O exemplo a seguir mostra um elemento mínimo ItemOut válido.

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

ItemOut tem os seguintes atributos:

quantity	Número dos itens desejados. As frações são permitidas a algumas unidades de medida. O valor provavelmente já foi verificado pelo fornecedor durante a sessão punchout. Nunca deve ser negativo.
lineNumber (opcional)	Posição desse item em um pedido. Esse valor ordinal aumenta uma vez por ItemOut em uma "nova" OrderRequest. Os clientes devem sempre especificar esse atributo em uma OrderRequest, embora isso não seja útil em outros contextos ItemOut.
requisitionID (opcional)	Identificador de requisição do comprador para esse item de linha. Não deve ser incluído, se requisitionID estiver especificada em OrderRequestHeader.
requestedDeliveryDate (opcional)	O item data foi solicitado para entrega, e isso, portanto, permite datas de entrega no nível de item, em OrderRequest. Deve estar de acordo com o formato ISO 8601.

O atributo lineNumber permanece constante para qualquer item, desde atualizações até o pedido. A exclusão de itens de um pedido nunca altera o atributo lineNumber de itens remanescentes. Os novos itens têm um número maior que aqueles previamente incluídos no pedido. Uma modificação em um item (por exemplo, aumentar a quantidade) não afeta o atributo lineNumber correspondente a esse item.

O exemplo a seguir mostra um ItemOut mais complicado.

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

```
        <UnitOfMeasure>EA</UnitOfMeasure>
        <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">
        Anything valid in XML can go here.
```

```
        </Comments>
    </ItemOut>
```

O elemento ItemDetail permite que se enviem dados adicionais ao fornecedor, em vez de somente o identificador do item representado por ItemID.

Os elementos ShipTo, Shipping, Tax, Contact, Comments e Extrinsic (alguns aninhamos em ItemDetail) são idênticos àqueles que podem estar em OrderRequestHeader. Esses elementos permitem a representação de dados por item, tais como entrega, tipo de entrega e custos correspondentes. Use esses elementos no nível OrderRequestHeader ou no nível ItemOut, mas não em ambos.

Distribution

Divide o custo de um item em múltiplas partes. Os fornecedores retornam o elemento Distribution em faturas para facilitar o processo de reconciliação do comprador.

Accounting

O elemento Accounting agrupa Segments para identificar quem será cobrado.

Accounting tem o seguinte atributo:

name	Nome do acordo contábil.

Segment tem os seguintes atributos:

type	Um nome de identificação para esse atributo Segment, com respeito aos outros no elemento Accounting.
id	Único identificador nesse tipo de atributo Segment. Esse valor pode de fato um código de conta se o tipo fosse "centro de custos".

Charge

Esse elemento especifica a quantia a ser cobrada da entidade representada pelo elemento Accounting.

Resposta a uma OrderRequest

Esta é a parte da resposta de uma transação síncrona de solicitação/resposta. O exemplo a seguir mostra uma resposta a um documento 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>
```

Como mostrado anteriormente, essa resposta é direta. Nesse caso, não há nenhum elemento denominado "OrderResponse" de fato, porque o único dado que necessita ser enviado ao solicitante é a parte Status de Response.

Response informa o solicitante que a OrderRequest foi analisada com sucesso e processada pela parte remota da conexão HTTP. Confirmações no nível do pedido — por exemplo, itens que foram entregues ou precisam ser solicitados novamente — não são comunicadas.

Transação punchout

As definições de mensagem punchout são mensagens de solicitação/resposta transportadas dentro dos elementos Request e Response. Todas as seguintes mensagens devem ser implementadas pelos fornecedores para que possam suportar o punchout.

PunchOutSetupRequest

PunchOutSetupRequest e PunchOutSetupResponse são o par solicitação/resposta usado para configurar uma sessão punchout em um sistema remoto. O cliente os utiliza para identificar o aplicativo de compra, enviar informações de configuração e receber uma resposta indicando o local a que deve ir para iniciar a sessão de pesquisa HTML no site remoto da Web.

O elemento PunchOutSetupRequest é transportado dentro do elemento Request. O exemplo a seguir mostra um documento PunchOutSetupRequest.

```
<PunchOutSetupRequest operation="create">
    <BuyerCookie>34234234ADFSDF234234</BuyerCookie>
    <Extrinsic name="department">Marketing</Extrinsic>
    <BrowserFormPost>
        <URL>http://orms.acme.com:1616/punchoutexit</URL>
    </BrowserFormPost>
    <SelectedItem>
        <ItemID>
            <SupplierPartID>54543</SupplierPartID>
        </ItemID>
    </SelectedItem>
    <SupplierSetup>
        <URL>http://workchairs.com/cxml</URL>
    </SupplierSetup>
</PunchOutSetupRequest>
```

PunchOutSetupRequest tem os seguintes atributos:

operation	Especifica o tipo de PunchOutSetupRequest: "create", "inspect" ou "edit".

Esse elemento contém também os seguintes elementos: BuyerCookie, Extrinsic, BrowserFormPost, Contact, ShipTo, SelectedItem, SupplierSetup e uma lista ItemOut. Somente o elemento BuyerCookie é exigido. A estrutura dos elementos Extrinsic, Contact e ShipTo é discutida mais detalhadamente em "OrderRequestHeader" na página 69. O elemento ItemOut é discutido em "ItemOut" na página 75. Nesse contexto (for a de um OrderRequest), os elementos Distribution e Comments e os atributos lineNumber, requisitionID e requestedDeliveryDate de um ItemOut acrescentam pouco ou nenhum valor e não devem ser incluídos. Em virtude de as sessões punchout ocorrerem antes do processamento de pedidos, essa informação não é relevante em um PunchOutSetupRequest.

Uma lista ItemOut descreve um carrinho de compras existente (itens de uma sessão punchout anterior). A operação inspect inicia uma sessão punchout somente leitura (executada tanto pelo cliente quanto pelo servidor) para exibir detalhes sobre os itens relacionados. A operação edit também se inicia no carrinho de compras anterior (descrita por meio da lista ItemOut), mas permite modificações. Suporte à operação edit implica o suporte inspect (consulte "PunchOutOrderMessageHeader" na página 83 e "Carrinhos de compra vazios" na página 83).

BuyerCookie

Esse elemento transmite informações que são obscuras ao site remoto da Web, mas deve retornar ao originador para todas as operações punchout subseqüentes. Esse elemento permite ao aplicativo de compra emparelhar múltiplas solicitações punchout em espera.

BrowserFormPost

Esse elemento é o destino dos dados em PunchOutOrderMessage. Contém um elemento URL cujo uso será posteriormente explicado na definição dePunchOutOrderMessage. Se o método Formulário URL Codificado não estiver sendo usado, esse elemento não tem de ser incluído.

Extrinsic

Esse elemento opcional contém quaisquer dados adicionais que o solicitante deseja transmitir ao site externo da Web. Esse exemplo passa o departamento do usuário, iniciando a operação punchout. A especificação cXML não define o conteúdo dos elementos Extrinsic — essa definição deve ser combinada entre cada solicitante e o site remoto da Web e implementada.

Os elementos Extrinsic destinam-se ao fornecimento de informações adicionais que possam ser lidas por máquinas. Esses elementos estendem o protocolo cXML a fim de suportar recursos não exigidos por todas as implementações. Nesse contexto, os novos dados adicionalmente descrevem o usuário, iniciando a solicitação punchout.

O elemento Extrinsic também pode aparecer nos elementos OrderRequestHeader, ItemDetail e ContractItem. Esses contextos são descritos mais pormenorizadamente em outro lugar neste documento.

SelectedItem

Esse elemento opcionalindica os itens que os usuários desejam comprar em uma sessão punchout. E contém um único ItemID exigido. Esse item permite que os usuários passem do local em que estão no catálogo ao site Web do fornecedor.

Esse elemento em geral está presente nas operações create. Aplicativos de compra que permitem ao usuário executar uma sessão punchout diretamente em uma lista do fornecedor devem excluir SelectedItem.

No caso das operações edit e inspect, SelectedItem deve aparecer somente se o usuário escolhesse retornar ao site Web do fornecedor, visualizando ao mesmo tempo novas informações no catálogo local, em vez de itens em uma requisição existente. Em ambos os casos, o carrinho de compras atual deve aparecer na lista ItemOut.

Os fornecedores podem criar catálogos de forma que SelectedItem direcione os usuários a um punchout do nível loja, seção- ou produto. Quanto mais específico for o item no catálogo, menor será o tempo de pesquisa a ser efetuada pelos usuários no site Web do fornecedor.

SupplierSetup

Esse elemento opcional especifica o URL para o qual será enviado o documento PunchOutSetupRequest. Entretanto, esse elemento não é necessário se o hub de e-commerce souber qual é o URL punchout do fornecedor.

PunchOutSetupResponse

Depois que receber um PunchOutSetupRequest, o site da Web responderá, usando um PunchOutSetupResponse, como é mostrado a seguir:

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

StartPage

Esse elemento contém um elemento URL que especifica o URL a ser transmitido ao navegador para ser iniciada a sessão punchout de pesquisa solicitada em PunchOutSetupRequest. Esse URL deve conter informações suficientes sobre estado a fim de executar a conexão com um contexto de sessão no site remoto da Web; por exemplo, identidade do solicitante e elemento BuyerCookie apropriado.

Nesse momento, o usuário que iniciou o PunchOutSetupRequest pesquisa o site externo da Web e seleciona itens a serem transferidos ao aplicativo de compra por meio de PunchOutOrderMessage.

PunchOutOrderMessage

Esse elemento envia o conteúdo da cesta remota de compras ao originador de PunchOutSetupMessage. Esse documento pode conter uma quantidade muito maior de dados que outras mensagens porque tem de mostrar o conteúdo de qualquer cesta concebível no site externo da Web. Essa mensagem não segue estritamente o modelo de solicitação/resposta.

O site remoto da Web gera um PunchOutOrderMessage quando o usuário registra a saída. Essa mensagem informa o conteúdo da cesta remota de compras ao aplicativo. Por exemplo:

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

Esses elementos são discutidos nas seções a seguir.

BuyerCookie

Esse elemento é o mesmo elemento que foi passado no PunchOutSetupRequest original. Deve ser retornado para permitir que o aplicativo de compra compare o PunchOutOrderMessage com um PunchOutSetupRequest anterior.

PunchOutOrderMessageHeader

Esse elemento contém informações sobre todo o conteúdo da cesta de compras que está sendo transferido. O único elemento exigido é Total, que é o custo geral dos itens que estão sendo acrescentados à requisição. Elementos adicionais permitidos são Shipping e Tax, os quais são a quantia e descrição de qualquer despesa de remessa ou impostos computada no site remoto da Web. ShipTo, também opcional, especifica as informações do endereço de remessa que o usuário selecionou no site remoto ou que foram transmitidas no original PunchOutSetupRequest original. Todas as quantias são apresentadas em um elemento Money, o qual sempre especifica a moeda em um formato padronizado.

PunchOutOrderMessageHeader tem o seguinte atributo:

operationAllowed	Especifica as operações PunchOutSetupRequest permitidas: create, inspect ou edit.

Esse atributo controla se o usuário poderá iniciar posteriormente transações PunchOutSetupRequest que contenham dados provenientes do PunchOutOrderMessage. Se operationAllowed="create", somente um OrderRequest posterior poderá conter esses itens. De outra maneira, o aplicativo de compra poderá inspecionar ou editar posteriormente o carrinho de compras (iniciando transações PunchOutSetupRequest subseqüentes com as operações apropriadas e os elementos ItemID correspondentes à lista ItemIn retornada nesse PunchOutOrderMessage). Suporte a edit implica suporte a inspect. O aplicativo de compra pode sempre usar os itens em um OrderRequest subseqüente.

Carrinhos de compra vazios

O documento PunchOutOrderMessage pode conter uma lista dos itens correspondentes ao carrinho de compras no site Web do fornecedor. Isso invariavelmente indica o fim da sessão punchout interativa. Os parágrafos seguintes detalham alguns casos em que não há uma lista de itens no documento PunchOutOrderMessage. Essas mensagens permitem aos clientes reiniciar imediatamente quando saem do site Web do fornecedor.

Se operação no PunchOutSetupRequest original foi inspecionada, a lista de itens de PunchOutOrderMessage deve ser ignorada pelo aplicativo de compra. O site do fornecedor não deve retornar nenhum elemento ItemIn, nesse caso. Se um PunchOutOrderMessage não contiver nenhum elemento ItemIn e a operação tiver sido criada, nenhum item deve ser adicionado à requisição. O site do fornecedor ou o usuário cancelou a sessão punchout sem criar um carrinho de compras. Se a operação foi editada e o PunchOutOrderMessage não contiver nenhum elemento ItemIn, os itens existentes nessa sessão punchout devem ser excluídos da requisição no aplicativo de compra.

O código de status "204/No Content" indica o fim da sessão, sem alteração no carrinho de compras. Novamente, o documento PunchOutOrderMessage (que é sempre necessário ao BuyerCookie) não deve conter elementos ItemIn. Esse código pode ser manipulado da mesma forma que no caso "vazio", detalhado anteriormente, a menos que a operação tenha sido editada. Nesse caso, o usuário cancelou a sessão sem efetuar modificações; portanto, nenhuma modificação deve ser feita na requisição no aplicativo de compra.

ItemIn

Esse elemento acrescenta um item contido na cesta de compras à requisição no aplicativo de compra. Contém uma série de elementos, dentre os quais apenas dois são necessários: ItemID e ItemDetail.

ItemIn tem os seguintes atributos:

quantity	Número de itens selecionados pelo usuário no site remoto da Web. Em virtude de o site do fornecedor possivelmente impor regras a unidades parciais, o protocolo permite quantidades fracionárias. Nunca ser negativo.
lineNumber (opcional)	Posição desse item no pedido. Em decorrência de as sessões punchout em geral ocorrerem antes do processamento de pedidos e de o servidor não conseguir controlar a disposição de itens em um pedido, em nenhuma situação, esse atributo não é relevante em um documento PunchOutOrderMessage.

Os elementos opcionais são ShipTo, Shipping e Tax, os quais são os mesmos elementos anteriormente especificados em PunchOutOrderMessage.

À exceção dos elementos Distribution e Comments e dos atributos requisitionID e requestedDeliveryDate disponíveis no elemento ItemOut, as estruturas ItemIn e ItemOut combinam-se entre si. O sistema de aquisição de origem pode converter diretamente uma lista ItemIn em ItemOut ao iniciar uma operação inspect ou edit. Os fornecedores podem converter uma em outra (transmitindo as extensões relacionadas, disponíveis no elemento ItemOut) no momento em que estiver sendo executada uma operação edit. O sistema de aquisição de origem pode executar a conversão direta e acrescentar informações e comentários adicionais sobre entrega e distribuição quando uma transação OrderRequest é iniciada. Dados ItemDetail (exceto, possivelmente, elementos Extrinsic) em elementos ItemIn não devem ser removidos em caso de conversão de ItemIn em ItemOut.

ItemID

Esse elemento apenas identifica o item para o site remoto da Web. Este é o único elemento que deve ser retornado ao site remoto da Web para que se possa identificar novamente o item que está sendo transferido.

ItemID contém dois elementos: SupplierPartID e SupplierPartAuxiliaryID. Somente SupplierPartID é exigido. SupplierPartAuxiliaryID auxilia o site remoto da Web a transportar configurações complexas ou informações sobre faturamento de produtos para uma nova identificação quando o item for posteriormente apresentado ao site remoto da Web.

Se SupplierPartAuxiliaryID contiver caracteres especiais (por exemplo, se contiver elementos XML adicionais não definidos no protocolo cXML), esses caracteres devem ser apropriadamente representados. Por causa da necessidade de transmitir informações SupplierPartAuxiliaryID por meio de aplicativos e retransmiti-las ao fornecedor de origem, um subconjunto interno contendo quaisquer elementos XML adicionais é suficiente.

ItemDetail

Esse elemento contém informações descritivas sobre o item que o aplicativo de compra apresenta aos usuários. O conteúdo de um elemento ItemDetail pode ser um tanto quanto complexo, mas os requisitos mínimos são simples: UnitPrice, Description, UnitOfMeasure e Classification.

No contexto de um elemento ItemIn, os elementos Extrinsic contidos em ItemDetail funcionam da mesma forma que aqueles em um Index (especificamente um IndexItemAdd).

Description

Esse elemento descreve o item usando uma forma textual. Por causa da possibilidade de esse texto exceder os limites de uma pequena tabela de itens de linha (ou outra interface de usuário comprimida) e de possíveis truncamentos aleatórios, o elemento Description contém um elemento ShortName opcional. Se isso ocorrer, os clientes devem apresentar ShortName, em vez de um truncamento do texto Description, em qualquer um dos arquivos restringidos. Os clientes devem continuar a truncar o texto Description se nenhum ShortName for fornecido.

Por exemplo:

```
<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>
```

deve aparecer como "Big Computer", se o espaço for pequeno, e "Big Computer: This wonder … lying around." (ou como dois campos distintos, mas completos) onde houver espaço para exibir uma quantidade maior de texto.

Alterações de status posteriores

Depois de concluída a transação OrderRequest, os fornecedores e os servidores intermediários podem ter de retransmitir informações adicionais ao sistema de aquisição. As transações descritas nesta seção são usadas com esse objetivo. Essas transações compartilham alguns elementos e semântica.

Da mesma forma que um OrderRequest (consulte "Resposta a uma OrderRequest" na página 78), nenhuma dessas transações contém um elemento Response específico. Em vez disso, o documento retornado contém um documento Response quase vazio (há somente um Status). Os documentos retornados têm a seguinte forma:

```
<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>
```

O código retornado será "200" somente se a operação for completada com sucesso.

DocumentReference

O elemento DocumentReference contém informações suficientes para vincular a solicitação atualizada a um determinado documento. Esse processo repete um atributo necessário do documento anterior e acrescenta um identificador opcional gerado pelo fornecedor. Por exemplo:

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

DocumentReference não contém nenhum elemento, mas os seguintes atributos:

payloadID	Um número único relacionado com o espaço e o horário usados para registro e identificação de documentos. Esse valor não deve alterar no caso de novas tentativas. A implementação recomendada é: datetime.process id.random number@hostname É tirada diretamente do elemento cXML do documento OrderRequest.
orderID	Identificador desse pedido. É o número da ordem de compra atual. É tirado diretamente de OrderRequestHeader do documento OrderRequest.

StatusUpdateRequest

Essatransação informa um ponto de interconexão mais recente de alterações no status de processamento de um pedido. Uma das alterações tem um significado especial: Quando um hub intermediário transmite com sucesso um OrderRequest avançado, é possível informar o remetente original ou um hub anterior de que a transmissão foi bem-sucedida. Transições entre várias filas e etapas de processamento no lado do fornecedor ou no hub podem também ser significativas para o comprador.

Essa solicitação atualiza o status de processamento de um único documento OrderRequest. Por exemplo:

```
<cXML version="1.1.007" xml:lang="en-US"
        payloadID="0c30050@supplier.com"
        timestamp="2000-01-08T23:00:06-08:00">
    <Header>
        Routing, identification and authentication information.
    </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>
```

Essa solicitação contém somente um DocumentReference e um elemento Status. Ambos são exigidos. Status pode transmitir um erro anterior de transporte encontrado por um hub intermediário. A semântica desse elemento é idêntica à de um Status que possivelmente tenha sido retornado na resposta HTTP inicial a um documento OrderRequest.

O código 200/OK é especialmente importante quanto os documentos são armazenados e enviados. Esse código indica que o fornecedor iniciou o processamento de OrderRequest ou que um hub enviou o documento. O receptor deve esperar documentos StatusUpdateRequest posteriores depois que o código200/OK surgir.

Fornecedores e hubs que estiverem usando a transação StatusUpdate devem retornar o código 201/Accepted quando um OrderRequest estiver na fila para processamento posterior. Depois de enviar 200/OK (em uma resposta imediata a um OrderRequest ou em um StatusUpdateRequest posterior), o servidor não deve enviar nenhuma transação StatusUpdate para esse pedido. Erros posteriores no processamento devem ser considerados uma exceção a essa regra.

Definições de catálogo

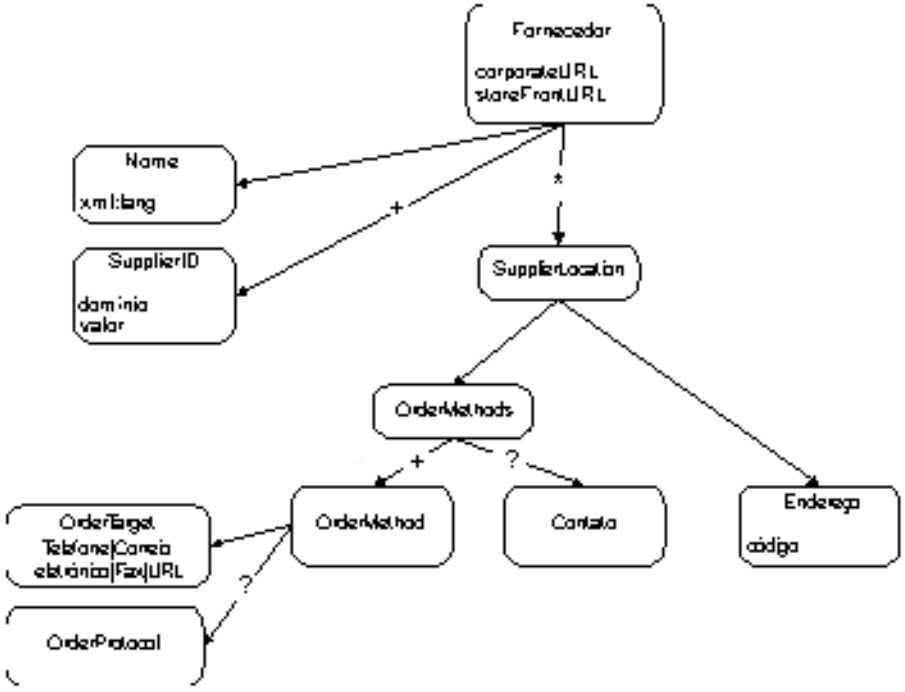
As definições cXML de catálogo consistem em três elementos principais: Supplier, Index e Contract. Esses três elementos descrevem dados que serão usados constantemente ou armazenados em cache em um hub ou em um sistema de aquisição de uma organização de compra.

- Supplier: Contém dados básicos sobre o fornecedor, tais como endereço, contato e informações sobre o processamento de pedidos.
- Index: Descreve dados sobre o estoque de produtos e serviços do fornecedor, tais como descrição, códigos dos itens e códigos de classificação.
- Contract: Descreve dados sobre aspectos flexíveis do estoque negociados pelo comprador e fornecedor; por exemplo: preço.

Observe que Index usa vários subelementos para descrever itens de linha no respectivo estoque dos fornecedores. Os fornecedores podem enviar informações sobre preço para serem armazenas em cache no sistema do comprador ou informações punchout para habilitar os compradores a entrar em um site remoto da Web para pesquisar preços ou obter outras informações.

Supplier

O elemento Supplier inclui um fornecedor de produtos e serviços designado. Esse elemento deve ter os elementos Name e SupplierID. Supplier também descreve informações adicionais de endereço e processamento de pedidos ao fornecedor:



Supplier tem os seguintes atributos:

corporateURL (opcional)	URL para o site Web do fornecedor.
storeFrontURL (opcional)	URL para site da Web destinado a compras ou pesquisa.

O exemplo a seguir mostra um perfil do elemento 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

Alguns fornecedores conduzem negócios em mais de uma localização. Um elemento SupplierLocation pode ser usado para cada localização. Esse elemento também inclui a forma como aquela localização faz negócios ou os tipos de pedidos aceitáveis. Um elemento SupplierLocation contém um Address e um conjunto de OrderMethods.

OrderMethods e OrderMethod

O elemento OrderMethods é um grupo de um ou mais elementos OrderMethod para um determinado elemento SupplierLocation. A posição de OrderMethods na lista é significativa — o primeiro elemento é o método de processamento de pedidos preferido, o segundo, a próxima prioridade, e assim por diante, em ordem decrescente de preferência.

OrderMethod inclui informações sobre o processamento de pedidos segundo o destino do pedido (por exemplo, telefone, fax ou URL) e um protocolo opcional para posteriormente esclarecer expectativas em relação ao processamento de pedidos no destino determinado; por exemplo, "cxml" para um destino de URL.

Index

Esse elemento é o elemento-raiz para atualização de catálogos nos sistemas de aquisição das organizações de compra.

Um elemento Index é associado com um único fornecedor. O elemento Index fornece uma lista de códigos de fornecedor, na qual cada código representa o respectivo fornecedor.

Index contém um ou mais elementos IndexItem, bem como um conjunto opcional de elementos SearchGroup, para a definição de dados paramétricos de pesquisa em relação aos itens. O elemento IndexItem contém elementos que acrescentam ou excluem itens do catálogo armazenado em cache da organização de compra. O exemplo a seguir mostra um esboço de um elemento Index:

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

IndexItem, IndexItemAdd, IndexItemDelete e IndexItemPunchout

O elemento IndexItem é o recipiente da lista de itens em um índice. IndexItem contém três tipos de elemento:

- IndexItemAdd: Insere um novo item ou atualiza um item existente no índice. Esse elemento contém os elementos ItemID, ItemDetail e IndexItemDetail.
- IndexItemDelete: Remove um item do índice. Esse elemento contém um elemento ItemID para identificação do item.
- IndexItemPunchout: Insere um item para iniciar a sessão punchout no site Web do fornecedor. Esse elemento contém os elementos PunchoutDetail e ItemID. É similar a um elemento IndexItemAdd, com a exceção de que não exige informações sobre preço. Os compradores obtêm detalhes sobre o item em tempo real no site Web do fornecedor.

ItemID

O elemento ItemID apenas identifica itens do fornecedor. Contém o elemento SupplierPartID e um elemento SupplierPartAuxiliaryID opcional.

SupplierPartAuxiliaryID

Se SupplierPartID não identificar unicamente o item, o fornecedor deve usar SupplierPartAuxiliaryID para especificar uma chave "auxiliar" que identifique apenas o item quando combinada com SupplierID e SupplierPartID. Por exemplo, um fornecedor provavelmente usaria o mesmo SupplierPartID para um item, mas diferentes preços para unidades de "EA" e "BOX". Nesse caso, In this case, um SupplierPartAuxiliaryID razoável para os dois itens seria "EA" e "BOX".

SupplierPartAuxiliaryID pode também ser usado como um cookie de fornecedor, o que habilita o fornecedor a se referir a dados de configuração complexa ou sobre itens. Pode ainda conter todos os dados de que o fornecedor necessita para reconstruir o que é o item em questão o sistema de computador (uma cesta ou cookie de dados que faça sentido apenas ao fornecedor). Para obter mais informações, consulte "Cookies do comprador e do fornecedor" na página 38.

ItemDetail

ItemDetail contém informações detalhadas sobre um item ou todos os dados, além dos dados essenciais representados em ItemID, que um usuário possa querer ver sobre um item. Esse elemento deve conter UnitPrice, UnitOfMeasure, um ou mais elementos Description e Classification, e pode, opcionalmente, conter ManufacturerPartID, ManufacturerName, URL e qualquer quantidade de elementos Extrinsic. Para obter mais informações, consulte "ItemDetail" na página 85.

No contexto de IndexItemAdd, os elementos Extrinsic divulgam informações sobre um determinado item. Essas informações não devem ser transmitidas a um fornecedor em um documento OrderRequest, porque o fornecedor pode recuperar os mesmos dados, usando o ItemID exclusivo.

IndexItemDetail

O elemento IndexItemDetail contém elementos específicos do índice que definem outros aspectos de um item, como LeadTime, ExpirationDate, EffectiveDate, SearchGroupData ou TerritoryAvailable.

PunchoutDetail

PunchoutDetail é similar a ItemDetail, com a exceção de que exige apenas um ou mais elementos Description e Classification. Pode também conter URL, ManufacturerName, ManufacturerPartID, ExpirationDate, EffectiveDate, SearchGroupData, TerritoryAvailable e elementos Extrinsic. Não contém informações sobre preço, tempo de avanço ou unidades de medida.

Contract

O elemento Contract representa um contrato entre o fornecedor e o comprador, para produtos e serviços descritos no índice do fornecedor. O contrato permite que o fornecedor "sobreponha" valores negociados com o comprador a determinados atributos do item (como preços) no índice. Além disso, permite que fornecedores e compradores segmentem essas sobreposições com base em uma "chave de segmento" combinada entre as partes, significativa para a organização de compra, como nome de uma planta ou de um centro de custos.

Contract tem os seguintes atributos:

effectiveDate	Data e duração reais do contrato, no formato ISO 8601.
expirationDate	Data de expiração e duração do contrato, no formato ISO 8601.

Contract contém um ou mais elementos ItemSegment; por exemplo:

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

ItemSegment

O elemento ItemSegment é o recipiente de uma lista de elementos ContractItem para um determinado "segmento", na qual um segmento representa uma divisão arbitrária de itens do contrato com base em uma chave de segmento combinada entre fornecedor e comprador.

ItemSegment tem o seguinte atributo:

segmentKey (opcional)	Caracteres combinados usados para segmentar preços personalizados.

ContractItem

Esse elemento é um item particularmente usado para sobrepor um item do índice. Contém um ItemID que apenas identifica o item do índice no sistema de aquisição para efetuar a sobreposição. Pode conter qualquer quantidade de elementos Extrinsic que compreendam o valor sobreposto relativo ao atributo do item do índice designado.

Definições de gerenciamento de assinaturas

Intermediários, como os hubs de redes de comércio eletrônico, podem gerenciar os fornecedores e os catálogos dos fornecedores usados pelos sistemas de aquisição das organizações de compra. Esses intermediários podem fornecer conexões diretas entre os sistemas de aquisição e os sistemas de fornecimento. Esta seção contém definições de elementos para gerenciamento de dados sobre fornecedores e o conteúdo dos catálogos. Essas definições baseiam-se em várias definições anteriores sobre solicitação/resposta cXML, mensagens unidirecionais e catálogos.

Dados sobre o fornecedor

As definições sobre gerenciamento de dados do fornecedor consistem principalmente nos elementos SupplierListRequest, SupplierListResponse, SupplierDataRequest, SupplierDataResponse e SupplierChangeMessage. Esses elementos são descritos a seguir com exemplos cujo intermediário é a Ariba Network.

SupplierListRequest

SupplierListRequest solicita uma lista de fornecedores com os quais o comprador estabeleceu relações comerciais.

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

SupplierListResponse

SupplierListResponse relaciona os fornecedores com os quais o comprador estabeleceu relações comerciais.

```
<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

SupplierDataRequest solicita dados sobre o fornecedor.

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

SupplierDataResponse

SupplierDataResponse contém dados sobre o fornecedor.

```
<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>
        </Supplier>
    </SupplierDataResponse>
</Response>
```

SupplierChangeMessage

Esse elemento destina-se à notificação de alterações nos dados do fornecedor.

```
<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>
```

Assinaturas de catálogos

As definições sobre gerenciamento de assinaturas de catálogos são descritas a seguir. Nos exemplos, o intermediário a Ariba Network.

Subscription

Esse elemento captura metadados sobre uma única assinatura de catálogo. Seus subelementos incluem:

- InternalID: código interno exclusivo do intermediário
- Name: nome da assinatura
- ChangeTime: a data e o horário, quando nenhum dado sobre a assinatura mencionada foi alterada
- SupplierID: código do fornecedor que está fornecendo o catálogo
- Format: formato do catálogo
- Description: descrição do catálogo

```
<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

Esse elemento solicita a lista atual de assinaturas de catálogos do comprador.

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

SubscriptionListResponse

Esse elemento relaciona as listas atuais de assinaturas de catálogos do comprador.

```
<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

Esse elemento solicita o conteúdo de um catálogo assinado. A solicitação inclui InternalID e SupplierID do catálogo.

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

SubscriptionContentResponse

Esse elemento contém o conteúdo de um catálogo. O formato do catálogo pode ser Catalog Interchange Format (CIF) ou xCML. Ser for CIF, será codificado usando-se base 64 e incluído como conteúdo de um elemento CIFContent. Se for cXML, os elementos Index e Contract serão incluídos diretamente.

```
<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

Esse elemento informa o sistema de aquisição da organização de compra de que o catálogo assinado foi alterado.

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

SubscriptionChangeMessage tem o seguinte atributo:

type	Refere-se ao tipo de alteração: new, delete ou update.

Definições de recuperação de mensagens

Algumas organizações de compra não têm pontos de entrada HTTP para receber mensagens cXML provenientes de fora das firewalls da empresa. A especificação cXML reconhece esses ambientes.

Esta seção apresenta as definições que reconhecem sistemas de origem para posicionar mensagens em fila de espera quando o destino não está apto a aceitar diretamente transmissões HTTP. Em vez disso, o destino extrai as mensagens como for conveniente.

GetPendingRequest

Esse elemento extrai um conjunto de mensagens que estão aguardando o solicitante. O elemento MessageType e os atributos lastReceivedTimestamp e maxMessages controlam o tipo e a contagem das mensagens recuperadas.

lastReceivedTimestamp (opcional)	O registro de horário das mensagens recebidas mais recentemente.
maxMessages (opcional)	Número máximo de mensagens em uma única resposta que o solicitante pode manipular.

Depois de receber a solicitação, o receptor retorna as mensagens mais antigas, dos tipos especificados, com registros de horário iguais ou posteriores aos registros especificados. Se houver múltiplas mensagens que atendam a esse critério, múltiplas mensagens poderão ser retornadas, dependendo do atributo maxMessages. O sistema de enfileiramento descarta todas as mensagens pendentes do tipo de mensagem especificado com registros de horário anteriores ao registro especificado.

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

GetPendingResponse

Esse elemento contém uma ou mais mensagens esperando pelo solicitante.

```
<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>
```

Apêndice B
Novos recursos em cXML 1.1

A cXML 1.1 contém as seguintes categorias de novos recursos:

- Alterações gerais em cXML
- Alterações em elementos Extrinsic
- Melhorias na transação punchout
- Novos recursos de ordem de compra
- Nova transação de status de ordem de compra

Para uma análise aprofundada de qualquer elemento ou atributo aqui mencionado, consulte Apêndice A, "Especificação da linguagem cXML."

Alterações gerais em cXML

As alterações descritas a seguir afetam grandes áreas da linguagem cXML. Esses aperfeiçoamentos tornam a linguagem cXML mais fácil de ser usada internacionalmente e são responsáveis pela compatibilidade da versão cXML. Além disso, uma nova transação de perfil possibilita aos clientes da cXML consultar as capacidades do servidor cXML.

Suporte aperfeiçoado a vários idiomas

Por motivo de consistência e para oferecer um suporte ainda melhor a vários idiomas, os elementos cXML, Status e ManufacturerName agora contam com um xml opcional :lang.

Para obter mais informações, consulte:

"Status" na página 57

Exemplo:

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

Esse atributo especifica o idioma que os clientes da cXML devem usar em respostas, e que os sites punchout devem exibir aos usuários.

DTDs centralizadas

Anteriormente, não havia nenhuma localização central estipulada para DTDs em cXML; portanto, os analisadores cXML não podiam recuperá-los automaticamente. Atualmente, as DTDs para todas as versões cXML estão disponíveis em localizações consistentes em cxm1.org.

Para encontrar as DTDs em cXML, consulte:

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

Para obter mais informações, consulte:

"Validação comparando com DTDs" na página 7

onde<*version*> corresponde ao número completo da versão cXML; por exemplo, 1.1.007.

Para obter o melhor desempenho possível, os clientes da cXML não devem recuperar DTDs toda vez que analisarem documentos cXML, mas devem armazená-las localmente. Depois de preencher o URL abaixo //xml.cxml.org/schemas/cXML, não há necessidade de alterar a localização. As DTDs nunca serão alteradas, porém novos desvios serão acrescentados.

Nova transação de perfil

Uma nova transação de perfil transmite informações básicas sobre servidores cXML. Essa transação consiste em dois novos documentos denominados ProfileRequest e ProfileResponse. Essa transação recupera a capacidade do servidor, incluindo a versão cXML compatível, transações compatíveis e opções para essas transações.

Para obter mais informações, consulte:

"Transação Perfil" na página 67

Nota: Todos os servidores cXML 1.1 **devem** suportar essa transação.

Os clientes podem ainda usar a transação de perfil para "fazer ping" dos servidores e verificar se estão disponíveis.

ProfileRequest

Não há conteúdo no documento ProfileRequest. Ele simplesmente é encaminhado ao servidor cXML apropriado, usando as credenciais Header.

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

O servidor responde com um documento ProfileResponse descrito abaixo.

ProfileResponse

O documento ProfileResponse relaciona as transações suportadas pelo servidor cXML, sua localização e quaisquer opções designadas por um valor de caracteres.

```
<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>
```

Novos códigos de status

Para obter mais informações, consulte:

"Status" na página 57

A especificação cXML 1.1 inclui novos códigos de status de transação com o objetivo de possibilitar uma comunicação mais precisa entre cliente e servidor. Além disso, a especificação contém descrições melhores de códigos de status HTTP e cXML existentes.

Novo tipo de atributo para membros do mercado

O elemento Credential tem um novo atributo type, que especifica se o emissor ou receptor é membro de um mercado. É possível haver múltiplos mercados, e cada mercado pode ter diferentes exigências em relação a credenciais.

Para obter mais informações, consulte:

"Credential" na página 56

O novo atributo type tem um único valor possível: marketplace. Use-o para diferenciar credenciais destinadas aos membros do mercado de credenciais destinadas a um comprador normal ou empresas fornecedoras. Elementos Credential sem um atributo type identificam empresas não associadas a um mercado.

Solicitações a um mercado ou provenientes dele devem identificar o mercado e a empresa-membro nos elementos To or From Credential.

Alterações em elementos Extrinsic

cXML 1.1 introduz outros elementos e atributos a dados previamente contidos em elementos Extrinsic. Essas adições movem as informações que foram enviadas anteriormente, usando Extrinsic dentro da especificação-base.

Além disso, o cXML 1.1 passa a suportar Extrinsic no nível do cabeçalho.

Novo elemento Contact

Os elementos OrderRequestHeader, PunchOutSetupRequest e ItemOut podem conter elementos Contact opcionais, que relacionam uma pessoa ou um grupo para obtenção de mais informações.

Para obter mais informações, consulte:

"Contact" na página 72

```
<ItemOut quantity="1">
    <ItemID>
        <SupplierPartID>5555</SupplierPartID>
    </ItemID>
    <ItemDetail>
        <UnitPrice>
            <Money currency="USD">134.00</Money>
        </UnitPrice>
        <Description xml:lang="en">
            <ShortName>Office Chair</ShortName>
            Black leather, with adjustable arms, adjustable height and back angle.
        </Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">12345</Classification>
    </ItemDetail>
```

```
<Contact role="customerService">
    <Address>
        <Name xml:lang="en-US">Joe Bob Emmet</Name>
        <Email>joebob@workchairs.com</Email>
        <Phone name="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>
```

O atributo role especifica a posição ou cargo da pessoa para contato. As funções permitidas são endUser, administrator, purchasingAgent, technicalSupport, customerService ou sales.

Atributo requisitionID suportado

O atributo requisitionID passou a ser totalmente suportado. É um atributo opcional para elementos OrderRequest e ItemOut que identifica a requisição de um item de linha do comprador.

Para obter mais informações, consulte:

"OrderRequestHeader" na página 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>
```

Não inclua requisitionID no nível ItemOut se for usá-lo no nível OrderRequestHeader.

Resumo das informações de elementos Extrinsic que foram movidas

O quadro a seguir relaciona as alterações feitas nos elementos Extrinsic geralmente usados.

Elementos Extrinsic antigos	Elementos ou atributos cXML novos ou existentes
Requested Ship Date	atributo ItemOut requestedDeliveryDate
ship complete	atributo shipComplete novo
ReqNumber	atributo requisitionID existente
Requisition #	atributo requisitionID existente
Name	elemento Contact novo
Phone	elemento Contact novo
E-mailAddress	elemento Contact novo
Buyer Name	elemento Contact novo
Buyer Phone	elemento Contact novo
OriginalRequester	elemento Contact novo
Requester Phone Number	elemento Contact novo
ETA	atributo requestedDeliveryDate existente

Elementos Extrinsic no nível de cabeçalho

Anteriormente, os elementos Extrinsic das ordens de compra podiam aparecer somente no nível de item de linha de produto. Agora, podem aparecer em qualquer lugar no documento OrderRequest.

Use esse novo recurso para dados Extrinsic que se apliquem à ordem de compra inteira.

Elementos Extrinic com nomes idênticos não podem aparecer ao mesmo tempo no nível de cabeçalho e no nível de item de linha, em um documento OrderRequest.

Melhorias na transação punchout

As transações punchout foram melhoradas para suportar melhor compras do nível de seção e do nível de produto. Atualmente, cXML também suporta sessões punchout que foram canceladas.

PunchOutSetupRequest melhorado

O documento PunchOutSetupRequest enviado pelos compradores foi alterado para otimizar a flexibilidade das transações punchout. O URL especificado em PunchOutSetupRequest foi desaprovado; no futuro, os servidores cXML ignorarão esse elemento.

A nova metodologia usa a identidade do fornecedor (presente em Credential). Os hubs de rede de e-commerce recebem o documento PunchOutSetupRequest, lêem o código do fornecedor, encontram o URL do site punchout, usando as informações da conta do fornecedor, e enviam o documento PunchOutSetupRequest ao URL. Os hubs de rede de-commerce — e não o comprador — especificam o URL do site punchout, que é mais flexível.

Além disso, os hubs de rede de e-commerce permitem aos fornecedores armazenar URLs dos respectivos sites punchout.

Elemento SelectedItem

Como parte da otimização de PunchOutSetupRequest, cXML passou a ter um suporte mais adequado para punchout dos níveis loja, seção e produto. Um novo elemento SelectedItem opcional nesse documento possibilita que os fornecedores especifiquem o punchout para toda uma loja ou qualquer subconjunto de ofertas de produtos. As aplicações de compra podem incluir o elemento SelectedItem em documentos PunchOutSetupRequest e os sites punchout podem usá-lo para determinar quais produtos serão expostos aos usuários. Se não houver nenhum SelectedItem, os fornecedores devem apresentar todos os produtos oferecidos (nível loja).

Para obter mais informações, consulte:

"SelectedItem" na página 80

Um SelectedItem contém um elemento ItemOut, que contém um ItemID; por exemplo:

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

No caso do conteúdo do elemento SelectedItem, as aplicações de compra usam o ItemID (SupplierPartID e o SupplierPartAuxiliaryID) de um catálogo punchout indexado. Não é exigida nenhuma modificação no catálogo.

As aplicações de compra devem inicialmente enviar tanto o novo elemento SelectedItem quanto o antigo URL punchout em PunchOutSetupRequest. Os hubs de rede de e-commerce usam o URL antigo somente para os fornecedores que ainda não armazenaram o respectivo destino do URL punchout.

PunchOutOrderMessage vazio

Atualmente, cXML permite documentos PunchOutOrderMessage vazio, que possibilitam aos usuários finalizar as sessões de compra punchout, sem selecionar nenhum item. Anteriormente, o documento PunchOutOrderMessage retornado tinha de conter pelo menos um item.

Os fornecedores podem inserir um botão "Cancelar" para criar um documento PunchOutOrderMessage vazio. Dessa forma, tanto o site punchout quanto a aplicação de compra sabem quando o usuário cancelou uma sessão de compra, o que lhes possibilita excluir um carrinho de compras, excluir itens da requisição e executar outras tarefas de manutenção.

cXML-base64: um novo campo oculto

O novo campo oculto cXML-base64 suporta documentos internacionais dentro do FORM POST retornado pelos sites punchout. Os documentos cXML que contêm símbolos que não pertençam a "us-ascii" devem usar esse campo, em vez de o campo oculto cXML-urlencoded. A semântica dessa alternativa é quase idêntica, mas todo o documento é codificado no formato base 64 ao longo do transporte, e não no formato de codificação HTML para o navegador ou no formato de codificação URL para o servidor da Web receptor.

Para obter mais informações, consulte:

"Codificação de formulário URL" na página 62

A codificação Base64 mantém a codificação original de caracteres de um documento cXML. Embora nenhum parâmetro "charset" apareça nas informações transmitidas, o documento decodificado (depois que a codificação de transferência é removida) pode ser tratado como um tipo de mídia "application/xml". Esse tratamento permite que o analisador receptor aceite qualquer atributo de "codificação" especificado na declaração XML. No caso desse campo, e de todos os documentos "application/xml", a codificação de caracteres padrão é UTF-8.

Novos recursos de ordem de compra

Os documentos cXML de ordem de compra foram melhorados para suportar recursos que foram solicitados.

Novo atributo lineNumber

lineNumber é um novo atributo opcional para o elemento ItemOut. Esse atributo especifica a localização de um item na ordem de compra do comprador. É escolhido assim que o pedido é aplicado (portanto, em geral não é relevante para sessões punchout).

Para obter mais informações, consulte:

"ItemOut" na página 75

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

Números de linha de itens devem permanecer constantes quando as ordens forem modificadas a fim de identificar a modificação.

Anexos de ordem de compra

Os compradores em geral precisam esclarecer as ordens de compra, usando memorandos, esboços ou fax. As aplicações de compra podem agora anexar arquivos de qualquer tipo a ordens de compra cXML, usando MIME (Multipurpose Internet Mail Extensions).

cXML contém apenas referências a itens MIME externos, enviados dentro de um envelope MIME contínuo (com o documento cXML, em uma mensagem de correio eletrônico ou fax).

O novo elemento Attachment contém referência aos anexos:

Para obter mais informações, consulte:

"Transmissão de anexos" na página 53

```
<Comments>
    <Attachment><URL>cid: uniqueCID@buyer.com</Attachment>
    Please see attached image for my idea of what this should look like
</Comments>
```

Os hubs de rede de e-commerce recebem os anexos e podem enviá-los ao fornecedor ou armazená-los para recuperação on-line.

Para obter mais informações sobre padrão MIME, consulte os seguintes sites da Web:

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

Novo atributo shipComplete

shipComplete é um novo atributo opcional para o elemento OrderRequestHeader que instrui os fornecedores a registrar um pedido somente quando todos os itens estiverem disponíveis. Esse atributo evita a entrega parcial de itens.

Para obter mais informações, consulte:

"OrderRequestHeader" na página 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>
```

Novo elemento ShortName

ShortName é um novo elemento opcional dentro dos elementos Description.

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

Para obter mais informações, consulte:

"ItemDetail" na página 85

ShortName é um nome abreviado (é recomendável 30 caracteres e no máximo 50 caracteres) de um item, que se ajusta às listas apresentadas aos usuários. Anteriormente, pelo motivo de haver apenas um elemento Description, descrições longas eram truncadas em pontos desconhecidos.

As aplicações de compra e outros clientes da cXML deveriam exibir ShortName, em vez de um texto Description truncado em qualquer campo cujo espaço é limitado. Se não for fornecido nenhum ShortName, os clientes da cXML podem continuar a truncar o texto Description.

Os criadores de catálogos não deveriam usar ShortName para duplicar as informações em Description. Em vez disso, deveriam usar ShortName para designar produtos e Description, para detalhá-los.

O formato de catálogo CIF 3.0 também foi otimizado para suportar ShortName. O nome do campo CIF é Short Name.

Nova transação de status de ordem de compra

cXML 1.1 introduz uma nova transação para enviar o status de ordens de compra a hubs de rede de e-commerce.

Essa transação depende do novo elemento OrderReference, que associa a atualização do status com o OrderRequest recebido mais recentemente do comprador.

Novo elemento OrderReference

O novo elemento OrderReference associa a atualização do status com um determinado documento OrderRequest. Esse elemento repete os atributos necessários dos elementos OrderRequestHeader e cXML do documento OrderRequest e acrescenta um identificador opcional criado pelo fornecedor.

Para obter mais informações, consulte:

"DocumentReference" na página 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"
</OrderReference>
```

A nova transação StatusUpdateRequest usa esse elemento.

Nova transação StatusUpdateRequest

Outros assistentes de processamento de ordens (por exemplo, fax ou provedores de serviços EDI) enviam a nova transação StatusUpdateRequest a hubs de rede de e-commerce para configurar o status das ordens. Isso afeta o indicador de status do pedido no hub, que é exibido tanto para os compradores quanto para os fornecedores. Além disso, os fornecedores podem enviar essa transação para permitir que os compradores visualizem o status do processamento do documento na organização do fornecedor.

Para obter mais informações, consulte:

"StatusUpdateRequest" na página 87

> **Nota:** Essa transação informa às partes interessadas sobre alterações na entrega e no status do processamento de *documentos de ordem de compra*, e não no status de remessa de itens atuais.

Uma alteração tem um significado especial: Quando um hub intermediário envia com sucesso um documento OrderRequest, é possível informar ao emissor original ou ao hub anterior que a transação foi bem-sucedida. Transações entre várias filas e etapas de processamento no lado do fornecedor ou no hub podem também ser significativas para o comprador.

Novo elemento Followup

Followup é um elemento dentro de OrderRequestHeader que especifica o URL para o qual documentos StatusUpdateRequest posteriores devem ser enviados. Essa é a localização de entrada para quaisquer documentos posteriores que mencionem o documento OrderRequest.

Para obter mais informações, consulte:

"Followup" na página 74

Índice

A

anexos de ordens de compra 45
anexos MIME 45, 53
atributo code 57
atributo corporateURL 89
atributo deploymentMode 61
atributo domain 56
atributo effectiveDate 68, 93
atributo expirationDate 93
atributo id 77
atributo inReplyTo 61
atributo lastReceivedTimestamp 102
atributo lineNumber 75, 84
atributo maxMessages 102
atributo operation 23, 79
atributo operationAllowed 83
atributo orderDate 71
atributo orderID 71, 87
atributo payloadID 51, 87
atributo quantity 75, 84
atributo requestedDeliveryDate 75
atributo requestName 69
atributo requisitionID 71, 75
atributo role 73
atributo segmentKey 94
atributo shipComplete 71
atributo storeFrontURL 89
atributo timestamp 51
atributo type 56
atributo version 51

C

codificação de caractere 52
codificação de formulário 63
codificação de formulário HTML 63
codificação, caractere 52
comprador e fornecedor, cookies de 28, 38
cookies, comprador e fornecedor 28, 38
cotando pedidos 18
cXML-base64, campo oculto 36, 64
cXML-urlencoded, campo oculto 36, 63

D

deploymentMode 57
DTDs (Document Type Definitions) 7

E

EDI (X.12 Electronic Data Interchange) 4
editores para XML 9
Elemento accounting 77
Elemento attachment 73
Elemento BillTo 72
Elemento BrowserFormPost 80
Elemento BuyerCookie 80, 82
Elemento charge 77
Elemento Classification 21
Elemento comments 73
Elemento contact 72
Elemento contract 93
Elemento credential 56
Elemento cXML 50
Elemento Description 21
Elemento description 85
Elemento distribution 77
Elemento DocumentReference 86
Elemento Extrinsic 39
Elemento extrinsic 74, 80

Elemento followup 74
Elemento GetPendingRequest 102
Elemento GetPendingResponse 103
Elemento header 54
Elemento index 91
Elemento IndexItemAdd 92
Elemento IndexItemDelete 92
Elemento IndexItemDetail 93
Elemento IndexItemPunchout 92
Elemento IsoCountryCode 65
Elemento IsoLanguageCode 65
Elemento ItemDetail 85, 92
Elemento ItemID 85
Elemento ItemIn 84
Elemento ItemOut 75
Elemento ItemSegment 94
Elemento message 61
Elemento OrderMethods 91
Elemento OrderRequest 69
Elemento OrderRequestHeader 69
Elemento payment 72
Elemento ProfileRequest 67
Elemento ProfileResponse 67
Elemento PunchoutDetail 93
Elemento PunchOutOrderMessage 82
Elemento PunchOutOrderMessageHeader 83
Elemento PunchOutSetupRequest 79
Elemento PunchOutSetupResponse 81
Elemento request 57
Elemento response 57
Elemento SelectedItem 80
Elemento shipping 72
Elemento ShipTo 72
Elemento ShortName 85
Elemento StartPage 81
Elemento status 57
Elemento StatusUpdateRequest 87
Elemento subscription 98
Elemento SubscriptionContentRequest 100
Elemento SubscriptionContentResponse 100
Elemento SubscriptionListRequest 99
Elemento SubscriptionListResponse 99
Elemento supplier 89
Elemento SupplierChangeMessage 97
Elemento SupplierDataRequest 96
Elemento SupplierDataResponse 96
Elemento SupplierListRequest 95
Elemento SupplierListResponse 95
Elemento SupplierLocation 90
Elemento SupplierPartAuxiliaryID (cookie do fornecedor) 28, 38, 92
Elemento SupplierSetup 81
Elemento tax 72
Elemento total 71
Elemento transaction 68
Elemento URL 66
Elemento xmllanguageCode 66
Elementos from, to e sender 55
Elementos sender, to e from 56
Elementos to, from e sender 55
Extrinsic, elemento 26

F

fazendo ping de servidores com a Transação de perfil 8
ferramentas para trabalhar com XML 9
formato data e horário 52
formato horário e data 52
formulário codificado 36
Formulário HTML codificado 36
fornecedor e comprador, cookies de 28, 38
From, elemento 23

I

idioma, no cabeçalho cXML 30

L

localidade, no cabeçalho cXML 30

O

ordens de compra 41–45
  anexos 45

P

Página de início 29
Página do receptor do pedido 37
Página do remetente 34
Página inicial 33
payloadID, atributo 23
PunchOutOrderMessage 27
PunchOutSetupRequest 22
PunchOutSetupResponse 26

R

registrando pedidos 19

S

SelectedItem, elemento 25
Sender, elemento 23
Site da Web cxml.org 8
SupplierID, elemento 21

T

timestamp, atributo 23
To, elemento 23
Transação de perfil 8

U

Unidade de medida 29
Unidades de medida 66
URL codificado 63
URL SupplierSetup 25
utilitários para serem usados com XML 9

V

validando a cXML 7

X

XML 15
xml:lang 30

www.cxml.org