



Guía del usuario de cXML

VERSIÓN 1.1

ABRIL DE 2000

Por el presente documento, Ariba, Inc. (en lo sucesivo Ariba) le concede la licencia y el derecho mundiales, libres de regalías (royalties), perpetuos y no exclusivos para utilizar la especificación cXML (en lo sucesivo la "Especificación") en virtud de los derechos de autor de Ariba respecto a la misma para su uso, copia, publicación, modificación y distribución. Además, Ariba le concede una licencia libre de regalías conforme a los derechos de propiedad intelectual de Ariba pertinentes para implementar y utilizar las directrices de esquema y etiquetas cXML incluidas en la Especificación con el fin de crear programas informáticos que cumplan dichas directrices. Una condición de esta licencia será su conformidad respecto a no hacer valer ningún derecho de propiedad intelectual en contra de Ariba y otras compañías en relación a su implementación de la Especificación. Ariba se reserva expresamente todos los demás derechos que pueda tener sobre la materia y el contenido de esta Especificación. Asimismo, Ariba renuncia expresamente a todas y cualesquiera garantías respecto a esta Especificación, incluidas las garantías de que esta Especificación o implementaciones de la misma no infringen los derechos de otras personas. Esta Especificación se proporciona "tal cual" sin ninguna garantía expresa o implícita. Si publica, copia o distribuye esta Especificación, deberá adjuntar este aviso de copyright; sin embargo, si modifica esta Especificación, el nombre de la especificación modificada no deberá incluir el término "cXML". Si envía comentarios o sugerencias a Ariba, y Ariba modifica la especificación cXML basándose en sus aportaciones, Ariba tendrá la propiedad de la versión de cXML modificada.

La información de este documento está sujeta a cambios sin previo aviso.

Contenido

Prólogo.....	v
Destinatarios y requisitos previos.....	v
Capítulos recomendados.....	vi
 Capítulo 1	
Introducción a cXML.....	1
Funciones de cXML.....	1
Catálogos.....	2
Punchouts.....	3
Pedidos de compra.....	4
Tipos de aplicaciones que utilizan cXML.....	5
Aplicaciones de adquisición.....	5
Plataformas de red de comercio.....	5
Catálogos de punchout.....	6
Sistemas de recepción de pedidos.....	6
Estrategia de entrega de contenido.....	7
Validación respecto a definiciones del tipo de documento (DTD)....	8
Cómo obtener DTD de cXML.....	8
Realización de la validación.....	8
Transacción de perfil.....	9
Utilidades de XML.....	10

Capítulo 2

Implementación de un sitio para punchouts 11

Requisitos de los punchouts	11
Compradores	11
Proveedores	13
Secuencia de eventos de punchout	16
Pasos 1 y 2: Solicitud de punchout	16
Paso 3: Selección de productos	17
Paso 4: Comprobación	18
Paso 5: Envío del pedido de compra	19
Documentos de punchout	20
Catálogo de índice de punchout	20
PunchOutSetupRequest	22
PunchOutSetupResponse	27
PunchOutOrderMessage	27
Modificaciones de las páginas Web	29
Página de inicialización	30
Página de inicio	33
Página del remitente	34
Página del receptor de pedidos	37
Sugerencias sobre el sitio Web para punchouts	37
Pautas de implementación	37
Cookies de comprador y proveedor	38
Personalización	39

Capítulo 3

Recepción de pedidos de compra cXML 41

Proceso de los pedidos de compra	41
Recepción de pedidos de compra	42
OrderRequest	42
OrderResponse	44
Aceptación de datos adjuntos a pedidos	45

Apéndice A	
Especificación del lenguaje cXML	47
Especificación del protocolo	48
Modelo Petición-respuesta	48
Convenciones de XML	49
Sobre de cXML	50
Niveles de ajuste	52
Header	54
Request	56
Response	56
Modelo Sentido único (asíncrono)	59
Elementos básicos	65
Entidades de tipo	65
Elementos base	66
Transacción de perfil	66
ProfileRequest	66
ProfileResponse	67
Definiciones de pedidos	68
OrderRequest	68
Respuesta a una OrderRequest	77
Transacción de punchout	77
PunchOutSetupRequest	77
PunchOutSetupResponse	80
PunchOutOrderMessage	80
Cambios de estado posteriores	85
DocumentReference	85
StatusUpdateRequest	86
Definiciones del catálogo	87
Supplier	88
Index	90
Contract	92
Definiciones de gestión de suscripciones	93
Datos del proveedor	93
Suscripciones del catálogo	97
Definiciones de recuperación de mensajes	100
GetPendingRequest	100
GetPendingResponse	101

Apéndice B

Nuevas funciones de cXML 1.1103

Cambios generales en cXML	103
Soporte mejorado para varios idiomas	103
DTD (definiciones del tipo de documento) centralizadas	104
Nueva transacción de perfil	104
Nuevos códigos de estado	105
Nuevo atributo Type para miembros de mercado	105
Cambios en Extrinsic	106
Nuevo elemento Contact	106
Atributo requisitionID admitido	107
Resumen de la información sobre Extrinsic que se ha modificado	108
Extrinsic de nivel de cabecera	108
Mejoras de la transacción de punchout	109
PunchOutSetupRequest mejorado	109
Elemento SelectedItem	109
PunchOutOrderMessage vacío	110
Nuevo campo oculto cXML-base64	110
Nuevas funciones para los pedidos de compra	111
Nuevo atributo lineNumber	111
Datos adjuntos de los pedidos de compra	111
Nuevo atributo shipComplete	112
Nuevo elemento ShortName	112
Nueva transacción para el estado de los pedidos de compra	113
Nuevo elemento OrderReference	113
Nueva transacción StatusUpdateRequest	114
Nuevo elemento Followup	114

Indice 115

Prólogo

En este documento se explica cómo utilizar cXML (commerce eXtensible Markup Language, lenguaje de marcado extensible comercial) para la transmisión de datos relativos al comercio electrónico.

Destinatarios y requisitos previos

Este documento va dirigido a programadores que diseñan aplicaciones compatibles con cXML. Está orientado a proveedores que desean modificar sus sitios Web de comercio electrónico para punchouts.

cXML es un lenguaje abierto y versátil para los requisitos de transacciones de:

- Catálogos de productos electrónicos
- Catálogos de punchout cXML
- Aplicaciones de adquisición
- Comunidades de compra

Los lectores deben tener conocimientos de los conceptos de comercio electrónico y del estándar de comunicación del Web HTTP.

En este documento no se explica cómo utilizar aplicaciones de adquisición ni concentradores de comercio electrónico de red específicos.

Capítulos recomendados

- **Gestores comerciales de comercio electrónico:** Si desea obtener información general sobre las funciones de cXML, consulte Capítulo 1, "Introducción a cXML".
- **Programadores de páginas Web:** Los programadores de páginas Web que implementen sitios de comercio electrónico deben consultar todos los capítulos.
- **Administradores de sitios para punchouts:** Los técnicos del Web con experiencia en sitios Web para punchouts deben consultar Apéndice B, "Nuevas funciones de cXML 1.1".

Capítulo 1

Introducción a cXML

Este capítulo es una introducción a cXML (commerce eXtensible Markup Language, lenguaje de marcado extensible comercial) para las transacciones de comercio electrónico.

En él se proporciona un resumen sobre cXML y se tratan los siguientes temas:

- Funciones de cXML
- Tipos de aplicaciones que utilizan cXML
- Estrategia de entrega de contenido
- Validación respecto a definiciones del tipo de documento (DTD)
- Transacción de perfil
- Utilidades de XML

Funciones de cXML

cXML permite la comunicación entre compradores, proveedores, agrupadores de productos e intermediarios mediante un único lenguaje abierto y estándar.

Los portales de comercio electrónico entre empresas (comercio electrónico B2B) de éxito dependen de un protocolo flexible y de uso generalizado. cXML constituye la clave para proporcionar el máximo acceso a sus productos y servicios, ya que se trata de un eficaz lenguaje bien definido, diseñado específicamente para el comercio electrónico B2B y elegido por proveedores y compradores de grandes volúmenes.

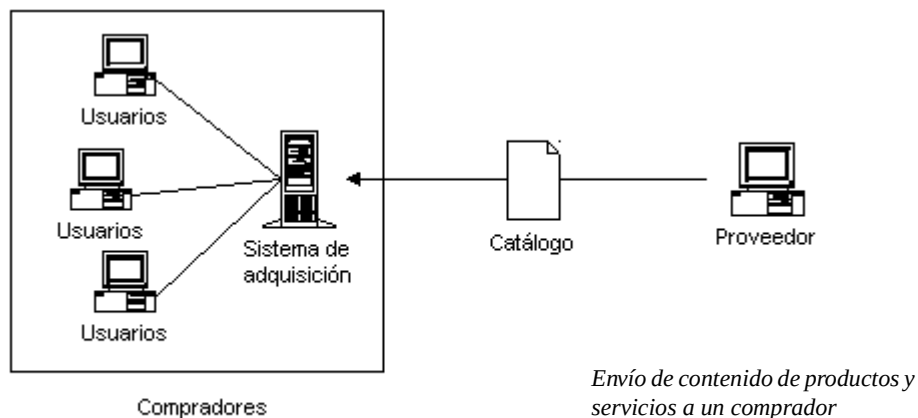
Las transacciones cXML consisten en *documentos*, que son archivos de texto sencillos con formato y contenido bien definidos. La mayoría de los tipos de documentos cXML son análogos a los documentos impresos empleados tradicionalmente en las empresas.

En las siguientes secciones se describen los principales tipos de documentos cXML.

Catálogos

Los catálogos son archivos que transmiten contenido de productos y servicios a compradores. Proporcionan una descripción de los productos y servicios ofrecidos y los precios correspondientes, y constituyen el principal canal de comunicación entre usted y sus clientes.

Cree catálogos para que las organizaciones que utilizan aplicaciones de adquisición puedan consultar y adquirir los productos y servicios que usted ofrece. Las aplicaciones de adquisición leen los catálogos y los almacenan internamente en sus bases de datos. Una vez que el comprador aprueba los catálogos, los usuarios podrán consultar su contenido, elegir los artículos y agregarlos a solicitudes de compra.



Puede crear catálogos de cualquier producto o servicio, independientemente de la unidad de medida, el método de precio y el modo de entrega utilizados.

Respecto a cada artículo de un catálogo, hay información básica necesaria e información opcional que activa características avanzadas del catálogo, como descripciones en varios idiomas.

Punchouts

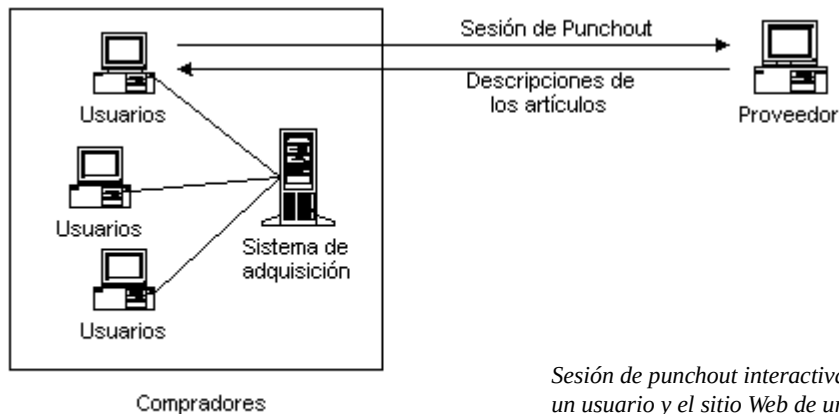
Los punchouts son una alternativa frente a los archivos de catálogo estáticos. Los sitios para punchouts son catálogos interactivos y actualizados que se ejecutan en el sitio Web.

Si posee un sitio Web de comercio electrónico, puede modificarlo para que admita punchouts. Los sitios para punchouts se comunican con los sistemas de adquisición a través de Internet con cXML.

Para obtener más información:

Capítulo 2, "Implementación de un sitio para punchouts".

En los sitios para punchouts, las aplicaciones de adquisición muestran un botón en lugar de detalles sobre productos o precios. Cuando los usuarios hacen clic en este botón, el navegador Web empleado muestra páginas de su sitio Web local. Según cómo implemente las páginas, los usuarios podrán navegar por las opciones de productos, especificar configuraciones y seleccionar métodos de entrega. Cuando los usuarios hayan terminado de seleccionar artículos, pueden hacer clic en un botón que devuelve la información del pedido a la aplicación de adquisición. Los productos configurados totalmente y sus precios aparecen en las solicitudes de compra de los usuarios.



Sesión de punchout interactiva entre un usuario y el sitio Web de un

El sitio Web puede ofrecer productos y precios previamente acordados mediante contrato.

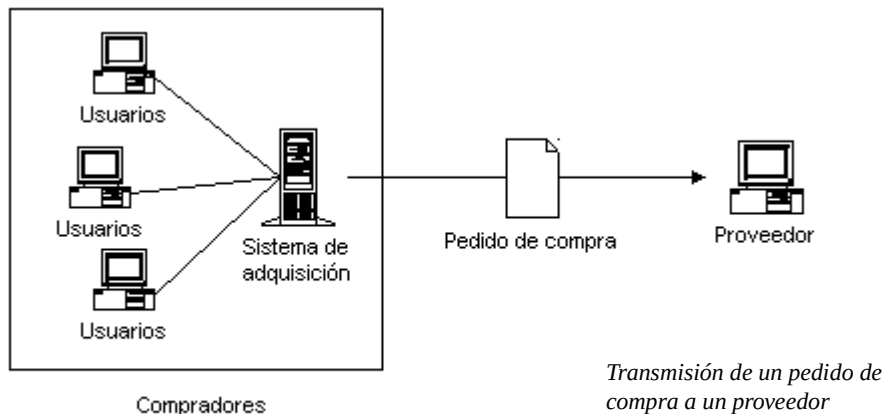
Pedidos de compra

Los compradores envían pedidos de compra a los proveedores para solicitar el cumplimiento de un contrato.

Los pedidos se pueden transmitir a los proveedores mediante una plataforma de red de comercio electrónico, como Ariba Network.

Para obtener más información:

Capítulo 3,
"Recepción de
pedidos de
compra cXML".



cXML es sólo uno de varios formatos posibles para pedidos de compra. Otros formatos son el correo electrónico, el fax y EDI (X.12 Electronic Data Interchange, intercambio electrónico de datos). No obstante, cXML es el formato ideal para los pedidos de compra, ya que es flexible, su implementación es económica, admite la más amplia variedad de datos y archivos adjuntos, y los sistemas automatizados pueden utilizarlo.

Tipos de aplicaciones que utilizan cXML

Cualquier aplicación de comercio electrónico puede utilizar cXML. Actualmente lo utilizan compradores, comunidades de compra verticales y horizontales, proveedores y distribuidores de aplicaciones.

En las siguientes subsecciones se describen los principales tipos de aplicaciones que usan actualmente cXML.

Aplicaciones de adquisición

Aplicaciones de adquisición, como Ariba ORMS (Operating Resource Management System, sistema de administración de recursos operativos) y Ariba IBX (Internet Business eXchange, intercambio entre empresas en Internet), utilizan cXML para las transacciones externas.

Ariba ORMS es una aplicación empresarial que poseen grandes organizaciones y que utilizan sus empleados a través de una red intranet.

Ariba IBX es un servicio basado en Internet que permite la creación de comunidades de compra formadas por numerosas empresas pequeñas y medianas.

Estas aplicaciones permiten a las comunidades de usuarios adquirir productos y servicios según contrato a proveedores que han sido aprobados por sus administradores de compras. Los administradores de las comunidades aprueban primero las compras solicitadas y luego los pedidos de compra aprobados se transmiten a los proveedores mediante varios canales posibles, incluido cXML a través de Internet.

Plataformas de red de comercio

Las plataformas de red de comercio, como Ariba Network, son servicios basados en la Web para la conexión de compradores y proveedores.

Estos servicios Web proporcionan características como la validación de catálogos y la administración de archivos, la publicación y la suscripción a catálogos, el direccionamiento automatizado de pedidos de compra y el historial de pedidos de compra.

La comunicación entre estos servicios Web, las aplicaciones de los compradores y las aplicaciones de los proveedores se pueden realizar en su totalidad con cXML a través de Internet.

Catálogos de punchout

Para obtener más información:

Capítulo 2, "Implementación de un sitio para punchouts".

Como se ha descrito anteriormente, los catálogos de punchout son catálogos interactivos, disponibles en los sitios Web de los proveedores. Estos catálogos son aplicaciones de servidor Web creadas en un lenguaje de programación como ASP (Active Server Pages, páginas Active Server), JavaScript o CGI, que administran las sesiones de punchout de los compradores.

Los catálogos de punchout aceptan solicitudes de punchout enviadas desde aplicaciones de adquisición, identifican al comprador y muestran los productos y los precios adecuados en formato HTML. Posteriormente, los usuarios pueden seleccionar artículos, configurarlos y seleccionar opciones, si corresponde.

Al final de la sesión de punchout, el sitio para punchouts envía descripciones de las selecciones de los usuarios, en formato cXML, a las aplicaciones de adquisición.

Sistemas de recepción de pedidos

Para obtener más información:

Capítulo 3, "Recepción de pedidos de compra cXML".

Los sistemas de recepción de pedidos son aplicaciones incluidas en los sitios de los proveedores que aceptan y procesan los pedidos de compra enviados por los compradores. Estos sistemas pueden ser sistemas automatizados, como sistemas de control de existencias, de cumplimentación o de procesamiento de pedidos.

Puesto que es sencillo extraer información de pedidos de compra cXML, resulta relativamente fácil crear los adaptadores que permiten que los sistemas de recepción de pedidos existentes los acepten.

Estrategia de entrega de contenido

Las aplicaciones de adquisición muestran contenido de productos y servicios a los usuarios. Los proveedores desean controlar el modo de presentación de sus productos y servicios a los clientes, ya que la presentación resulta decisiva para el proceso de ventas. Asimismo, los compradores desean facilitar el acceso y la búsqueda de contenido para asegurar un alto grado de cumplimiento del contrato.

Compradores y proveedores pueden elegir entre varios métodos de entrega del contenido de los productos y servicios. El método concreto que se utilice está determinado por el acuerdo entre un comprador y un proveedor, y la naturaleza de los productos y servicios comercializados.

La siguiente tabla contiene ejemplos de categorías de productos y servicios adquiridos con frecuencia y sus métodos de entrega de contenido preferidos.

Artículo	Propiedades	Método de entrega de contenido
Material de oficina, material de uso interno	Contenido estático, precios fijos	Catálogos estáticos
Material de laboratorio, MRO (mantenimiento, reparación y operaciones), piezas electrónicas	Requiere la normalización para su utilidad	Punchout en un portal de artículos verticales
Libros, productos químicos	Gran número de artículos de línea	Punchout en un sitio de un proveedor
Ordenadores, equipos de red, dispositivos periféricos	Muchas configuraciones posibles	Punchout en una herramienta de configuración de un proveedor
Servicios, materiales impresos	El contenido tiene atributos muy variables	Punchout en un formulario electrónico en el sitio de un proveedor

Los compradores pueden almacenar el contenido localmente dentro de la organización o pueden acceder a él de forma remota en Internet, mediante punchouts. Los catálogos cXML admiten ambas estrategias de almacenamiento.

Como se indica en la tabla anterior, los punchouts ofrecen un marco flexible en el que los proveedores, según sus artículos o clientes, pueden proporcionar contenido personalizado. El objetivo de esta estrategia de contenido es permitir que los compradores y los proveedores intercambien datos de los catálogos con el método más conveniente.

Validación respecto a definiciones del tipo de documento (DTD)

Puesto que cXML es un lenguaje XML, un conjunto de definiciones del tipo de documento (DTD) lo definen a fondo. Estos DTD son archivos de texto que describen la sintaxis y el orden precisos de los elementos de cXML. Los DTD permiten a las aplicaciones validar el cXML que leen o escriben.

No se necesitan aplicaciones cXML para validar documentos cXML, aunque se recomienda.

Cómo obtener DTD de cXML

Hay disponibles DTD para todas las versiones de cXML en ubicaciones coherentes en cxml.org:

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

Donde `<versión>` es el número completo de versión de cXML, por ejemplo, 1.1.007.

Realización de la validación

Las aplicaciones pueden utilizar estos DTD para validar todos los documentos cXML entrantes y salientes. Hay disponibles aplicaciones de validación de XML en la Web. Microsoft Internet Explorer 5 tiene integrada una función de validación de XML.

Para la gestión de transacciones más eficaz, valide todos los documentos cXML recibidos. Si detecta errores, emita el código de error adecuado para que el remitente pueda volver a transmitirlo.

Para obtener el mejor rendimiento, los clientes cXML no deben recuperar los DTD cada vez que analicen documentos cXML. En su lugar, deben consultar la versión de cXML indicada en las cabeceras de los documentos y recuperar los DTD que todavía no se han almacenado localmente.

Transacción de perfil

La transacción de perfil transmite información básica sobre los servidores cXML. Es la única transacción que deben admitir todos los servidores cXML.

Esta transacción está formada por dos documentos, ProfileRequest y ProfileResponse. Juntos, estos documentos recuperan capacidades del servidor, incluidas la versión de cXML admitida, las transacciones compatibles y las opciones para estas transacciones.

Nota: Todos los servidores cXML 1.1 **deben** admitir la transacción de perfil.

Los clientes pueden utilizar la transacción de perfil para ejecutar el comando "ping" en servidores con el fin de comprobar si están disponibles.

ProfileRequest

El documento ProfileRequest no tiene contenido. Simplemente direcciona al servidor cXML especificado.

ProfileResponse

El servidor responde con el documento ProfileResponse, que muestra las transacciones compatibles con el servidor cXML, sus ubicaciones y las opciones con un valor de cadena.

Utilidades de XML

Hay disponibles utilidades para editar y validar archivos XML tanto gratuitamente como mediante pago en la Web. A continuación se describen algunas de estas utilidades:

- **Internet Explorer 5** de Microsoft. Un navegador Web compatible con XML que puede validar archivos XML respecto a DTD.
www.microsoft.com/windows/ie/default.htm
- **XML Notepad** de Microsoft. Un sencillo editor de XML.
msdn.microsoft.com/xml/notepad/intro.asp
- **XML Authority** de Extensibility. Un editor de DTD de XML basado en Java, con vistas jerárquicas y gráficas.
www.extensibility.com
- **XML Spy** de Icon Information Systems. Una herramienta para mantener DTD y archivos XML, con vistas de cuadrícula, origen y navegador.
www.icon-is.com
- **XMetaL** de Softquad Software. Una herramienta de creación de XML personalizable.
www.softquad.com
- **CLIP** de Techno2000 USA. Una herramienta de creación de XML fácil de utilizar, con edición guiada.
www.t2000-usa.com
- **XMLwriter** de Wattle Software. Una herramienta gráfica de creación de XML diseñada para administrar proyectos de XML.
www.xmlwriter.net

Además, los siguientes sitios Web contienen más herramientas de XML:

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

Capítulo 2

Implementación de un sitio para punchouts

Los punchouts permiten a los usuarios de aplicaciones de adquisición acceder a contratos con proveedores sobre productos o servicios que se encuentran en el sitio Web. Asimismo, eliminan la necesidad de enviar catálogos completos a los compradores. En su lugar, basta con enviar archivos pequeños que contienen una descripción del escaparate, las categorías de productos o los productos.

En este capítulo se muestra cómo modificar un sitio Web para que admita punchouts y se tratan los siguientes temas:

- Requisitos de los punchouts
- Secuencia de eventos de punchout
- Documentos de punchout
- Modificaciones de las páginas Web
- Sugerencias sobre el sitio Web para punchouts

Requisitos de los punchouts

Antes de que los compradores configuren sus aplicaciones de adquisición para punchouts o los proveedores implementen sitios Web para punchouts, ambas partes deben evaluar los beneficios y los requisitos de los punchouts.

Compradores

Los procesos de instalación y comprobación de aplicaciones de adquisición compatibles con cXML con un proveedor que admite punchouts se pueden completar en menos de un día.

Puesto que los límites a la integración técnica son mínimos para los compradores, la decisión de utilizar punchouts debe basarse en las prácticas comerciales y los tipos de artículos adquiridos. (Consulte "Estrategia de entrega de contenido" en la página 7 para ver una lista de los artículos adecuados para punchouts).

Cuestiones empresariales

Los compradores deben tener en cuenta las siguientes cuestiones:

- ¿Tienen los solicitantes y aprobadores acceso a Internet? En caso contrario, ¿se permitiría el acceso controlado a Internet?
- ¿Desea el comprador que sus proveedores creen y mantengan contenido de catálogos (incluidos los precios)?
- ¿Obtienen los solicitantes actualmente productos mediante Internet? En tal caso, ¿requieren estos productos una herramienta de configuración del proveedor o contienen atributos únicos que no se ajustan a un modelo de contenido estático?
- ¿Utiliza el comprador agrupadores de contenido para catálogos (por ejemplo, Aspect, TPN Register o Harbinger)?
- ¿Obtiene el comprador actualmente servicios (por ejemplo, asesores, servicios de agencias temporales o mantenimiento) mediante Internet?
- ¿Adquiere el comprador actualmente productos y servicios en línea?

Si la respuesta a cualquiera de estas preguntas es afirmativa, los punchouts pueden resultar adecuados para el comprador.

Cuestiones técnicas

Los compradores deben cumplir los siguientes requisitos técnicos:

- Acceso directo a Internet: Los compradores deben tener acceso directo a Internet. Los punchouts se basan en sesiones normales del navegador Web donde el usuario interactúa con sitios Web actualizados de proveedores. Esta comunicación se produce mediante la infraestructura normal de internet/Internet, no mediante la aplicación de adquisición.
- Conexión a Internet fiable: El acceso a Internet debe funcionar y ser fiable constantemente. Si los usuarios no pueden obtener productos debido a interrupciones del acceso a Internet, es probable que realicen compras sin seguir todos los trámites necesarios.
- Contratos con proveedores que admiten punchouts: Los agentes de compras deben haber realizado contratos con proveedores que admiten punchouts. Los sitios Web para punchouts permiten el acceso sólo a compradores conocidos y autenticados.

Proveedores

El término *proveedor* en el contexto de punchouts abarca más que su definición tradicional. El protocolo de punchouts fue diseñado como un marco flexible capaz de transmitir datos sobre prácticamente cualquier tipo de producto o servicio de cualquier tipo de proveedor, distribuidor, agrupador de productos o fabricante.

Algunos ejemplos de productos y servicios son:

- Ordenadores adquiridos directamente a un fabricante o distribuidor
- Productos químicos y reactivos de un agrupador de productos
- Material de oficina de un distribuidor
- Servicios de contratos de una agencia temporal

Es posible que ya disponga de un sitio Web para transacciones capaz de alojar contenido y recibir pedidos de compra. Con esta capacidad, debe tener en cuenta sus prácticas comerciales y recursos técnicos a la hora de decidir la implementación de punchouts.

Cuestiones empresariales

Los proveedores deben tener en cuenta las siguientes cuestiones:

- ¿Vende actualmente sus productos y servicios a través de Internet? En tal caso, ¿ofrece contenido específico para clientes (precios según contrato) mediante el sitio Web?
- ¿Corresponden sus productos y servicios a una de las categorías de punchouts que se describen en la tabla mostrada en "Estrategia de entrega de contenido" en la página 7? Como revisión, estas categorías incluyen:
 - Productos con un gran nivel de configurabilidad (por ejemplo, ordenadores)
 - Gran número de artículos de línea (como libros)
 - Atributos de producto únicos (por ejemplo, productos químicos)
 - Datos normalizados (como material de MRO)
- ¿Prefiere recibir los pedidos de compra y/o pagos mediante el sitio Web?

Si la respuesta a cualquiera de estas preguntas es afirmativa, los punchouts pueden resultar adecuados para su organización.

Cuestiones técnicas

Los proveedores deben cumplir los siguientes requisitos técnicos:

- **Conexión a Internet fiable:** La infraestructura del servidor Web y la conexión a Internet deben ser muy fiables. Si los usuarios no pueden acceder a contenido remoto, es probable que acudan a otro proveedor.
- **Administradores de sitios Web competentes:** El sitio Web para punchouts y las aplicaciones compatibles requieren operaciones periódicas de modificación y mantenimiento. Las necesidades de los usuarios y sus ofertas de productos cambiarán, por lo que necesita personal para modificar la infraestructura de punchouts.
- **Soporte de transacciones básicas:** No es necesario que los sitios Web para punchouts admitan toda la funcionalidad de cXML, pero deben admitir las siguientes transacciones necesarias:

Transacción de perfil
PunchOutSetupRequest
PunchOutSetupResponse
PunchOutOrderMessage

Previsiones de trabajo

En la siguiente tabla se muestran las previsiones del trabajo necesario para la integración de punchouts cXML según las previsiones de los proveedores:

Nivel de infraestructura previa existente	Tiempo previsto para la finalización
Sitio para transacciones con infraestructura XML	3 semanas con personal de TI interno 3-4 semanas con contratistas
Sitio para transacciones sin infraestructura XML	4 semanas con personal de TI interno 4-5 semanas con contratistas

Qué es XML

El primer paso para utilizar punchouts es comprender en qué consiste XML. XML es un lenguaje para describir lenguajes. Los documentos cXML se crean basándose en definiciones del tipo de documento XML (DTD, Document Type Definitions, definiciones de tipos de documentos). Estas DTD actúan como plantillas y se pueden utilizar para definir modelos de contenido en documentos cXML (por ejemplo, el orden y el anidamiento de elementos válidos) y los tipos de datos de atributos.

Acerca de XML

XML (eXtensible Markup Language, lenguaje de marcado extensible) es un estándar para la transmisión de datos entre aplicaciones de Internet. Los documentos XML contienen datos con un formato de binomio etiqueta/valor. XML tiene una estructura similar a HTML (Hypertext Markup Language, lenguaje de marcado de hipertexto), pero las aplicaciones de Internet pueden extraer y utilizar datos de documentos XML más fácilmente que de documentos HTML.

A medida que se generaliza el uso de aplicaciones de Internet, el uso de XML será más frecuente.

Para implementar un sitio Web para punchouts, debe tener conocimientos básicos sobre cómo crear, analizar, consultar, recibir y transmitir datos XML a y desde una fuente remota.

Las herramientas básicas para procesar documentos XML son analizadores de XML. Estos analizadores se pueden adquirir gratuitamente de Microsoft y otras compañías (por ejemplo, Microsoft Internet Explorer 5 incluye un analizador de XML estándar). Para ver una lista de herramientas de XML, consulte "Utilidades de XML" en la página 10.

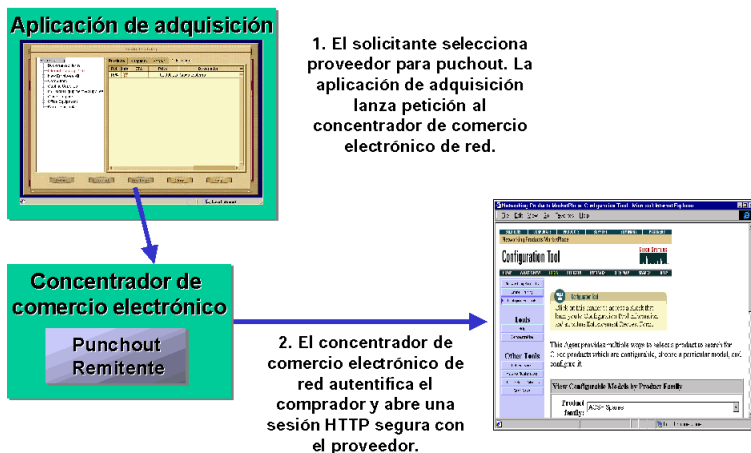
Secuencia de eventos de punchout

Una sesión de punchout se compone de varios pasos distintos.

Pasos 1 y 2: Solicitud de punchout

Los usuarios inician una sesión en una aplicación de adquisición y abren nuevas solicitudes de compra. Buscan los productos que desean en los catálogos locales por artículos, proveedores o descripción de productos. Cuando seleccionan un artículo de punchout, la aplicación de adquisición abre una nueva ventana del navegador e inicia una sesión en sus cuentas en el sitio Web.

En la siguiente ilustración se muestran los pasos para la solicitud de punchout:



¿Cómo funciona? Cuando un usuario hace clic en un artículo de punchout, la aplicación de adquisición envía el documento cXML PunchOutSetupRequest a un concentrador de red de comercio electrónico. Este concentrador actúa como el tercero de confianza y acepta la solicitud, comprueba el comprador y envía la solicitud al sitio Web para punchouts.

Nota: Todos los documentos cXML enviados a través de Internet pueden transmitirse mediante conexiones seguras HTTPS con cifrado 3.0 de SSL (Secure Socket Layer, nivel de socket seguro).

La finalidad de esta solicitud es notificar al sitio Web la identidad del comprador y comunicar la operación que se va a realizar. Las operaciones admitidas incluyen las siguientes:

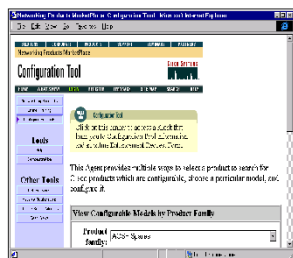
- **create:** Inicia una nueva sesión de punchout.
- **edit:** Vuelve a abrir una sesión de punchout para su edición.
- **inspect:** Vuelve a abrir una sesión de punchout para su inspección (no se pueden realizar cambios en los datos).

Después de que el sitio Web recibe una solicitud, devuelve el documento PunchOutSetupResponse que contiene una dirección URL que indica a la aplicación de adquisición dónde debe ir para iniciar una sesión de navegación en el sitio Web.

La aplicación de adquisición abre una nueva ventana del navegador, que muestra una sesión iniciada en una cuenta en el sitio Web. Esta cuenta puede ser específica para una región, compañía, departamento o usuario.

Paso 3: Selección de productos

Los usuarios seleccionan artículos de las existencias utilizando todas las características y servicios que proporciona el sitio Web:



3. El solicitante utiliza el sitio del proveedor para encontrar y configurar productos.

Según el producto o el cliente, estas características pueden incluir lo siguiente:

- Herramientas de configuración para crear productos personalizados (por ejemplo, ordenadores, compuestos orgánicos o artículos personalizados)
- Motores de búsqueda para buscar productos deseados en catálogos de gran tamaño
- Vistas de datos normalizados para comparar productos según precios, características o disponibilidad (por ejemplo, productos de MRO)
- Vistas de atributos únicos de un artículo concreto (por ejemplo, materiales impresos, productos químicos y reactivos, o servicios)

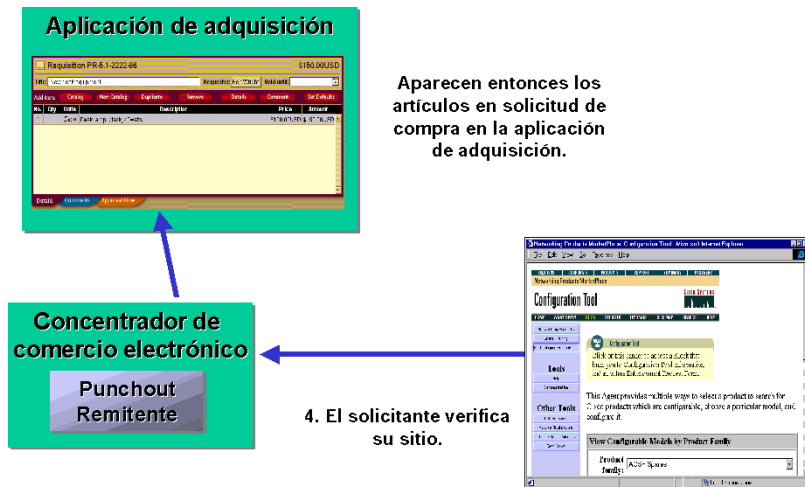
- Determinación de precios, existencias y comprobación de la disponibilidad en tiempo real
- Cálculo automático de impuestos y flete basado en el destino de envío, tamaño y número de artículos (no es necesario realizar el cálculo durante la sesión de punchout)

¿Cómo funciona? Después de que la aplicación de adquisición dirige a los usuarios al sitio Web, el procedimiento de compra es el mismo que si hubieran iniciado una sesión directamente en el sitio Web. Por tanto, no es necesario modificar ninguna de las características y servicios indicados anteriormente.

Paso 4: Comprobación

El sitio Web calcula el coste total de las selecciones del usuario, incluidos los impuestos, el flete y los descuentos específicos del cliente. A continuación, los usuarios hacen clic en el botón "Check Out" del sitio Web para enviar el contenido del carro de compra a sus solicitudes de compra en la aplicación de adquisición.

En la siguiente ilustración se muestran los pasos para la comprobación:



¿Cómo funciona? Cuando los usuarios hacen clic en el botón "Check Out", el sitio Web envía a la aplicación de adquisición el documento cXML PunchOutOrderMessage que contiene detalles de los productos y sus precios. También puede enviar cookies de proveedor ocultos, que posteriormente pueden asociar artículos a una sesión de compra específica.

En efecto, ha proporcionado un presupuesto de los artículos solicitados, pero no ha recibido todavía un pedido de compra, por lo que no puede registrar el pedido.

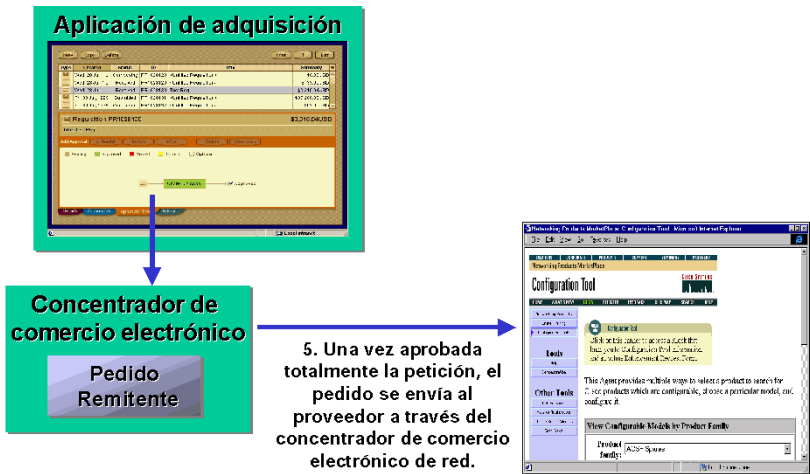
Si más adelante los usuarios necesitan editar cualquiera de los artículos en una solicitud de compra, puede permitirles un nuevo punchout en el sitio Web. La aplicación de adquisición devuelve el contenido del carro de compra original al sitio Web y los usuarios realizan los cambios allí. Tras la comprobación, el sitio Web devuelve los artículos a la solicitud de compra.

El sitio Web es la fuente de información de todos los artículos de punchout. Los cambios en la cantidad o la adición de nuevos artículos a la solicitud pueden variar los cargos de impuestos o envío, por lo que sería necesario realizar un nuevo cálculo en el sitio Web. Así, los cambios en los artículos originales se deben realizar en el sitio Web, no en la aplicación de adquisición, de ahí la necesidad de nuevos punchouts. Un nuevo punchout es simplemente un documento PunchOutSetupRequest con la operación "edit".

Paso 5: Envío del pedido de compra

Una vez enviado el contenido del carro de compra desde el sitio Web a la solicitud de compra del usuario, tiene lugar el proceso de aprobación en la aplicación de adquisición. Cuando se ha aprobado la solicitud de compra, la aplicación de adquisición la convierte en un pedido de compra, que envía al sitio Web para su cumplimentación. Los datos de la tarjeta de compra se pueden transmitir junto con el pedido o se puede facturar el pedido después del envío.

En la siguiente ilustración se muestra el envío de un pedido de compra:



¿Cómo funciona? La aplicación de adquisición envía todos los pedidos de compra al concentrador de comercio electrónico en formato cXML. A continuación, el concentrador se los envía a usted, utilizando el método de direccionamiento de pedidos que prefiera. Cuando acuse el recibo de un pedido de compra, habrá registrado realmente el pedido.

Los proveedores que admitan punchouts deben utilizar el método de direccionamiento de pedidos mediante cXML, por los siguientes motivos:

- Los pedidos de compra cXML permiten la devolución de información incrustada de cookies de proveedor. Como el tipo de datos de cookies de proveedor puede ser "cualquiera", no se pueden aplicar fácilmente a otros métodos de direccionamiento de pedidos como el fax, el correo electrónico o EDI.
- Los proveedores que admiten punchouts también tienen la compatibilidad con cXML, por lo que la aceptación de pedidos de compra cXML supone sólo un pequeño esfuerzo adicional.

Los pedidos de compra se tratan en detalle en Capítulo 3, "Recepción de pedidos de compra cXML".

Documentos de punchout

Hay cuatro tipos de documentos cXML:

- Catálogo de índice de punchout
- PunchOutSetupRequest
- PunchOutSetupResponse
- PunchOutOrderMessage

Catálogo de índice de punchout

Los catálogos de índice de punchout son archivos que muestran artículos de punchout y dirigen hacia el sitio Web para punchouts.

En el siguiente ejemplo se muestra un catálogo de índice de punchout:

Tipo de documento cXML y dirección URL de DTD	<pre><?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> <Description xml:lang="fr-FR">Chaises de Bureau</Description> <URL>http://www.workchairs.com/punchout.asp</URL> <Classification domain="UNSPSC">5136030000</Classification> </PunchoutDetail> </IndexItemPunchout> </IndexItem> </Index></pre>
Identificador del artículo de punchout	
Dirección URL del sitio Web para punchouts	

SupplierID identifica al proveedor. Puede utilizar cualquier dominio de identificación, pero los recomendados son DUNS (Dun & Bradstreet Universal Naming System, sistema de nomenclatura universal de Dun & Bradstreet) y NetworkID. Para obtener más información sobre los números DUNS, visite el sitio Web www.dnb.com.

Description especifica el texto que la aplicación de adquisición muestra en catálogos de productos. Puede proporcionar la descripción en varios idiomas y la aplicación de adquisición mostrará la descripción adecuada según el idioma del usuario.

Classification especifica el grupo de artículos al que corresponde el artículo de línea del comprador. Todos los productos y servicios deben estar asignados y estandarizados según el esquema UNSPSC. Para catálogos de índice de punchout, la clasificación determina la ubicación del artículo de punchout en los catálogos mostrados a los usuarios. Para ver una lista de los códigos UNSPSC, visite el sitio Web www.unspsc.com.

Creación y publicación de catálogos de índice

Cree estos catálogos y publíquelos en un concentrador de comercio electrónico para sus clientes. El administrador de catálogos que tengan los compradores los descargará y almacenará para utilizarlos con aplicaciones de adquisición.

Los usuarios consultan el contenido de los catálogos de índice de punchout junto a artículos de catálogos estáticos normales.

Granularidad de los artículos de punchout

Puede crear catálogos organizados por tiendas, divisiones o productos.

- Los catálogos de tienda muestran todos los productos y servicios. Los usuarios deben realizar búsquedas para encontrar el artículo deseado.
- Los catálogos de división muestran productos y servicios relacionados.
- Los catálogos de producto muestran sólo un producto o servicio. Los usuarios no necesitan realizar búsquedas.

Para determinar el nivel de amplitud de los artículos de punchout, tenga en cuenta el modelo empresarial que utiliza, la composición de las ofertas de sus productos y servicios, y la estructura de su sitio Web para punchouts.

Cuanto mayor sea el número de herramientas de búsqueda y configuración en el sitio Web, más amplios pueden ser los artículos de punchout incluidos en los catálogos de índice.

PunchOutSetupRequest

Para iniciar una sesión de punchout, el usuario debe seleccionar el artículo de punchout. A continuación, la aplicación de adquisición genera el documento PunchOutSetupRequest y lo envía a un concentrador de comercio electrónico, que lo reenvía al sitio Web para punchouts.

A continuación se muestra un ejemplo del documento PunchOutSetupRequest:

		<code><?xml version="1.0"?></code>
		<code><!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd"></code>
		<code><cXML version="1.1.007" xml:lang="en-US"</code>
		<code>payloadID="933694607118.1869318421@jlee" timestamp="2000-08-15T08:36:47-</code>
		<code>07:00"></code>
		<code><Header></code>
Origen (comprador)	_____	<code><From></code>
		<code><Credential domain="DUNS"></code>
		<code><Identity>65652314</Identity></code>
		<code></Credential></code>
		<code></From></code>
Destino (proveedor)	_____	<code><To></code>
		<code><Credential domain="DUNS"></code>
		<code><Identity>83528721</Identity></code>
		<code></Credential></code>
		<code></To></code>
Entidad de confianza	_____	<code><Sender></code>
anterior (en este caso,		<code><Credential domain="AribaNetworkUserId"></code>
Ariba Network)		

		<Identity>sysadmin@ariba.com</Identity> <SharedSecret>abracadabra</SharedSecret> </Credential> <UserAgent>Ariba ORMS 6.1</UserAgent> </Sender> </Header> <Request>
Tipo de solicitud	_____	<PunchOutSetupRequest operation="create"> <BuyerCookie>1CX3L4843PPZO</BuyerCookie> <Extrinsic name="CostCenter">610</Extrinsic> <Extrinsic name="User">john_smith</Extrinsic>
Destino del documento	_____	<BrowserFormPost> <URL>https://aribaorms:26000/punchout.asp</URL> </BrowserFormPost> <SupplierSetup> <URL>http://www.workchairs.com/punchout.asp</URL> </SupplierSetup>
Artículo seleccionado por el usuario	_____	<SelectedItem> <ItemOut quantity="1"> <ItemID> <SupplierPartID>5555</SupplierPartID> </ItemID> </ItemOut> </SelectedItem> </PunchOutSetupRequest> </Request> </cXML>

Los clientes cXML utilizan los atributos payloadID y timestamp que aparecen casi al principio para realizar un seguimiento de los documentos y detectar documentos duplicados.

Los elementos From, To y Sender permiten a los sistemas de recepción identificar y autorizar las partes que intervienen. Los elementos From y To de un documento no cambian. No obstante, durante el envío del documento a su destino, nodos intermedios (como Ariba Network) cambian el elemento Sender.

Operaciones de creación, edición e inspección

El atributo operation especifica el tipo de sesión que inicia el comprador. Puede ser create, edit o inspect.

- Las sesiones de create generan nuevos carros de compra, que corresponden a nuevas solicitudes de compra.
- Las sesiones de edit vuelven a abrir carros de compra creados anteriormente para su modificación. La aplicación de adquisición envía datos de los artículos de línea como parte del documento PunchOutSetupRequest. El sitio Web para punchouts puede utilizar estos datos para volver a crear la operación del carro de compra creado durante la sesión original.
- Las sesiones de inspect vuelven a abrir carros de compra creados anteriormente sólo para su visualización. Como con la operación edit, la aplicación de adquisición envía datos de los artículos de línea como parte del documento PunchOutSetupRequest. Sin embargo, después de volver a crear la operación del carro de compra, el sitio Web para punchouts no permite modificar su contenido.

A continuación se muestra un ejemplo de la solicitud 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>
    </PunchOutSetupRequest>
  </Request>
</cXML>
```



```
<Extrinsic name="User">john_smith</Extrinsic>
<BrowserFormPost>
  <URL>https://aribaorms:26000/punchout.asp</URL>
</BrowserFormPost>
<SupplierSetup>
  <URL>http://www.workchairs.com/punchout.asp</URL>
</SupplierSetup>
<ItemOut quantity="2">
  <ItemID>
    <SupplierPartID>220-6338</SupplierPartID>
    <SupplierPartAuxiliaryID>E000028901
    </SupplierPartAuxiliaryID>
  </ItemID>
</ItemOut>
</PunchOutSetupRequest>
</Request>
</cXML>
```

Si el usuario inició la sesión de edit al seleccionar un artículo de un catálogo, el documento PunchOutSetupRequest contendrá un elemento SelectedItem, como en una sesión de create.

Autenticación mediante un concentrador de comercio electrónico

Todos los documentos PunchOutSetupRequest se envían mediante un concentrador de comercio electrónico para su autenticación y para buscar la dirección URL del sitio Web para punchouts. Los pasos son:

1. El concentrador recibe el documento PunchOutSetupRequest del usuario.
2. El concentrador comprueba el ID del comprador (elementos From y Shared Secret) con la cuenta de comercio electrónico de ese comprador. También identifica el proveedor solicitado (elemento To).
3. El concentrador busca el secreto compartido en su cuenta y lo inserta (elemento Shared Secret) en el elemento Sender.
4. El concentrador encuentra la dirección URL del sitio Web para punchouts en su cuenta y envía el documento PunchOutSetupRequest a esa dirección.
5. El sitio Web recibe el documento cXML y sabe que está autenticado porque contiene su secreto compartido.
6. El sitio Web utiliza la información del elemento From para identificar al solicitante por la compañía a la que pertenece (por ejemplo, acme.com).
7. Puede usar la información de Contact y datos extrínsecos en el cuerpo de la solicitud para identificar de forma única al usuario (por ejemplo, Juan Sánchez del departamento de Finanzas de acme.com).

Los documentos PunchOutSetupRequest y PunchOutSetupResponse pasan por el concentrador de comercio electrónico para su autenticación. El documento PunchOutOrderMessage (que devuelve el contenido de la cesta de la compra a la aplicación de adquisición) se transmite directamente entre el sitio Web y el usuario a través de HTTPS o HTTP estándar.

Dirección URL de configuración del proveedor y elemento SelectedItem

En anteriores versiones de cXML, el elemento SupplierSetup proporcionaba la única manera de especificar la dirección URL del sitio Web para punchouts. Desde la versión cXML 1.1, el concentrador de comercio electrónico ya conoce la dirección URL del sitio Web para punchouts.

También desde la versión cXML 1.1, las aplicaciones de adquisición pueden utilizar el elemento SelectedItem para especificar punchouts de tienda, división o producto.

El elemento SupplierSetup se ha dejado de usar. No obstante, su sitio Web para punchouts debe admitir ambos métodos hasta que todos los sitios Web para punchouts reconozcan el elemento SelectedItem.

Información de contacto para datos extrínsecos e identificación del usuario

El documento PunchoutSetupRequest puede contener información detallada del usuario en el elemento Contact que el sitio Web puede utilizar para autenticar y dirigir a los usuarios, por ejemplo:

- Nombre y puesto del usuario
- Dirección de correo electrónico

Además, el documento PunchOutSetupRequest también puede contener datos *extrínsecos* que puede utilizar para identificar mejor a los usuarios, por ejemplo:

- Centro de costes del usuario y subcuenta
- Región
- Supervisor
- Divisa por defecto

Los compradores configuran sus aplicaciones de adquisición para insertar el elemento Contact y datos extrínsecos. Pregunte a los clientes los datos que van a proporcionarle.

PunchOutSetupResponse

Después de recibir el documento PunchOutSetupRequest, el sitio Web envía el documento PunchOutSetupResponse. El documento PunchOutSetupResponse tiene dos funciones:

- Indica si el documento PunchOutSetupRequest era correcto.
- Proporciona a la aplicación de adquisición una dirección URL que lleva a la página de inicio.

Este documento contiene un elemento <URL> que especifica la dirección URL de la página de inicio que se va a proporcionar al navegador Web del usuario para la sesión de navegación interactiva. Esta dirección URL debe contener suficiente información de estado para el enlace a un contexto de sesión en el sitio Web, como la identidad del solicitante y el contenido del elemento BuyerCookie.

A continuación se muestra un ejemplo del 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

Después de que el usuario seleccione artículos en el sitio Web, los configure y haga clic en el botón "Check Out", usted enviará el documento PunchOutOrderMessage para comunicar el contenido de la cesta de la compra a la aplicación de adquisición del comprador. Este documento puede contener muchos más datos que los demás documentos, ya que necesita poder expresar totalmente el contenido de cualquier cesta de la compra imaginable. Este documento no sigue de manera estricta el paradigma solicitud-respuesta; su uso se explicará en detalle.

A continuación se muestra un ejemplo del 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 permite a la aplicación de adquisición asociar un documento PunchOutOrderMessage específico al documento PunchOutSetupRequest de origen correspondiente. Por tanto, el sitio Web deberá devolver este elemento siempre que aparezca. No utilice BuyerCookie para realizar un seguimiento de las sesiones de punchout, ya que cambia para cada sesión, de creación, a inspección, a edición.

SupplierPartAuxiliaryID actúa como cookie de proveedor. Este campo permite transmitir datos adicionales, como el número de presupuesto u otro documento cXML. La aplicación de adquisición se los vuelve a enviar en cualquier sesión posterior de edit o inspect de PunchOutSetupRequest, y en el pedido de compra cXML resultante. Puede utilizar cookies de proveedor para asociar artículos de una solicitud de compra a los artículos correspondientes de un carro de compra en el sitio Web.

UnitOfMeasure especifica cómo se comercializa o se envía el producto. Debe ajustarse a los códigos comunes de unidad de medida UN/CEFACT. Para ver una lista de los códigos UN/CEFACT, visite el sitio Web www.unece.org/cefact.

Classification muestra el código de artículo UNSPSC (United Nations Standard Product and Service Codes, códigos estándar de productos y servicios de las Naciones Unidas) correspondiente a cada artículo seleccionado. Los sistemas de servicios de fondo de los compradores y proveedores utilizan estos códigos para la contabilidad y la generación de informes. Para ver una lista de los códigos UNSPSC, visite el sitio Web www.unspsc.org.

Modificaciones de las páginas Web

Para recibir o enviar los tres documentos de punchout cXML, puede que necesite modificar o crear cuatro páginas en el sitio Web:

- Página de inicialización
- Página de inicio
- Página del remitente
- Página del receptor de pedidos

Para ilustrar cómo puede implementar estas páginas se van a utilizar ejemplos sencillos de códigos ASP (Active Server Pages, páginas Active Server) y el analizador de XML de Microsoft Internet Explorer 5. La implementación real de estas páginas varía según el entorno de desarrollo del proveedor (por ejemplo, CGI, JavaScript o WebObjects).

Página de inicialización

La página de inicialización recibe todos los documentos PunchOutSetupRequest autenticados del concentrador de comercio electrónico. Lee la secuencia de HTTP enviada desde el concentrador y valida la solicitud cXML incrustada en dicha secuencia respecto a la DTD de cXML (en el caso de ASP, con llamadas de métodos al analizador de XML de Internet Explorer 5).

Tras la validación, la página de inicialización extrae elementos del documento para:

1. Identificar al usuario y determinar a dónde va a dirigirle.
2. Procesar un documento PunchOutSetupResponse y devolverlo al remitente.

La página de inicialización debe contener los siguientes datos para que los utilice la página de inicio:

- Identidad del solicitante (Sender)
- Identidad del idioma del usuario (xml:lang) para que pueda proporcionar el contenido traducido
- Tipo de solicitud (create, edit o inspect)
- Datos extrínsecos que identifiquen mejor al usuario y su ubicación

A continuación se muestra un ejemplo de una página de inicialización. Este código no utiliza un analizador de XML para generar de forma dinámica el documento PunchOutSetupResponse, sino que utiliza una plantilla XML estática donde se introducen los datos de artículos de línea. **Este código sólo se proporciona con fines ilustrativos.**

```
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 Create data needed in prolog.
%<
Randomize
randStr = Int( 100000001 * Rnd )
prologStr = genProlog( "1.0", randStr )
Response.ContentType = "text/xml"
Response.Charset = "UTF-8"
%>
<%
REM This receives the PunchOutSetup request coming from the e-commerce hub.
REM It takes the ORMSURL and buyercookie, attaches them to the Start Page URL,
REM and sends the response back to the requester.
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 This command reads the incoming HTTP cXML request
    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 This formats the cXML PunchOutSetupReponse
<% if IsEmpty(cookie) then %>
<%= prologStr %>
<Response>

```



```

        <Status code="400" Text="Bad Request">Invalid Document. Unable to extract
        BuyerCookie.</Status>
    </Response>
</cXML>
<% else %>
<%= prologStr %>
    <Response>
        <Status code="200" text="OK"/>
        <PunchOutSetupResponse>
            <StartPage>
                <URL>http://<%=
Request.ServerVariables("LOCAL_ADDR")%>/<%= dir%>/punchoutredirect.asp?bc=<%=
cookie%>&amp;url="<%= url%>"&amp;from=<%= fromUser%>&amp;to=<%=
toUser%>&amp;redirect=<%= StartPage%></URL>
            </StartPage>
        </PunchOutSetupResponse>
    </Response>
</cXML>
<%end if%>

```

La página de inicialización debe devolver una dirección URL StartPage que sea única para esa sesión de punchout. Además, esta dirección URL debe ser válida sólo durante un intervalo de tiempo limitado. Al desactivar esta dirección URL, resulta más difícil el acceso de usuarios no autorizados a la página de inicio.

Recuerde implementar la funcionalidad para posteriores sesiones de edit y inspect. Los usuarios no pueden cambiar detalles de los pedidos sobre artículos de punchout (como la cantidad) con la aplicación de adquisición que utilicen. Deben crear nuevos punchouts con una sesión de edit. Para que los usuarios obtengan el mayor beneficio posible, las sesiones de inspect que se inicien después de que reciban los pedidos deben mostrar el estado de los pedidos.

Página de inicio

La página de inicio abre una sesión para el solicitante en una cuenta en el sitio Web. Desde la página de inicio, los usuarios comienzan sus compras. Puede que ya exista esta página en su sitio Web, pero debe modificarla para que se solicite la información de nombre de usuario y contraseña indicada en el documento PunchOutSetupRequest.

Permita sólo el acceso de usuarios autorizados a la página de inicio. Si espera hasta el paso de comprobación para autenticarlos, no protegerá su información confidencial de precios o condiciones.

Si utiliza cookies HTTP de navegador para realizar un seguimiento de las preferencias de los usuarios, elimínelos después de enviar el documento PunchOutOrderMessage a los compradores. La eliminación de estos cookies evita la posibilidad de ofrecer características privilegiadas a usuarios no autorizados.

Página del remitente

La página del remitente envía al usuario el contenido de su carro de compra. Como se ha indicado anteriormente, después de que los usuarios llenan sus carros de compra, hacen clic en el botón "Check Out".

A continuación se muestra una sencilla implementación de esta característica en ASP. Este código no utiliza un analizador de XML para generar de forma dinámica el documento PunchOutOrderMessage, sino que utiliza una plantilla XML estática donde se introducen los datos de artículos de línea. **Este código sólo se proporciona con fines ilustrativos.**

Esto es una parte de una página de productos del sitio Web de un proveedor:

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

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

La función AddBuyButton devuelve el documento PunchOutOrderMessage al usuario.

La siguiente lista corresponde al archivo para incluir (punchoutitem.inc) al que se hace referencia anteriormente:

```
<%
REM This asp is included in items.asp, which specifies the item parameters, formats
REM a cXML document, and allows the user to proceed with a checkout of the item.
function CreateCXML(toUser, fromUser, buyerCookie, unitPrice, supPartId, desc)
%>
<?xml version="1.0" encoding="UTF-8"?>
```

```

<!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
%>

```

La función AddBuyButton contiene el elemento FORM POST que devuelve al usuario el documento PunchOutOrderMessage codificado con una dirección URL.

Codificación de formularios HTTP

Para el envío del documento PunchOutOrderMessage se utiliza la *codificación de formularios HTML*, que es un modelo de transporte diferente al modelo tradicional de solicitud-respuesta HTTP. Este transporte diferente facilita la integración entre el sitio Web y la aplicación de adquisición. También permite a los compradores recibir datos en formato XML sin que necesiten tener un servidor Web disponible mediante un servidor de seguridad.

En lugar de enviar el documento PunchOutOrderMessage directamente a la aplicación de adquisición, el sitio Web lo codifica como un campo de formulario HTML oculto y lo envía a la dirección URL especificada en el elemento BrowserFormPost del documento PunchOutSetupRequest. El nombre de este campo debe ser cxml-urlencoded o cxml-base64 (estos nombres no distinguen mayúsculas y minúsculas).

Esta codificación le permite diseñar una página Web de comprobación que contenga el documento cXML. Cuando los usuarios hagan clic en el botón "Check Out", el sitio Web mostrará los datos (que no pueden ver los usuarios) a la aplicación de adquisición como el envío de un formulario HTML.

Cancelación de punchouts

Puede que desee agregar un botón "Cancelar" a sus páginas para que los usuarios puedan cancelar la sesión de punchout. Este botón "Cancelar" envía un documento PunchOutOrderMessage vacío que comunica a la aplicación de adquisición que no se van a devolver artículos y que suprima los artículos de punchout existentes de la solicitud. También puede utilizarlo para efectuar tareas de mantenimiento necesarias en el sitio Web, como eliminar el carro de compra y cerrar la sesión del usuario.

Página del receptor de pedidos

La página del receptor de pedidos acepta pedidos de compra cXML enviados por compradores. Puede ser similar a la página de inicialización descrita anteriormente.

Para obtener información sobre la recepción de pedidos de compra, consulte Capítulo 3, "Recepción de pedidos de compra cXML".

Sugerencias sobre el sitio Web para punchouts

Cuando planifique la implementación de un sitio Web para punchouts, tenga en cuenta estas sugerencias.

Pautas de implementación

Siga estas pautas cuando desarrolle el sitio Web para punchouts:

- Estudie la especificación cXML.
- Utilice un analizador de XML y valide los documentos respecto a la DTD de cXML.
- Use la propiedad `xml:lang=` para identificar los idiomas de los usuarios y poder proporcionar contenido traducido.
- Utilice la credencial `From` para identificar a los compradores.
- Envíe una única dirección URL temporal para el direccionamiento en la sesión.

- No mantenga cookies de navegador.
- No sobrecargue a los clientes con requisitos de datos extrínsecos.
- Utilice UNUOM (United Nations Units of Measure, unidades de medida de las Naciones Unidas) y UNSPSC (United Nations Standard Product and Service Codes, códigos estándar de productos y servicios de las Naciones Unidas) para cada artículo de línea.
- Proporcione valor real a los clientes. Muestre la disponibilidad de los productos, el estado de los pedidos y promociones especiales.
- La comprobación debe ser fácil e intuitiva. Lo ideal sería que los usuarios sólo tuvieran que hacer clic en tres botones para realizar sus compras.
- Codifique las posteriores sesiones de edición e inspección. Los usuarios no pueden cambiar detalles de los pedidos sobre artículos de punchout (como la cantidad) con la aplicación de adquisición que utilicen. Deben crear nuevos punchouts con una sesión de edición.
- Para que los usuarios obtengan el mayor beneficio posible, las sesiones de inspección deben mostrar el estado de los pedidos.
- Realice pruebas del sitio Web para punchouts. Permita un tiempo para pruebas con las aplicaciones de adquisición de los clientes.
- Las transacciones de punchout generan presupuestos, no pedidos de compra. Implemente una página de recepción de pedidos de compra cXML para aceptar pedidos.

Cookies de comprador y proveedor

Los cookies de comprador y proveedor permiten a estos volver a crear las operaciones de datos de artículos de línea para los sistemas de servicios de fondo que utilicen.

- Devuelva el cookie de comprador (BuyerCookie) que reciba. No lo modifique.
- Utilice el cookie de proveedor (SupplierPartAuxiliaryID).

El cookie de comprador es análogo a un número de solicitud de compra; transmite el estado que permite al sistema del comprador mantener la relación entre una solicitud y una cesta de la compra.

De igual manera, el cookie de proveedor es análogo a un número de presupuesto; transmite el estado que permite a su sistema mantener la relación entre una cesta de la compra y la solicitud y el pedido de compra del comprador. Las aplicaciones de adquisición le devuelven el cookie de proveedor en posteriores sesiones de edit o inspect de punchout, y en el pedido de compra resultante. Su sitio Web debería beneficiarse del cookie de proveedor para que no sea necesario transmitir datos visibles específicos del proveedor al comprador.

Personalización

La cabecera del documento PunchOutSetupRequest identifica siempre al comprador, pero la solicitud también puede contener datos de los elementos Contact y Extrinsic (como el centro de costes y la ubicación del usuario, o la categoría de los productos) que puede utilizar para determinar la dirección URL dinámica de atención al usuario.

Aunque no todos los compradores envían estos datos extrínsecos, esta información puede permitirle personalizar su tienda Web más allá del sencillo nivel corporativo. Por ejemplo, puede proporcionar una tienda Web independiente para cada centro de costes del comprador (o cada categoría de productos o cada usuario).

También puede almacenar y mostrar los presupuestos anteriores del usuario. Puede permitir a los usuarios volver a utilizar los presupuestos, comprobar el estado de los pedidos y crear informes sobre las operaciones anteriores. Para evitar problemas relativos a la seguridad, almacene el historial de presupuestos sólo respecto a cada usuario.

Una consideración clave durante la planificación es el esfuerzo que supone implementar un sitio Web para punchouts muy dinámico y personalizado. Debe encontrar el equilibrio entre personalización y complejidad: lleva más tiempo implementar y mantener un sitio Web complejo, pero puede ser más valioso para los usuarios. Se recomienda que comience por un sitio Web para punchouts sencillo y que lo vaya mejorando con el tiempo.

Capítulo 3

Recepción de pedidos de compra cXML

En este capítulo se explica cómo configurar un sitio Web para recibir pedidos de compra en formato cXML. También se explica cómo enviar mensajes sobre el estado de los pedidos de compra a compradores o mercados.

Proceso de los pedidos de compra

Las aplicaciones de adquisición convierten las solicitudes de compra aprobadas en uno o varios pedidos de compra. Un pedido de compra es una solicitud formal que realiza un comprador a un proveedor para el cumplimiento de un contrato.

cXML es sólo uno de los formatos para transmitir pedidos de compra. Otros formatos comunes son el correo electrónico, el fax y EDI (X.12 Electronic Data Interchange, intercambio electrónico de datos). No obstante, cXML es el mejor formato para los pedidos de compra, ya que permite automatizar fácilmente el procesamiento de pedidos. La estructura bien definida de cXML permite a los sistemas de procesamiento de pedidos interpretar fácilmente los elementos que contiene un pedido de compra. Con poca o ninguna intervención del usuario, se pueden transmitir los datos adecuados de los pedidos de compra a los departamentos de envío, facturación y ventas, según sea necesario.

Además, el método de direccionamiento de pedidos mediante cXML permite la transmisión de cookies de proveedores (SupplierPartAuxiliaryID) y datos adjuntos a pedidos de compra.

Cuando configure su cuenta en un concentrador de red de comercio electrónico, debe especificar una dirección URL a la que se van a enviar todos los pedidos de compra cXML. Tras recibir un pedido de compra, lo enviará a su sistema de administración de pedidos interno y lo cumplimentará de la manera habitual. El sitio Web también debe enviar un acuse de recibo de respuesta del pedido al concentrador de red de comercio electrónico, que comunicará al comprador que usted ha recibido y analizado correctamente el pedido de compra.

No necesita un sitio Web para punchouts para poder recibir pedidos de compra cXML; los punchouts y la recepción de pedidos cXML son capacidades distintas. No obstante, la infraestructura y las aplicaciones necesarias para admitir punchouts son las mismas que para recibir pedidos de compra cXML.

Recepción de pedidos de compra

Existen dos tipos de documentos cXML que se utilizan para transmitir pedidos de compra. La aplicación de adquisición envía el documento OrderRequest y usted responde con el documento OrderResponse. Estos documentos se envían a través del concentrador de red de comercio electrónico.

OrderRequest

El documento OrderRequest es similar a un pedido de compra. En el siguiente ejemplo se muestra el documento OrderRequest correspondiente a un artículo:

```
<?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>
      </OrderRequestHeader>
    </OrderRequest>
  </Request>
</cXML>
```

```

</Total>
<ShipTo>
  <Address isoCountryCode="US" addressID="1000467">
    <Name xml:lang="en">Acme, Inc.</Name>
    <PostalAddress name="default">
      <DeliverTo>John Q. Smith</DeliverTo>
      <DeliverTo>Buyers Headquarters</DeliverTo>
      <Street>123 Main Street</Street>
      <City>Mountain View</City>
      <State>CA</State>
      <PostalCode>94089</PostalCode>
      <Country>United States</Country>
    </PostalAddress>
    <Email name="default">john_smith@acme.com</Email>
    <Phone name="work">
      <TelephoneNumber>
        <CountryCode isoCountryCode="US">1
        </CountryCode>
        <AreaOrCityCode>800</AreaOrCityCode>
        <Number>5555555</Number>
      </TelephoneNumber>
    </Phone>
  </Address>
</ShipTo>
<BillTo>
  <Address isoCountryCode="US" addressID="12">
    <Name xml:lang="en">Acme Accounts Payable</Name>
    <PostalAddress name="default">
      <Street>124 Union Street</Street>
      <City>San Francisco</City>
      <State>CA</State>
      <PostalCode>94128</PostalCode>
      <Country isoCountryCode="US">US</Country>
    </PostalAddress>
    <Phone name="work">
      <TelephoneNumber>
        <CountryCode isoCountryCode="US">1
        </CountryCode>
        <AreaOrCityCode>415</AreaOrCityCode>
        <Number>6666666</Number>
      </TelephoneNumber>
    </Phone>
  </Address>
</BillTo>
<Shipping>
  <Money currency="USD">12.34</Money>
  <Description xml:lang="en-us">FedEx 2-day</Description>
</Shipping>
<Tax>
  <Money currency="USD">10.74</Money>
  <Description xml:lang="en">CA State Tax</Description>
</Tax>

```

```

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

```

OrderResponse

El documento OrderResponse acusa un recibo del pedido de compra y que su análisis es correcto. No es un compromiso de ejecución del pedido de compra, confirma que lo ha recibido y que es un 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>

```

Aceptación de datos adjuntos a pedidos

A menudo los compradores necesitan aclarar los pedidos de compra con notas, dibujos o faxes asociados. Para ello, pueden adjuntar archivos de cualquier tipo a pedidos de compra cXML mediante MIME (Multipurpose Internet Mail Extensions, extensiones multipropósito de correo Internet).

cXML contiene sólo referencias a componentes MIME externos enviados en un sobre MIME (con el documento cXML, por correo electrónico o por fax).

El concentrador de red de comercio electrónico recibe los datos adjuntos y puede reenviarlos al proveedor o almacenarlos para su posterior recuperación en línea.

Para obtener más información sobre los datos adjuntos a pedidos de compra, consulte "Transmisión de los datos adjuntos" en la página 53.

Para obtener más información sobre el estándar MIME, visite los siguientes sitios Web:

www.hunnysoft.com/mime

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

Apéndice A

Especificación del lenguaje cXML

En este apéndice se describen los formatos de los datos y el protocolo de cXML (commerce eXtensible Markup Language, lenguaje de marcado extensible comercial). Contiene toda la información necesaria para implementar cualquier transacción admitida desde la perspectiva del sistema del cliente o del servidor. Las interacciones del protocolo y los documentos comerciales que contienen las transacciones se estudian con mayor profundidad más adelante.

Además, los ejemplos con implementaciones reales ilustran y clarifican el uso de cXML.

Este apéndice consta de los siguientes apartados:

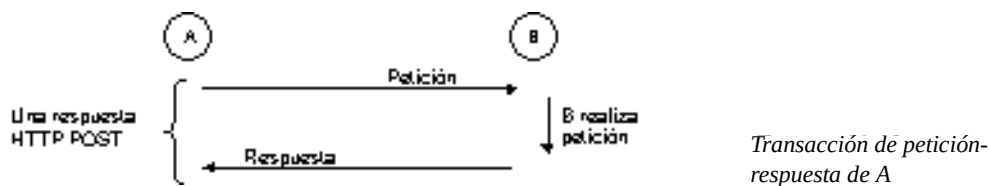
- Especificación del protocolo
- Elementos básicos
- Transacción de perfil
- Definiciones de pedidos
- Transacción de punchout
- Cambios de estado posteriores
- Definiciones del catálogo
- Definiciones de gestión de suscripciones
- Definiciones de recuperación de mensajes

Especificación del protocolo

Existen dos modelos de comunicación para las transacciones cXML: el Petición-respuesta y el Sentido único. Estos dos tipos de modelo permiten realizar de manera sencilla la implementación, ya que se describen con todo detalle las operaciones requeridas. Ambos modelos son necesarios porque se producen situaciones en que uno solo no sería adecuado.

Modelo Petición-respuesta

Las transacciones de petición-respuesta sólo se pueden realizar mediante una conexión HTTP. La siguiente imagen ilustra los pasos que se deben realizar en una interacción de petición-respuesta entre las partes A y B:



Esta transacción consta de los siguientes pasos:

1. A inicia una conexión HTTP/1.x con B en una URL predeterminada que representa la dirección de B.
2. A utiliza una operación POST para enviar el documento cXML mediante la conexión HTTP.
3. A espera que se devuelva una respuesta a través de la conexión HTTP.
4. B tiene un servidor compatible con HTTP/1.x que remite la petición de HTTP al recurso especificado por la URL que se ha especificado en el paso 1. Este recurso puede ser cualquier ubicación conocida por el servidor HTTP de B, como por ejemplo un programa CGI o una página ASP.
5. El recurso de B, identificado en el paso 4, lee el contenido del documento cXML y asigna la petición al gestor adecuado.
6. El gestor de B encargado de esa petición cXML realiza la tarea que la petición especifica y genera un documento cXML como respuesta.

7. B envía la respuesta cXML a A mediante la conexión HTTP establecida en el paso 1.
8. A lee la respuesta cXML y la devuelve al proceso que ha iniciado la petición.
9. A cierra la conexión HTTP establecida en el paso 1.

Este proceso se repetirá en futuros ciclos de petición/respuesta.

Para simplificar la tarea en los pasos anteriores, los documentos cXML se dividen en dos partes diferenciadas:

- Cabecera: Contiene la información sobre la verificación y la dirección.
- Datos de la petición o la respuesta: Contienen una petición o respuesta específica, así como la información que se debe transferir.

Ambos elementos se incluyen en un elemento de sobre principal. En el siguiente ejemplo se muestra la estructura de un documento de petición cXML:

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

En los siguientes ejemplos se muestra la estructura de un documento de respuesta cXML:

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

La estructura de la respuesta no utiliza el elemento de cabecera porque la respuesta siempre viaja en la misma conexión HTTP que la petición.

Convenciones de XML

cXML utiliza los elementos para describir los distintos artículos, que normalmente son propiedades en los documentos empresariales tradicionales. También la información que contiene claras subdivisiones y relaciones entre éstas, como por ejemplo una dirección, se describe utilizando estos elementos.

cXML hace un amplio uso de los atributos.

En cXML, todos los nombres de los elementos y los atributos utilizan palabras completas en mayúsculas (sin guiones) para separar las palabras. Los nombres de los elementos comienzan por una letra en mayúsculas y los nombres de los atributos por minúsculas. Por ejemplo:

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

Sobre de cXML

El elemento de sobre es la raíz de la estructura de los documentos cXML y contiene a los demás elementos. El elemento cXML está presente en todas las transacciones cXML.

En el siguiente ejemplo se muestra un elemento cXML totalmente 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 tiene los siguientes atributos:

version (opcional)	Especifica la versión del protocolo cXML. Un analizador XML de validación podría también determinar el atributo de la versión mediante las definiciones del tipo de documento (DTD) a las que hace referencia. Sin embargo, los documentos cXML deberían incluir la versión explícitamente para ayudar a las aplicaciones que no utilizan analizadores de validación.
xml:lang (opcional)	Lugar utilizado para todos los textos libres que se han enviado en este documento. El receptor debe responder o mostrar la información en el mismo lugar o en uno similar. Por ejemplo, un cliente que especifique xml:lang="en-UK" en una petición debería recibir datos "en" de vuelta.
payloadID	Número único, en relación con el espacio y el tiempo, que se utiliza en las operaciones de conexión para identificar documentos que pueden haberse perdido o tener problemas. Este valor no debería cambiar en caso de que se vuelva a intentar. La implementación recomendada es la siguiente: datetime.process id.random number@hostname
timestamp	Fecha y hora en que se envió el mensaje (con formato ISO 8601). Este valor no debería cambiar en caso de que se vuelva a intentar. El formato es DD-MM-YYYYThh:mm:ss-hh:mm (por ejemplo, 16-07-1997T19:20:30+01:00).

Lugar especificado en xml:lang

El atributo `xml:lang` también aparece en la mayoría de los elementos del texto libre (como por ejemplo en `Description` y `Comments`). Mientras que la especificación XML permite que el lugar de un elemento de cualquier elemento principal establezca por defecto los ya especificados, estos valores por defecto dan lugar a consultas no eficientes del árbol del documento. cXML procura mantener juntos los identificadores de lugar y las cadenas afectadas.

Los atributos `xml:lang` que aparecen en el protocolo cXML no tienen efecto alguno sobre los datos con formato, como por ejemplo los números, las fechas y las horas. Como se mencionaba anteriormente con respecto al atributo `timestamp`, estos valores distintos adoptan el formato de acuerdo con sus tipos de datos. Las cadenas más grandes (y las páginas Web a las que se hace referencia) que no están concebidas para el procesamiento de máquina deberían contener un formato de número o fecha de un lugar específico que coincida aproximadamente con un atributo `xml:lang`.

Hora y otros tipos de datos

El atributo `timestamp` (y el resto de las fechas y horas en cXML) deben seguir el formato que dicta el subconjunto restringido de ISO 8601 descrito en la nota emitida por el Word Wide Web Consortium (W3C) con título "Date and Time Formats" (formatos de fecha y hora) y que se encuentra disponible en www.w3.org/TR/NOTE-datetime-970915.html.

Los registros horarios requieren como mínimo una fecha completa más hora, minutos y segundos. Las fracciones de segundo son opcionales. Este protocolo necesita que las horas estén expresadas en hora local con un desfase de zona horaria respecto de la UTC (Coordinated Universal Time, hora coordinada universal) también denominada como Hora media de Greenwich. El indicador de zona horaria "Z" no está permitido.

Por ejemplo, `14-04-2000T013:36:00-08:00` corresponde al 14 de abril de 2000, 1:36 p.m., hora oficial del Pacífico (EE.UU.).

Otras referencias acerca del formato de la fecha, la hora y otros tipos de datos que cXML utiliza pueden ser las siguientes:

- Sitio XML Data Types Reference (Referencia de tipos de datos) de Microsoft, msdn.microsoft.com/xml/reference/schema/datatypes.asp
- La propuesta original acerca de los datos XML dirigida a Word Wide Web Consortium (W3C), www.w3c.org/TR/1998/NOTE-XML-data-0105

Niveles de ajuste

El elemento cXML es el cuerpo de un documento XML normal. Los documentos deben comenzar del siguiente modo:

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

Este documento se transfiere normalmente a través de HTTP con un tipo de medio MIME de text/xml y un parámetro charset que coincide con la codificación del documento. Como HTTP es de ocho bits, cualquier codificación que admita el analizador de recepción se puede utilizar sin una codificación de transferencia de contenidos, como por ejemplo Base64 o quoted-printable. Todos los analizadores XML admiten la codificación UTF-8, en la que se incluyen todos los caracteres Unicode. Por este motivo, las aplicaciones deben utilizarla para transferir los documentos cXML.

Nota: De acuerdo con los RFC 2376 "XML Media Types" (tipos de medios XML), el parámetro MIME charset sustituye cualquier codificación que se haya especificado en la declaración XML. Además, la codificación por defecto para el tipo de medio text/xml es us-ascii, no UTF-8 como aparecía en el apartado 4.3.3 de la especificación de XML. Para mayor clarificación, los documentos cXML deberían incluir una codificación explícita en la declaración XML. Los sobres MIME deberían utilizar un parámetro charset coincidente para los tipos de medio text/xml o application/xml.

Una transmisión HTTP de un documento cXML puede incluir las siguientes cabeceras MIME y 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"?>
...
```

Transmisión de los datos adjuntos

Cuando se envía una OrderRequest que hace referencia a archivos externos, éstos pueden residir en un servidor accesible al proveedor o se pueden transferir junto con el documento cXML. Esta segunda opción requiere el uso de un sobre MIME con varias partes. Un requisito cXML para este sobre (según los conceptos básicos descritos en RFC 2046 "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types") es que se incluyan las cabeceras Content-ID con cada archivo adjunto.

En el siguiente ejemplo se muestra la estructura necesaria de un documento cXML con una imagen JPEG adjunta (sin las cabeceras HTTP que se han visto 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--
```

Además, esta estructura contiene toda la información que un analizador MIME de recepción es capaz de procesar. Las aplicaciones que hacen uso del tipo de medio descrito en RFC 2387 "The MIME Multipart/Related Content-type" pueden obtener más información si la estructura se amplía del siguiente modo:

```
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--
```

Los analizadores MIME de recepción que no comprenden el tipo de medio multipart/related debe tratar los dos ejemplos anteriores del mismo modo. Cada parte de la transmisión MIME puede tener de manera adicional una codificación de transferencia de contenidos y la utiliza. Esta adición no es necesaria en las transmisiones HTTP. Las cabeceras Content-description y Content-disposition son opcionales en el protocolo cXML, aunque proporcionan información útil.

Para obtener más información acerca de los archivos externos para pedidos de compra, consulte "Attachment" en la página 72.

Header

El elemento Header contiene información sobre la dirección y la verificación. Este elemento es el mismo independientemente de la petición o respuesta específicas que se contengan en el cuerpo del mensaje cXML. Las aplicaciones necesitan la identidad del solicitante, pero no es necesario comprobar que la información sobre la identidad proporcionada es correcta.

En el siguiente ejemplo se muestra el 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>
```

Los elementos From y To son sinónimos de los elementos From y To (De y Para) de los mensajes de correo SMTP, que son la fuente lógica y el destino de los mensajes. Sender es la parte que inicia la conexión HTTP y envía el documento cXML.

Sender contiene el elemento Credential, mediante el cual la parte receptora puede verificar la parte emisora. Este elemento permite realizar verificaciones de gran envergadura sin necesidad de una infraestructura de certificado digital integral de clave pública. La parte receptora tan sólo debe proporcionar el nombre de usuario y la clave para que la parte emisora pueda realizar las peticiones.

Inicialmente, los elementos Sender y From son idénticos. Sin embargo, si el documento cXML se transfiere a través de los concentradores de red de comercio electrónico, el elemento Sender se modifica para indicar cuál es la parte emisora actual.

From

Este elemento identifica al autor de la petición cXML. De manera opcional, puede contener más de un elemento Credential, que permite al solicitante identificarse utilizando varios métodos de identificación. Usar varias credenciales es igual que especificar las direcciones SMTP y X.400 en un mensaje de correo electrónico.

To

Este elemento identifica el destino de la petición cXML. Al igual que el elemento From, puede contener más de una Credential para facilitar la identificación del objetivo.

Sender

Este elemento permite a la parte receptora identificar y verificar la parte que ha iniciado la conexión HTTP. Contiene una Credential de verificación más exhaustiva que las de los elementos From o To, porque la parte receptora debe verificar qué persona solicita realizar una tarea.

Credential

Este elemento contiene valores de identificación y verificación utilizados en los mensajes cXML.

Credential tiene los siguientes atributos:

domain	Especifica el tipo de credencial. Este atributo permite que los documentos contengan varios tipos de credenciales para varios dominios de verificación. Para los mensajes enviados con Ariba Network, por ejemplo, el dominio normalmente es AribaNetworkUserId o DUNS.
type (opcional)	Las peticiones dirigidas o procedentes de un mercado identifican tanto al mercado como a la compañía en los elementos From o To Credential. En este caso, la credencial del mercado utiliza el atributo Type, que se ha establecido con el valor "marketplace".

Credential contiene un elemento Identity y, de manera opcional, un elemento SharedSecret o DigitalSignature. El elemento Identity indica a quién representa la Credential, mientras que los elementos opcionales de verificación constatan la identidad de la parte.

El elemento SharedSecret se utiliza cuando el Sender tiene una combinación de nombre de usuario/contraseña que reconoce el solicitante.

El elemento DigitalSignature se puede emplear si las dos partes han acordado una autoridad y formato de certificado comunes. El atributo type de un elemento DigitalSignature especifica el tipo de certificado utilizado.

Nota: No utilice los elementos de verificación con los documentos enviados mediante comunicación de sentido único. Este transporte se mueve a través de los navegadores de los usuarios, de manera que éstos podrán consultar las fuentes del documento (incluidos los elementos Credential).

Request

Los clientes envían peticiones para solicitar operaciones. Sólo se permite un elemento Request para cada elemento de sobre cXML, lo que simplifica las implementaciones del servidor, porque no es necesario que se produzca ninguna desmultiplexación al leer los documentos cXML. El elemento Request puede contener virtualmente cualquier tipo de datos XML.

Request tiene los siguientes atributos:

deploymentMode (opcional)	Indica si la petición es de prueba o de producción. Los valores permitidos son "production" (por defecto) o "test".
-------------------------------------	---

Response

Los servidores envían respuestas para informar a los clientes de los resultados de las operaciones. Como puede que los resultados de algunas peticiones no tengan datos, el elemento Response puede contener de manera opcional sólo un elemento Status. El elemento Response también puede contener datos de cualquier nivel de la aplicación. En los escenarios de punchout, es un elemento PunchOutSetupResponse.

Status

Este elemento indica el éxito o fallo de una operación de petición.

Status tiene los siguientes atributos:

code	Código del estado de la petición. Sigue el modelo de códigos de estados HTTP. Por ejemplo, 200 representa una petición correcta.
text	Texto del mensaje de estado. Este texto facilita al usuario la lectura de los registros y está compuesto de cadenas canónicas en inglés.
xml:lang (opcional)	Idioma de los datos en el elemento Status. Resulta opcional con respecto a la compatibilidad con cXML 1.1. Puede que sea necesario en futuras versiones de cXML.

Los atributos del elemento Status indican lo que le ha sucedido a la petición.

Los datos del elemento Status pueden ser cualquier información que necesite el solicitante. Para el código de estado 200/OK, puede que no haya datos. Sin embargo, para el código de estado 500/Internal Server Error, se recomienda que se especifique el error de análisis XML o el error de aplicación. Este error permite realizar mejor la depuración unilateral y las pruebas de interoperabilidad.

Los servidores no deberían incluir elementos de respuesta adicionales (por ejemplo, un elemento PunchOutSetupResponse) a menos que el código de estado se encuentre en el rango 200 (Por ejemplo, 200/OK).

La especificación HTTP 1.1 incluye muchos códigos de estado que no son adecuados para cXML. Como en la mayoría de los casos cXML se superpone a HTTP, se producen muchos errores (como por ejemplo 404/Not Found) debido al transporte. Los códigos de estado 200/OK y 500/Internal Server Error son los más comunes. Los errores de validación al analizar un documento Request provocaría normalmente un error de transporte, como el error 400/Bad Request de HTTP.

La siguiente tabla incluye otros códigos HTTP que se pueden utilizar.

cXML incluye pocos códigos de estado que no son HTTP:

- 550: No se puede llegar al siguiente servidor cXML para finalizar la transacción que requiere conexiones de dirección ascendente. Un concentrador intermedio devuelve este código cuando no se puede acceder al sitio de un proveedor. (Cuando finalizan las conexiones de dirección ascendente, los concentradores intermedios deben devolver errores directamente al cliente.)
- 551: No se puede reenviar la petición debido a un problema de configuración del proveedor. Por ejemplo, un concentrador intermedio falla al realizar su verificación ante un proveedor. Los clientes no pueden rectificar este error, pero se puede resolver antes de que el cliente vuelva a intentarlo.

- 560: Error temporal de servidor. Por ejemplo, puede que un servidor esté fuera de servicio por cuestiones de mantenimiento. El cliente debe volver a intentarlo más tarde.

Estado	Texto canónico	Significado
200	OK	El servidor puede ejecutar esta petición, aunque puede que la respuesta devuelta contenga advertencias o errores de aplicación.
201	Accepted	Puede que no hayan finalizado aún algunos procesos. Como se ha mencionado en "StatusUpdateRequest" en la página 86, el cliente debe esperar transacciones StatusUpdate posteriores si se devuelve este estado como respuesta de una OrderRequest.
204	No Content	La información de todas las peticiones se ha validado y reconocido. El servidor no tiene datos de respuesta del tipo solicitado. En un PunchOutOrderMessage, este estado indica que la sesión de punchout ha finalizado sin que se produzcan cambios en el carro de la compra (o solicitudes del cliente).
400	Bad Request	Petición no aceptable para el servidor, aunque se ha analizado correctamente.
401	Unauthorized	El servidor no ha reconocido las credenciales proporcionadas en Request (el elemento Sender).
402	Payment Required	Esta Request debe incluir un elemento Payment completo.
403	Forbidden	El usuario no tiene privilegios suficientes para ejecutar esta Request.
406	Not Acceptable	Un alias del código 400: Request no aceptable para el servidor, aunque se ha analizado correctamente.
409	Conflict	El estado actual del servidor o de sus datos internos ha evitado la operación de petición (o su actualización). Puede que se realice una Request idéntica en el futuro, especialmente después de que se haya ejecutado otra operación.

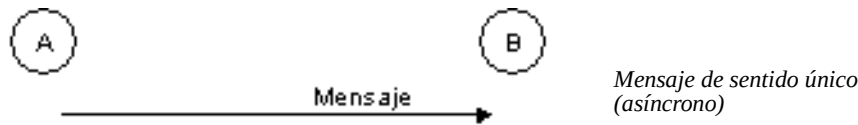
412	Precondition Failed	No se ha cumplido una de las condiciones previas de la Request (por ejemplo, una sesión de punchout adecuada para una edición de PunchOutSetupRequest). Este estado normalmente indica que el cliente ha ignorado parte de una transmisión previa del servidor (por ejemplo, el atributo operationAllowed de una PunchOutOrderMessageHeader).
417	Expectation Failed	Request indica que no se ha cumplido la condición de un recurso. Un ejemplo podría ser una SupplierDataRequest que solicita información sobre un proveedor que el servidor desconoce. Este estado implica la pérdida de información por parte del cliente o del servidor.
500	Internal Server Error	El servidor no ha podido finalizar la Request.
501	Not Implemented	El servidor no implementa esta Request. Por ejemplo, PunchOutSetupRequest o no se admite la operación solicitada. El estado normalmente indica que el cliente ha ignorado el perfil del servidor.

Cuando se reciben códigos no reconocidos, los clientes cXML deben procesarlos de acuerdo con sus clases. Por tanto, los clientes antiguos deben tratar todos los códigos nuevos 2xx como 200, los 4xx como 400 y los 5xx como 500. Este comportamiento hace posible que se realicen expansiones futuras del protocolo cXML y de los códigos específicos del servidor sin obtener pérdidas de interoperabilidad.

Modelo Sentido único (asíncrono)

A diferencia de las transacciones de petición-respuesta, los mensajes de sentido único no se limitan al transporte HTTP.

Estos mensajes se emplean cuando no es adecuado utilizar un canal HTTP (una operación del tipo petición-respuesta sincrónico). La siguiente imagen muestra un ejemplo de cómo A y B se pueden comunicar con mensajes en lugar de hacerlo a través de una transacción de petición-respuesta.



En este caso, se presenta el siguiente escenario:

1. A da formato y codifica un documento cXML con un transporte que resulte comprensible para B.
2. A envía el documento utilizando el transporte conocido. A no espera de manera activa (ya que no puede) una respuesta procedente de B.
3. B recibe el documento cXML y lo descodifica fuera de la corriente de transporte.
4. B procesa el documento.

En el modelo Sentido único, A y B *no* tienen un ciclo explícito de petición-respuesta. Por ejemplo, entre mensajes de sentido único, pueden llegar mensajes procedentes de terceros y se pueden producir conversaciones con otras personas.

Para especificar completamente una transacción de sentido único, también se debe documentar el transporte utilizado para el mensaje. Para las transacciones cXML que utilizan una aproximación de sentido único, se especifican el transporte y la codificación. Un ejemplo habitual de transacción que hace uso de esta modalidad podría ser el `PunchOutOrderMessage`.

Los mensajes de sentido único tienen una estructura similar a la del modelo Petición-respuesta:

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

El elemento `Header` recibe un tratamiento idéntico al del caso Petición-respuesta. El elemento `cXML` también es igual al descrito anteriormente. El modo más sencillo de apreciar la diferencia entre un mensaje de sentido único y uno de petición-respuesta es observar la presencia de un elemento `Message` (en lugar de un elemento `Request` o `Response`). En el siguiente apartado se estudia detenidamente el elemento `Message`.

Message

Este elemento conlleva toda la información de nivel de cuerpo en un mensaje cXML. Puede contener un elemento Status opcional, idéntico al encontrado en un elemento Response. Se podría utilizar con mensajes que son respuestas lógicas a mensajes de solicitud.

Message tiene los siguientes atributos:

deploymentMode (opcional)	Indica si la petición es de prueba o de producción. Los valores permitidos son "production" (por defecto) o "test".
inReplyTo (opcional)	Especifica a qué mensaje responde este mensaje. El contenido del atributo inReplyTo podría ser el payloadID de un mensaje recibido anteriormente. Se podría utilizar para construir una conversación de dos direcciones con muchos mensajes.

El atributo inReplyTo también puede hacer referencia al payloadID de una petición anterior o de un documento de respuesta. Cuando una transacción de petición-respuesta inicia una "conversación" mediante varias interacciones de sentido único, el primer mensaje puede incluir el payloadID de la última petición importante o de una respuesta que iba en dirección contraria. Por ejemplo, un mensaje que contiene un PunchOutOrderMessage puede incluir un atributo inReplyTo que contiene el payloadID de la PunchOutSetupRequest que ha iniciado la sesión punchout. (La BuyerCookie incluida en los documentos de punchout realiza una función parecida a la del atributo inReplyTo.)

Opciones de transporte

Existen dos transportes que se suelen utilizar con los mensajes de sentido único: HTTP y URL-Form-Encoding. Son dos de los transportes mejor definidos actualmente. Puede que se lleguen a admitir en el futuro.

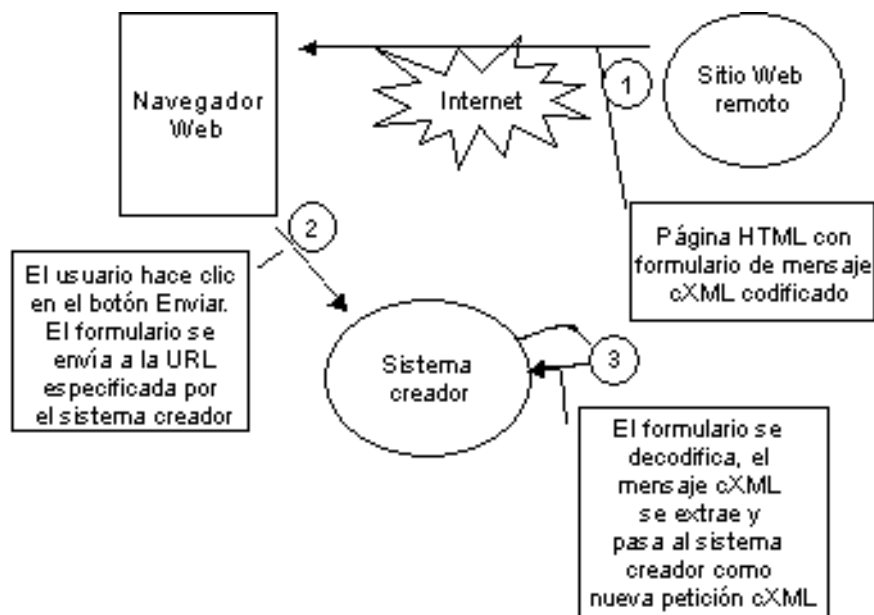
HTTP

HTTP se utiliza en la comunicación de sentido único para permitir que las aplicaciones de adquisición obtengan información. Uno de los tipos de transacción que utiliza la comunicación HTTP de sentido único es GetPendingRequest, que se estudia en página 100.

URL-Form-Encoding

Este transporte se comprende mejor examinando cómo se realiza la transacción PunchOutOrderMessage. URL-Form-Encoding permite la integración de un sitio Web remoto con las aplicaciones de adquisición. También se utiliza como medio para evitar la necesidad de un servidor de escucha en el sistema del comprador al que se puede acceder directamente a través de Internet.

El mensaje cXML PunchOutOrderMessage no se envía directamente a la aplicación de adquisición a través del sitio Web remoto, sino que se codifica como campo oculto del formulario HTML y se envía a la URL que se ha especificado en el elemento BrowserFormPost de la PunchOutSetupRequest. Cuando el usuario hace clic en Check Out, el sitio Web envía los datos a la aplicación de adquisición como una emisión de formulario HTML. El siguiente diagrama ilustra lo que sucede:



A continuación se describe la semántica del empaquetado y desempaquetado:

Empaquetado del formulario

El documento PunchOutOrderMessage se ha codificado mediante la URL (según la especificación HTTP) y se ha asignado a un campo oculto del formulario que se denomina cXML-urlencoded. El elemento HTML Form se asigna a un METHOD de POST y a una ACTION que incluye la URL transferida en el elemento BrowserFormPost de la PunchOutSetupRequest. Por ejemplo:

```
<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>
```

Las etiquetas HTML adicionales de la página pueden contener los fragmentos anteriores para describir con detalle el contenido de la cesta de la compra.

Nota: Cuando los servidores Web envían el `campocXML-urlencoded`, aún no se ha codificado mediante la URL. Esta codificación sólo es necesaria cuando el formulario se ejecuta desde los navegadores Web (cuando los usuarios hacen clic en Check Out en el ejemplo anterior). Los propios navegadores Web cumplen este requisito. El servidor Web debe codificar mediante HTML sólo el valor del campo, evitando las comillas y otros caracteres especiales, de manera que el formulario se muestra al usuario correctamente.

El nombre `cXML-urlencoded` no distingue mayúsculas y minúsculas.

Para los datos `cXML-urlencoded`, el analizador de recepción no puede asumir un parámetro `charset` por debajo del valor establecido por defecto para el tipo de medio `text/xml`. Un POST de HTTP no incluye información sobre codificación de caracteres para los datos enviados. El servidor Web de recepción no puede determinar la codificación de la página HTML que contiene el campo oculto. El documento `cXML` reenviado de esta forma debe por tanto utilizar la codificación de caracteres `us-ascii`. Todos los caracteres (incluidos los "URI codificados" como `"%XX"`) encontrados en el documento fuente XML deben encontrarse en el juego "us-ascii". Se pueden codificar otros símbolos Unicode utilizando las entidades de carácter del documento fuente.

Codificación mediante Base64

El campo oculto `cXML-base64` facilita la gestión de los documentos internacionales. Los documentos `cXML` que contienen símbolos distintos de "us-ascii" deben utilizar este campo en lugar del campo oculto `cXML-urlencoded`. Esta alternativa tiene una semántica casi idéntica, pero todo el documento está codificado mediante Base64 en todo el transporte y no mediante HTML para el navegador o mediante la URL para el servidor Web de recepción. La codificación mediante Base64 se describe en RFC 2045 "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

El nombre `cXML-base64` no distingue mayúsculas y minúsculas.

La codificación mediante Base64, que procede del sitio del proveedor, pasa a través del navegador y llega hasta el servidor Web de recepción y al cliente, mantiene la codificación de caracteres original de los documentos `cXML`. A pesar de que no llega ningún parámetro `charset` con la información enviada, el documento descodificado (después de que se elimine la codificación de la transferencia) se puede tratar como un tipo de medio `application/xml`. Esta codificación permite al analizador de recepción respetar todos los atributos de la codificación especificados en la declaración XML. Para este campo (así como para todos los documentos de `application/xml`), la codificación de caracteres por defecto es UTF-8.

Uno de estos campos ocultos (cXML-urlencoded o cXML-base64) debe aparecer en los datos enviados a la aplicación de adquisición. Aunque los receptores deben en primer lugar buscar cXML-base64 en los datos, resulta una pérdida de tiempo enviar ambos campos.

Empaquetamiento y desempaquetamiento de formularios

La aplicación de adquisición, que ha proporcionado anteriormente la URL adecuada, recibe un POST del formulario HTML que contiene los datos de éste, como se ha descrito anteriormente. El procesador POST del formulario buscará primero la variable cXML-base64, extraerá el valor y descodificará mediante Base64 el contenido. Si en los datos no existe ese campo, el procesador POST buscará la variable cXML-urlencoded, extraerá el mensaje cXML codificado mediante la URL y lo descodificará mediante la URL. A continuación se procesará el contenido descodificado como si se hubiese recibido a través de un ciclo normal de petición-respuesta de HTTP.

A continuación se presenta el tipo de medio del documento una vez descodificadas las variables, con posibles codificaciones de caracteres:

- La variable cXML-urlencoded es un tipo de medio text/xml que no tiene atributo charset. Se limita por tanto a la codificación de caracteres us-ascii. El analizador de recepción debe ignorar todos los atributos de codificación de la declaración XML incluida en el documento cXML, porque puede que el navegador haya cambiado de codificación.
- La variable cXML-base64 es un tipo de medio application/xml y, por lo tanto, debe tener alguna codificación de caracteres (indicada por el atributo de codificación de la declaración XML, en caso de que hubiera).

La diferencia principal entre esta transacción y una transacción normal de petición-respuesta es que no se puede generar ninguna respuesta, porque no existe la conexión HTTP a través de la cual se puede enviar.

Elementos básicos

Las siguientes entidades y elementos se utilizan a lo largo de toda la especificación cXML. La mayoría de las definiciones presentan un vocabulario básico con el que se describen los documentos comerciales de los pedidos mayores. Aquí se definen las entidades de tipo común y los elementos más habituales que representan objetos de bajo nivel.

Entidades de tipo

La mayoría de estas definiciones proceden del envío de la nota de los datos XML al World Wide Web Consortium (W3C). Un número muy reducido de entidades de tipo de alto nivel que también se definen aquí no son datos XML. Estos tipos se estudian en "Sobre de cXML" en la página 50.

isoLangCode

Código de idioma ISO incluido en la normativa ISO 639.

isoCountryCode

Código de país ISO incluido en la normativa ISO 3166.

xmlLangCode

Código de idioma como se define en la Especificación XML 1.0 (en www.w3.org/TR/1998/REC-xml-19980210.html). En el caso más frecuente, incluye un código de idioma ISO 639 y (opcionalmente) un código de país ISO 3166 separado por un guión. A diferencia de la recomendación XML, los códigos de idioma privados o IANA no se deben utilizar en cXML. Los subcódigos privados y IANA están permitidos, a pesar de que deberían colocarse después de un código de país ISO 3166 válido.

El formato del código de idioma cXML recomendado es `xx[-YY[-zzz]*]?`, donde `xx` es un código de idioma ISO 639, `YY` es un código de país ISO 3166 y `zzz` es un subcódigo privado o IANA para el idioma en cuestión. De nuevo, se recomienda siempre utilizar el código de país. Por convención, el código de idioma va en minúsculas y el código de país en mayúsculas, aunque no es necesario para que se produzca una coincidencia correcta de los códigos.

unitOfMeasure

`UnitOfMeasure` describe el modo en que el producto se empaqueta o envía. Debe ajustarse a los códigos comunes de unidad de medida de UN/CEFACT. Para obtener una lista con los códigos UN/CEFACT, consulte www.unece.org/cefact.

URL

Una URL (Uniform Resource Locator, localizador uniforme de recursos) tal como se define en la normativa HTTP/1.1.

Elementos base

Estos elementos, utilizados en la especificación, van desde los más genéricos, como por ejemplo Name y Extrinsic, a los más específicos, como por ejemplo Money.

Transacción de perfil

Las implementaciones del servidor cXML 1.1 deben admitir los documentos ProfileRequest y ProfileResponse. Esta transacción se debe utilizar para recuperar las capacidades del servidor, incluyendo la versión cXML admitida, las transacciones y las opciones de éstas.

La respuesta debe contener todas las peticiones admitidas en un sitio Web concreto, no necesariamente todas las que admite la empresa. Los proveedores que pueden recibir documentos OrderRequest y enviar varios mensajes o iniciar transacciones de petición-respuesta describen su soporte a OrderRequest en la transacción de perfil.

La transacción de perfil se puede utilizar para realizar un "ping" al servidor dentro del protocolo cXML.

ProfileRequest

Este elemento no tiene contenido. Tan sólo se dirige al servidor cXML adecuado utilizando Header. El servidor responde con una sola ProfileResponse como se describe más abajo. Las únicas partes dinámicas de esta respuesta son los atributos payloadId y timestamp del propio elemento cXML. En este caso concreto, no es necesario que un proveedor proporcione respuestas en varios lugares.

Un ejemplo de petición de este tipo sería el siguiente:

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

ProfileResponse

Este elemento contiene una lista de transacciones admitidas, sus ubicaciones y las opciones admitidas. Mientras no se hayan definido aún las opciones, el siguiente ejemplo es una posible 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>
```

Una ProfileResponse más frecuente procedente de un proveedor actual podría ser la siguiente:

```
<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 tiene el siguiente atributo:

effectiveDate	Fecha y hora en que estos servicios están disponibles. Las fechas no pueden ser futuras.
----------------------	--

Option

Valor para una opción definida (para todos los servicios o para una transacción específica). Aún no se ha definido ninguna opción.

Option tiene el siguiente atributo:

name	Nombre de esta opción. Este atributo no se puede consultar directamente (porque este perfil está concebido para uso de la máquina).
-------------	---

Transacción

La descripción de una transacción admitida por este servicio. La definición de perfil actualmente indica las ubicaciones a las que se envían las peticiones específicas. Las versiones futuras de cXML añadirán definiciones de Option y ampliarán la información del perfil para incluir más datos sobre las peticiones admitidas.

El elemento Transaction debe contener un elemento URL.

Transaction tiene el siguiente atributo:

requestName	Petición específica que este servidor acepta en la URL dada. Los valores pueden ser los siguientes:	
	ProfileRequest PunchOutSetupRequest GetPendingRequest SupplierListRequest SupplierDataRequest	OrderRequest StatusUpdateRequest SubscriptionListRequest SubscriptionContentRequest

Definiciones de pedidos

Los documentos de pedidos cXML son OrderRequest y una respuesta genérica. OrderRequest es similar a un pedido de compra. La respuesta es una confirmación de que el proveedor ha recibido el pedido de compra. No es necesario ejecutar el pedido de compra, sino la confirmación de que se ha recibido correctamente.

OrderRequest

En el siguiente ejemplo se muestra la estructura del elementoOrderRequest:

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

OrderRequestHeader

En el siguiente ejemplo se muestra una OrderRequestHeader con todo detalle:

```
<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 tiene los siguientes atributos:

orderID	Identificador de este pedido. Similar al número de pedido de compra.
orderDate	Fecha y hora en que se ha generado este pedido (en formato ISO 8601).
type (opcional)	Tipo de petición: new (por defecto), update o delete.
requisitionID (opcional)	Identificador de la solicitud del comprador para todo el pedido. Puede que sea el mismo que orderID o puede que no se incluya. No se incluye si requisitionID se especifica en todos los elementos ItemOut.
shipComplete (opcional)	Preferencia sobre envíos parciales. El único valor permitidos es "yes". Por defecto, los artículos se envían cuando están disponibles. Puesto que los pedidos pueden incluir artículos con elementos ShipTo que varían, sólo deberían mantenerse hasta completarse los grupos de artículos que tienen ubicaciones de envío comunes cuando shipComplete="yes".

OrderRequestHeader y ItemOut (cuando se amplía con ItemDetail) contienen información similar. Donde OrderRequestHeader incluye información sobre la facturación global (BillTo) y el pago (Payment), ItemOut describe todos los artículos (en ItemID, ItemDetail y Distribution).

No utilice esta información en OrderRequestHeader como valor por defecto para los elementos específicos del artículo. Si están presentes, ShipTo, Shipping, Contact y todos los denominados Extrinsic deben aparecer con todos los ItemOut o en la OrderRequestHeader. Los elementos Comments y Tax pueden aparecer simultáneamente en ambos niveles. Sin embargo, los distintos elementos Comments no deben duplicar la información y el elemento Tax de nivel de cabecera contiene un total para el pedido.

Total

Este elemento contiene la cantidad monetaria total del pedido. Es un contenedor para el elemento Money.

ShipTo/BillTo

Estos elementos contienen las direcciones de las entidades de envío y de facturación de la OrderRequest.

Un pedido se puede facturar a una sola entidad. Por consiguiente, el elemento BillTo aparece sólo en la OrderRequestHeader. Los artículos de un pedido se pueden enviar a varias ubicaciones. Al igual que el elemento Shipping (consulte el próximo apartado), el elemento ShipTo podrá aparecer en la OrderRequestHeader o en todos los elementos ItemOut.

Shipping

Este elemento describe cómo enviar los artículos de la petición y el coste de su realización. Si el elemento Shipping está presente en la OrderRequestHeader, no debe aparecer en todos los elementos ItemOut. Si no está no presente en la OrderRequestHeader, debe aparecer en los elementos ItemOut.

Tax

Este elemento contiene el impuesto asociado al pedido. Este elemento está presente si el comprador realiza el cálculo de los impuestos. Si aparece en la OrderRequestHeader, Tax describe el total de impuestos que tiene un pedido. Los elementos Tax a nivel de artículo pueden describir todos los importes de los impuestos.

Payment

Este elemento describe el instrumento de pago empleado para pagar los artículos solicitados. En el ejemplo anterior, el elemento Payment contiene un elemento PCard que codifica la tarjeta de compra estándar en un documento cXML. En el futuro se definirán y admitirán otros instrumentos de pago.

Contact

Información de contacto que puede utilizar el proveedor para realizar el seguimiento de un pedido. Este elemento identifica a la persona y proporciona una lista de maneras de contactar con esa persona o entidad. El único elemento necesario es el elemento Name del contacto. Entre las posibilidades opcionales y repetidas se incluyen PostalAddress (no se recomienda en la corrección inmediata de problemas relacionados con los pedidos), Email, Phone, Fax y URL.

Puede que los compradores seleccionen este elemento para identificar al solicitante original, al administrador del sistema de la aplicación de adquisición o a otro contacto que asuma la responsabilidad de solucionar los problemas de los pedidos. Contact puede no coincidir con la información de BillTo y ShipTo para un pedido.

Contact tiene el siguiente atributo:

role (opcional)	Posición de la persona en el proceso de adquisición. Puede tomar los valores endUser, administrator, purchasingAgent, technicalSupport, customerService o sales.
---------------------------	---

No debe aparecer la misma Contact role en los niveles de cabecera y artículo.

No existe una función por defecto, debido a los diferentes contenidos del elemento Contact. Por este motivo, las aplicaciones cXML tratan unContact que no tenga atributo role como una función adicional.

Comments

Información arbitraria que se puede leer y que los compradores pueden enviar junto con los pedidos de compra. Estos datos de cadena no están concebidos para los sistemas automáticos de los sitios de los proveedores.

El elemento Comments puede contener un elementoAttachment para incluir archivos externos.

Attachment

Comments puede adjuntar archivos externos para incrementar los pedidos de compra. El elemento Attachment aparece en Comments y contiene sólo una referencia a la parte externa de MIME de los datos adjuntos. Estos datos se deberían enviar mediante una sola transmisión de varias partes junto con el documento OrderRequest. Si no fuera posible, el contentID que se ha especificado en el elemento Attachment se puede utilizar para recuperar los datos adjuntos.

Para obtener más información acerca de la transferencia de archivos adjuntos, consulte "Transmisión de los datos adjuntos" en la página 53.

Attachment contiene una sola URL con el esquema "cid:". Puede que aparezca un archivo adjunto en un documento cXML:

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

El elemento Comments aparece en muchos lugares del protocolo cXML, pero sólo contiene el elemento Attachment en los documentos OrderRequest.

Followup

Especifica la URL a la que se enviarán los futuros documentos StatusUpdateRequest. Ésta es la ubicación de entrada de todos los documentos posteriores que hagan referencia al documento OrderRequest actual.

Extrinsic

Este elemento contiene información sobre el pedido que puede leer la máquina, pero que el protocolo cXML no ha definido. A diferencia de él, el elemento Comments transfiere la información para la lectura humana. Los elementos Extrinsic que contienen datos aparecen con más frecuencia en los documentos posteriores, a diferencia del elemento Comments. En este nivel, Extrinsic amplía la descripción de todos los artículos contenidos en el pedido. Parte de la información Extrinsic puede describir también el pedido de compra global sin que esto afecte a la descripción de todos los ItemOut contenidos.

Cada nombre Extrinsic puede aparecer solo una vez en las listas asociado a la OrderRequestHeader y a todos los elementos ItemOut (en el elemento ItemDetail contenido). Este mismo nombre no puede aparecer al mismo tiempo en la lista OrderRequestHeader y en cualquier lista asociada a los elementos ItemOut. Si el mismo nombre Extrinsic y su valor se repiten en todas las listas de ItemOut, se debe cambiar a la OrderRequestHeader.

El elemento Extrinsic también puede aparecer en los elementos IndexItem, PunchOutSetupRequest y ContractItem. Estos contextos se describen más adelante en este documento.

ItemOut

En el siguiente ejemplo se muestra un elemento ItemOut mínimamente válido:

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

ItemOut tiene los siguientes atributos:

quantity	Número de artículos deseados. Se permiten las fracciones para algunas unidades de medida. Puede que el proveedor ya haya comprobado el valor durante la sesión de punchout. Nunca debe ser negativa.
lineNumber (opcional)	Posición de este artículo en un pedido. Este valor ordinal aumenta con cada ItemOut en una "nueva" OrderRequest. Los clientes deben siempre especificar este atributo en la OrderRequest, aunque puede que no sea útil en otros contextos ItemOut.
requisitionID (opcional)	Identificador de la solicitud del comprador para este artículo de línea. No se debe incluir si se ha especificado requisitionID en OrderRequestHeader.
requestedDeliveryDate (opcional)	Fecha en que se ha solicitado la entrega del artículo, que permite fechas de entrega en nivel de artículo en la OrderRequest. Debe presentar el formato ISO 8601.

El atributo lineNumber permanece constante para todos los artículos a lo largo de las actualizaciones del pedido. Si se suprimen artículos de un pedido no cambia el lineNumber de los artículos restantes. Los artículos nuevos tienen números más altos que los que se han incluido previamente en el pedido. Cuando se produce una modificación en un artículo existente (por ejemplo un aumento de cantidad) no afecta al lineNumber de ese artículo.

En el siguiente ejemplo se muestra un elemento ItemOut más complejo:

```
<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>

```

El elemento `ItemDetail` permite enviar datos adicionales a proveedores en lugar de enviar el identificador único del artículo, indicado por el `ItemID`.

Los elementos `ShipTo`, `Shipping`, `Tax`, `Contact`, `Comments` y `Extrinsic` (algunos anidados en el `ItemDetail`) son idénticos a los que aparecen en la `OrderRequestHeader`. Estos elementos permiten que se representen los datos de cada artículo, como por ejemplo el envío, el tipo de envío y el coste asociado. Utilícelos sólo como nivel de `OrderRequestHeader` o sólo como nivel de `ItemOut`.

Distribution

La distribución divide el coste de un elemento entre sus múltiples partes. Los proveedores devuelven el elemento `Distribution` sobre las facturas para facilitar el proceso de conciliación del comprador.

Accounting

El elemento `Accounting` agrupa `Segments` para identificar a quién se ha efectuado el cargo.

`Accounting` tiene el siguiente atributo:

name	Nombre de la combinación contable.
-------------	------------------------------------

`Segment` tiene los siguientes atributos:

type	Nombre de identificación de este <code>Segment</code> con respecto a los otros en el elemento <code>Accounting</code> .
id	Identificador único en este tipo de <code>Segment</code> . Puede que este valor sea el código de cuenta real si el tipo fuera "centro de costes".

Charge

Este elemento especifica el importe que se carga a la entidad que indica el elemento `Accounting`.

Respuesta a una OrderRequest

Ésta es la parte de respuesta de la transacción de petición-respuesta sincrónica. En el siguiente ejemplo se muestra la respuesta de un 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 se mostraba anteriormente, esta respuesta es sencilla. En este caso, no existen elementos reales denominados "OrderResponse", porque el único dato que se debe devolver al solicitante es la parte Status de la respuesta.

La respuesta comunica al solicitante que la parte remota de la conexión HTTP ha analizado y ha cumplido correctamente su OrderRequest. Sin embargo, no comunica la confirmación en el nivel de pedido, como por ejemplo qué artículos se pueden enviar o cuáles se deben reponer.

Transacción de punchout

Las definiciones de los mensajes de punchout son mensajes de petición-respuesta que están incluidos en los elementos Request y Response. Los proveedores deben implementar los siguientes mensajes para admitir el punchout.

PunchOutSetupRequest

PunchOutSetupRequest y PunchOutSetupResponse son el par petición/respuesta utilizado para establecer una sesión de punchout con un sistema remoto. El cliente los utiliza para identificar la aplicación de adquisición, enviar la información de configuración y recibir una respuesta que indique dónde dirigirse para iniciar una sesión de exploración HTML en el sitio Web remoto.

Un elemento PunchOutSetupRequest está incluido en el elemento Request. En el siguiente ejemplo se muestra una 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 tiene el siguiente atributo:

operation	Especifica el tipo de PunchOutSetupRequest: "creación", "inspección" o "edición".
------------------	---

Este elemento también contiene los siguientes elementos: BuyerCookie, Extrinsic, BrowserFormPost, Contact, ShipTo, SelectedItem, SupplierSetup y una lista de ItemOut. Sólo es necesario el elemento BuyerCookie. La estructura de los elementos Extrinsic, Contact y ShipTo se estudia con más detalle en "OrderRequestHeader" en la página 69. El elemento ItemOut se estudia en "ItemOut" en la página 74. En este contexto (distinto de una OrderRequest), los elementos Distribution y Comments, así como los atributos lineNumber, requisitionID y requestedDeliveryDate de un ItemOut no añaden ningún valor (o muy poco) y no se deberían incluir. Como la sesión de punchout tiene lugar antes de realizar el pedido, esta información no es relevante para una PunchOutSetupRequest.

En una lista de ItemOut se describe un carro de la compra existente (los artículos procedentes de una sesión de punchout anterior). La operación inspect inicia una sesión de punchout de sólo lectura (entre el cliente y el servidor) para consultar los detalles sobre los artículos de la lista. La operación edit también se inicia desde el carro de la compra anterior (descrito mediante la lista de ItemOut), pero permite realizar modificaciones. Si se admite la operación edit significa que se admite inspect (consulte "PunchOutOrderMessageHeader" en la página 81 y "Carros de la compra vacíos" en la página 82).

BuyerCookie

Este elemento transmite información poco clara para el sitio Web remoto, pero se debe devolver al autor para todas las siguientes operaciones de punchout. Este elemento permite a la aplicación de adquisición hacer coincidir varias peticiones de punchout pendientes.

BrowserFormPost

Este elemento es el destino de los datos del PunchOutOrderMessage. Contiene un elemento URL cuyo uso queda explicado en la definición del PunchOutOrderMessage. Si no se utiliza el método URL-Form-Encoded, no se debe incluir este elemento.

Extrinsic

Este elemento opcional contiene todos los datos adicionales que el solicitante desea transferir al sitio Web externo. Este ejemplo se transmite al departamento del usuario que inicia la operación de punchout. En la especificación cXML no se define el contenido de los elementos Extrinsic. Es una cuestión que todos los solicitantes y sitios Web remotos deben acordar e implementar.

Los elementos Extrinsic están concebidos para proporcionar información adicional que puede leer la máquina. Amplían el protocolo cXML para admitir funciones que no necesitan todas las implementaciones. En este contexto, los nuevos datos describen al usuario que inicia la petición de punchout.

El elemento Extrinsic también puede aparecer en los elementos OrderRequestHeader, ItemDetail y ContractItem. Estos contextos se describen más adelante en este documento.

SelectedItem

Este elemento opcional indica a qué artículos los usuarios desean realizar un punchout para adquirirlos. Contiene un solo ItemID necesario. Este artículo dirige a los usuarios de su catálogo local al sitio Web del proveedor.

Este elemento suele estar presente en las operaciones create. Las aplicaciones de adquisición que permiten a los usuarios realizar un punchout directo desde la lista del proveedor no deben incluir el SelectedItem.

En las operaciones edit e inspect, SelectedItem sólo debe aparecer si el usuario elige volver al sitio Web del proveedor mientras consulta información nueva en el catálogo local, en lugar de los artículos de una solicitud existente. En cualquier caso, el carro de la compra actual debe aparecer en la lista ItemOut.

Los proveedores pueden crear sus catálogos de manera que el SelectedItem conduce a un punchout en nivel de tienda, de sección- o de producto. Cuanto más se especifique el artículo en el catálogo, más sencilla resultará la búsqueda que los usuarios deben realizar en el sitio Web del proveedor.

SupplierSetup

Este elemento opcional especifica la URL a la que se debe enviar la PunchOutSetupRequest. Este elemento no es necesario si el concentrador de red de comercio electrónico reconoce la URL del punchout del proveedor.

PunchOutSetupResponse

Una vez haya recibido el sitio Web una PunchOutSetupRequest, responde con una PunchOutSetupResponse, como se muestra a continuación:

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

StartPage

Este elemento contiene un elemento URL que especifica la URL que se transmite al navegador para iniciar la sesión de exploración de punchout que se ha solicitado en la PunchOutSetupRequest. Esta URL debe contener suficiente información sobre el estado como para enlazar con el contexto de una sesión en el sitio Web remoto, como puede ser por ejemplo la identidad del solicitante y el elemento BuyerCookie adecuado.

Llegados a este punto, el usuario que ha iniciado la PunchOutSetupRequest examina el sitio Web externo y selecciona los artículos que se han de transferir de vuelta a la aplicación de adquisición mediante un PunchOutOrderMessage.

PunchOutOrderMessage

Este elemento envía el contenido de la cesta de la compra remota al autor del PunchOutSetupMessage. Puede contener más datos que los otros mensajes porque debe expresar completamente el contenido de cualquier cesta de la compra que se pueda concebir en el sitio Web externo. Este mensaje no sigue de manera estricta el modelo Petición-respuesta.

El sitio Web remoto genera un PunchOutOrderMessage cuando el usuario realiza la comprobación. Este mensaje comunica el contenido de la cesta de la compra remota a la aplicación de adquisición. Por ejemplo:

```
<PunchOutOrderMessage>
  <BuyerCookie>34234234ADFSD234234</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>
```

En los siguientes apartados se estudian estos elementos.

BuyerCookie

Este elemento es el mismo que se ha transferido en la PunchOutSetupRequest original. Se debe devolver aquí para que la aplicación de adquisición pueda coincidir con el PunchOutOrderMessage con una PunchOutSetupRequest anterior.

PunchOutOrderMessageHeader

Este elemento contiene información sobre todos los contenidos de la cesta de la compra que se han transferido. El único elemento necesario es Total, que indica el coste global de los artículos que se han añadido a la solicitud. Los elementos adicionales que están permitidos son Shipping y Tax, que indican el importe y la descripción de los cargos de envío o impuestos que se han calculado en el sitio Web remoto. ShipTo es también opcional y especifica la información sobre la dirección

de envío que el usuario ha seleccionado en el sitio remoto o que se ha transmitido en la PunchOutSetupRequest original. Todos los importes monetarios se encuentran en un elemento Money que siempre especifica la divisa en formato estándar.

PunchOutOrderMessageHeader tiene el siguiente atributo:

operationAllowed	Especifica las operaciones de PunchOutSetupRequest permitidas: create, inspect o edit.
-------------------------	--

Este atributo controla si el usuario puede iniciar transacciones de PunchOutSetupRequest posteriores que contengan datos de este PunchOutOrderMessage. Si operationAllowed="create", sólo una OrderRequest posterior puede contener estos artículos. En caso contrario, la aplicación de adquisición puede inspeccionar o editar más adelante el carro de la compra (iniciando transacciones PunchOutSetupRequest posteriores con las operaciones adecuadas y los elementos ItemID que corresponden a la lista ItemIn que se ha devuelto a este PunchOutOrderMessage). Si se admite la operación edit significa que se admite inspect. La aplicación de adquisición siempre puede utilizar los artículos en la siguiente OrderRequest.

Carros de la compra vacíos

El PunchOutOrderMessage puede contener una lista de artículos correspondientes al carro de la compra del sitio Web del proveedor. Siempre indica el término de la sesión punchout interactiva. En los siguientes párrafos se explican detalladamente algunos casos sin una lista de artículos del PunchOutOrderMessage. Estos mensajes permiten a los clientes continuar inmediatamente cuando el usuario abandona el sitio Web del proveedor.

Si la operación del PunchOutSetupRequest original era inspect, la aplicación de adquisición debe ignorar la lista de artículos del PunchOutOrderMessage. En este caso, el sitio del proveedor no debería devolver ningún elemento ItemIn. Si un PunchOutOrderMessage no contiene ningún elemento ItemIn y la operación era create, no se debe añadir ningún artículo a la solicitud. El sitio del proveedor o el usuario ha cancelado la sesión de punchout sin crear un carro de la compra. Si la operación era edit y el PunchOutOrderMessage no contiene ningún elemento ItemIn, los artículos existentes procedentes de la sesión de punchout se deben suprimir de la solicitud en la aplicación de adquisición.

El código de estado "204/No Content" indica un fin de sesión sin cambios en el carro de la compra. De nuevo, el PunchOutOrderMessage (siempre necesario para la BuyerCookie) no debe contener elementos ItemIn. Este código se debe tratar igual que los casos "vacíos" comentados anteriormente, a no ser que la operación sea edit. En ese caso, el usuario ha cancelado la sesión sin realizar ningún cambio y no se debe modificar la solicitud en la aplicación de adquisición.

ItemIn

Este elemento añade un artículo de una cesta de la compra a la solicitud en la aplicación de adquisición. Puede contener una gran variedad de elementos, pero sólo dos de ellos son necesarios: ItemID e ItemDetail.

ItemIn tiene los siguientes atributos:

quantity	Número de artículos que ha seleccionado el usuario en el sitio Web remoto. Como el sitio del proveedor admite reglas para unidades parciales, el protocolo permite cantidades fraccionarias. Nunca puede ser negativa.
lineNumber (opcional)	Posición de este artículo en el pedido. Como las sesiones de punchout suelen tener lugar antes de la realización de los pedidos y el servidor no puede en ningún caso controlar la colocación de los artículos en el pedido, este atributo no es relevante en un PunchOutOrderMessage.

Los elementos opcionales son ShipTo, Shipping y Tax, que son los mismos elementos que los especificados en el PunchOutOrderMessage que aparece más arriba.

A excepción de los elementos Distribution y Comments, así como los atributos requisitionID y requestedDeliveryDate disponibles en el elemento ItemOut, las estructuras ItemIn e ItemOut coinciden de una en una. El sistema comprador que se origina puede convertir directamente las listas ItemIn en ItemOut al iniciar la operación inspect o edit. Los proveedores pueden convertir una a la otra (colocando las extensiones disponibles en el elemento ItemOut) al ejecutar una operación edit. El sistema comprador que se origina puede realizar las conversiones directamente y añadir información adicional sobre los envíos y la distribución así como comentarios cuando se inicia la transacción OrderRequest. No se deben borrar los datos ItemDetail (con la posible excepción de los elementos Extrinsic) contenidos en los elementos ItemIn al realizar la conversión de ItemIn a ItemOut.

ItemID

Este elemento identifica de forma única el artículo en el sitio Web remoto. Es el único elemento que se debe devolver al sitio remoto para volver a identificar el artículo que se transfiere.

ItemID contiene dos elementos: SupplierPartID y SupplierPartAuxiliaryID. Sólo se necesita SupplierPartID. SupplierPartAuxiliaryID ayuda a que la información sobre la configuración compleja de transportes de sitio Web o sobre la lista de productos pueda volver a identificar el artículo cuando se presente en el sitio Web remoto en otra ocasión.

Si `SupplierPartAuxiliaryID` contiene caracteres especiales (por ejemplo, si contiene elementos XML adicionales que no están definidos en el protocolo cXML), se debe solucionar su presentación correctamente. Debido a la necesidad de transferir la información `SupplierPartAuxiliaryID` mediante las aplicaciones y devolverlas al proveedor que las ha originado, se hace insuficiente un subconjunto interno con información adicional sobre elementos XML.

ItemDetail

Este elemento contiene información descriptiva sobre el artículo que las aplicaciones de adquisición presentan a los usuarios. El contenido de un elemento `ItemDetail` puede ser bastante complejo, pero los requisitos mínimos son sencillos: `UnitPrice`, `Description`, `UnitOfMeasure` y `Classification`.

En el contexto de un elemento `ItemIn`, los elementos `Extrinsic` contenidos en una función `ItemDetail` son idénticos a los encontrados en un índice (específicamente un `IndexItemAdd`).

Description

Este elemento describe el artículo con formato de texto. Como puede que este texto sobrepase los límites de una tabla breve con artículos de línea (u otras interfaz de usuario limitada) y se pueden producir truncamientos aleatorios, el elemento `Description` contiene el elemento opcional `ShortName`. Si existe, los clientes deben presentar el `ShortName` en lugar del truncamiento del texto de la `Description` en todos los campos restringidos. Los clientes deben seguir truncando el texto de la `Description` si no se ha proporcionado el `ShortName`.

Por ejemplo:

```
<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>
```

puede aparecer como "Big Computer" cuando el espacio está limitado y "Big Computer: This wonder ... lying around." (o como dos campos completos distintos) cuando hay espacio para introducir un texto mayor.

Cambios de estado posteriores

Una vez finalizada la transacción de OrderRequest, puede que los proveedores y los servidores intermediarios tengan que proporcionar información adicional al sistema comprador. Las transacciones descritas en este apartado se utilizan para este fin. Estas transacciones comparten algunos rasgos semánticos y elementos comunes.

Al igual que sucedía con las respuestas a una OrderRequest (consulte "Respuesta a una OrderRequest" en la página 77), ninguna de estas transacciones incluye un elemento Response específico. En su lugar, el documento devuelto contiene una Response casi vacía (sólo un Status). Los documentos devueltos presentan el siguiente formato:

```
<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>
```

El código devuelto es "200" sólo si la operación ha finalizado correctamente.

DocumentReference

El elemento DocumentReference contiene información suficiente como para asociar la petición actualizada con un documento concreto. Repite un atributo necesario del documento anterior y añade un identificador opcional que el proveedor genera. Por ejemplo:

```
<DocumentReference
  payloadID="0c300508b7863dccb_14999"
  orderID="DO4321" />
```

DocumentReference no contiene elementos, pero sí los siguientes atributos:

payloadID	Número único, en relación con el espacio y el tiempo, que se utiliza en las operaciones de conexión para identificar documentos. Este valor no debería cambiar en caso de que se vuelva a intentar. La implementación recomendada es la siguiente: datetime.process id.random number@hostname Se toma directamente del elemento cXML del documento OrderRequest.
orderID	Identificador de este pedido. Número actual del pedido de compra. Se toma directamente del elemento OrderRequestHeader del documento OrderRequest.

StatusUpdateRequest

Esta transacción comunica a un nodo anterior acerca de los cambios realizados en el estado del procesamiento de un pedido. Existe un cambio de especial importancia: cuando un concentrador intermedio transmite correctamente una OrderRequest hacia delante, puede informar al emisor original o a un concentrador previo acerca de lo ocurrido. Asimismo puede que las transiciones a través de varias colas y fases de procesamiento de un proveedor o concentrador sean también de gran importancia para el comprador.

Esta petición actualiza el estado de procesamiento de un solo documento OrderRequest. Por ejemplo:

```
<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>
```

Esta petición contiene sólo un elemento DocumentReference y Status. Ambos son necesarios. El Status puede comunicar un error de transporte posterior que ha detectado un concentrador intermedio. La semántica de este elemento es idéntica a la del Status que probablemente se ha devuelto en la respuesta HTTP inicial a un documento OrderRequest.

El código 200/OK resulta de especial importancia cuando los documentos se almacenan y se reenvían. Este código indica que un proveedor ha comenzado a procesar la OrderRequest o que un concentrador ha reenviado el documento. El receptor no debería esperar más documentos StatusUpdateRequest después de recibir el código 200/OK.

Los proveedores y concentradores que utilizan la transacción StatusUpdate deben devolver el código 201/Accepted cuando la OrderRequest espera en cola para procesarse más tarde. Después de enviar el código 200/OK (como Response inmediata a una OrderRequest o como una StatusUpdateRequest posterior), el servidor no debería enviar más transacciones StatusUpdate para ese pedido. Los errores que se producen posteriormente durante el procesamiento pueden constituir la excepción de esta regla.

Definiciones del catálogo

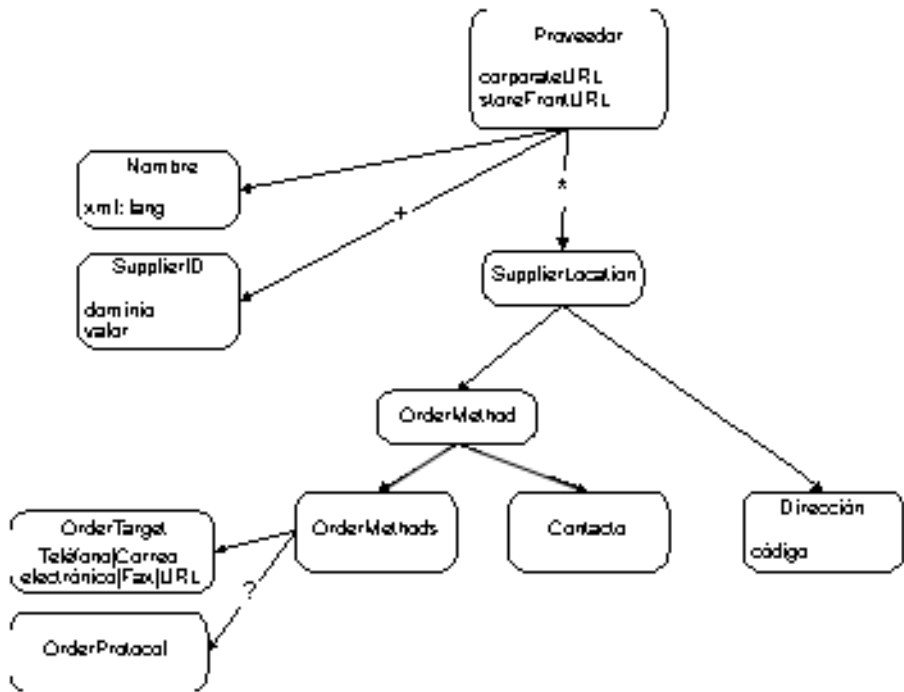
Las definiciones del catálogo cXML constan de tres elementos principales: Supplier, Index y Contract. Todos estos elementos describen los datos concebidos para uso permanente o temporal en el sistema de adquisición de un comprador o un concentrador.

- **Supplier:** Contiene datos básicos acerca del proveedor, como la información sobre la realización de pedidos, la dirección y el contacto.
- **Index:** Describe los datos acerca del inventario de productos del proveedor, como por ejemplo descripciones, números de pieza y códigos de clasificación.
- **Contract:** Describe los datos acerca de aspectos flexibles del inventario acordado entre el comprador y el proveedor, como es el precio.

Tenga en cuenta que Index utiliza varios subelementos para describir los artículos de línea en los inventarios del proveedor. Los proveedores pueden enviar la información sobre los precios para entrar en los sistemas del comprador o bien la información sobre punchout para que los compradores puedan realizar un punchout en los sitios Web remotos para obtener información sobre los precios y otros datos.

Supplier

El elemento Supplier contiene un proveedor de productos o servicios. Debe tener un elemento Name y un elemento SupplierID. Además, especifica al proveedor información sobre la realización de pedidos y direcciones opcionales:



Supplier tiene los siguientes atributos:

corporateURL (opcional)	URL del sitio Web del proveedor.
storeFrontURL (opcional)	URL del sitio Web para realizar compras o exploraciones.

En el siguiente ejemplo se muestra la estructura del elementoSupplier:

```
<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>
        </OrderMethods>
        <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

Algunos proveedores realizan sus actividades comerciales en más de una ubicación. Se puede utilizar un elemento `SupplierLocation` para cada ubicación. Este elemento también contiene el modo en que cada ubicación realiza las actividades comerciales o el modo en que se aceptan los pedidos. El elemento `SupplierLocation` contiene una `Address` y un conjunto de `OrderMethods`.

OrderMethods y OrderMethod

El elemento `OrderMethods` es una agrupación de uno o varios elementos `OrderMethod` para un determinado elemento `SupplierLocation`. La posición de `OrderMethods` en la lista es significativa: el primer elemento es el método de pedido preferido, el segundo elemento es la siguiente prioridad y así sucesivamente siguiendo un orden decreciente de preferencia.

`OrderMethod` contiene información sobre la realización de pedidos en la forma de objetivo del pedido (como teléfono, fax o URL) así como un protocolo opcional para aclarar más las expectativas sobre los pedidos en relación con ese objetivo. Por ejemplo: "cxml" para un objetivo URL.

Index

Éste es el elemento raíz para actualizar los catálogos de los sistemas de adquisición del comprador.

Se asocia un elemento `Index` a un solo proveedor. Este elemento facilita una lista de ID de proveedor, donde cada ID se considera como sinónimo del proveedor.

El `Index` contiene uno o varios elementos `IndexItem` así como un conjunto opcional de elementos `SearchGroup` para definir los datos de búsqueda paramétrica de los artículos. El elemento `IndexItem` contiene elementos que se añaden o suprimen del catálogo almacenado del comprador. En el siguiente ejemplo se muestra el esquema de un elemento `Index`:

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

IndexItem, IndexItemAdd, IndexItemDelete y IndexItemPunchout

El elemento `IndexItem` es un contenedor de la lista de artículos de un índice. Consta de tres tipos de elementos:

- `IndexItemAdd`: Inserta un nuevo artículo o actualiza uno ya existente del índice. Contiene un elemento `ItemID`, un elemento `ItemDetail` y un elemento `IndexItemDetail`.
- `IndexItemDelete`: Elimina un artículo del índice. Contiene un elemento `ItemID` que identifica al artículo.
- `IndexItemPunchout`: Inserta un artículo para iniciar un punchout en el sitio Web del proveedor. Contiene un elemento `PunchoutDetail` y un elemento `ItemID`. Es similar a un elemento `IndexItemAdd` salvo que no necesita la información sobre los precios. Los compradores obtienen información sobre los artículos en tiempo real a través del sitio Web del proveedor.

ItemID

El elemento ItemID identifica de forma única a un artículo del proveedor. Contiene un elemento SupplierPartID y un elemento opcional SupplierPartAuxiliaryID.

SupplierPartAuxiliaryID

Si SupplierPartID no identifica el artículo de forma exclusiva, el proveedor debería usar SupplierPartAuxiliaryID para especificar una clave "auxiliar" que identifique a la parte de forma única cuando se combina con SupplierID y SupplierPartID. Por ejemplo, puede que un proveedor utilice el mismo SupplierPartID para un artículo, pero tiene distinto precio por unidades de "EA" y "BX". En este caso, un SupplierPartAuxiliaryID razonable para dos artículos podría ser "EA" y "BX".

SupplierPartAuxiliaryID también se podría utilizar como cookie del proveedor, permitiendo a éste que haga referencia a configuraciones complejas o datos de la parte. Podría contener todos los datos necesarios para que el proveedor reconstruya los que el artículo en cuestión representa en su sistema informático (una cesta o cookie de datos sólo comprensibles para el proveedor). Para obtener más información, consulte "Cookies de comprador y proveedor" en la página 38.

ItemDetail

ItemDetail contiene información detallada acerca de un artículo o todos los datos que un usuario desee consultar acerca de un artículo siguiendo las pautas básicas representadas en el ItemID. Debe contener un UnitPrice, una UnitOfMeasure, uno o varios elementos Description y una Classification y, de manera opcional, puede contener un elemento ManufacturerPartID, uno ManufacturerName, uno URL y varios Extrinsic. Para obtener más información, consulte "ItemDetail" en la página 84.

En el contexto de un IndexItemAdd, los elementos Extrinsic amplían la información acerca de un artículo concreto. Estas ampliaciones no se deberían transmitir a un proveedor en una OrderRequest, porque el proveedor puede recuperar los mismos datos utilizando el ItemID único.

IndexItemDetail

El elemento IndexItemDetail contiene elementos específicos del índice que definen aspectos adicionales de un artículo, como por ejemplo LeadTime, ExpirationDate, EffectiveDate, SearchGroupData o TerritoryAvailable.

PunchoutDetail

PunchoutDetail es similar a ItemDetail, salvo que sólo necesita uno o varios elementos Description y una Classification. También puede contener los elementos URL, ManufacturerName, ManufacturerPartID, ExpirationDate, EffectiveDate, SearchGroupData, TerritoryAvailable y Extrinsic. No contiene información sobre la determinación de precios, los plazos ni las unidades de medida.

Contract

El elemento Contract representa un contrato establecido entre el proveedor y el comprador acerca de los productos o servicios descritos en el índice del proveedor. Permite al proveedor "superponer" en el índice atributos del artículo (como el precio) con valores acordados con el comprador. Además, permite a los proveedores y compradores dividir estas superposiciones basadas y estipuladas según una "clave de segmento" que resulte significativa para el comprador, como pueda ser el nombre de una planta o centro de costes.

Contract tiene los siguientes atributos:

effectiveDate	Fecha y hora en vigor del contrato, con formato ISO 8601.
expirationDate	Fecha y hora de caducidad del contrato, con formato ISO 8601.

Contract contiene uno o varios elementos ItemSegment, por ejemplo:

```
<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

El elemento ItemSegment contiene una lista de elementos ContractItem para un "segmento" concreto, donde un segmento representa una partición arbitraria de los artículos del contrato basada en una clave de segmento que se ha acordado entre el proveedor y el comprador.

ItemSegment tiene el siguiente atributo:

segmentKey (opcional)	Cadena acordada que se utiliza para dividir los precios personalizados.
---------------------------------	---

ContractItem

El elemento de artículo de contrato es una superposición concreta de artículo para un artículo del índice. Contiene un ItemID que identifica de forma única al artículo de índice en el sistema de adquisición que se va a superponer. Puede contener cualquier número de elementos Extrinsic que contenga el valor superpuesto para el atributo de artículo de índice.

Definiciones de gestión de subscripciones

Los intermediarios, como los concentradores de red de comercio electrónico, pueden gestionar catálogos de los proveedores que utilizan los sistemas de adquisición de los compradores. Estos intermediarios pueden proporcionar vínculos directos entre los sistemas de adquisición y los sistemas del proveedor. Este apartado contiene definiciones de elementos para la gestión de los datos y el contenido de los catálogos del proveedor. Estas definiciones se crean a partir de las definiciones anteriores para las respuestas cXML de petición-respuesta, los mensajes de sentido único y las definiciones de los catálogos.

Datos del proveedor

Las definiciones para la gestión de datos del proveedor están compuestas principalmente de los elementos SupplierListRequest, SupplierListResponse, SupplierDataRequest, SupplierDataResponse y SupplierChangeMessage. Estos elementos se describen a continuación con ejemplos en los que el intermediario es Ariba Network.

SupplierListRequest

SupplierListRequest solicita una lista de los proveedores con quienes el comprador ha establecido relaciones comerciales.

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

SupplierListResponse

SupplierListResponse contiene a los proveedores con quienes el comprador ha establecido relaciones comerciales.

```
<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 datos acerca de un proveedor.

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

SupplierDataResponse

SupplierDataResponse contiene los datos acerca de un proveedor.

```
<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

Este elemento se utiliza para notificar los cambios realizados en los datos del proveedor.

```

<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>

```


Suscripciones del catálogo

A continuación se describen las definiciones para la gestión de suscripciones del catálogo. En estos ejemplos, Ariba Network es el intermediario.

Subscription

Este elemento captura los metadatos de una única suscripción del catálogo. Sus subelementos son los siguientes:

- InternalID: ID interno único para el intermediario
- Name: Nombre de la suscripción
- ChangeTime: Fecha y hora en que se produjo algún cambio relacionado con la suscripción
- SupplierID: ID del proveedor que proporciona el catálogo
- Format: Formato del catálogo
- Description: Descripción del 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

Este elemento solicita la lista actual del comprador con las suscripciones del catálogo.

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

SubscriptionListResponse

Este elemento contiene la lista actual del comprador con las suscripciones del catálogo.

```
<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

Este elemento solicita el contenido del catálogo suscrito. La petición incluye el InternalID y el SupplierID del catálogo.

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

SubscriptionContentResponse

Este elemento recoge el contenido de un catálogo. El formato del catálogo puede ser CIF (formato de intercambio de catálogos) o cXML. Si es CIF, se codifica utilizando Base64 y se incluye como contenido de un elemento CIFContent. Si es cXML, los elementos Index y Contract se incluyen directamente.

```
<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

Este elemento indica al sistema de adquisición del comprador que ha cambiado un catálogo suscrito.

```
<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 tiene el siguiente atributo:

type	Tipo de cambio: new (por defecto), update o delete.
------	---

Definiciones de recuperación de mensajes

Algunos compradores no tienen puntos de entrada HTTP para recibir mensajes cXML que no proceden de los servidores de seguridad de la empresa. La especificación cXML permite estos entornos.

Este apartado es una introducción a las definiciones que permiten a los sistemas de origen dejar los mensajes en cola cuando los objetivos no pueden aceptar directamente los envíos HTTP. Los objetivos, a su vez, van extrayendo los mensajes como consideren conveniente.

GetPendingRequest

Este elemento extrae un conjunto de mensajes en espera del solicitante. El elemento `MessageType` y los atributos `lastReceivedTimestamp` y `maxMessages` controlan el tipo y hacen un recuento de los mensajes recuperados.

lastReceivedTimestamp (opcional)	Registro horario del último mensaje recibido.
maxMessages (opcional)	Número máximo de mensajes para una sola respuesta que el solicitante puede gestionar.

Una vez recibida la petición, el receptor devuelve el mensaje más antiguo (de los tipos especificados) con un registro horario igual o posterior al especificado. Si existen varios mensajes que cumplen este criterio, se pueden devolver varios mensajes, gracias al atributo `maxMessages`. El sistema de cola descarta todos los mensajes pendientes que sean de los tipos de mensaje especificados y que tengan registros horarios anteriores al especificado.

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

GetPendingResponse

Este elemento contiene uno o varios mensajes que esperan al 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>
```

Apéndice B

Nuevas funciones de cXML 1.1

cXML 1.1 contiene las siguientes categorías para las nuevas funciones:

- Cambios generales en cXML
- Cambios en Extrinsic
- Mejoras de la transacción de punchout
- Nuevas funciones para los pedidos de compra
- Nueva transacción para el estado de los pedidos de compra

Para obtener información detallada sobre cualquiera de los elementos o atributos aquí mencionados, consulte Apéndice A, "Especificación del lenguaje cXML".

Cambios generales en cXML

Los siguientes cambios afectan a amplias áreas del lenguaje cXML. Estas mejoras hacen que resulte más sencillo utilizar cXML a nivel internacional y dirigen la compatibilidad de la versión de cXML. Además, una nueva transacción de perfil permite a los clientes cXML consultar las posibilidades del servidor cXML.

Soporte mejorado para varios idiomas

Para obtener una mejor consistencia y soporte para varios idiomas, los elementos cXML, Status y ManufacturerName cuentan ahora con un atributo opcional `xml:lang`.

Ejemplo:

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

Para obtener más información:

"Status" en la página 56.

Este atributo especifica el idioma que los clientes cXML deberían usar con las respuestas y el que los sitios Web de punchouts deberían mostrar a los usuarios.

DTD (definiciones del tipo de documento) centralizadas

Anteriormente, no había ninguna ubicación central establecida para las DTD de cXML, por lo que los analizadores de cXML no podían recuperarlas automáticamente. Ahora, las DTD de todas las versiones de cXML están disponibles en ubicaciones permanentes en [cxml.org](http://xml.cxml.org).

Para obtener más información sobre las DTD de cXML, vaya a:

<http://xml.cxml.org/schemas/cXML/<versión>/cXML.dtd>

Para obtener más información:

"Validación respecto a definiciones del tipo de documento (DTD)" en la página 8.

donde *<versión>* es el número de versión completo de cXML, como por ejemplo 1.1.007.

Para obtener un rendimiento óptimo, los clientes cXML no deben extraer las DTD cada vez que se analizan los documentos cXML, si no que deben recogerlas a nivel local. Después de incluir una URL en [//xml.cxml.org/schemas/cXML](http://xml.cxml.org/schemas/cXML), no es necesario cambiar esa ubicación. Las DTD no se deben cambiar localmente, si no que se deben añadir nuevas ramificaciones.

Nueva transacción de perfil

Esta nueva transacción comunica la información básica acerca de los servidores cXML. Consta de dos documentos denominados ProfileRequest y ProfileResponse. Mediante esta transacción se recuperan las capacidades del servidor, incluidas la versión de cXML admitida, las transacciones admitidas y las opciones de esas transacciones.

Para obtener más información:

"Transacción de perfil" en la página 66

Nota: Todos los servidores cXML 1.1 **deben** admitir esta transacción.

Los clientes pueden también utilizar la transacción de perfil para realizar un "ping" en los servidores y verificar que están disponibles.

ProfileRequest

El documento ProfileRequest no tiene contenido. Tan sólo se dirige al servidor cXML adecuado mediante las credenciales de 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>
```

Este servidor responde mediante un documento ProfileResponse, descrito más abajo.

ProfileResponse

El documento ProfileResponse contiene transacciones que admite el servidor cXML, así como sus ubicaciones y muchos nombres de opciones que tienen un valor de cadena.

```
<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>
```

Nuevos códigos de estado

Para obtener más información:

"Status"
en la página 56

La especificación cXML 1.1 incluye nuevos códigos de estado para las transacciones para así lograr una comunicación más precisa entre el cliente y el servidor. Además, la especificación contiene descripciones más exactas de los códigos de estado de HTTP y cXML.

Nuevo atributo Type para miembros de mercado

El elemento Credential tiene un nuevo atributo type que especifica si el emisor o receptor es miembro de un mercado. Existen varios mercados y cada uno de ellos puede necesitar requisitos distintos para las credenciales.

Para obtener más información:

"Credential"
en la página 55

Este nuevo atributo type tiene un solo valor posible: marketplace. Se utiliza para diferenciar las credenciales de los miembros de mercado de las credenciales de los compradores o los proveedores. Los elementos Credential que no tienen el atributo type identifican a las compañías que no están asociadas a un mercado.

Las peticiones que proceden o se dirigen a un mercado deben identificar el mercado y la compañía a la que el miembro pertenece en los elementos de Credential To o From.

Cambios en Extrinsic

cXML 1.1 presenta elementos y atributos adicionales para los datos contenidos anteriormente en los elementos Extrinsic. Con estas adiciones se modifica la información que se había enviado previamente mediante Extrinsic en la especificación base.

Asimismo cXML 1.1 ofrece soporte para Extrinsic en el nivel de cabecera.

Nuevo elemento Contact

Los elementos OrderRequestHeader, PunchOutSetupRequest e ItemOut incluyen ahora elementos Contact opcionales, que especifican a una persona o un grupo con los que ponerse en contacto para obtener más información.

Para obtener más información:

"Contact"
en la 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>
```

El atributo `role` especifica la posición o puesto de la persona de contacto. Las funciones admitidas son `endUser`, `administrator`, `purchasingAgent`, `technicalSupport`, `customerService` y `sales`.

Atributo `requisitionID` admitido

`requisitionID` se admite ahora completamente. Es un atributo opcional de los elementos `OrderRequest` e `ItemOut` que identifica la solicitud del comprador para un artículo de línea.

Para obtener más información:

"OrderRequestHeader"
en la 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>
```

Si utiliza `requisitionID` en el nivel de `OrderRequestHeader`, no lo incluya en el nivel de `ItemOut`.

Resumen de la información sobre Extrinsic que se ha modificado

La siguiente tabla contiene los cambios realizados a los elementos Extrinsic más utilizados.

Extrinsic anterior	Elemento o atributo cXML existente o nuevo
Requested Ship Date	Atributo ItemOut requestedDeliveryDate
ship complete	Atributo nuevo shipComplete
ReqNumber	Atributo existente requisitionID
Requisition #	Atributo existente requisitionID
Name	Elemento nuevo Contact
Phone	Elemento nuevo Contact
E-mailAddress	Elemento nuevo Contact
Buyer Name	Elemento nuevo Contact
Buyer Phone	Elemento nuevo Contact
OriginalRequester	Elemento nuevo Contact
Requester Phone Number	Elemento nuevo Contact
ETA	Atributo existente requestedDeliveryDate

Extrinsic de nivel de cabecera

Anteriormente, los elementos Extrinsic del pedido de compra sólo podían aparecer en el nivel de artículo de línea del producto. Ahora pueden aparecer en cualquier lugar del documento OrderRequest.

Utilice esta nueva función para los datos Extrinsic que afectan a todo el pedido de compra.

Los elementos Extrinsic que tienen el mismo nombre no pueden aparecer al mismo tiempo en el nivel de cabecera y en el de artículo de línea dentro del documento OrderRequest.

Mejoras de la transacción de punchout

Las transacciones de punchout se han ampliado para poder admitir las compras de nivel de sección y de nivel de producto. Asimismo, cXML también admite las sesiones de punchout canceladas.

PunchOutSetupRequest mejorado

El documento PunchOutSetupRequest que envían los compradores se ha modificado para mejorar la flexibilidad de las transacciones de punchout. La URL especificada en PunchOutSetupRequest se ha dejado de utilizar. Los servidores cXML ignorarán este elemento en el futuro.

La nueva metodología utiliza la identidad (de la Credential) del proveedor. Los concentradores de red de comercio electrónico reciben el documento PunchOutSetupRequest, leen el ID de proveedor, extraen la URL del sitio Web de punchout a través de la información de la cuenta del proveedor y envían el documento PunchOutSetupRequest a esa URL. Este concentrador (no el comprador) especifica la URL del sitio Web de punchout, que resulta más flexible.

Los concentradores de red de comercio electrónico permiten a los proveedores almacenar las URL de sus sitios Web de punchout.

Elemento SelectedItem

Como parte de las mejoras de PunchOutSetupRequest, cXML admite mejor el punchout de nivel de tienda, sección y producto. El nuevo elemento SelectedItem opcional de este documento permite a los proveedores especificar el punchout de todo un establecimiento o cualquier oferta de subconjuntos de productos. Las aplicaciones de adquisición pueden incluir el elemento SelectedItem en los documentos PunchOutSetupRequest y los sitios de punchout pueden emplearlo para determinar qué productos mostrar a los usuarios. Si no existe ningún SelectedItem, los proveedores deben presentar todas las ofertas de productos (nivel de tienda).

El SelectedItem contiene un elemento ItemOut, que incluye un ItemID. Por ejemplo:

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

Para obtener más información:

"SelectedItem" en la página 79

Para obtener el contenido del elemento `SelectedItem`, las aplicaciones de adquisición utilizan el `ItemID` (`SupplierPartID` y `SupplierPartAuxiliaryID`) del catálogo de índices de punchout. No es necesario realizar cambios en el catálogo.

Las aplicaciones de adquisición deben enviar al comienzo el nuevo elemento `SelectedItem` y la antigua URL de punchout en la `PunchOutSetupRequest`.

Los concentradores de red de comercio electrónico sólo utilizan la antigua URL para los proveedores que no han almacenado aún los destinos de la URL de punchout.

PunchOutOrderMessage vacío

cXML admite ahora documentos con un `PunchOutOrderMessage` vacío para que los usuarios puedan finalizar las sesiones de punchout de compra sin necesidad de seleccionar ningún artículo. Anteriormente, el `PunchOutOrderMessage` se tenía que devolver con al menos un artículo.

Los proveedores pueden implementar un botón "Cancelar" para generar un documento `PunchOutOrderMessage` vacío. De este modo, el sitio de punchout y la aplicación de adquisición sabrán en qué momento un usuario ha cancelado una sesión de compra y podrán suprimir el carro de la compra, eliminar los artículos de la solicitud y realizar otras tareas de reorganización.

Nuevo campo oculto cXML-base64

El nuevo campo oculto `cXML-base64` admite documentos internacionales en el FORM POST que devuelven los sitios de punchout. Los documentos cXML que contienen símbolos distintos de "us-ascii" deben utilizar este campo en lugar del campo oculto `cXML-urlencoded`. Esta operación alternativa presenta casi la misma semántica, pero todo el documento está codificado mediante Base64 para todo el transporte y no lo está mediante HTML para el navegador, ni tampoco mediante la URL para el servidor Web de recepción.

Para obtener más información:

"URL-Form-Encoding"
en la página 61

La codificación mediante Base64 mantiene la codificación de caracteres original de los documentos cXML. A pesar de que no llega ningún parámetro "charset" con la información enviada, el documento descodificado (después de que se elimine la codificación de la transferencia) se puede tratar como un tipo de medio "application/xml". Esta codificación permite al analizador de recepción respetar todos los atributos de "codificación" especificados en la declaración XML. Para este campo, así como para todos los documentos "application/xml", la codificación de caracteres por defecto es UTF-8.

Nuevas funciones para los pedidos de compra

Se han mejorado los documentos de pedidos de compra cXML para admitir funciones de petición.

Nuevo atributo lineNumber

lineNumber es un nuevo atributo opcional para el elemento ItemOut. Especifica la ubicación de un artículo incluido en el pedido del comprador. Se puede elegir cuándo se realiza el pedido (por eso no es significativo para las sesiones de punchout).

Para obtener más información:

"ItemOut" en la página 74

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

Los números de línea de los artículos no se deben modificar en los pedidos de cambio, para de este modo poder identificar el cambio.

Datos adjuntos de los pedidos de compra

A menudo, los compradores necesitan aclarar los pedidos de compra con notas, dibujos o faxes asociados. Para ello, las aplicaciones de adquisición pueden adjuntar archivos de cualquier tipo a los pedidos de compra cXML a través de MIME (Multipurpose Internet Mail Extensions, extensiones de correo Internet de múltiples objetivos).

cXML sólo contiene referencias a partes externas MIME enviadas en un sobre MIME de varias partes (junto con el documento cXML, mediante un fax o un mensaje de correo electrónico).

El nuevo elemento Attachment contiene referencias a los datos adjuntos:

Para obtener más información:

"Transmisión de los datos adjuntos" en la página 53

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

Los concentradores de red de comercio electrónico reciben los datos adjuntos y los reenvían al proveedor o los almacenan para su posterior recuperación en línea.

Para obtener más información sobre la normativa MIME, consulte los siguientes sitios Web:

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

Nuevo atributo shipComplete

shipComplete es un nuevo atributo opcional para el elemento OrderRequestHeader que indica a los proveedores cómo rellenar un pedido cuando todos los artículos están disponibles. Este atributo no permite los envíos de artículos parciales.

Para obtener más información:

"OrderRequestHeader" en la 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>
```

Nuevo elemento ShortName

ShortName es un nuevo elemento opcional para los elementos Description de los artículos.

```
<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 obtener más información:

"ItemDetail" en la página 84

ShortName es el nombre abreviado del artículo, que debe tener un extensión de entre 30 caracteres (recomendado) y 50 (máximo), para que quepa en la lista de productos que reciben los usuarios. Anteriormente, como sólo había un elemento Description, las descripciones de mayor extensión se truncaban en cualquier punto.

Las aplicaciones de adquisición y otros clientes cXML deben mostrar el ShortName en lugar de truncar el texto de la Description en los campos que presentan un espacio limitado. Si no se proporciona el ShortName, los clientes cXML pueden seguir truncando el texto de la Description.

Los creadores de catálogos no deben utilizar el ShortName para duplicar la información de Description, sino que deben usar el ShortName para asignar un nombre a cada producto y la Description para describir los detalles de ese producto.

El formato de los catálogos CIF 3.0 se ha ampliado para admitir el ShortName. El nombre del campo CIF es Short Name.

Nueva transacción para el estado de los pedidos de compra

cXML 1.1 ofrece una nueva transacción para enviar el estado de los pedidos de compra a los concentradores de red de comercio electrónico.

Esta transacción se basa en el nuevo elemento OrderReference, que asocia la actualización del estado con la última OrderRequest que ha enviado el comprador.

Nuevo elemento OrderReference

El nuevo elemento OrderReference asocia la actualización del estado a un documento OrderRequest concreto. Repite los atributos necesarios de los elementos OrderRequestHeader y cXML a partir del documento OrderRequest y añade un identificador opcional que ha generado el proveedor.

Para obtener más información:

"DocumentReference" en la página 85

```
<OrderReference
  payloadID="0c300508b7863dccb_14999"
  timestamp="2000-01-08T14:36:05-07:00"
  orderID="DO4321"
  orderDate="2000-01-08T13:56:23-07:00"
  supplierOrderID="27-33-00-08-01"
/>
```

La nueva transacción StatusUpdateRequest utiliza este elemento.

Nueva transacción StatusUpdateRequest

Para obtener más información:

"StatusUpdateRequest" en la página 86

Los socios de procesamiento de pedidos (como los proveedores de servicios de fax o EDI) envían la nueva transacción StatusUpdateRequest a los concentradores de red de comercio electrónico para establecer el estado del pedido de compra. Esta operación afecta al indicador de estado del pedido que incluye el concentrador, visible para los compradores y para los proveedores. Asimismo, los proveedores pueden enviar esta transacción para que los compradores puedan consultar el estado del procesamiento de documentos de su organización.

Nota: Esta transacción informa a las partes implicadas de los cambios realizados en el estado de la entrega y el procesamiento de los *documentos de los pedidos de compra*, no del estado de compra de los artículos reales.

Existe un cambio de especial importancia: cuando un concentrador intermedio reenvía un documento OrderRequest correctamente, puede informar al emisor original o a un concentrador anterior acerca de lo ocurrido. Además, puede que las transiciones a través de varias colas y fases de procesamiento de un proveedor o concentrador sean también de gran importancia para el comprador.

Nuevo elemento Followup

Para obtener más información:

"Followup" en la página 73

Followup es un elemento de OrderRequestHeader que especifica la URL a la que se deben enviar los documentos StatusUpdateRequest futuros. Ésta es la ubicación de entrada de todos los documentos posteriores que hagan referencia al documento OrderRequest actual.

Indice

A

Accounting, elemento 76
atributo payloadID 85
Attachment, elemento 72

B

BillTo, elemento 71
BrowserFormPost, elemento 79
BuyerCookie, elemento 78, 81

C

cabecera cXML, idioma 30
caracteres, codificación 52
Charge, elemento 76
Classification, elemento 21
code, atributo 57
codificación de caracteres 52
codificación de formularios 36
codificación mediante URL 62
Comments, elemento 72
comprador y proveedor, cookies 29, 38
Contact, elemento 72
Contract, elemento 92
cookies, comprador y proveedor 29, 38
corporateURL, atributo 88
Credential, elemento 55
cXML, elemento 50
cXML-base64, campo oculto 36, 63
cXML-urlencoded, campo oculto 36, 62

D

datos adjuntos a pedidos de compra 45
deploymentMode, atributo 56, 61
Description, elemento 21, 84
Distribution, elemento 76
DocumentReference, elemento 85
domain, atributo 55
DTDs (Document Type Definitions,
definiciones del tipo de
documento) 8

E

EDI (X.12 Electronic Data Interchange,
intercambio electrónico de datos) 4
editores de XML 10
effectiveDate, atributo 67, 92
ejecución del comando “ping” en servidores
con la transacción de perfil 9
expirationDate, atributo 92
Extrinsic, elemento 26, 39, 73, 79

F

fecha y hora, formato 51
Followup, elemento 73
formulario, codificación 62
formularios HTML, codificación 36
From, elemento 23
From, To y Sender, elementos 55

G

GetPendingRequest, elemento 100
GetPendingResponse, elemento 101

H

Header, elemento 54
herramientas para trabajar con XML 10
hora y fecha, formato 51
HTML, codificación de formulario 62

I

id, atributo 76
idioma, en la cabecera cXML 30
Index, elemento 90
IndexItemAdd, elemento 90
IndexItemDelete, elemento 90
IndexItemDetail, elemento 91
IndexItemPunchout, elemento 90
inReplyTo, atributo 61
IsoCountryCode, elemento 65
IsoLanguageCode, elemento 65
ItemDetail, elemento 84, 91
ItemID, elemento 83
ItemIn, elemento 83
ItemOut, elemento 74
ItemSegment, elemento 92

L

lastReceivedTimestamp, atributo 100
lineNumber, atributo 74, 83

M

maxMessages, atributo 100
Message, elemento 61
MIME, datos adjuntos 45, 52

O

operation, atributo 24, 78
operationAllowed, atributo 82

orderDate, atributo 70
orderID, atributo 70, 85
OrderMethods, elemento 89
OrderRequest, elemento 68
OrderRequestHeader, elemento 69

P

Página de inicialización 30
Página de inicio 33
Página del receptor de pedidos 37
Página del remitente 34
payloadID, atributo 23, 50
Payment, elemento 71
pedidos de compra 41–45
 datos adjuntos 45
presupuesto de pedidos 19
ProfileRequest, elemento 66
ProfileResponse, elemento 67
proveedor y comprador, cookies 29, 38
PunchoutDetail, elemento 91
PunchOutOrderMessage 27
PunchOutOrderMessage, elemento 81
PunchOutOrderMessageHeader,
 elemento 81
PunchOutSetupRequest 22
PunchOutSetupRequest, elemento 77
PunchOutSetupResponse 27
PunchOutSetupResponse, elemento 80

Q

quantity, atributo 74, 83

R

registro de pedidos 20
Request, elemento 56
requestedDeliveryDate, atributo 74
requestName, atributo 68
requisitionID, atributo 70, 74
Response, elemento 56
role, atributo 72

S

segmentKey, atributo 92
 SelectedItem, elemento 26, 79
 Sender, elemento 23
 Sender, To y From, elementos 55
 shipComplete, atributo 70
 Shipping, elemento 71
 ShipTo, elemento 71
 ShortName, elemento 84
 sitio Web, cxml.org 8
 StartPage, elemento 80
 Status, elemento 56
 StatusUpdateRequest, elemento 86
 storeFrontURL, atributo 88
 Subscription, elemento 97
 SubscriptionContentRequest, elemento 98
 SubscriptionContentResponse, elemento 99
 SubscriptionListRequest, elemento 97
 SubscriptionListResponse, elemento 98
 Supplier, elemento 88
 SupplierChangeMessage, elemento 96
 SupplierDataRequest, elemento 94
 SupplierDataResponse, elemento 94
 SupplierID, elemento 21
 SupplierListRequest, elemento 93
 SupplierListResponse, elemento 94
 SupplierLocation, elemento 89
 SupplierPartAuxiliaryID, elemento (cookie de proveedor) 29, 38
 SupplierPartAuxiliaryID, elemento (cookie del proveedor) 91
 SupplierSetup, dirección URL 26
 SupplierSetup, elemento 80

T

Tax, elemento 71
 timestamp, atributo 23, 50
 To, elemento 23
 To, From y Sender, elementos 55
 Total, elemento 71
 transacción de perfil 9
 Transacción, elemento 68
 type, atributo 55

U

Unidad de medida 29, 65
 URL, elemento 66
 utilidades para utilizar con XML 10

V

validación de cXML 8
 version, atributo 50

X

XML 15
 xml:lang 30
 xmlLanguageCode, elemento 65



www.cxml.org