

SUPPLIER INTEGRATION SPECIFICATION

CXML INTEGRATIONS

PRODUCT VERSION 17.3



Table of Contents

WELCOME	1
OVERVIEW	1
SUPPLIER INTEGRATIONS TEAM	2
NETWORK MEMBER SERVICES	2
Implementation Services	2
Integration Testing	2
Integration Support (Post Go-Live)	3
NETWORK MEMBER TOOLS	3
Supplier Network Portal	3
Catalog Management Template	3
Customer Enablement Settings	4
JAGGAER SOLUTIONS & CLIENT CONFIGURATIONS	5
JAGGAER SOLUTIONS	5
Spend Director	5
Requisition Manager	5
Order Manager	5
Accounts Payable Director	5
Total Contract Manager	6
Inventory Management	6
Spend Radar	6
Total Supplier Manager	6
Sourcing Director	7
Advanced Sourcing Optimizer	7
CLIENT CONFIGURATIONS	7
Full Suite Configuration	7
Spend Director / Order Manager Configuration	8
Spend Director Only Configuration	9
INTEGRATION RESOURCES	11
GENERAL INTEGRATION REQUIREMENTS	11
SUPPORTED CXML VERSIONS	11
Obtaining the DTD	12
SUPPLIER DEVELOPMENT RESOURCES	12

cXML.org	13
W3C.org	13
cXML Validation	14
UNSPSC.org	14
Third Party Solutions	14
DATA STANDARDS	16
UNITS OF MEASURE (UOM)	16
JAGGAER UOM Standards	16
UOM Mapping	16
Unmapped UOM Values	17
Product Size in UOM	17
SUPPORTED CURRENCIES	18
ISO 4217 Currency Standards	19
DATE/TIMESTAMP FORMAT	19
NUMERICAL FORMATS & CALCULATIONS	19
Numerical Formats	19
Calculations	20
PRODUCT CATEGORIZATION	21
PRODUCT FLAGS	22
EH&S Flags	22
Additional Flags	24
DATA PARSING (PCDATA VS. CDATA)	25
PCDATA	25
CDATA	26
Encoding & Special Characters	26
SUPPLIER INTEGRATION POINTS	27
PUNCHOUT	28
What is a punch-out?	28
PunchOut Requirements	28
MOBILE PUNCHOUT	30
Mobile Punchout Requirements	31
P3P COMPLIANCE	31
P3P Defined	31

Importance of P3P Compliance	32
Becoming P3P Compliant	32
P3P and Internet Explorer	32
JAGGAER Policy	33
PUNCHOUT SITE BEST PRACTICES	33
TRACKING USERS	34
COOKIES & SUPPLIERPARTAUXILIARYID	35
Browser Cookies	35
SupplierPartAuxiliaryID (SPAID)	35
PUNCHOUTSETUPREQUEST	36
Sample Message (JAGGAER Default)	36
Sample Message (Expanded)	38
PunchoutSetup Message Elements	40
PUNCHOUTSETUPRESPONSE	42
Sample Message - PunchOutSetupResponse	42
PUNCHOUTORDERMESSAGE	43
Sample Message (No Tax or Shipping Estimates)	43
Sample Message (Order Level Tax & Shipping Estimates)	45
Sample Message (Line Level Tax & Shipping Estimates)	47
Sample Message (Shipping as Line Item in Cart)	49
Sample Form Submit (Browser Form Post)	51
DIGITAL SIGNATURE	54
Sample Digital Signature XML	54
PUNCHOUTORDER MESSAGE ELEMENTS	56
EDIT / INSPECT	59
Edit - Expected Behavior	59
Inspect - Expected Behavior	59
PunchOutOrderMessage Changes	59
PunchOutSetupRequest Message Changes	60
Sample Message (Edit/Inspect)	60
SMARTFIND	64
PROXY CATALOG & ACCESSIBILITY FILES	64
Proxy Catalog	64
Proxy Catalog Accessibility	64

LEVEL 2 PUNCHOUT	65
Sample Message (Level 2 PunchOut)	66
LIVEPRICE	69
LIVEPRICE INTEGRATION REQUIREMENTS	69
SUPPLIER NETWORK SETTINGS	69
LivePrice Support	70
Items Per LivePrice Call	70
LivePrice Cache Lifespan	70
LivePrice Request Message	70
Sample Message- LivePrice Request	71
Message Elements - LivePrice Request	72
Sample Message - LivePrice Response	73
Message Elements - LivePrice Response	74
AVAILABILITY CALL	76
AVAILABILITY CALL INTEGRATION REQUIREMENTS	76
AVAILABILITY CALL REQUEST MESSAGE	76
Sample Message - Availability Call (Request)	77
Message Elements - Availability Call (Request)	78
AVAILABILITY CALL RESPONSE MESSAGE	79
Sample Message - Availability Call (Response)	79
Message Elements - Availability Call (Response)	80
PURCHASE ORDER DELIVERY	82
CXML PURCHASE ORDER REQUIREMENTS	82
SHIPTO ADDRESSES	83
Dynamic vs. Static Address Lines	83
Address Code Lookup	84
Address Formatting	84
Address Variances	85
PAYMENT METHOD	87
Purchase Order (Direct Billing)	87
BillTo Addresses	87
Credit Card	88
NON-CATALOG ORDER DELIVERY	89

ORDER FAILURE MANAGEMENT	90
CONTACT ROLES	90
EMAIL CONFIRMATIONS	91
ORDERREQUEST MESSAGE	91
Sample Message (Order Request)	91
Message Elements - Order Request	96
RESPONSE TO AN ORDER REQUEST	105
Sample Message (Response)	105
CONFIRMATION REQUEST	106
CONFIRMATIONREQUEST MESSAGE	106
Confirmation Request Integration Requirements	106
Sample Message (Line Detail)	107
Message Elements - Confirmation Request	111
Response to Confirmation Request	113
SHIP NOTICE REQUEST	115
SHIP NOTICE REQUEST INTEGRATION REQUIREMENTS	115
SHIPNOTICEREQUEST MESSAGE	116
Sample Message (Single order, multiple lines)	116
Sample Message (multiple tracking numbers)	118
Sample Message (consolidated shipment)	120
Message Elements - ShipNotice Request	121
RESPONSE TO SHIPNOTICEREQUEST	125
INVOICES AND CREDIT MEMOS	126
Integration Requirements for Invoice Integration	126
DOCUMENT STRUCTURE	127
Category	127
Level	127
Purpose	128
Additional Fees	128
Miscellaneous Fees	129
DOCUMENT ACCEPTANCE PROCESS	130
Non-PO Definitions	131
Part Substitutions	132

SETTLEMENT EXTRINSICS	132
Standard Extrinsic	132
Custom Extrinsic	133
INVOICEDetailRequest MESSAGE	134
Sample Message (Most Common Structure)	134
Sample Message (Line Level Additional Fees)	140
Sample Message (Header Invoice, Summary Level Fees)	149
Sample Message (Header Invoice, Line Level Fees)	155
Sample Message (Summary Invoice, Line Level Fees)	161
Sample Message (Miscellaneous Fees)	170
Sample Message (Credit, Most Common Structure)	177
Sample Message (Negative Invoice, Credit Alternative)	183
Message Elements - Invoice Detail Request	189
RESPONSE TO INVOICEDetailRequest MESSAGE	204
APPENDIX	206
JAGGAER RESPONSE CODES	206
Code Definitions	206
Codes by Integration Type	206
TROUBLESHOOTING GUIDES	225
PunchOut Troubleshooting Guide	226
Edit/Inspect Troubleshooting Guide	232
Level 2 PunchOut Troubleshooting Guide	232
LivePrice Troubleshooting Guide	234
Availability Call Troubleshooting Guide	235
Purchase Order (OrderRequest) Troubleshooting Guide	237
Confirmation Request Troubleshooting Guide	238
ShipNotice Request Troubleshooting Guide	238
Invoices/Credit Memos Request Troubleshooting Guide	238

Welcome

The primary purpose of this document is to define the functional and technical specifications for a standard cXML integration. cXML (commerce eXtensible Markup Language) is a protocol that defines eCommerce document formats and is intended for the exchange of transaction data between procurement applications and suppliers. cXML is an open source language that is published and freely available on the Internet.

Included in this document is an overview of supplier services & tools, the JAGGAER modules and how they integrate with the client ERP system, integration requirements, sample messages by integration point, and suggested supplier development resources.

Overview

This section of the handbook gives an overview of JAGGAER Supplier Integrations team and JAGGAER products in general, including basic information about products and typical client configurations.

Supplier Integrations Team

The JAGGAER Supplier Integrations team is dedicated to helping our customers integrate with the key suppliers they do business with. Our resources are specialized in the area of supplier enablement and have a solid understanding of both the technical and business processes required to successfully integrate catalog, purchase order, and invoicing transactions.

Once a supplier has determined integration requirements can be satisfied and will move forward with customer integration, JAGGAER will provide the supplier with the tools and services available to members of the JAGGAER Supplier Network.

Network Member Services

Implementation Services

Before a project can be initiated, the customer must first formally request the supplier integration with JAGGAER. This is done by contacting their Client Partner or logging a ticket in Salesforce. Once submitted, a JAGGAER resource will be assigned within two weeks and will contact the supplier to determine readiness to proceed.

Before the implementation can begin, the supplier must have completed development and be prepared to test the integration, as well as have available resources to implement the project. A kick-off call will be held with all parties to identify and review the integration requirements, determine the implementation schedule/timeline, and develop the Project and Test Plans. The Project Plan defines the scope of the project, lists the test scenarios to be completed, and outlines the roles and responsibilities of the project team members. The assigned JAGGAER resource is responsible for maintaining both the Project and Test Plans and will serve as the overall Project Manager for the implementation.

Any proposed project delays will be evaluated for impact and addressed by the entire project team. Repeated, lengthy delays may significantly impact resource availability for all parties

Integration Testing

JAGGAER will provide the Integration Test Plan and coordinate with the supplier to execute integration testing. The test plan will consist of a set of basic user scenarios as well as any specific test scenarios as required by the customer, the supplier, and JAGGAER.

JAGGAER maintains both a test and production environment. This allows the customer to test new integrations without concern for erroneous data winding up in the production environment. In order to integrate with customers, the supplier must be able to execute end-to-end testing in a separate environment in order to ensure that test data is not processed as genuine, production transactions.

The integration must meet minimum requirements and have successfully completed all test scenarios outlined in the Test Plan before promotion to production can be considered. The fully

tested integration will be validated against the Project Plan to ensure that the agreed upon scope has been delivered.

Integration Support (Post Go-Live)

During the implementation phase, the assigned JAGGAER Project Manager is the primary contact; however, once the integration is live and in the production environment, the integration will move to a support status. For any production integration performance issues, the supplier can contact JAGGAER Customer Support at support@JAGGAER.com, or (800) 233-1121, option #3.

Network Member Tools

Supplier Network Portal

During implementation, the supplier will be provided the opportunity to register for and access the Supplier Network Portal. Suppliers can use the Network Portal to perform a variety of tasks during implementation and post go-live. These tasks include:

- Access documentation on JAGGAER services, tools, and integration specifications
- Manage catalog data (Hosted Catalog Suppliers)
- Maintain and update UOM mappings
- View reports on catalog spend and customer activity
- View order delivery information & manage order failures
- Process orders delivered to the supplier via the JAGGAER Supplier Portal
- Invoice orders from the JAGGAER Supplier Portal (upon client request only)
- View cXML Invoice failures
- Manage integration credentials and URL's
- Test punch-out connectivity

If there are questions about the JAGGAER Supplier Network Portal, refer to the **Online Training and Support** page within the portal or contact a member of the JAGGAER Support team at (800) 233-1121, option #3 or support@JAGGAER.com.

Catalog Management Template

The Catalog Management Template (CMT) is an Excel template designed to help suppliers format catalog content with pricing, and is provided to those that will participate in the network with a hosted catalog. A hosted catalog is supplier catalog content that is hosted on the JAGGAER servers and is made accessible to requesting client organizations. The CMT includes reference files and information that outline the required data sets for the supplier hosted product offering.

The CMT also applies in part to suppliers who will participate in Level 2 PunchOut or LivePrice integration options as part of SmartFind.

The CMT is available via the Online Training and Support Page in the Supplier Network Portal, or can be provided to the supplier directly upon request.

Customer Enablement Settings

In the JAGGAER Supplier Network, you may access data related to the enablements you have in place with JAGGAER customers. This includes data for punch out, purchase order, and invoice integrations. You may also use this area of the JAGGAER Supplier Network Portal to enter settings and other important enablement details that customers know as part of the supplier implementation process directly into the Network. JAGGAER can then apply this information to the system during the supplier enablement process with the customer.

In addition, the JAGGAER Supplier Network, there are tools provided for testing integrations. For punch out integrations, you have the ability to test the configuration settings and see the cXML response message returned to confirm the configurations are correct.

For more information about the customer enablement menu in the Supplier Network Portal, please refer to the **Supplier Network Handbook** found on the **Online Training and Support** page in the portal.

JAGGAER Solutions & Client Configurations

JAGGAER's on-demand eProcurement software provides clients with the control and visibility needed to streamline catalog management, direct spend to preferred vendors, drive higher compliance with contracts, and tighten control throughout the entire source-to-settle process. Products can be deployed together or separately in various configurations to meet the client's specific needs. The following sections briefly describe the JAGGAER solutions and illustrate several common client configurations.

JAGGAER Solutions

Spend Director

Spend Director is a powerful and configurable application that enables clients to achieve supplier enablement, catalog management, and spend management goals. This module provides clients with the tools necessary to direct spend to preferred suppliers, categorize and customize content views, and to provide users with a consistent and optimal approach to the procurement process. This module is the repository for user profile information and other custom client data. Many client configurations include this module and is the user interface to the shopping experience.

Requisition Manager

Requisition Manager is a configurable application that automates and streamlines requisition creation, submission, and approval processes. This module offers clients the ability to customize requisition workflow to meet specific business process needs, including budget confirmation and hazardous material clearance.

Order Manager

Order Manager is a configurable application that automates and streamlines purchase order placement, tracking, and management processes. Clients with this module have licensed JAGGAER to deliver purchase orders to their supplier base on their behalf through a single, secure integration point.

Accounts Payable Director

Accounts Payable Director automates and streamlines receiving, matching, and invoicing processes. Clients with this module have licensed JAGGAER to receive electronic invoices on their behalf through a single, secure integration point. Suppliers that are not yet able to invoice electronically will continue to route invoices directly to their clients as they do today.

Total Contract Manager

Total Contract Manager is a contract lifecycle management software solution that is designed to improve the contracting process and sets the standard for enterprise contract management software best practices. With over 130 customers across many industries, JAGGAER's Total Contract Manager helps organizations increase visibility into every aspect of their contracts, reduce manual touch-points throughout the lifecycle of the contracts, and mitigate the exposure to legal and regulatory risk. Total Contract Manager enables organizations to truly manage the complete lifecycle of their contracts.

Inventory Management

JAGGAER offers unique inventory management solutions for managing internal stockrooms and inventory, integrating a customer's ERP item master to present ERP inventory to buyers, and for managing enterprise reagents and chemical stock globally.

- JAGGAER **Supplies Manager** allows an organization to streamline all inventory management processes related to internal order fulfillment. Customers can manage multiple stockrooms using a simple, single inventory tracking application that is fully integrated with our eProcurement software application, providing a unified inventory management system for all internal and external goods and ordering.
- JAGGAER offers the **Enterprise Reagent Manager** as a material management and procurement tool for chemical reagents and other materials used in the research process. It optimizes inventory use and promotes regulatory compliance while managing the risks associated with hazardous materials. Supplier catalogs and, where negotiated, preferred pricing are available to scientific staff alongside internal inventory, libraries and commercial databases through a single search with JAGGAER's chemical inventory software.

Spend Radar

Spend Analysis management solutions allow procurement and accounting professionals to conduct the comprehensive analysis necessary to drive and measure cost savings. JAGGAER eliminates the traditional roadblocks to spend management by delivering clean, actionable spend data based on consistent and accurate catalog content.

Total Supplier Manager

JAGGAER Total Supplier Manager spans the supplier life cycle, using intelligent questionnaires, streamlined workflows and process automation to manage vendor registration and onboarding, reduce supplier risk and enhance a customer's sourcing process. It also powers supplier diversity programs by giving an organization complete visibility into local and global diverse and non-diverse suppliers.

Sourcing Director

JAGGAER Sourcing Director is an on-demand sourcing and bid management solution that helps customers achieve significant process efficiencies and cost savings over traditional paper-based bid and auction processes. This solution includes secure support for sealed bidding, informal quotes and reverse auctions. From a standard web browser, suppliers register and respond to bid opportunities providing increased accuracy and easy capture of award criteria.

Advanced Sourcing Optimizer

JAGGAER Advanced Sourcing Optimizer is an on-demand advanced sourcing software that enables supply-chain driven companies to create highly customizable requests for proposals and run scenarios that balance a number of factors including supplier cost, risk, value, and other important criteria with internal business rules and preferences. This sourcing optimization capability enables users to make better supplier award decisions faster, split awards and uncover the real cost impact of sourcing decisions.

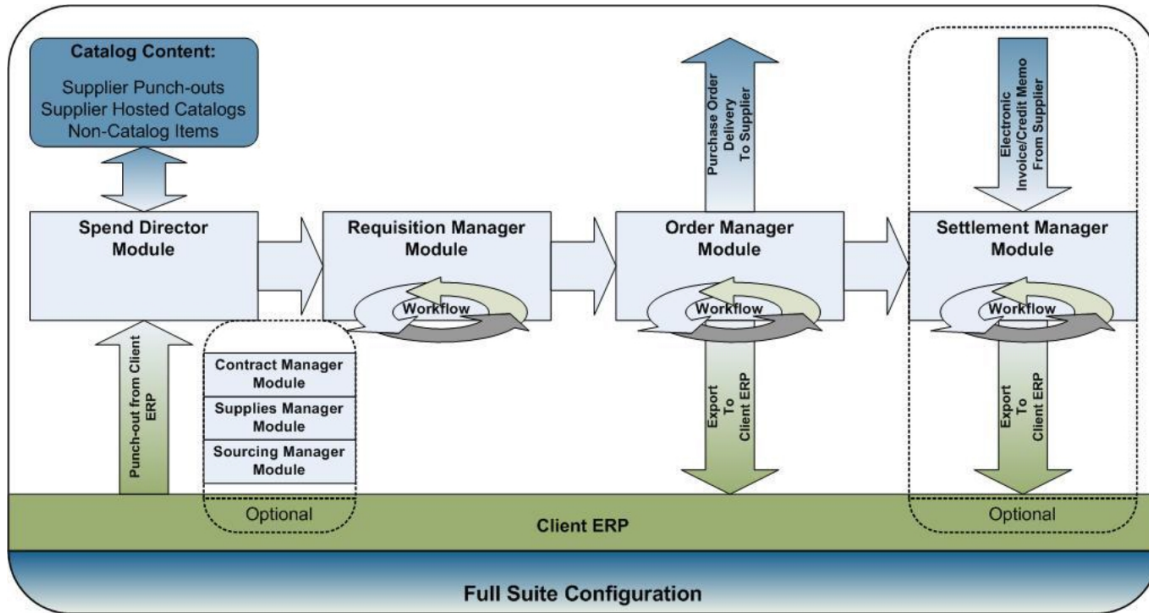
Client Configurations

Full Suite Configuration

In the Full Suite configuration, the JAGGAER application generates the requisition and moves through the client-defined workflow approval steps before creating the purchase order document. Purchase orders are delivered to suppliers on behalf of the client via email, fax, cXML, or to the Supplier Portal. Client ShipTo and BillTo addresses are stored in the JAGGAER application and are passed to the supplier on the purchase order document.

If the client has licensed an optional Accounts Payable product, invoices from suppliers can be received electronically via cXML or via the Supplier Portal. If the client configuration does not include an Accounts Payable product or the supplier is unable to deliver electronically, invoices are delivered directly from the supplier to the client, bypassing the JAGGAER application.

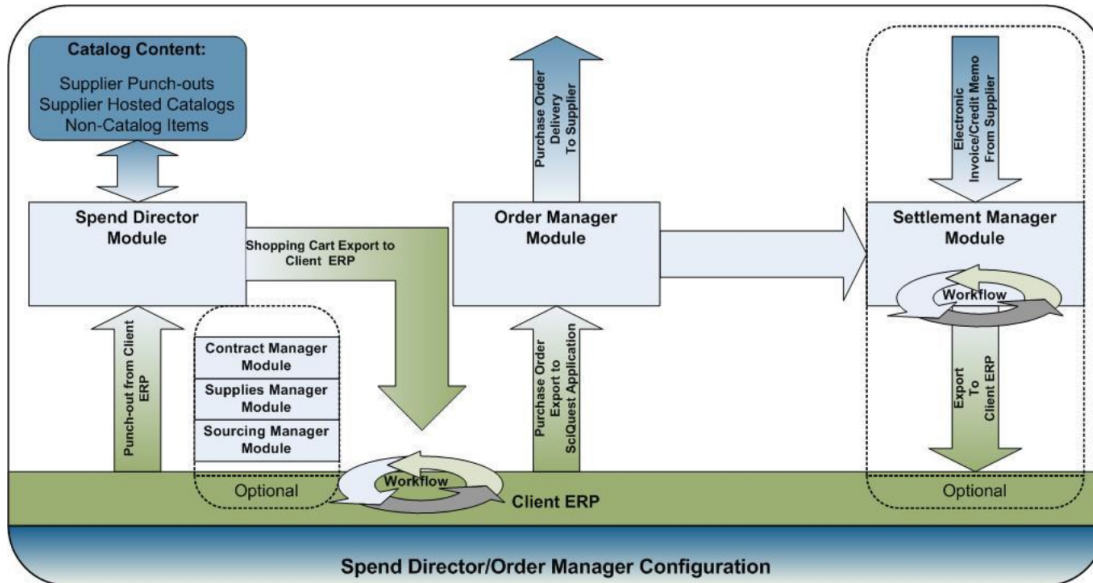
In most configurations, transaction data is exported from the JAGGAER application to the client's system of record (ERP).



Spend Director / Order Manager Configuration

In the Spend Director/Order Manager configuration, the JAGGAER application exports the shopping cart to the client ERP system, where it generates a requisition, moves through workflow approvals, and is then converted to a purchase order. The purchase orders are then exported from the client ERP and imported into the JAGGAER application, where they are delivered to suppliers on behalf of the client via fax, email, cXML, or to the Supplier Portal. ShipTo and BillTo addresses are **not** stored in the JAGGAER application; however, address information is imported with the purchase order document and will be passed to the supplier upon delivery.

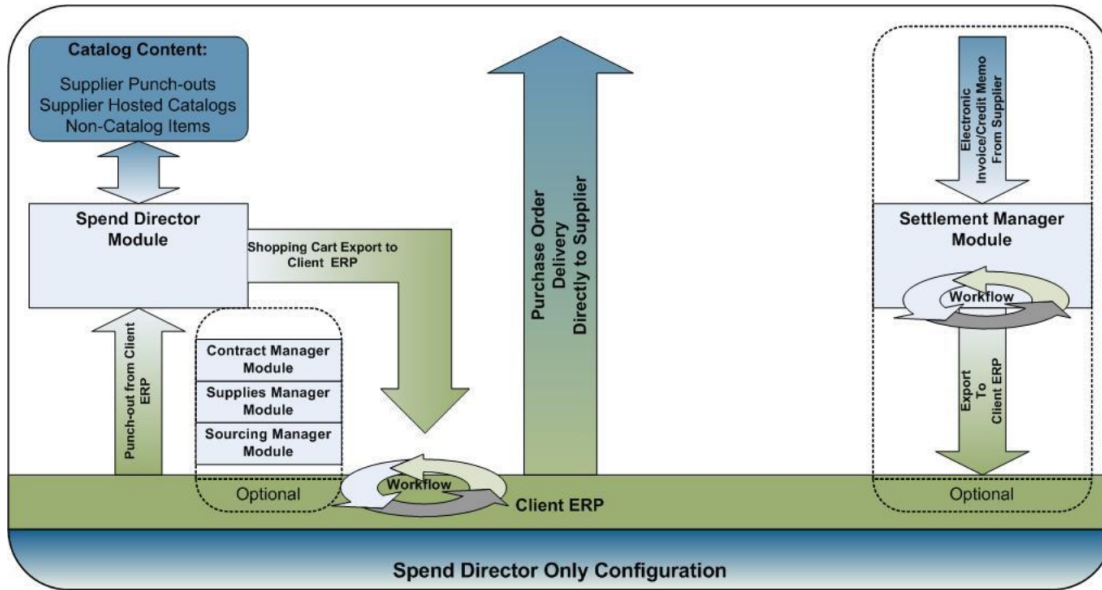
If the client has licensed an optional Accounts Payable product, invoices from suppliers can be received electronically via cXML or via the Supplier Portal. If the client configuration does not include an Accounts Payable product or the supplier is unable to deliver electronically, invoices are delivered directly from the supplier to the client, bypassing the JAGGAER application.



Spend Director Only Configuration

In the Spend Director Only configuration, the JAGGAER application exports the shopping cart to the client ERP, where it generates a requisition, moves through workflow approvals, and is then converted to a purchase order. In this configuration, ShipTo and BillTo addresses are **not** stored in the JAGGAER application and the client system delivers purchase orders **directly to the supplier outside of the JAGGAER application**. The client will coordinate with the supplier to negotiate the delivery method and execute purchase order testing as applicable.

If the client has licensed an optional Accounts Payable product, invoices from suppliers can be received electronically via cXML or via the Supplier Portal. If the client configuration does not include the Accounts Payable product or the supplier is unable to deliver electronically, invoices are delivered directly from the supplier to the client, bypassing the JAGGAER application.



Integration Resources

General Integration Requirements

All integrations between JAGGAER and the supplier have specific technical requirements. The following standards are for all integrations; however, please refer to the specific integration for additional requirements.

- All integrations are based on cXML message format standards. Applications must be able to authenticate cXML messages to ensure transactions are secure and to prevent unauthorized access.
- All integrations are required to be encrypted using standard **https** internet protocol over port **443** by default. **Hypertext Transfer Protocol Secure (HTTPS)** is a combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encryption and secure identification of the server.
- SSL digital certificates must be issued against the same domain as the transactive URL. Certificates that are issued by an internal authority or are self-signed must be provided to JAGGAER for installation and should be valid for a minimum of three years.
- All cXML documents must not contain byte order marks (BOM) as they cause JAGGAER parsing failures.
- All integration testing is to be executed in a supplier-designated test environment, separate from the production environment.
- The **Supplier** DUNS/NetworkID values should remain static across all customer integrations. The client DUNS/NetworkID values will vary by customer.

Supported cXML Versions

Each Supplier Integration Point chapter indicates the JAGGAER supported Document Type Definition (DTD) version for that integration type. **JAGGAER does not support all elements of any particular DTD version, and in some cases does support specific, individual elements of later DTD versions.** For detailed information about JAGGAER message element support, please refer to the integration chapters and the message element tables within each one.

JAGGAER understands that suppliers may have developed support for later DTD versions. JAGGAER will ignore the supplier supported version provided in the DOCTYPE header element; however, unless otherwise indicated within this document the supplier message must not contain any elements from a version later than JAGGAER's to avoid a failure response. If unsupported elements cannot be omitted, please contact JAGGAER to further investigate the possibility of integration.

The JAGGAER application will naturally develop and change over time; therefore, DTD support may also change. As a result, suppliers should refrain from rejecting documents based solely upon the DTD version indicated in the DOCTYPE header element.

Obtaining the DTD

DTDs are publically available via the internet and can be easily retrieved by the supplier. JAGGAER codes based upon DTD only; therefore, suppliers interested in obtaining schemas will need a conversion application or an alternate source.

A copy of each DTD can be retrieved as follows:

- Follow the web link(s) listed below using Mozilla Firefox or Safari (IE will truncate content in Notepad).
- Upon arrival, an xml parsing error message will be displayed. Right click on the page and select "view source".
- The DTD source will be revealed and can be reviewed and/or copied.

PUNCHOUT, EDIT/INSPECT, LEVEL 2 PUNCHOUT:

<http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd>

LIVEPRICE & AVAILABILITY CALL:

http://solutions.sciquest.com/app_docs/dtd/priceAvailability/PriceAvailabilityMessage.dtd

ORDERREQUEST:

<http://xml.cXML.org/schemas/cXML/1.2.011/cXML.dtd>

CONFIRMATIONREQUEST & SHIPNOTICEREQUEST:

<http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd>

INVOICES/CREDIT MEMOS:

<http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd>

Supplier Development Resources

Suppliers should be capable of cXML messaging before engaging JAGGAER in new integration projects. While JAGGAER can provide general integration guidance, development services such as code authoring and debugging are not provided to the supplier. The supplier may elect to engage a third party provider, or JAGGAER can recommend a provider upon request. Any fees assessed by a third party provider are the sole responsibility of the supplier.

Consider the following:

- Electronic B2B channels provide the means to save significant transaction costs if done efficiently and in a manner consistent with business processes. A thorough understanding of

the technology and protocols involved in cXML procurement transactions is invaluable in the planning and execution of the implementation process.

- Review existing business processes and evaluate how the proposed integration fits within that process. Consider the resources that may be needed to support the integration once in place.
- The size and expertise of any existing technical staff is paramount to implementation success. Engaging a third party to bridge any technical or resource limitations may expedite implementation and significantly reduce the risk of failed or substandard development projects.
- Review internal customer account setup and evaluate the impact an electronic cXML order or invoice might have on that process. Consider that a single customer procurement channel may require a change to the account structure currently in place. Plan for a seamless transition from the old to new structures and advise sales staff of any changes in account procedures.
- Consider that with most JAGGAER clients, ShipTo address information is not available at the time of punch-out. This information is typically not available until the purchase order document has been created and is ready for distribution to the supplier. This may have a significant impact to the punch-out development process.
- Evaluate how the supplier system currently stores and delivers product information to customers and how that might best fit either a punch-out or hosted catalog. If a planned punch-out site will share code with an existing eCommerce site, consider how changes to the eCommerce site might impact the punch-out site and vice versa.
- Evaluate any existing system integrations and apply any common technologies, resources, successes, and lessons learned wherever possible.
- Discuss the integration with the customer and understand any specific requirements they may have before proceeding with development.

cXML.org

The cxml.org website outlines cXML standards and makes available a public users guide. The user guide is the most comprehensive resource available for programmers designing cXML applications, and should be the supplier's primary tool in developing and troubleshooting cXML messaging.

W3C.org

The W3C.org website is the home of the **Platform for Privacy Preferences Project (P3P)**. The project was initiated by industry professionals to define a standard format for the conveyance of website privacy practices that can be read and interpreted by web browsers (compact policy header). This site is a primary reference for suppliers developing punch-out applications and provides a tool to validate the compliance of developed sites.

Please see the [P3P Compliance](#) section for more information.

cXML Validation

There are many applications available to developers that can aide in the validation of cXML documents. JAGGAER can make no recommendations about specific applications; however, an internet search will provide a vast list of both free and retail applications available.

UNSPSC.org

The United Nations Standard Products and Services Code (UNSPSC) is the global mechanism with which products are grouped and categorized within the JAGGAER application. JAGGAER clients require that suppliers provide either an 8 or 10 digit UNSPSC code for all products in order to track and report spend by commodity, and to facilitate availability of products by category to the user community.

For more information about UNSPSC and a list of codes, please visit www.unspsc.org as well as the [Product Categorization](#) section in this document.

Third Party Solutions

There are many 3rd party providers that specialize in helping suppliers connect electronically with their customer base. From punch-out connectivity to purchase order and invoice delivery, there are a variety of options available to suppliers via a basic internet search.

A JAGGAER partner, Liaison Technologies, is a global integration, data management and data security company. Liaison connects businesses by providing a complete business-to-business solution for companies of all sizes. Liaison offers tailored integration packages that help JAGGAER suppliers integrate with buyers on the JAGGAER network, regardless of the type of software the supplier uses. Liaison's connector packages for JAGGAER include solutions for Order-to-Invoice, PunchOut Integration, PunchOut Site Enablement, and Price and Availability Call. Headquartered in Atlanta, Liaison has offices in the Netherlands, Finland, Sweden and the United Kingdom. For more information, please contact JAGGAER@liaison.com or 425-247-0935.



Additional 3rd party providers that have previously built eProcurement solutions for supplier integrations on the JAGGAER platform include (in alphabetical order):

Greenwing Technology, Inc.

Jeremy Friedman / jfriedman@greenwingtechnology.com

877-415-3237 / 302-295-5690

<http://www.greenwingtechnology.com/>

Pebblejet

JAGGAER-suppliers@pebblejet.com

415-895-0203

<http://pebblejet.com/>

Punchout2Go

Brady Behrman, CEO / brady@punchout2go.com

888-623-2374

<http://www.punchout2go.com/>

PunchoutCatalogs.com (multiple customer connections only)

Brian Shuey, Director of Sales / brian@punchoutcatalogs.com

724-433-5265

www.punchoutcatalogs.com

Supplier Solutions

Matthew Petersman, Director of Sales & Marketing / matt.petersman@suppliersolutions.com

513-400-4657

<http://www.suppliersolutions.com/>

***NOTE:** JAGGAER does not advocate or provide references for any third party providers. Selection of a third party provider is solely at the discretion of the supplier organization. Any fees assessed by a third party provider are the sole responsibility of the supplier.

Data Standards

The SpendDirector Module is the user interface to the shopping experience and is the repository for supplier catalog data. Users enter the Market Place to locate products and services, searching and comparing products across a multitude of supplier catalogs in order to locate the best product match. Diversity in supplier data formats, terms, and style can obscure the process and undermine the goal of a user friendly shopping experience. As a result, data must be standardized in order to provide shoppers a consistent experience within a single environment. The following sections outline how the JAGGAER application standardizes and presents data to users.

Units of Measure (UOM)

Users must understand the quantity or units of the supplier product offering in order to effectively compare products across a variety of catalogs. Because suppliers and buying organizations maintain individual standards, JAGGAER has developed tools to apply a single standard while preserving the supplier's ability to maintain their own.

JAGGAER UOM Standards

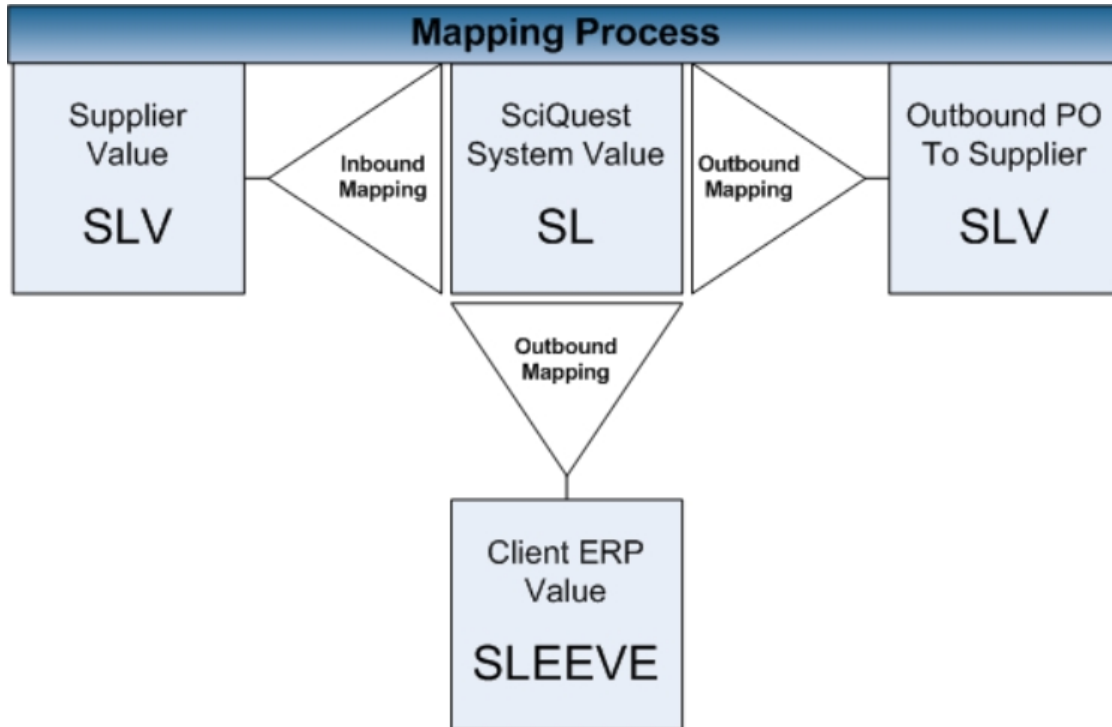
The JAGGAER UOM standards are a composition of ANSI, ISO, and common industry-driven standards that cover over 99% of all UOM values used by suppliers. In the event a supplier utilizes values not included in the JAGGAER standards, unrecognized values will require mapping.

Refer to JAGGAER UOM Standards in the **Standard Units and Unit Mappings** file found in the Supplier Network.

UOM Mapping

In order to display varying supplier standards uniformly to users, JAGGAER employs a mapping tool that effectively translates inbound supplier UOM values to the JAGGAER UOM standard values. Users are then able to shop across multiple supplier catalogs with confidence that the displayed units of measure have been consistently applied. While not overtly displayed to users, the original values provided by the supplier are preserved for outbound documents to both the supplier and client ERP system.

Upon shopping completion, the JAGGAER standard values are then mapped to the expected client and supplier values respectively. Outbound purchase orders from JAGGAER will pass the original supplier UOM value, while outbound exports to the client ERP system pass both the original supplier value as well as the expected client ERP value. Clients that deliver purchase orders directly to suppliers outside of the JAGGAER application will then have the original supplier UOM value to pass on any outbound documents as required.



Suppliers should review the JAGGAER UOM standards list provided in the previous section to compare and correlate supplier values to JAGGAER values, as well as review the **Standard Units and Unit Mappings** spreadsheet which illustrates common mappings already in place. If required, the supplier can provide the JAGGAER implementation resource a list of additional mappings for execution or discuss any subsequent values that may be needed. Following go-live, the maintenance of UOM mappings is completed by the supplier using the mapping tool provided in the Supplier Network.

Unmapped UOM Values

In the event that UOM mapping has not been completed, or a particular UOM has not been accounted for, the JAGGAER system will map any unrecognized supplier values to "EA" by default. This may present the user with invalid catalog data and could cause the supplier processing and/or accounting issues upon purchase order receipt. Unmapped values will also negatively impact the cXML invoice process. To prevent this, ensure that UOMs are discussed and reviewed with the JAGGAER Implementation Project Manager before taking the catalog to the production environment.

Once in production, the supplier is responsible to maintain and update the existing mappings with any new or revised values through the Supplier Network Portal.

Product Size in UOM

If required, suppliers can provide product/package size information for punch-out catalog products. This is most commonly used to denote the number of individual units contained

within a larger package unit (e.g. 100 per Box). This can be provided in two ways:

- **Recommended:** Include the product/packaging size information as part of the product description.

For example:

```
<ItemIn quantity="1">  
<Description xml:lang="en-US">White gloves, 100/Box</Description>  
<UnitOfMeasure>BX</UnitOfMeasure>
```

- **Not Recommended:** Include the product/packaging size information as part of the unit of measure. Shoppers can often misinterpret the below example to mean they have ordered 100 boxes.

For example:

```
<ItemIn quantity="1">  
<UnitOfMeasure>100/BX</UnitOfMeasure>  
<Description xml:lang="en-US">White gloves</Description>
```

In both of the above examples, the shopper has ordered 1 box containing 100 units.

For suppliers providing a hosted catalog, there is a unique field for product size. See the [Catalog Management Template](#) section for more information.

Supported Currencies

Many of JAGGAER's clients do business with both domestic and international suppliers. The support of multiple currencies allows JAGGAER's clients to maintain these relationships and continue to do business in the required currencies.

Client organizations can set a default base currency for the procurement application, but may also choose to enable additional currencies by supplier as needed. Should the supplier operate in a single currency that is not the client base currency, the JAGGAER application employs a conversion tool that allows client organizations to set the currency exchange rates. This provides suppliers the ability to provide pricing in their base currency, while allowing users the choice to view pricing in either their own base currency or the supplier's. Purchase orders are delivered to the supplier in the supplier's base currency.

Currencies are not enabled by default; therefore, the supplier should coordinate with the client to discuss the possibility of enabling additional currencies.

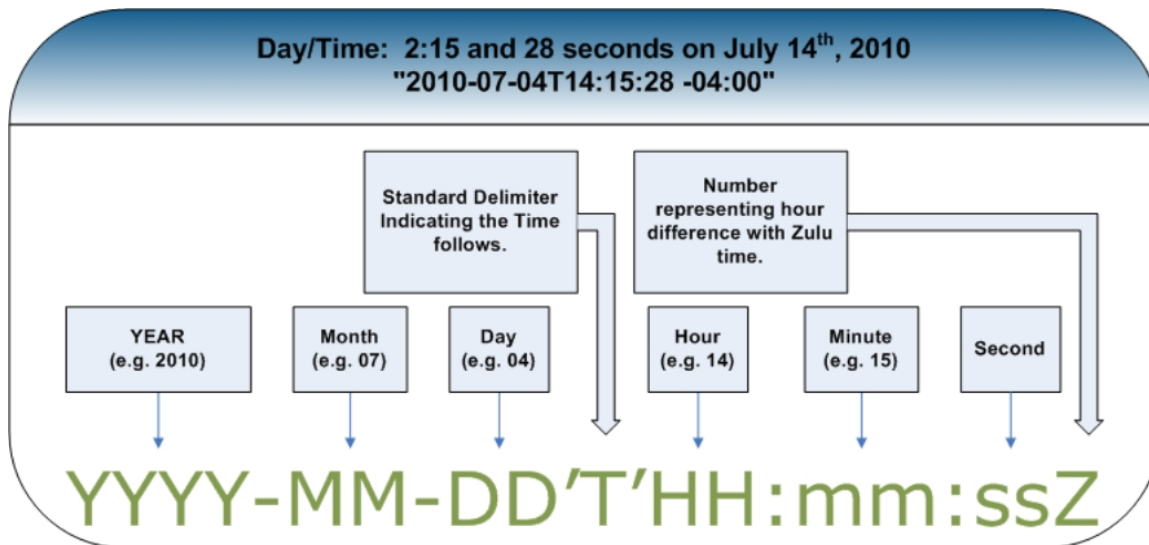
ISO 4217 Currency Standards

JAGGAER supports the ISO 4217 standard list of currencies. They can be found at <http://www.currency-iso.org/en/home/tables/table-a1.html>

Date/Timestamp Format

All JAGGAER outbound documents will contain a date/timestamp in the format illustrated below, representing the International Standard ISO 8601.

JAGGAER will ignore the date/timestamp on inbound documents in favor of the day/time recorded by the JAGGAER system at the time of receipt.



Numerical Formats & Calculations

The following sections outline the JAGGAER accepted numerical formats as well as the calculation of extended prices and order totals.

Numerical Formats

QUANTITIES

- Quantities must be presented in whole numbers (e.g. "1"). JAGGAER can accept decimal quantities only when the client organization is configured to accept them.

- Leading zeros are accepted (e.g. "01"); however, JAGGAER will not display or return a quantity value with leading zeros on any outbound documents.
- Quantities 1K and over can be delimited by a comma (e.g. "1,000"); however, many popular ERP systems do not accept this character. As a result, comma delimiters will be removed from any outbound documents.
- Quantities may not be negative.

UNIT PRICE

- JAGGAER can accept up to 4 decimal places (e.g. "1.6894"). The unit prices provided in hosted and punch-out catalogs should be consistent with the unitPrice format that will be provided by the supplier on invoice documents.
- Comma delimiters and currency symbols are not accepted.

Correct Format:

```
<UnitPrice>
  <Money currency="USD">1099.00</Money>
</UnitPrice>
```

Incorrect Format:

```
<UnitPrice>
  <Money currency="USD">$1,099.00</Money>
</UnitPrice>
```

Calculations

GENERAL CALCULATION

JAGGAER ignores order totals as sent by the supplier in favor of calculation by unit price. The extended price is calculated based upon the multiplication of [quantity] x [unit price] + [shipping] + [tax].

<pre><Total> <Money currency="USD">199.99</Money> </Total></pre>	← Ignored
--	-----------

```
<Shipping>
  <Money currency="USD">4.99</Money>
</Shipping>
<Tax>
  <Money currency="USD">0.82</Money>
</Tax>
<ItemIn quantity="1">
  <UnitPrice>
    <Money currency="USD">9.99</Money>
  </UnitPrice>
```

In the above example, JAGGAER would calculate the total as 15.80. The incorrect amount of 199.99 that appears in the <total> tag is ignored.

ROUNDING

In the event the supplier provides a unit price with more than two decimal places, JAGGAER will apply standard rounding rules when calculating the extended price.

```
<Total>
  <Money currency="USD">199.99</Money>
</Total>
```

← Ignored

```
<ItemIn quantity="8">
  <UnitPrice>
    <Money currency="USD">1.6894</Money>
  </UnitPrice>
```

In the above example, JAGGAER would first calculate the extended price before applying standard rounding rules. The order total above would therefore be $[8] * [1.6894] = [13.5152]$. Once rounded, the price displayed to the user and sent on outbound documents would be 13.52.

DECIMAL QUANTITIES

In the event the client has agreed to accept decimal quantities, JAGGAER will calculate the extended price as previously stated: $[quantity] \times [unit\ price]$.

```
<Total>
  <Money currency="USD">199.99</Money>
</Total>
```

← Ignored

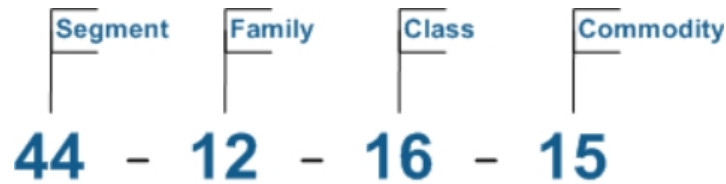
```
<ItemIn quantity="1.5">
  <UnitPrice>
    <Money currency="USD">2.00</Money>
  </UnitPrice>
```

In the above example, JAGGAER would calculate the extended price as 3.00.

Product Categorization

The United Nations Standard Products and Services Code (UNSPSC) is the global mechanism with which products are grouped and categorized within the JAGGAER application. JAGGAER clients require that suppliers provide either an 8 or 10 digit UNSPSC code for all products in order to track, analyze, and report spend by commodity, as well as to facilitate availability of products by category to the user community.

The code is hierarchical in nature, with four to five levels determining the depth of product classification.



The code should be provided to JAGGAER as part of the item detail without the dash delimiters. In the above example, a stapler with UNSPSC 44-12-16-15 should be passed as 44121615.

For more information about UNSPSC and a list of codes, please visit www.unspsc.org.

Product Flags

In addition to product categorization mentioned in the previous section, the JAGGAER applications offers suppliers additional data elements to identify certain attributes about the products they offer. These attributes can be merely informational, but more importantly can be used to identify, or “flag” products of a regulatory nature. Many clients require that regulatory or hazardous products pass through certain workflow authorizations, or they may use product flags to make products of a certain nature unavailable to groups of users.

In recent times, organizations have become increasingly more apt toward driving spend to environmentally friendly and energy efficient products. In order to meet this demand, JAGGAER offers end users the ability to search and give preference to products that have these attributes.

Products that present environmental health and safety (EH&S) concerns may **require** flagging for applicable products based upon the categories outlined in sections below. All other flags are typically optional, but may be required by the requesting client.

For suppliers providing hosted catalogs, there are additional elements to consider. Please refer to the **Product Flag Guidelines** reference guide located in the **Content Management Template**.

EH&S Flags

Products that present environmental health and safety (EH&S) concerns may **require** flagging for applicable products.

RADIOACTIVE

Definition: Some products have a radionuclide in their structure. Such 'radiolabeled' substances have radioactivity associated with them above the background level of radiation. NRC has set regulations and restrictions for the usage of such substances.

Usage: This Product Flag should be passed for any radioactive product whose radioactivity is high enough such that its usage is governed by NRC regulations.

RADIOACTIVE MINOR

Definition: Some products have a radionuclide in their structure. Such 'radiolabeled' substances have radioactivity associated with them; however, the radioactivity levels are below the background level of radiation. As a result, such products can generally be handled without need of any specific regulatory license unless purchased in high volumes that would exceed levels as outlined above.

HAZARDOUS MATERIAL

Definition: Some products are hazardous due to their physical/chemical properties. OSHA has set regulations to identify specific health and physical hazards.

Usage: This Product Flag should be passed for any product that has hazards associated with them. Additional details about regulations governing hazardous material can be accessed at following websites:

<http://www.dot.gov>

<http://www.osha.gov>

<http://www.nfpa.org>

http://www.access.gpo.gov/nara/cfr/waisidx_03/40cfr261_03.html

http://www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=STANDARDS&p_id=10100

CONTROLLED SUBSTANCES

Definition: Some products are addictive drugs or can be used in the manufacturing addictive drugs. Such products are regulated under the Controlled Substances Act and are maintained by the FDA and USDOJ. Federal regulations have been set for the identification, handling and usage of such products.

Usage: This Product Flag should be passed for any product that is classified into one of the five controlled substance schedules, or is on the list of substances that can be used to manufacture addictive drugs.

Additional details about federal regulations, including specific lists of controlled substances can be accessed at following websites:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/2108cfrt.htm>

<http://www.usdoj.gov/dea/pubs/csa/802.htm>

TOXINS

Definition: Some products, such as bacterial toxins or animal venoms, are toxic to various biological processes and are used to study specific cell processes (e.g. signal transduction mechanism).

Usage: This Product Flag should be passed for any toxin that is used to study various cell processes.

SELECT AGENTS

Definition: Some products are live cells of extremely hazardous pathogens such as the Ebola and Marburg viruses. The CDC has set stringent regulations about the handling of these products, such as transport, usage, and disposal.

Usage: This Product Flag is required and should be passed for any product that is considered to be a select agent per CDC definitions. In many cases, clients will request the supplier eliminate Select Agents from any online catalogs to ensure the utmost safety of their users. In these cases, client EH&S departments will order Select Agents directly from the supplier.

Additional details about CDC regulations can be accessed at following websites:

<http://www.cdc.gov/od/sap/docs/salist.pdf>

<http://www.cdc.gov/od/sap/>

Additional Flags

The flags outlined in the following sections are typically optional, but if provided, would allow users to search products based upon these highly desirable attributes. When unflagged, users may shift focus to alternative suppliers that offer environmentally friendly or energy efficient products.

RECYCLABLE

Definition: Products that can be recycled through established recycling programs or have been manufactured from recycled products are considered to be environmentally friendly.

Usage: This Product Flag should be passed for any product that can be considered environmentally friendly per the above definition.

ENERGY STAR

Definition: The Environmental Protection Agency and Department of Energy jointly run a program to identify various energy efficient equipment/appliances. Qualified products are given an Energy Star rating so that consumers can quickly identify them.

Usage: This Product Flag should be passed for any product that has an Energy Star rating per federal guidelines.

Guidelines and references can be accessed at following website: <http://www.energystar.gov>

GREEN PRODUCTS

Definition: A product can be considered 'green' for reasons such as an environmentally friendly production method, reduced energy consumption, or an improved disposal method. While a precise definition of a green product is not available, some guidelines can be found on the NAEP website: <http://www.naepnet.org>

Usage: Green Purchasing based on NAEP guidance:

- Green Purchasing is the method wherein environmental and social considerations are taken with equal weight to the price, availability, and performance criteria that colleges and universities use in making purchasing decisions.
- Green Purchasing is a serious consideration in supply chain management.
- Green Purchasing is also known as "environmentally preferred purchasing (EPP), green procurement, affirmative procurement, eco-procurement, and environmentally responsible purchasing," particularly within U.S. Federal government agencies.
- Green Purchasing minimizes negative environmental and social impact through the use of environmentally friendly products.
- Green Purchasing attempts to identify and reduce environmental impact and to maximize resource efficiency.

Data Parsing (PCDATA vs. CDATA)

All text passed in a cXML document will be parsed. This causes issues with specific characters in general text that are used in cXML standards to identify new elements or character entities. Characters such as "&" and "<" are two such characters. These characters, if present in product descriptions and other general text elements, generate errors and will cause cXML messaging to fail.

PCDATA

Parsed Character Data (PCDATA) is a term that represents the text characters that will be parsed. When a cXML element is parsed, the text between tags is also parsed. For example:

```
<Description xml:lang="en">Generic Product and Size Description</Description>
```

In the above example, the product description of "Generic Product and Size Description" will be parsed correctly as it contains no characters used in cXML elements. However, consider the same description in the next example:

```
<Description xml:lang="en">Generic Product & Size Description</Description>
```


In this example, the entire message will fail parsing at the "&" sign as it represents a new character entity in cXML standards. To avoid message failures, general text that should not be parsed can be contained in what is called a CDATA section.

CDATA

Character Data (CDATA) refers to text in a cXML message that is considered general character data and will not be parsed as a cXML element. Characters such as "<" and "&" contained within a CDATA section bypass parsing and as a result will prevent cXML message failures.

The following example represents a CDATA section within a cXML standard element:

```
<Description xml:lang="en"><![CDATA[Generic Product Size &
Description]]></Description>
```

In the above example, the product description of "Generic Product Size & Description" will be successfully passed to JAGGAER as it is contained within a CDATA section. Even though successful, there are characters that common client ERP systems cannot handle however they are sent. For more information about these characters, refer to the section on Special Characters.

Encoding & Special Characters

All cXML documents must be UTF-8 encoded. Suppliers should avoid the use of any non UTF-8 character.

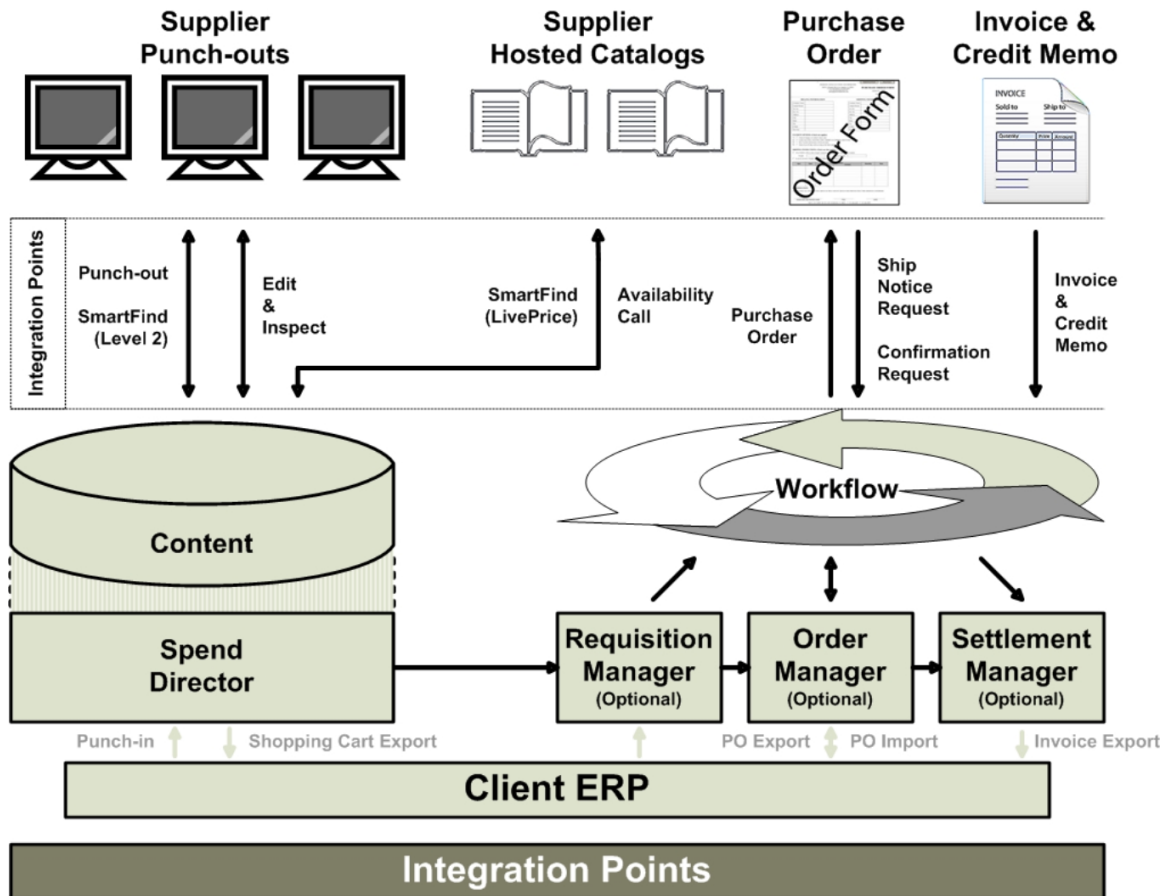
Many ERP systems encounter issues with special characters; therefore, they should not be used in data value sets. Data sets such as part numbers, product descriptions, and manufacturer name should be free from special characters. Here are just a few examples of characters to be avoided in any data set:

Character	Description
&	Ampersand
®	Registered Trademark
™	Unregistered Trademark
©	Copyright
Δ	Delta
±	Plus-Minus
^	Circumflex Accent
~	Tilde

Supplier Integration Points

The following diagram depicts the e-procurement process and the potential cXML integration points between suppliers and the JAGGAER modules. The remainder of this document outlines the JAGGAER requirements for each of the following integration points:

- Supplier PunchOuts
- Edit/Inspect
- SmartFind (Level 2, LivePrice)
- Availability Call
- Purchase Order (Delivery)
- Confirmation Request
- Ship Notice Request
- Invoices and Credit Memos



PunchOut

What is a punch-out?

A punch-out is a web based catalog available to buyers through an eProcurement application. The buyer accesses the online catalog by “punching out” from the procurement application to the supplier site, shops and requisitions products, and then returns the cart contents to the procurement application. A punch-out site can look similar to a standard eCommerce site; however, a significant difference is that punch-out session cXML messages are passed between the procurement application and the supplier network/site. In a traditional eCommerce setting, an online shopping session results in an order being placed directly with the supplier, while in a punch-out session the cart contents are returned to the procurement application such that it can pass through budget checks and other approval processes before a purchase order is routed to the supplier.

DTD: CXML.dtd

Version: 1.1.010

Guides: [Punchout Troubleshooting Guide](#), [cXML User’s Guide](#)

JAGGAER conforms to the cXML **1.1.010** DTD standard for punch-out integrations. JAGGAER should be able to integrate with any supplier using this version; however, if the supplier is using another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, see [Supported cXML Versions](#).

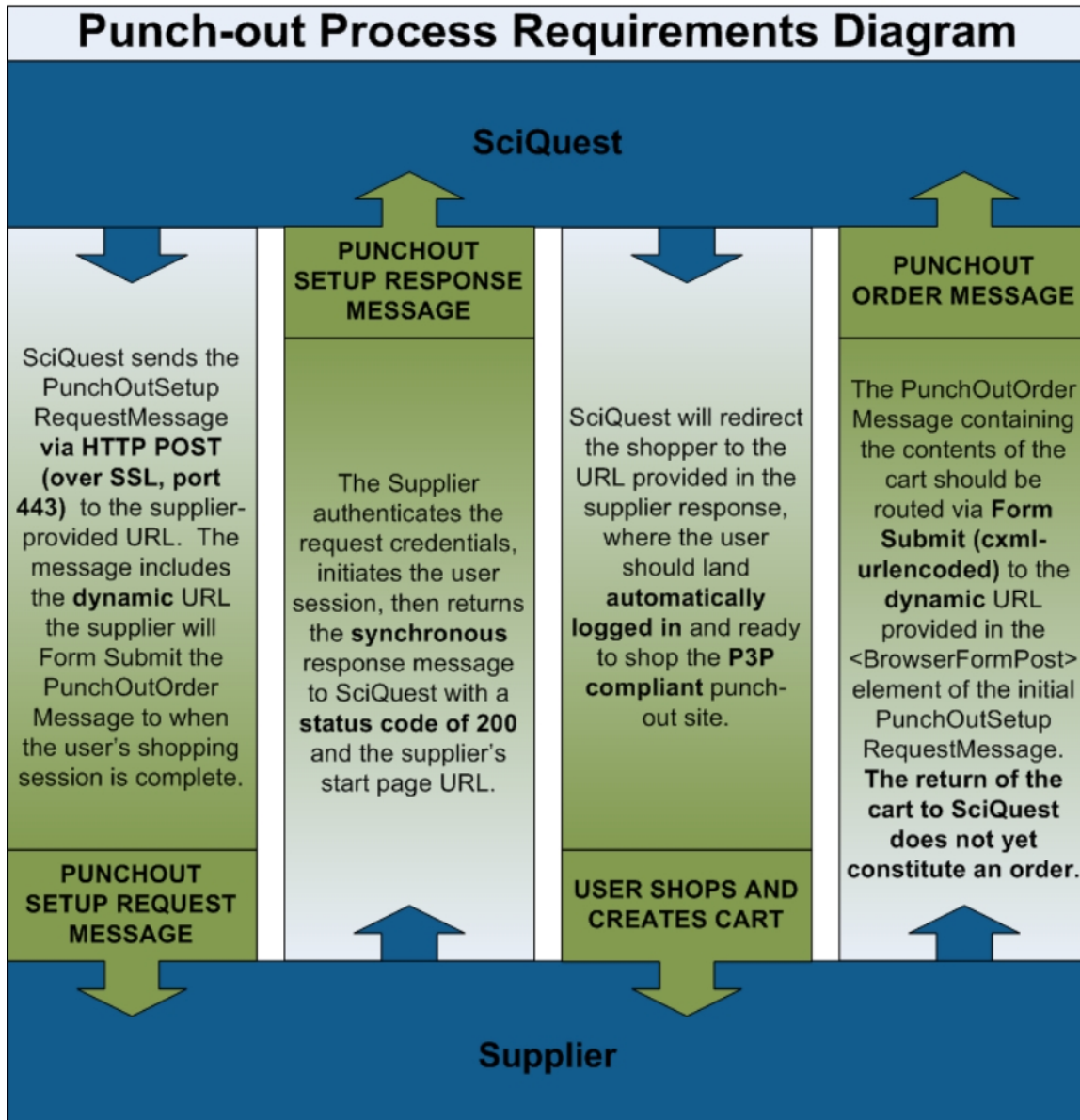
PunchOut Requirements

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the punch-out integration that ensure the supplier’s product offering is presented to the shopper in the most beneficial manner possible:

- The punch-out site must support multiple, concurrent users.
- The initial post URL and the redirect URL contained within the <StartPage><URL> tag set of the PunchOutSetupResponse message must be https (port 443).
- The punch-out site must be P3P compliant before promotion to production can be considered. Please see the [P3P Compliance](#) section for more detailed information.
- The supplier must support the three major internet browsers: IE, Firefox , Safari
- Should the punch-out require cookies, the cookie must be unique and separate from those being used on the supplier’s eCommerce site. Separate cookies ensure that user activity between the two sites is not inadvertently affected by the other.
- The punch-out site must not require any additional user login, selection of account, or require the user to provide a ShipTo address or payment information on the site itself. Once a shopping cart is returned to JAGGAER, the cart will move through a client approvals

process before a purchase order is generated. Once approved, the purchase order delivery to the supplier will specify the ShipTo address and payment information.

- The punch-out site will be displayed within a frame session, and therefore should not contain any code that would prevent proper function within frames. Scripts or code that designates the supplier site as the parent/top window, or code that would otherwise force break a frame session must be removed.
- The punch-out site should not contain links to external sites that would break the punch-out session and remove the user from the shopping experience. Any necessary links to internal sites/pages should open in a new window.
- The PunchOutOrderMessage must be routed via Form Submit, cxml-urlencoded. JAGGAER does not accept cxml-base64 encoding.
- The punch-out site must not generate an order for fulfillment. The return of a shopping cart to JAGGAER does not yet constitute an order.
- All other requirements as listed in the Message Elements Table.
- Successful integration testing per the provided PunchOut Test Plan is required before promotion to production can be considered.



Mobile Punchout

Mobile Punchout functionality allows an organization to utilize a supplier's existing punchout technology on their Android and/or iOS mobile device through the JAGGAER Mobile Application. JAGGAER uses the same PunchoutSetupRequest format and supplier URL as the desktop application to initiate a mobile punchout session.

Mobile Punchout Requirements

- Desktop punchout must be configured prior to mobile functionality being enabled.
- Supplier's webpages **must be able to consume the following data** passed by JAGGAER:
 - Browser Information
 - Device Type
 - Device Resolution
 - Screen Size
- Use of the following technologies within a punchout **will prevent users from successfully completing the checkout process**:
 - Flash
 - Silverlight
 - Java Plugins
 - Iframes
 - Popups
 - Active X
 - Browser Plugins
- A responsive punchout website is recommended for optimal user experience.

P3P Compliance

P3P non-compliance is one of the most common root causes of punch-out performance issues, particularly for users in Internet Explorer. As a result, all punch-out supplier catalogs must be P3P compliant. Because JAGGAER has no insight or control over the supplier site source code, image locations, or web server setup; it is the supplier's responsibility to ensure the punch-out site is compliant. While JAGGAER is unable to troubleshoot compliance issues on the supplier's behalf, JAGGAER can direct suppliers to the tools listed in the following sections.

P3P Defined

P3P stands for the **Platform for Privacy Preferences Project**. The project was initiated by industry professionals to define a standard format for the conveyance of website privacy practices that can be read and interpreted by web browsers (compact policy header). With standards defined, browsers can then develop behavior that aligns the site privacy practices with the user-defined privacy settings, thereby eliminating or significantly reducing the need for the user to read a privacy policy at every site they visit.

For more information about P3P, please visit the [W3C Platform for Privacy Preferences Project](http://www.w3.org/2001/06/P3P/) website.

Importance of P3P Compliance

P3P Compliance is key to the success of eCommerce sites for many reasons. Industry research has shown that users/shoppers that are required to accept 3rd party cookies or provide personal information are less likely to complete a transaction. Furthermore, organizations that provide users access to eCommerce sites through a secure network often do not, for network security reasons, provide the administrator rights necessary for shoppers to adjust browser settings. In most instances, the user/shopper is unaware of these complexities, and likely cannot troubleshoot site privacy issues on their own. For organizations with large groups of users, it can be very difficult to manage privacy settings for every eCommerce site available, particularly when most organization IT departments do not allow 3rd party cookies as a policy. This naturally results in a loss of revenue for the supplier and does not provide for a positive shopping experience.

Sites that are P3P compliant provide a seamless shopping experience that are generally free from pesky pop-ups advising of unsecure content, as well as shopping cart malfunctions due to blocked or rejected cookies.

Becoming P3P Compliant

There are many sources available on the web, as well as many policy authoring software providers. JAGGAER does not provide or endorse any of these services; however an internet search of "P3P" will provide additional information for consideration. Below are some key resources for the development and testing of P3P:

The Platform for Privacy Preferences 1.0 (P3P1.0) Specification : Please visit the W3C site for the P3P specification.

<http://www.w3.org/TR/P3P/>

Troubleshooting P3P Compliance : Please visit the Webmasters Guide to Troubleshooting P3P for troubleshooting tips.

<http://www.oreillynet.com/pub/a/javascript/2002/11/19/p3p.html>

P3P Validation : Please visit the W3C site for access to a validator.

<http://www.w3.org/P3P/validator.html>

P3P and Internet Explorer

Internet Explorer is the most widely used browser by a large margin. As a result, careful consideration of the way in which IE handles privacy is paramount to the success of P3P compliance. JAGGAER utilizes frame sessions to maintain the external link to a supplier's punch-out site, thereby allowing the external shopping cart to be returned to the JAGGAER procurement application. Internet Explorer treats content within a frame session as third party content, and as a result, will reject cookies from non compliant sites using IE's default settings (Medium). The shopping experience will cease, as the end user will be unable to add items to their shopping cart - often without an understanding as to why.

As previously mentioned, many of our clients users do not have the administrative ability or the experience to identify the problem and adjust browser settings to accommodate, nor do most organization IT departments allow 3rd party cookies to be accepted as a policy. To correct this problem, a compact policy header must be added to **all** displayed content & images, including source content that resides on another site, server, or other alternative source.

For more information about this IE specific issue, please visit the Microsoft site.
<http://support.microsoft.com/kb/323752>

JAGGAER Policy

JAGGAER will require all punch-out supplier sites that require cookies to be P3P compliant. In addition to this document, please also review the P3P specification when evaluating punch-out catalog participation. A compact policy header must be added to **all** displayed content & images, including content that resides on another site, server, or other alternative source that may be accessed via the punch-out.

To validate P3P compliance, check the site URL at the **W3C** site.
<http://www.w3.org/P3P/validator.html>

PunchOut Site Best Practices

PunchOut sites that are not easily navigated and do not offer functionality that is typically available on eCommerce sites can negatively impact the user experience. JAGGAER has the following recommendations to ensure the supplier site is user friendly:

- **Logo and Welcome Text:** Displaying the client specific logo and a user welcome message helps the shopper to feel comfortable that they have arrived at the appropriate site.
- **Product Images:** Shoppers are much more likely to purchase products where an image is available. Displaying images on the punch-out site can be one of the largest contributors to a user friendly experience.
- **Product Description:** Provide a comprehensive and clear product description free from special characters (256 char limit). Users are less likely to purchase products where the description is incomplete or does not appropriately describe the product.
- **Product Size:** Include the product size in the product description where possible. This ensures users fully understand the product packaging before making purchasing decisions.
- **Product Availability/Lead Time:** Providing accurate product availability and lead time sets the appropriate expectations for product delivery and can influence purchasing decisions.
- **Saved Cart:** Offering users an automatic save or the ability to save the cart manually can reduce user frustration when punch-out idle time has expired.
- **Internal/Product Links:** Links that take the user outside the punch-out experience cause confusion and break the procurement process. Internal links such as product inform-

ation should display in a new window; however, the new window should not introduce new pathways to other catalog items or purchasing opportunities.

- **Order History:** Allowing users to retrieve order history can be beneficial; however, consider that the buying organization may wish to restrict users to personal history rather than providing history organization-wide. Suppliers are encouraged to make order history optional by client.
- **Logout:** JAGGAER offers users the ability to cancel the punch-out session from within the procurement application; therefore, logout links are unnecessary on the supplier site. In the event the supplier is unable to remove site logout links, the supplier should return an empty PunchOutOrderMessage to JAGGAER when the logout link is accessed. This returns the user to the procurement application without issue.
- **Favorites:** Favorite features provide users with easy access to the most frequently purchased products and can be offered at both the user and organizational levels. Offering “favorites” features can be one of the largest contributors to a user friendly experience; however, when offering organizational level favorites, consider that clients may wish to control the products that appear to all users.
- **ShipTo Address:** Consider that a user’s ShipTo address is not typically available at the time of punch-out. Suppliers should not require the passing of the address or ShipTo code for successful punch-out connectivity. The ShipTo address will be passed to the supplier upon purchase order delivery.
- **Email Address:** JAGGAER will pass the user’s email address when available at the time of punch-out; however, the availability of this field is not guaranteed and should not be required for successful punch-out connectivity. Please review the [Tracking Users](#) section for more information.
- **Submit Order Button:** Traditional eCommerce sites display a button for users to click when submitting their order. Because the return of a shopping cart to JAGGAER does not constitute an order, JAGGAER recommends that the button text read “Return to Procurement Application” to avoid confusion.

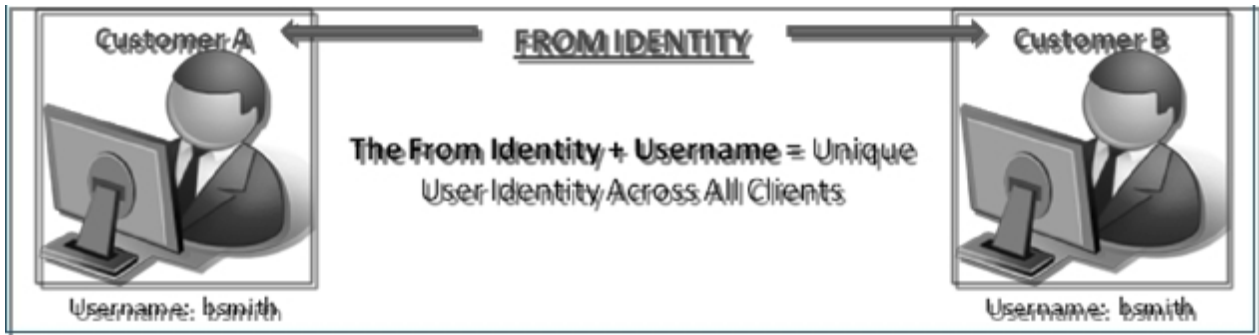
Tracking Users

Suppliers may choose to offer punch-out functionality that requires specific user identification. Functionality such as favorites, order history, and quote retrieval often rely upon the supplier having setup a user “account” that will be accessed with every punch-out connection without requiring the user to perform a login action (see [Punchout Requirements](#) section for more information). Consider the following when tracking users:

- Email addresses are unique by user; however, this field is not always available to pass to the supplier at the time of punch-out.
- Usernames are unique by user within a single client organization; however, may not be unique across multiple clients.

Recommendation:

Suppliers should consider tracking users via a combination of the "From Identity" in the header credentials and the Username as passed on the PunchOutSetupRequest Message. This provides a unique user identifier across all client organizations.



Cookies & SupplierPartAuxiliaryID

Browser Cookies

Browser cookies represent text that is stored by the user's web browser. Cookies can be used to store site preferences, the contents of a shopping cart, and facilitate user authentication among others. In punch-out applications, the cookie is used to maintain user session data during site navigation.

The cookie is sent as an HTTP header by the supplier's web server to the user's browser; however, the cookie must be sent following the redirect of the user to the shopping site. **The cookie cannot be set as part of the supplier response message to the initial PunchOutSetupRequest.**

Note: each user connection should initiate a new session cookie. This ensures that previous shopping session cart contents are cleared upon subsequent connections.

SupplierPartAuxiliaryID (SPAID)

SupplierPartAuxiliaryID (SPAID) can serve as a supplier cookie and is typically used to reference a particular item configuration or any other additional product data as needed. The SPAID is either stored by JAGGAER or exported to the client ERP system and is returned to the supplier on any outbound documents; including purchase orders and PunchOutSetupRequest Messages with an operation of "edit" or "inspect".

JAGGAER users have the ability within the application to copy previous requisitions to new carts. As part of this copy process, the SPAID is also copied. This may cause conflicts, as many

suppliers use the SPAID field in ways that are unique to the initial shopping session. JAGGAER can restrict by supplier if needed; therefore, suppliers should inform JAGGAER at the time of implementation that the copy cart functionality should be disabled.

PunchOutSetupRequest

JAGGAER will send the PunchOutSetupRequest message via HTTP POST (over SSL, port 443) to the supplier-provided URL. The <BrowserFormPost> tag contains the URL the supplier should Form Submit the PunchOutOrderMessage to when the users shopping session is complete. This URL should be dynamically read from each request message received as it **varies by customer and may change without notification**.

Sample Message (JAGGAER Default)

The below sample message represents the default PunchOutSetupRequest Message that JAGGAER will pass to the supplier. The user data passed is data that the supplier can reasonably expect to receive from each client user upon connection (see message elements table for more information). The data values bolded in green represent default values that will be passed for every client user. Suppliers should plan/develop based upon the following sample:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML payloadID="[payload ID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ" xml:lang="en-US" version="1.1.010">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>
      </Credential>
```

```

<UserAgent>JAGGAER</UserAgent>
</Sender>
</Header>
<Request>
<PunchOutSetupRequest operation="create">
<BuyerCookie>JAGGAER</BuyerCookie>
<Extrinsic name="FirstName">[punch-out User's First Name, if available]</Extrinsic>
<Extrinsic name="LastName">[punch-out User's Last Name, if available]</Extrinsic>
<Extrinsic name="UniqueName">[punch-out User's Login Name, always
available]</Extrinsic>
<Extrinsic name="UserPrintableName">[punch-out User's First Name & Last Name, if
available]</Extrinsic>
<Extrinsic name="UserOrgName" />
<Extrinsic name="UserOrgId" />
<Extrinsic name="UserOrgDept" />
<Extrinsic name="UserDept" />
<Extrinsic name="UniqueUsername">[punch-out User's Login Name, always
available]</Extrinsic>
<Extrinsic name="ReturnFrame">_self</Extrinsic>
<Extrinsic name="UserEmail">[punch-out User's Email Address, if available]</Extrinsic>
<Extrinsic name="UserPhoneNumber" />
<Extrinsic name="User">[punch-out User's Login Name, always available]</Extrinsic>
<BrowserFormPost>
<URL>[JAGGAER Dynamic URL Initiating punch-out Session]</URL>
</BrowserFormPost>
<Contact>
<Name xml:lang="en-US" />
</Contact>
<SupplierSetup>
<URL>[Supplier's punch-out URL]</URL>
</SupplierSetup>
<ShipTo>
<Address addressID="PunchoutUser">
<Name xml:lang="en-US">[Organization Name]</Name>
<PostalAddress>
<Street />
<City />
<Country isoCountryCode="" />

```

```

</PostalAddress>
<Phone>
<TelephoneNumber>
<CountryCode isoCountryCode="" />
<AreaOrCityCode />
<Number />
</TelephoneNumber>
</Phone>
</Address>
</ShipTo>
</PunchOutSetupRequest>
</Request>
</cXML>

```

Sample Message (Expanded)

The following sample represents all the available data fields for the PunchOutSetupRequest Message. Refer to the message elements table for information about limited user data availability.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML payloadID="[payload ID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ" xml:lang="en-US" version="1.1.010">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>

```

```

</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request>
<PunchOutSetupRequest operation="create">
<BuyerCookie>[Buyer Cookie]</BuyerCookie>
<Extrinsic name="FirstName">[punch-out User's First Name]</Extrinsic>
<Extrinsic name="LastName">[punch-out User's Last Name]</Extrinsic>
<Extrinsic name="UniqueName">[punch-out User's Login Name]</Extrinsic>
<Extrinsic name="UserPrintableName">[punch-out User's First Name & Last
Name]</Extrinsic>
<Extrinsic name="UserOrgName">[punch-out User's Organization Name]</Extrinsic>
<Extrinsic name="UserOrgId">[punch-out User's Organization Identifier]</Extrinsic>
<Extrinsic name="UserOrgDept">[punch-out User's Organization Dept]</Extrinsic>
<Extrinsic name="UserDept">[punch-out User's Department]</Extrinsic>
<Extrinsic name="BusinessUnit">[punch-out User's Business Unit]</Extrinsic>
<Extrinsic name="UniqueUsername">[punch-out User's Login Name]</Extrinsic>
<Extrinsic name="ReturnFrame">_self</Extrinsic>
<Extrinsic name="UserEmail">[punch-out User's Email Address]</Extrinsic>
<Extrinsic name="PhoneNumber" >[punch-out User's Phone Number]</Extrinsic>
<Extrinsic name="User">[punch-out User's Login Name]</Extrinsic>
<BrowserFormPost>
<URL>[JAGGAER Dynamic URL Initiating punch-out Session]</URL>
</BrowserFormPost>
<Contact>
<Name xml:lang="en-US" >[punch-out User's Name]</Name>
<Email>[punch-out User's Email Address]</Email>
</Contact>
<SupplierSetup>
<URL>[Supplier's punch-out URL]</URL>
</SupplierSetup>
<ShipTo>
<Address addressID="[AddressID]">
<Name xml:lang="en-US">[Organization Name]</Name>
<PostalAddress>
<DeliverTo>[Organization Name]</DeliverTo>
<Street >[Primary Street Address]</Street>

```

```

<City>[City]</City>
<State>[State Abbreviation]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="[Code]">[Country Code]</CountryCode>
</PostalAddress>
<Phone>
<TelephoneNumber>
<CountryCode isoCountryCode="[Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
<Extension>[Phone Extension]</Extension>
</TelephoneNumber>
</Phone>
</Address>
</ShipTo>
</PunchOutSetupRequest>
</Request>
</cXML>

```

Note: ShipTo addresses are not typically available to pass at the time of punch-out. As a result, the address tags will contain either no value or a single hard-coded value by default.

PunchoutSetup Message Elements

The supplier can specify the variables listed below. If a variable is not provided, the default value will be used.

Element/Attribute	Status	Description
<Header>		
From Domain	Always Passed	Default = NetworkID, DUNS accepted
From Identity	Always Passed	Typically the Institution Abbreviated Name or DUNS #
To Domain	Always Passed	Default = NetworkID, DUNS accepted
To Identity	Always Passed	NetworkID or DUNS (supplier preference)
Sender Domain	Always Passed	Default = NetworkID
Sender Identity	Always Passed	Default = JAGGAER (can be updated to reflect customer organization)

Element/Attribute	Status	Description
SharedSecret	Always Passed	Specified by the Supplier
UserAgent	Always Passed	Default = JAGGAER
<PunchOutSetupRequest>		
Operation	Always Passed	JAGGAER will pass "create" upon initial user connection. If the supplier will support the edit/inspect integration, other possible values are "edit" and "inspect". Please see the Edit/Inspect section for more information.
BuyerCookie	Always Passed	Default = JAGGAER (This is currently a fixed value in the POSR from JAGGAER and will not be variable with each transaction.)
Extrinsic name 1-13	Passed when available	<p>The UniqueUsername value will always be passed, and should be used in combination with the From Identity for account/session creation. All other extrinsic values will be passed only when available (See JAGGAER Default sample message).</p> <p>By default these values will map to their logical extrinsic name (e.g. "FirstName" maps to the User's First Name). Although the extrinsic names themselves are not configurable, the values are. Therefore, if the User's E-mail Address in the Extrinsic is called "User", this could be accommodated. If none of these Extrinsic names are valid in your system, JAGGAER can add custom Extrinsics. However, this requires custom code which extends the overall project schedule. JAGGAER will work with the supplier to determine if custom code is required.</p>
BrowserFormPostURL	Always passed	The value passed is dynamic and contains the URL the supplier should Form Submit the PunchOutOrderMessage to when the user's shopping session is complete.
Contact Name & Contact Email	Passed when available	By default these values are not populated in the PunchoutSetupRequest Message. However, if these values are present, they will be passed.
SupplierSetup - URL	Always passed	The supplier-specified URL for PunchOutSetupRequest message post and authentication.
<ShipTo> - addressID Name xml:lang <PostalAddress> - DeliverTo, Street,	Rarely passed	These values are not typically available to pass at the time of punch-out. Should the supplier require a value, a single generic address can be sent by default. JAGGAER does not send dynamic ShipTo information at the time of punch-out based on user. The default value passed for

Element/Attribute	Status	Description
City, State, Postal Code, Country		addressID is "PunchoutUser". The actual ShipTo address and corresponding address code will be sent to the supplier on the purchase order document.
General Information		
TelephoneNumber & Email Address	Passed when available	These values are not always available to pass at the time of punch-out and should not be required. Due to the unreliability of these fields, the use of email address to create user accounts/sessions is not recommended. Alternatively, the combination of the From Identity and the Username provides a unique user identify across all clients.
HTTP Header	Always Passed	When posting the PunchoutSetupRequest, the HTTP header contains a Content Type of "text/xml".

PunchOutSetupResponse

The supplier should authenticate the request credentials, initiate the user session, then return the synchronous response message to JAGGAER (via the same HTTP request path) with a status code of 200 and the https start page URL. JAGGAER will redirect the shopper to the URL provided in the supplier response, where the user should land automatically logged in and ready to shop the P3P compliant punch-out site.

Suppliers that will create individual user profiles/accounts by organization in order to provide functionality such as favorites or order history should accommodate automatic "on-the-fly" setup, such that the user is able to immediately connect. Unrecognized users should never experience a failed connection.

Sample Message - PunchOutSetupResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"
version="1.1.010" xml:lang="en-US">
  <Response>
    <Status code="200" text="[Successful Response Text]" xml:lang="en-US"/>
    <PunchOutSetupResponse>
      <StartPage>
        <URL>[https URL for Supplier PunchOut]</URL>
      </StartPage>
```

```
</PunchOutSetupResponse>
</Response>
</cXML>
```

PunchOutOrderMessage

The PunchOutOrderMessage containing the contents of the cart should be routed via Form Submit (cxml-urlencoded) to the dynamic URL provided in the initial PunchOutSetupRequest message. The return of the cart to JAGGAER does not yet constitute an order.

Sample Message (No Tax or Shipping Estimates)

The following message example is the most common PunchOutOrderMessage. JAGGAER calculates the extended price and order total using the UnitPrice and quantity elements for each line rather than parsing the amount listed in the header <Total> as sent by the supplier; however, the supplier must include <Total> as it is a required cXML element.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML version="1.1.010" payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>
      </Credential>
    <UserAgent>[User Agent]</UserAgent>
  </Sender>
</Header>
```

```

<Message deploymentMode="test" or "production">
<PunchOutOrderMessage>
<BuyerCookie>[Buyer Cookie]</BuyerCookie>
<PunchOutOrderMessageHeader operationAllowed="create">
<Total>
<Money currency="USD">[Total Price]</Money>
</Total>
</PunchOutOrderMessageHeader>
<ItemIn quantity="[Quantity]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en">[Product Description]</Description>
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<ManufacturerPartID>[Manufacturer Part Number]</ManufacturerPartID>
<ManufacturerName xml:lang="en">[Manufacturer Name]</ManufacturerName>
<LeadTime>[Number]</LeadTime>
<Classification domain="UNSPSC">[UNSPSC Code]</Classification>
<Classification domain="Hazardous">[yes/no]</Classification>
<Classification domain="Radioactive">[yes/no]</Classification>
<Classification domain="Controlled">[yes/no]</Classification>
<Classification domain="RadioactiveMinor">[yes/no]</Classification>
<Classification domain="SelectAgent">[yes/no]</Classification>
<Classification domain="Toxin">[yes/no]</Classification>
<Classification domain="EnergyStar">[yes/no]</Classification>
<Classification domain="GreenProduct">[yes/no]</Classification>
<Classification domain="Recycled">[yes/no]</Classification>
<Extrinsic name="CASNumber">[CAS Number]</Extrinsic>
</ItemDetail>
</ItemIn>
</PunchOutOrderMessage>
</Message>

```

</cXML>

Sample Message (Order Level Tax & Shipping Estimates)

JAGGAER clients typically expect that any tax and shipping costs will be included on the invoice only; however, the client may request that the supplier pass these at the order level on the PunchOutOrderMessage.

The following message example illustrates the passing of order level tax and shipping estimates. If shipping and tax is also included at the line level (shown in next example), any amounts at the header level will be ignored. Handling charges cannot be passed on the PunchOutOrderMessage, but with client approval, could be included in the shipping total. JAGGAER calculates the extended price and order total using the UnitPrice, quantity, and order level shipping and tax elements rather than parsing the amount listed in the header <Total> as sent by the supplier; however, the supplier must include <Total> as it is a required cXML element.

```
<cXML version="1.1.010" payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
```

```
<Header>
```

```
<From>
```

```
<Credential domain="[From Domain]">
```

```
<Identity>[From Identity]</Identity>
```

```
</Credential>
```

```
</From>
```

```
<To>
```

```
<Credential domain="[To Domain]">
```

```
<Identity>[To Identity]</Identity>
```

```
</Credential>
```

```
</To>
```

```
<Sender>
```

```
<Credential domain="[Sender Domain]">
```

```
<Identity>[Sender Identity]</Identity>
```

```
<SharedSecret>[Shared Secret]</SharedSecret>
```

```
</Credential>
```

```
<UserAgent>[User Agent]</UserAgent>
```

```
</Sender>
```

```
</Header>
```

```
<Message deploymentMode="test" or "production">
```

```
<PunchOutOrderMessage>
```

```

<BuyerCookie>[Buyer Cookie]</BuyerCookie>
<PunchOutOrderMessageHeader operationAllowed="create">
<Total>
<Money currency="USD">[Total Price]</Money>
</Total>
<Shipping>
<Money currency="USD">[Total Shipping]</Money>
<Description xml:lang="en">[Shipping Description]</Description>
</Shipping>
<Tax>
<Money currency="USD">[Total Tax]</Money>
<Description xml:lang="en">[Tax Description]</Description>
</Tax>
</PunchOutOrderMessageHeader>
<ItemIn quantity="[Quantity]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en">[Product Description]</Description>
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<ManufacturerPartID>[Manufacturer Part Number]</ManufacturerPartID>
<ManufacturerName xml:lang="en">[Manufacturer Name]</ManufacturerName>
<LeadTime>[Number]</LeadTime>
<Classification domain="UNSPSC">[UNSPSC Code]</Classification>
<Classification domain="Hazardous">[true/false]</Classification>
<Classification domain="Radioactive">[true/false]</Classification>
<Classification domain="Controlled">[true/false]</Classification>
<Classification domain="RadioactiveMinor">[true/false]</Classification>
<Classification domain="SelectAgent">[true/false]</Classification>
<Classification domain="Toxin">[true/false]</Classification>
<Classification domain="EnergyStar">[true/false]</Classification>
<Classification domain="GreenProduct">[true/false]</Classification>

```

```

<Classification domain="Recycled">[true/false]</Classification>
<Extrinsic name="CASNumber">[CAS Number]</Extrinsic>
</ItemDetail>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>

```

Sample Message (Line Level Tax & Shipping Estimates)

JAGGAER clients typically expect that any tax and shipping costs will be included on the invoice only; however, the client may request that the supplier pass these at the line level on the PunchOutOrderMessage.

The following message example illustrates the passing of line level tax and shipping estimates. If shipping and tax is also included at the order level (shown in previous example), any tax or shipping amounts at the order level will be ignored. Handling charges cannot be passed on the PunchOutOrderMessage, but with client approval, could be included in the shipping total. JAGGAER calculates the extended price and order total using the UnitPrice, quantity, and shipping and tax elements by line rather than parsing the amount listed in the header <Total> as sent by the supplier; however, the supplier must include <Total> as it is a required cXML element.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML version="1.1.010" payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">

```

```

<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Message deploymentMode="test" or "production">
<PunchOutOrderMessage>
<BuyerCookie>[Buyer Cookie]</BuyerCookie>
<PunchOutOrderMessageHeader operationAllowed="create">
<Total>
<Money currency="USD">[Total Price]</Money>
</Total>
<Shipping>
<Money currency="USD">[Total Shipping]</Money>
<Description xml:lang="en">[Shipping Description]</Description>
</Shipping>
<Tax>
<Money currency="USD">[Total Tax]</Money>
<Description xml:lang="en">[Tax Description]</Description>
</Tax>
</PunchOutOrderMessageHeader>
<ItemIn quantity="[Quantity]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en">[Product Description]</Description>
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<ManufacturerPartID>[Manufacturer Part Number]</ManufacturerPartID>
<ManufacturerName xml:lang="en">[Manufacturer Name]</ManufacturerName>
<LeadTime>[Number]</LeadTime>
<Classification domain="UNSPSC">[UNSPSC Code]</Classification>

```

```

<Classification domain="Hazardous">[true/false]</Classification>
<Classification domain="Radioactive">[true/false]</Classification>
<Classification domain="Controlled">[true/false]</Classification>
<Classification domain="RadioactiveMinor">[true/false]</Classification>
<Classification domain="SelectAgent">[true/false]</Classification>
<Classification domain="Toxin">[true/false]</Classification>
<Classification domain="EnergyStar">[true/false]</Classification>
<Classification domain="GreenProduct">[true/false]</Classification>
<Classification domain="Recycled">[true/false]</Classification>
<Extrinsic name="CASNumber">[CAS Number]</Extrinsic>
</ItemDetail>
<Shipping>
<Money currency="USD">[Total Shipping]</Money>
<Description xml:lang="en">[Shipping Description]</Description>
</Shipping>
<Tax>
<Money currency="USD">[Total Tax]</Money>
<Description xml:lang="en">[Tax Description]</Description>
</Tax>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>

```

Sample Message (Shipping as Line Item in Cart)

As an alternative to providing shipping charges at the order or line levels, the supplier can include shipping charges as if it were a catalog item upon client request. This is typically done when the client ERP system is unable to handle charges at the order or line levels. Tax amounts should not be passed in this manner.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML version="1.1.010" payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
<Header>
<From>
<Credential domain="[From Domain]">
<Identity>[From Identity]</Identity>
</Credential>

```



```

</From>
<To>
<Credential domain="[To Domain]">
<Identity>[To Identity]</Identity>
</Credential>
</To>
<Sender>
<Credential domain="[Sender Domain]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Message deploymentMode="test" or "production">
<PunchOutOrderMessage>
<BuyerCookie>[Buyer Cookie]</BuyerCookie>
<PunchOutOrderMessageHeader operationAllowed="create">
<Total>
<Money currency="USD">[Total Price]</Money>
</Total>
</PunchOutOrderMessageHeader>
<ItemIn quantity="[Quantity]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en">[Product Description]</Description>
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<ManufacturerPartID>[Manufacturer Part Number]</ManufacturerPartID>
<ManufacturerName xml:lang="en">[Manufacturer Name]</ManufacturerName>
<LeadTime>[Number]</LeadTime>
<Classification domain="UNSPSC">[UNSPSC Code]</Classification>

```

```

<Classification domain="Hazardous">[yes/no]</Classification>
<Classification domain="Radioactive">[yes/no]</Classification>
<Classification domain="Controlled">[yes/no]</Classification>
<Classification domain="RadioactiveMinor">[yes/no]</Classification>
<Classification domain="SelectAgent">[yes/no]</Classification>
<Classification domain="Toxin">[yes/no]</Classification>
<Classification domain="EnergyStar">[yes/no]</Classification>
<Classification domain="GreenProduct">[yes/no]</Classification>
<Classification domain="Recycled">[yes/no]</Classification>
<Extrinsic name="CASNumber">[CAS Number]</Extrinsic>
</ItemDetail>
</ItemIn>
<ItemIn quantity="[Quantity]">
<ItemID>
<SupplierPartID>ShippingCharge</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en">Shipping Costs</Description>
<UnitOfMeasure>EA</UnitOfMeasure>
<Classification domain="UNSPSC">[UNSPSC Code]</Classification>
</ItemDetail>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>

```

Sample Form Submit (Browser Form Post)

The PunchOutOrderMessage is not HTTP posted to JAGGAER, but is sent as a Form Submit (cxml- urlencoded) to the URL specified in the PunchOutSetupRequest message. The execution of the form action typically occurs when the user clicks to submit the shopping cart; and upon submittal, the web browser (rather than the web server) will inherently URL encode the message.

Note: The form action URL must not be encoded. The web server must HTML-encode the field values (bolded in green below) to ensure that all quotation marks and special characters are handled appropriately. The following is a basic HTML page illustrating the form submit:

```
<html>
<body>
<form method=post action="URL in BrowserFormPost Element of PunchOutSetupRequest
Message">
<input type="hidden" name="cxml-urlencoded" value="<?xml version="1.0" encoding="UTF-
8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML version="1.1.010" payloadID="010-08-26T00:00:00@127.0.0.1" timestamp="2010-08-
26T00:00:00">
<Header>
<From>
<Credential domain="From Domain">
<Identity>From Identity</Identity>
</Credential>
</From>
<To>
<Credential domain="To Domain">
<Identity>To Identity</Identity>
</Credential>
</To>
<Sender>
<Credential domain="Sender Domain">
<Identity>Sender Identity</Identity>
<SharedSecret>Shared Secret</SharedSecret>
</Credential>
<UserAgent>User Agent</UserAgent>
</Sender>
</Header>
<Message deploymentMode="Test">
<PunchOutOrderMessage>
<BuyerCookie>Buyer Cookie</BuyerCookie>
<PunchOutOrderMessageHeader operationAllowed="edit">
<Total>
<Money currency="USD">0.99</Money>
</Total>
```

```

</PunchOutOrderMessageHeader>
<ItemIn quantity="1">
<ItemID>
<SupplierPartID>PartID</SupplierPartID>
<SupplierPartAuxiliaryID>Auxiliary PartID</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">0.99</Money>
</UnitPrice>
<Description xml:lang="en">Part Description</Description>
<UnitOfMeasure>EA</UnitOfMeasure>
<ManufacturerPartID>Manufacturer Part Number</ManufacturerPartID>
<ManufacturerName xml:lang="en">Manufacturer Name</ManufacturerName>
<LeadTime>2</LeadTime>
<Classification domain="UNSPSC">UNSPSC Code</Classification>
<Classification domain="Hazardous">yes</Classification>
<Classification domain="Radioactive">yes</Classification>
<Classification domain="Controlled">no</Classification>
<Classification domain="RadioactiveMinor">yes</Classification>
<Classification domain="SelectAgent">no</Classification>
<Classification domain="Toxin">yes</Classification>
<Classification domain="EnergyStar">no</Classification>
<Classification domain="GreenProduct">no</Classification>
<Classification domain="Recycled">no</Classification>
<Extrinsic name="CASNumber">CAS Number</Extrinsic>
</ItemDetail>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>
">
<input type=submit value="Return Cart to Procurement Application">
</form>
</body>
</html>

```

Digital Signature

Digital Signature allows JAGGAER to confirm the sender's identity of an electronic document and is used to confirm a document was not modified after it was generated by the signer. An effective digital signature uses specific options with particular elements present to sign or not sign certain section of the XML Document.

JAGGAER utilizes the default oracle digital signature library. Digital Signature Core Validation consists of four elements:

- cXMLSignedInfo
- cXMLData
- #XAdESSignedProps
- Signature Validation.

Suppliers must be able to test and validate a digitally signed form post within their application prior to testing with JAGGAER. We recommend the following reference material for suppliers to assist with Digital Signature development and troubleshooting

- XML Signature Syntax and processing
 - <https://www.w3.org/TR/xmlsig-core/#sec-o-Simple>
- Oracle Digital Signature Library
 - <https://docs.oracle.com/javase/tutorial/security/apisign/index.html>
- cXML Digital Signatures cXML user guide
 - <http://xml.cxml.org/current/cXMLUsersGuide.pdf>
- Programming With the Java XML Digital Signature API
 - <http://www.oracle.com/technetwork/articles/javase/dig-signature-api-140772.html>
- Can the Supplier stop the website before it posts the message?
- A method to validate the signed document within their own application.

Sample Digital Signature XML

The example below would be appended after the </message> element in the XML of the punchout order message:

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="cXMLSignature">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#cXMLSignedInfo">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>/haY/N5RYEKiOT5knvLRLvnSk9I=</ds:DigestValue>
```

```

</ds:Reference>
<ds:Reference URI="#XAdESSignedProps">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>s3wLBaM4LuoXJBIpUPGYICWWL5U=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>[Encrypted Signature Value here]</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509SubjectName>CN=b2bdirect.dell.com, OU=Information Technology, O=Dell Inc,
L=Round Rock, S=Texas, C=US</ds:X509SubjectName>
<ds:X509Certificate>[Encrypted Certificate data here]</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
<ds:Object>
<xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
Target="#cXMLSignature">
<xades:SignedProperties Id="XAdESSignedProps">
<xades:SignedSignatureProperties>
<xades:SigningTime>2016-11-01T18:59:14Z</xades:SigningTime>
<xades:SigningCertificate>
<xades:Cert>
<xades:CertDigest>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:DigestValue>R97WVYkRkx3ckU2HwxcT0ASypBw=</ds:DigestValue>
</xades:CertDigest>
<xades:IssuerSerial>
<ds:X509IssuerName>CN=Symantec Class 3 Secure Server SHA256 SSL CA, OU=Symantec
Trust Network, O=Symantec Corporation, C=US</ds:X509IssuerName>
<ds:X509SerialNumber>2FDC4B3DCEEC219C5A293ED5305D0E69</ds:X509SerialNumber>
</xades:IssuerSerial>
</xades:Cert>
</xades:SigningCertificate>
<xades:SignaturePolicyIdentifier>
<xades:SignaturePolicyImplied />
</xades:SignaturePolicyIdentifier>
</xades:SignedSignatureProperties>
</xades:SignedProperties>

```

```

</xades:QualifyingProperties>
</ds:Object>
<ds:Object>
<cXMLSignedInfo Id="cXMLSignedInfo" payloadID="2016-11
01T13:59:14.12349832.13284838@dell.com" signatureVersion="1.0" />
</ds:Object>
</ds:Signature>
</cXML>

```

PunchoutOrder Message Elements

Element/Attribute	Status	Description
<Header> (same values as PunchoutSetupRequestMessage)		
From Domain	Mandatory	Default = NetworkID, DUNS accepted
From Identity	Mandatory	Typically the Institution Abbreviated Name or DUNS #
To Domain	Mandatory	Default = NetworkID, DUNS accepted
To Identity	Mandatory	NetworkID or DUNS (supplier preference)
Sender Domain	Mandatory	Default = NetworkID
Sender Identity	Mandatory	Default = JAGGAER
SharedSecret	Optional	Specified by the Supplier
UserAgent	Optional	Default = JAGGAER
<Message>		
deploymentMode	Optional	Accepted values are "test" or "production". Case sensitive.
<PunchOutOrderMessage>		
BuyerCookie	Optional	Default = JAGGAER (This is currently a fixed value in the POSR from JAGGAER and will not be variable with each transaction.)
<PunchOutOrderMessageHeader>		
operationAllowed	Mandatory	Accepted values are "create" and "edit". If the supplier will support the edit/inspect integration, the expected value is "edit"; otherwise, the expected value is "create". For more information please see the Edit/Inspect section.
<Total> Money currency	Mandatory	The supplier must include these tags in the PunchOutOrderMessage; however, JAGGAER will ignore the value passed by the supplier in favor of calculating the order total as follows:
Total	Mandatory	Extended Price (by line) = quantity x UnitPrice

Element/Attribute	Status	Description
		Total = Sum of Extended Price + applicable tax and shipping fees For more information, please see the Calculations section of the Data Standards chapter.
<ItemIn>		
quantity	Mandatory	Item quantity. JAGGAER can accept leading zeros on quantity if needed. For more information about numerical formats and calculations, please see the Data Standards chapter.
<ItemID>		
SupplierPartID	Mandatory	Defines the catalog number for the ordered product. Character limit is 100 and should contain no special characters.
SupplierPartAuxiliaryID	Supplier-Required	SupplierPartAuxiliaryID can serve as a supplier cookie and is typically used to reference a particular item configuration or any other additional product data as needed. For more information, please see the SupplierPartAuxiliaryID section. Character limit is 100 and should contain no special characters.
<ItemDetail>		
UnitPrice/Money @ currency	Mandatory	Value in two decimal places, USD by default. (e.g. \$1.99). Some client configurations may support additional decimal places. For more information about numerical formats and calculations, please see the Data Standards section.
Description xml:lang	Mandatory	Product description. Character limitation is 256. Supplier should avoid any special characters. For more information about special characters, please see the Data Standards section.
UnitOfMeasure	Mandatory	The item UOM code. For more information about expected system UOM values, please see the Data Standards section.
ManufacturerPartID	Client-Required	Field to denote the manufacturer part number of the ordered product. This may be the same value as the SupplierPartID in cases where the supplier manufactures their own products. Character limit is 100 and should contain no special characters.
ManufacturerName	Client-Required	Field to denote the manufacturer name of the ordered product. If the supplier manufactures their own products, include the supplier name in this field. Character limit is 100.
LeadTime	Optional	The number of days before the ordered product is received

Element/Attribute	Status	Description
		by the user. (e.g. 3). This is a numerical value only and should not contain any text.
Classification domain: UNSPSC	Client-Required	Supplier to return a valid 8 digit code without the dash delimiters. For more information about UNSPSC values, please see the Data Standards chapter.
Classification domain: Hazardous, Radioactive, Controlled, RadioactiveMinor, SelectAgent, Toxin, EnergyStar, GreenProduct, Recycled	Client Required	Suppliers offering products of a regulatory nature may be required, upon client request, to "flag" such products. The cXML classification standard is used to achieve this. JAGGAER will configure to accept based upon the supplier's domain value nomenclature. For example: <Classification domain="HazMat">yes</Classification> is an acceptable format. JAGGAER will configure "HazMat" internally to correlate this to the Hazardous system flag. Note: Sending product flags as a standard extrinsic is not sufficient – classification domain must be used. Additionally, suppliers must maintain the same domain value nomenclature across all customer integrations.
Extrinsic name: CASNumber	Client-Required	CAS Number is applicable only to chemical products, and is typically a client-required field. The CAS number should be passed without dash delimiters using cXML extrinsic standards. For example: <Extrinsic name="CAS">12345</Extrinsic> JAGGAER will configure "CAS" internally to correlate this to the CAS Number field within the JAGGAER application. Note: Suppliers must maintain the same extrinsic value nomenclature across all customer integrations.
Additional Elements		
<Shipping> Money currency Description xml:lang	Client-Required	Typically customers expect that any shipping costs will be included on the invoice only; however, the client may request shipping charges be passed at the order level, line level, or as a catalog line item on the PunchOutOrderMessage.
<Tax> Money currency Description xml:lang	Client-Required	Typically customers expect that any applicable tax charges will be included on the invoice only; however, the client may request that these charges be passed at the order or line level as part of the PunchOutOrderMessage. Tax as a line item in the cart is not acceptable.
Handling	NOT ACCEPTED	Handling charges cannot be passed on the PunchOutOrder. Message, but with client approval, could be included in the shipping total.

Edit / Inspect

The edit/inspect integration applies to punch-out catalog integrations and allows end users to access a previous punch-out shopping session to either edit cart contents, or inspect the items ordered (without cart changes). While these are individual operations, the JAGGAER application does not deploy the edit operation without also deploying the inspect operation; however, the inspect operation can be deployed by itself. JAGGAER recommends that the supplier either support both the edit and inspect operations OR perform the “edit” function for both operations.

DTD: CXML.dtd

Version: 1.1.010

Guides: [Edit/Inspect Troubleshooting Guide](#), [PunchOut Troubleshooting Guide](#), [cXML User's Guide](#)

JAGGAER conforms to the cXML 1.1.010 DTD standard for punch-out integrations. JAGGAER should be able to integrate with any supplier using this version; however, if the supplier is using another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, see the [Supported cXML Versions](#) section.

Edit - Expected Behavior

The expected behavior for the edit operation is that the shopper will be immediately returned to the punch-out site shopping cart from the original session to make changes. When changes are complete, the shopper will perform the same actions to return the updated PunchOutOrderMessage. The supplier should update all data fields as necessary, including the SupplierPartAuxiliaryID where applicable.

Inspect - Expected Behavior

The expected behavior for the inspect operation is that the shopper will be immediately returned to the punch-out site shopping cart from the original session to simply view the contents (no changes). If the supplier does not truly support the inspect function, the supplier can perform the same action the edit operation would perform when an operation of “inspect” is requested.

The following sections outline how the initial PunchOutOrderMessage from the supplier and the subsequent PunchOutSetupRequest message changes when edit/inspect is in operation.

PunchOutOrderMessage Changes

The supplier can return the PunchOutOrderMessage during the initial user shopping session as outlined in the previous chapter, with one change. If the supplier supports both edit and inspect, the operationAllowed attribute should be returned with a value of “edit”. If only the inspect function is supported, the operationAllowed attribute should be returned with a value of

"inspect". This will trigger the JAGGAER system to display the edit and inspect links accordingly to the user within the JAGGAER procurement application.

```
<PunchOutOrderMessageHeader operationAllowed="edit">
```

Or

```
<PunchOutOrderMessageHeader operationAllowed="inspect">
```

PunchOutSetupRequest Message Changes

Should the shopper wish to edit or inspect the shopping cart initially returned by the supplier, a link from the JAGGAER application will be accessed that triggers the re-initiation of the previous punch-out shopping session. A PunchOutSetupRequest message will be passed to the supplier as outlined in the previous chapter, with four changes. The operationAllowed attribute value of either "edit" or "inspect" will be passed (depending upon the operation requested by the shopper), the original punch-out session ID will be passed in <BuyerCookie>, the line item details of the original cart contents will be included, and the <BrowserFormPost> URL will dynamically change. See the bolded selections in the below sample message.

Sample Message (Edit/Inspect)

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

```
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
```

```
<cXML payloadID="[payload ID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ" xml:lang="en-US" version="1.1.010">
```

```
<Header>
```

```
<From>
```

```
<Credential domain="[From Domain]">
```

```
<Identity>[From Identity]</Identity>
```

```
</Credential>
```

```
</From>
```

```
<To>
```

```
<Credential domain="[To Domain]">
```

```
<Identity>[To Identity]</Identity>
```

```
</Credential>
```

```
</To>
```

```
<Sender>
```

```

<Credential domain="[Sender Domain]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>JAGGAER</UserAgent>
</Sender>
</Header>
<Request>
<PunchOutSetupRequest operation="edit">
<BuyerCookie>[Punchout Session ID]</BuyerCookie>
<Extrinsic name="FirstName">[punch-out User's First Name]</Extrinsic>
<Extrinsic name="LastName">[punch-out User's Last Name]</Extrinsic>
<Extrinsic name="UniqueName">[punch-out User's Login Name]</Extrinsic>
<Extrinsic name="UserPrintableName">[punch-out User's First Name & Last
Name]</Extrinsic>
<Extrinsic name="UserOrgName" />
<Extrinsic name="UserOrgId" />
<Extrinsic name="UserOrgDept" />
<Extrinsic name="UserDept" />
<Extrinsic name="UniqueUsername">[punch-out User's Login Name]</Extrinsic>
<Extrinsic name="ReturnFrame">_self</Extrinsic>
<Extrinsic name="UserEmail">[punch-out User's Email Address]</Extrinsic>
<Extrinsic name="UserPhoneNumber" />
<Extrinsic name="User">[punch-out User's Login Name]</Extrinsic>
<BrowserFormPost>
<URL>[JAGGAER Dynamic URL re-initiating punch-out Session]</URL>
</BrowserFormPost>
<Contact>
<Name xml:lang="en-US" />

```

```

</Contact>
<SupplierSetup>
<URL>[Supplier's punch-out URL]</URL>
</SupplierSetup>
<ItemOut quantity="[Quantity]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary ID]</SupplierPartAuxiliaryID>
</ItemID>
</ItemOut>
<ShipTo>
<Address addressID="PunchoutUser">
<Name xml:lang="en-US" />
<PostalAddress>
<Street />
<City />
<Country isoCountryCode="" />
</PostalAddress>
<Phone>
<TelephoneNumber>
<CountryCode isoCountryCode="" />
<AreaOrCityCode />
<Number />
</TelephoneNumber>
</Phone>
</Address>
</ShipTo>
</PunchOutSetupRequest>
</Request>

```

</cXML>

Note: ShipTo addresses are not typically available to pass at the time of punch-out. As a result, the address tags will contain either no value or a single hard-coded value by default. JAGGAER can optionally omit the ShipTo section for edit & inspect operations

SmartFind

SmartFind enables suppliers to display punch-out items and items with real-time pricing in search results within the JAGGAER application, which are typically only visible with a traditional hosted catalog. Through SmartFind options, suppliers can provide a **Proxy Catalog** and buyer organization accessibility information in order to have products visible in JAGGAER search results.

When suppliers provide item data through the SmartFind **Proxy Catalog**, buyer organizations can view content to make purchasing decisions and can be displayed with the following SmartFind pricing sources:

- **Level 2 PunchOut** – allows items that can normally only be found through a punch-out session to display in the JAGGAER search results. Users are presented with a link within the search results that directs the user to the supplier site for purchase and pricing information. This allows buying organizations to fully leverage SpendDirector features such as product compare and favorites against punch-out catalog items.
- **LivePrice** – allows the user to check for the real-time price of an item, which is provided by the supplier dynamically upon user request.

More information is available in the Online Training and Support page within the Supplier Network Portal.

Proxy Catalog & Accessibility Files

In order for the Level 2 PunchOut and LivePrice integrations to function, the supplier must provide the content that will be hosted and displayed within the JAGGAER search results. Instead of static pricing, an accessibility file is loaded in its place to facilitate the display of the proxy content by requesting client. Suppliers that have already provided hosted catalog data need only provide the accessibility file.

Proxy Catalog

The Proxy Catalog can be generated through the use of the Content Management Tool (CMT) and is loaded to JAGGAER by the supplier via the Supplier Network Portal. The CMT and supporting documentation is available on the Online Training and Support page in the Supplier Network portal.

As new products are introduced or retired, the supplier is responsible to update the proxy catalog content accordingly.

Proxy Catalog Accessibility

The Proxy Catalog Accessibility (PCA) file is loaded in place of static pricing and controls the visibility of products by client. The PCA template and supporting documentation is available on

the Online Training and Support page in the Supplier Network portal.

Level 2 PunchOut

The Level 2 PunchOut integration takes the standard punch-out integration a step further in that it brings punch-out content closer to buying organizations. In addition to the functionality and requirements listed in the PunchOut chapter, suppliers can provide a Proxy Catalog and accessibility information for non-configurable punch-out items that display within the JAGGAER search results. Supplier content will display with a link to the supplier punch-out directly in the user's search results screen. When accessed, the PunchOutSetupRequest Message passes the selected product data to the supplier based upon supplier capabilities for parsing this additional data:

- **Store Level** – directs the user to the standard punch-out site landing page. This option does not pass product data on the PunchOutSetupRequest Message and requires that the user re-search for the item on the supplier site. Shoppers can add items to the cart as in normal operation.
- **Aisle Level** – directs the user to the product category page of the requested item upon initial user connection. To facilitate this, JAGGAER will pass either the UNSPSC code or the Category Name of the product selected in the PunchOutSetupRequest Message for supplier parsing based upon supplier preference. Shoppers can add category items to the cart and check-out, or continue to shop the punch-out site as in normal operation.
- **Product Level (Preferred)** – directs the user to the specific page of the requested item upon initial user connection. JAGGAER will pass the part number, the clean part number (no dashes or delimiters), or the manufacturer part number in the PunchOutSetupRequest Message for supplier parsing based upon supplier preference. Shoppers can add the item to the cart and check-out, or continue to shop the punch-out site as in normal operation.

The supplier can choose only one of the above options that will apply for all client Level 2 integrations. In the event the product data passed is no longer valid, the supplier must either land the user on the home page, or a custom page that indicates the selected product is no longer available. The shopper must never experience connection failure due to product obsolescence - the supplier should ensure that the proxy catalog data is up to date.

Successful integration testing per the provided PunchOut Test Plan is required before promotion to production can be considered.

DTD: CXML.dtd

Version: 1.1.010

Guides: [Level 2 PunchOut Troubleshooting Guide](#), [PunchOut Troubleshooting Guide](#), [cXML User's Guide](#)

Sample Message (Level 2 PunchOut)

The following is a sample Level 2 PunchOutSetupRequest Message that passes either the UNSPSC code, Category Name, Part Number, Clean Part Number (no delimiters), or the Manufacturer Part Number based upon supplier preference as outlined in the previous section:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.1.010/cXML.dtd">
<cXML payloadID="[payload ID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ" xml:lang="en-US" version="1.1.010">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>
      </Credential>
      <UserAgent>JAGGAER</UserAgent>
    </Sender>
  </Header>
  <Request>
    <PunchOutSetupRequest operation="create">
      <BuyerCookie>JAGGAER</BuyerCookie>
      <Extrinsic name="FirstName">[punch-out User's First Name]</Extrinsic>
      <Extrinsic name="LastName">[punch-out User's Last Name]</Extrinsic>
      <Extrinsic name="UniqueName">[punch-out User's Login Name]</Extrinsic>
      <Extrinsic name="UserPrintableName">[punch-out User's First Name & Last Name]</Extrinsic>
      <Extrinsic name="UserOrgName" />
      <Extrinsic name="UserOrgId" />
    </PunchOutSetupRequest>
  </Request>
</cXML>
```

```

<Extrinsic name="UserOrgDept" />
<Extrinsic name="UserDept" />
<Extrinsic name="UniqueUsername">[punch-out User's Login Name]</Extrinsic>
<Extrinsic name="ReturnFrame">_self</Extrinsic>
<Extrinsic name="UserEmail">[punch-out User's Email Address]</Extrinsic>
<Extrinsic name="UserPhoneNumber" />
<Extrinsic name="User">[punch-out User's Login Name]</Extrinsic>
<BrowserFormPost>
<URL>[JAGGAER Dynamic URL re-initiating punch-out Session]</URL>
</BrowserFormPost>
<Contact>
<Name xml:lang="en-US" />
</Contact>
<SupplierSetup>
<URL>[Supplier's punch-out URL]</URL>
</SupplierSetup>
<ShipTo>
<Address addressID="PunchoutUser">
<Name xml:lang="en-US" />
<PostalAddress>
<Street />
<City />
<Country isoCountryCode="" />
</PostalAddress>
<Phone>
<TelephoneNumber>
<CountryCode isoCountryCode="" />
<AreaOrCityCode />
<Number />
</TelephoneNumber>
</Phone>
</Address>
</ShipTo>
<SelectedItem>
<ItemID>
<SupplierPartID>[Supplier Preferred Value]</SupplierPartID>
</ItemID>

```

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

LivePrice

LivePrice allows buying organizations to retrieve current contracted pricing in real time for proxy catalog items. LivePrice is an alternative to providing static hosted pricing, particularly for commodity items where pricing changes frequently. The proxy catalog items appear in the user's search results along with a notification that the dynamic pricing is being retrieved from the supplier. Once the supplier response is received, the pricing is displayed to the user and is valid for a supplier-specified amount of time.

DTD: PriceAvailabilityMessage.dtd

Guides: [LivePrice Troubleshooting Guide](#), [cXML User's Guide](#)

JAGGAER conforms to the PriceAvailabilityMessage.dtd standard for LivePrice integrations. JAGGAER should be able to integrate with any supplier using this version; however, if the supplier is using another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, see the [Supported cXML Versions](#) section.

The following sections outline the requirements and settings necessary for the LivePrice integration to function.

LivePrice Integration Requirements

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the LivePrice integration:

- The client organization's current, contracted pricing must be returned.
- The supplier must support synchronous response messaging.
- The request messageID must be retained for return to JAGGAER on the LivePrice response message.
- The supplier must set the number of products allowed per request message as well as the cache lifespan via the Supplier Portal.
- All other requirements as listed in the [Message Elements](#) table.
- Successful integration testing is required before promotion to production can be considered.

Supplier Network Settings

The supplier has the ability to control several LivePrice settings via the Supplier Network. The settings apply to all LivePrice integrations enabled and are not configurable by client.

LivePrice Support

The supplier indicates support of LivePrice via the Supplier Network. If this setting is not active, the LivePrice functionality will not be available to any requesting client.

Items Per LivePrice Call

The supplier can set the maximum number of items to be included in a single cXML request message. A lower number of products per message would require the supplier to process a higher number of overall requests, while a larger number reduces the overall number of requests.

Care should be taken in determining this number as a balance between performance and efficiency is ideal. The amount of time required for the supplier to execute price lookup and send back a response message increases with more items. This may appear to the user that the supplier is slow to respond and could potentially lead to message timeout errors.

JAGGAER recommends the supplier support up to 10 items per request message. JAGGAER will wait 2 seconds before resending the request and will display an error to the user if 3 consecutive attempts fail.

LivePrice Cache Lifespan

When a LivePrice for an item is returned from a supplier, it is stored in a cache on JAGGAER's server. The next time the price is requested for this item, it will come directly from the cache instead of initiating another LivePrice call to the supplier.

The LivePrice Cache Lifespan setting controls how long a price that has been cached is valid until it becomes stale or expires. Once a price has expired, the next request for item pricing will initiate a new LivePrice message to the supplier. The lifespan for LivePrice cache is 24 hours by default, but can be adjusted by the supplier within the Supplier Portal. The supplier has the option to set the cache lifespan in hours, days, or weeks.

Care must be taken when determining how long prices should remain in the LivePrice cache. The longer the lifespan, the fewer the overall number of request messages will be sent to the supplier and the faster the prices will display to the user. On the other hand, the longer the prices stay in the cache, the higher potential they have to become stale.

For more detailed information, please see the SmartFind information in the [Supplier Network Handbook](#) found in the Supplier Network Portal.

LivePrice Request Message

When a product appears in search results and pricing is not cached, JAGGAER will pass the LivePrice request message to the supplier via HTTP post (over SSL, port 443) to the supplier-provided URL.

Sample Message- LivePrice Request

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

<!DOCTYPE PriceAvailabilityRequest SYSTEM "http://solutions.sciquest.com/app_
docs/dtd/priceAvailability/PriceAvailabilityMessage.dtd">

<PriceAvailabilityRequest>

<Header>

<MessageId>[MessageID]</MessageId>

<Timestamp>YYYY-MM-DD'T'HH:mm:ssZ</Timestamp>

<Authentication>

<Identity>[Sender Identity]</Identity>

<SharedSecret>[Shared Secret]</SharedSecret>

</Authentication>

</Header>

<RequesterInfo>

<Customer code="[Customer Code]" />

<Location code="[Location Code]" />

<PreferredLanguage>English</PreferredLanguage>

<PreferredCountry>[Country Code]</PreferredCountry>

</RequesterInfo>

<ProductRequest itemNumber="1">

<SupplierPartNumber>[Part Number]</SupplierPartNumber>

<UnitOfMeasure quantity="[Product Size]" unit="[UOM]" />

<Quantity>[Quantity]</Quantity>

</ProductRequest>

<ProductRequest itemNumber="2">

<SupplierPartNumber>[Part Number]</SupplierPartNumber>

<UnitOfMeasure quantity="[Product Size]" unit="[UOM]" />

<Quantity>[Quantity]</Quantity>
```

</ProductRequest>

</PriceAvailabilityRequest>

Message Elements - LivePrice Request

Element/Attribute	Description
<Header>	
MessageId	JAGGAER assigns a MessageId to each outbound message. This value must be returned in the supplier response message in <RelatedMessageId>
Timestamp	JAGGAER will pass the timestamp per the ISO-8601 format outlined in the Date/Timestamp Format section.
<Authentication>	
Identity	Default = JAGGAER; however, this value is configurable
SharedSecret	Specified by the Supplier
<RequesterInfo>	
Customer code	Attribute value used to identify the buying organization. The required string attribute "code" is the supplier's reference number/code for the requesting client's pricing.
Location code	Attribute value used to identify the user's location. The required string attribute "code" is the supplier's reference/code for the requesting user's location. This is optional, but is typically used with clients that have multiple locations across states/countries.
PreferredLanguage	Preferred language for the text display in the supplier response. The default value passed is "English". Some languages may not be supported by some suppliers.
PreferredCountry	Element value passed is the two-letter country code per ISO 3166 standards (e.g. "US") and is applicable to data provided in the "OtherInfo" response elements. Some languages may not be supported by some suppliers.
<ProductRequest>	
ItemNumber	Attribute values are sequential, starting at 1. The number of products included in the message will dictate the number of itemNumber elements included (minimum is 1)
SupplierPartNumber	The supplier's part number as defined in the supplier-provided Proxy Catalog.
UnitofMeasure quantity	This quantity attribute indicates the number of units per UOM and is only passed when the supplier has provided this value for the product in the Proxy Catalog. For example: <UnitOfMeasure quantity="12" unit="CS" /> indicates there are 12 units per case.
UnitofMeasure unit	The supplier's UOM for the product as defined in the supplier-provided Proxy Catalog. <UnitOfMeasure quantity="12" unit="CS" />
Quantity	The user's requested quantity of the product per UOM. The supplier can use this to determine the extent of product availability.
HTTP Header	When posting the LivePrice request, the HTTP header contains a Content Type of "text/xml".

Sample Message - LivePrice Response

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PriceAvailabilityResponse SYSTEM "PriceAvailabilityMessage.dtd">
<PriceAvailabilityResponse>
  <Header>
    <MessageId>[Supplier MessageID]</MessageId>
    <RelatedMessageId>[MessageID from Original Request]</RelatedMessageId>
    <Timestamp>YYYY-MM-DD'T'HH:mm:ssZ</Timestamp>
  </Header>
  <ProductResponse itemNumber="1" status="OK">
    <SupplierPartNumber>[Part Number]</SupplierPartNumber>
    <UnitPrice currency="USD">[UnitPrice]</UnitPrice>
    <QuantityAvailable>[Qty Available]</QuantityAvailable>
    <IsAvailable>["true", "false", or "unknown"]</IsAvailable>
    <QuantityBackordered>[Backordered Quantity]</QuantityBackordered>
    <ShipDate>[Ship Date]</ShipDate>
    <BackorderShipDate>[Backordered Ship Date]</BackorderShipDate>
    <SubstitutePartNumber>[Substitute Part Number]</SubstitutePartNumber>
    <SubstituteUnitOfMeasure quantity="[Product Size]" unit="[UOM]"/>
    <OtherInfo name="MSDS" type="url">[URL]</OtherInfo>
    <OtherInfo name="Description" type="text">[Description]</OtherInfo>
    <OtherInfo name="Legal Disclaimer" type="text">[Legal Mumbo Jumbo]</OtherInfo>
  </ProductResponse>
  <ProductResponse itemNumber="2" status="OK">
    <SupplierPartNumber>[Part Number]</SupplierPartNumber>
    <UnitPrice currency="USD">[UnitPrice]</UnitPrice>
    <QuantityAvailable>[Qty Available]</QuantityAvailable>
```



```

<IsAvailable>["true", "false", or "unknown"]</IsAvailable>
<QuantityBackordered>[Backordered Quantity]</QuantityBackordered>
<ShipDate>[Ship Date]</ShipDate>
<BackorderShipDate>[Backordered Ship Date]</BackorderShipDate>
<SubstitutePartNumber>[Substitute Part Number]</SubstitutePartNumber>
<SubstituteUnitOfMeasure quantity="[Product Size]" unit="[UOM]"/>
<OtherInfo name="MSDS" type="url">[URL]</OtherInfo>
<OtherInfo name="Description" type="text">[Description]</OtherInfo>
<OtherInfo name="Legal Disclaimer" type="text">[Legal Mumbo Jumbo]</OtherInfo>
</ProductResponse>
</PriceAvailabilityResponse>

```

Message Elements - LivePrice Response

Element/Attribute	Status	Description
<Header>		
MessageId	Mandatory	The supplier response MessageId.
RelatedMessageId	Mandatory	The value for this element must be the MessageId of the originating request sent by JAGGAER
Timestamp	Mandatory	The timestamp per the ISO-8601 format outlined in the Date/Timestamp Format section.
<ProductResponse>		
ItemNumber	Mandatory	The value for this attribute must match the line item number from the original request sent by JAGGAER.
Status	Mandatory	Expected attribute values are either "OK" for a successful product match, or "Error" for an unsuccessful match.
SupplierPartNumber	Mandatory	The supplier part number for the product. This value must match the part number of the original request sent by JAGGAER.
UnitPrice	Mandatory	The client's current, contracted price for the related part number. The value must be US-formatted, even in cases where the currency is not "USD". The currency attribute must contain a client-supported currency. Please see the Supported Currencies section within this document.
QuantityAvailable	Conditional Requirement	The quantity of product that is currently available. This element is required when the <IsAvailable> element is not present. This element takes precedence if both

Element/Attribute	Status	Description
		<QuantityAvailable> and <IsAvailable> are present.
IsAvailable	Conditional Requirement	An indicator of availability with accepted values of "true", "false", or "unknown". This element is required when the <QuantityAvailable> element is not present. The <QuantityAvailable> takes precedence when both <QuantityAvailable> and <IsAvailable> are present. JAGGAER recommends this element be used only in cases where the supplier is unable to provide a numeric value for quantity available.
QuantityBackordered	Optional	The quantity of product on backorder or is not available.
ShipDate	Optional	The estimated date the product will be available to ship (YYYY-MM-DD).
BackorderShipDate	Optional	The estimated date backordered product will be available to ship (YY-MM-DD)
SubstitutePartNumber	Optional	If the requested part number is backordered, this value represents a suggested alternative product.
SubstituteUnitOfMeasure	Optional	The UOM for the suggested, substitute part number.
Otherinfo name type	Optional	<p>Zero or more of these elements can be provided at supplier discretion, and represent additional information about the requested product.</p> <p>If OtherInfo elements will be sent, the name attribute will serve as the display name to the user/shopper and must be present. This type attribute is optional with "text" and "url" as accepted values. When absent, a "text" type is assumed.</p> <p>OtherInfo element values can be enclosed with CDATA tags if desired. The name attribute and element values do not support HTML tags.</p>

Availability Call

The Availability Call allows buying organizations the option to retrieve current product availability in real time for hosted catalog items. A link to request product availability from the supplier is present in search results and will display to the user upon receipt of the response message.

The Availability Call integration is based upon the LivePrice message format, with two significant differences. Any pricing provided by the supplier in the response message will not be displayed or parsed in favor of the static, contracted pricing loaded as part of the hosted catalog. Additionally, the Availability Call message will contain a single product per request and does not require any supplier configuration or maintenance through the Portal. The Availability Call is intended only to provide users/shoppers with product availability.

DTD: PriceAvailabilityMessage.dtd

Guides: [Availability Call Troubleshooting Guide](#), [cXML User's Guide](#)

JAGGAER conforms to the PriceAvailabilityMessage.dtd standard for Availability Call integrations. JAGGAER should be able to integrate with any supplier using this version; however, if the supplier is using another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, see the [Supported cXML Versions](#) section.

The following sections outline the requirements for the Availability Call integration.

Availability Call Integration Requirements

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the **Availability Call** integration:

- The supplier must support synchronous response messaging.
- The request messageID must be retained for return to JAGGAER on the Availability Call response message.
- All other requirements as listed in the Message Elements table.
- Successful integration testing is required before promotion to production can be considered.

Availability Call Request Message

When a product appears in search results, an availability link will be displayed to the user/shopper. When selected, JAGGAER will pass the Availability Call request message to the supplier via HTTP post (over SSL, port 443) to the supplier-provided URL.

Sample Message - Availability Call (Request)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PriceAvailabilityRequest SYSTEM
"http://solutions.sciquest.com/app_docs/dtd/priceAvailability/PriceAvailabilityMessage.dtd">
<PriceAvailabilityRequest>
  <Header>
    <MessageId>[MessageID]</MessageId>
    <Timestamp>YYYY-MM-DD'T'HH:mm:ssZ</Timestamp>
    <Authentication>
      <Identity>[Sender Identity]</Identity>
      <SharedSecret>[Shared Secret]</SharedSecret>
    </Authentication>
  </Header>
  <RequesterInfo>
    <Customer code="[Customer Code]" />
    <Location code="[Location Code]" />
    <PreferredLanguage>English</PreferredLanguage>
    <PreferredCountry>[Country Code]</PreferredCountry>
  </RequesterInfo>
  <ProductRequest itemNumber="1">
    <SupplierPartNumber>[Part Number]</SupplierPartNumber>
    <UnitOfMeasure quantity="[Product Size]" unit="[UOM]" />
    <Quantity>[Quantity]</Quantity>
  </ProductRequest>
</PriceAvailabilityRequest>
```

Message Elements - Availability Call (Request)

Element/Attribute	Description
<Header>	
MessageId	JAGGAER assigns a MessageId to each outbound message. This value must be returned in the supplier response message in <RelatedMessageId>
Timestamp	JAGGAER will pass the timestamp per the ISO-8601 format outlined in the Date/Timestamp Format section within this document.
<Authentication>	
Identity	Default = JAGGAER; however, this value is configurable
SharedSecret	Specified by the Supplier
<RequesterInfo>	
Customer code	Attribute value used to identify the buying organization. The required string attribute "code" is the supplier's reference number/code for the requesting client.
Location code	Attribute value used to identify the user's location. The required string attribute "code" is the supplier's reference/code for the requesting user's location. This is optional, but is typically used with clients that have multiple locations across states/countries.
PreferredLanguage	Preferred language for the text display in the supplier response. The default value passed is "English". Some languages may not be supported by some suppliers.
PreferredCountry	Element value passed is the two-letter country code per ISO 3166 standards (e.g. "US") and is applicable to data provided in the "OtherInfo" response elements. Some languages may not be supported by some suppliers.
<ProductRequest>	
ItemNumber	Attribute value is "1".
SupplierPartNumber	The supplier's part number as defined in the supplier-provided Hosted Catalog.
UnitofMeasure quantity	This quantity attribute indicates the number of units per UOM and is only passed when the supplier has provided this value for the product in the Hosted Catalog. For example: <UnitOfMeasure quantity="12" unit="CS" /> indicates there are 12 units per case.
UnitofMeasure unit	The supplier's UOM for the product as defined in the supplier-provided Hosted Catalog. <UnitOfMeasure quantity="12" unit="CS" />
Quantity	The user's requested quantity of the product per UOM. The supplier can use this to determine the extent of product availability.
HTTP Header	When posting the LivePrice request, the HTTP header contains a Content Type of "text/xml".

Availability Call Response Message

The supplier response is synchronous in nature and should be returned to JAGGAER via the same HTTP request path. In the event the response message is not returned, an error will be displayed to the user.

[Click here to see a Sample Message](#)

[Click here to see Message Elements](#)

Sample Message - Availability Call (Response)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PriceAvailabilityResponse SYSTEM "PriceAvailabilityMessage.dtd">
<PriceAvailabilityResponse>
  <Header>
    <MessageId>[Supplier MessageID]</MessageId>
    <RelatedMessageId>[MessageID from Original Request]</RelatedMessageId>
    <Timestamp>YYYY-MM-DD'T'HH:mm:ssZ</Timestamp>
  </Header>
  <ProductResponse itemNumber="1" status="OK">
    <SupplierPartNumber>[Part Number]</SupplierPartNumber>
    <UnitPrice currency="USD">[UnitPrice – This value will be ignored]</UnitPrice>
    <QuantityAvailable>[Qty Available]</QuantityAvailable>
    <IsAvailable>["true", "false", or "unknown"]</IsAvailable>
    <QuantityBackordered>[Backordered Quantity]</QuantityBackordered>
    <ShipDate>[Ship Date]</ShipDate>
    <BackorderShipDate>[Backordered Ship Date]</BackorderShipDate>
    <SubstitutePartNumber>[Substitute Part Number]</SubstitutePartNumber>
    <SubstituteUnitOfMeasure quantity="[Product Size]" unit="[UOM]"/>
    <OtherInfo name="MSDS" type="url">[URL]</OtherInfo>
    <OtherInfo name="Description" type="text">[Description]</OtherInfo>
```

```

<OtherInfo name="Legal Disclaimer" type="text">[Legal Mumbo Jumbo]</OtherInfo>

</ProductResponse>

</PriceAvailabilityResponse>

```

Message Elements - Availability Call (Response)

Element/Attribute	Status	Description
<Header>		
MessageId	Mandatory	The supplier response MessageId.
RelatedMessageId	Mandatory	The value for this element must be the MessageId of the originating request sent by JAGGAER
Timestamp	Mandatory	The timestamp per the ISO-8601 format outlined in the Date/Timestamp Format section within this document.
<ProductResponse>		
ItemNumber	Mandatory	Value is "1".
Status	Mandatory	Expected attribute values are either "OK" for a successful product match, or "Error" for an unsuccessful match.
SupplierPartNumber	Mandatory	The supplier part number for the product. This value must match the part number of the original request sent by JAGGAER.
UnitPrice	Optional	Pricing will be ignored in favor of static, hosted pricing. The client's current, contracted price for the related part number. The value must be US-formatted, even in cases where the currency is not "USD". The currency attribute must contain a client-supported currency. Please see the Supported Currencies section within this document.
QuantityAvailable	Conditional Requirement	The quantity of product that is currently available. This element is required when the <IsAvailable> element is not present. This element takes precedence if both <QuantityAvailable> and <IsAvailable> are present.
IsAvailable	Conditional Requirement	An indicator of availability with accepted values of "true", "false", or "unknown". This element is required when the <QuantityAvailable> element is not present. The <QuantityAvailable> takes precedence when both <QuantityAvailable> and <IsAvailable> are present. JAGGAER recommends this element be used only in cases where the supplier is unable to provide a numeric value for quantity available.
QuantityBackordered	Optional	The quantity of product on backorder or is not available.
ShipDate	Optional	The estimated date the product will be available to ship (YYYY-MM-DD).
BackorderShipDate	Optional	The estimated date back-ordered product will be available to

Element/Attribute	Status	Description
		ship (YY-MM-DD)
SubstitutePartNumber	Optional	If the requested part number is back-ordered, this value represents a suggested alternative product.
SubstituteUnitOfMeasure	Optional	The UOM for the suggested, substitute part number.
Otherinfo name type	Optional	<p>Zero or more of these elements can be provided at supplier discretion, and represent additional information about the requested product.</p> <p>If OtherInfo elements will be sent, the name attribute will serve as the display name to the user/shopper and must be present. This type attribute is optional with "text" and "url" as accepted values. When absent, a "text" type is assumed.</p> <p>OtherInfo element values can be enclosed with CDATA tags if desired. The name attribute and element values do not support HTML tags.</p>

Purchase Order Delivery

A purchase order is a formal document request to the supplier authorizing the purchase of goods and services. In cases where JAGGAER will deliver the purchase order on behalf of the client, the supplier has four options for purchase order delivery:

- **cXML**: JAGGAER's preferred method of purchase order delivery.
- **Fax**: JAGGAER's default method of purchase order delivery when cXML is not supported by the supplier.
- **Email**: JAGGAER can deliver purchase orders via email in cases where the supplier does not support delivery via cXML or fax. There are 3 available email styles:
 - **HTML** (in the body of the email message)
 - **HTML** (as an attachment)
 - Plain Text
- **Supplier Network Portal**: JAGGAER can deliver purchase orders to the Supplier Network Portal for supplier pick-up and processing if preferred.

If the supplier is unable to support any of the above listed methods of delivery, please contact JAGGAER to discuss available options.

DTD: CXML.dtd

Version: 1.2.011

Guides: [Purchase Order \(OrderRequest\) Troubleshooting Guide](#), [cXML User's Guide](#)

JAGGAER conforms to the cXML 1.2.011 DTD standard for purchase order integrations. JAGGAER should be able to integrate with any supplier using this version; however, if the supplier is using another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, see the [Supported cXML Versions](#) section.

cXML Purchase Order Requirements

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the purchase order integration:

- A synchronous cXML response (200 series range) to an OrderRequest message is required. The absence of a cXML response to the OrderRequest will be identified as a delivery failure in the JAGGAER application.
- The supplier must be able to accept dynamic address lines. The supplier process for handling ShipTo addresses in general will be thoroughly tested before the cXML PO integration can be taken to production.
- Unrecognized ShipTo or BillTo address codes must not generate a failure response. The supplier should have a process in place to accept the order and move the request to an exception queue for supplier intervention and resolution.

- Electronic order processing must not be reliant upon the passing of a client account number. Client accounts can be associated on the supplier side to the "From Identity", BillTo address code, or other element in the cXML OrderRequest standard. JAGGAER recommends that where possible, the supplier create a master account number for the client to associate with the "From Identity" as passed in the OrderRequest message.
- All other requirements as listed in the Message Elements Table.

If any of the above requirements cannot be met, JAGGAER will request the supplier receive purchase orders via fax, email, or via the Supplier Portal. In cases where the electronic order is not received directly into the supplier ERP or requires the manual entry of order information to the supplier system, JAGGAER will discuss the value proposition of electronic integration with the supplier before proceeding.

ShipTo Addresses

A ShipTo address will be passed to the supplier on every purchase order delivered by JAGGAER. During implementation, the process for handling ShipTo addresses must be successfully tested before integration promotion to production can be considered. The following sections outline the formatting, variances, and considerations for ShipTo addresses.

Dynamic vs. Static Address Lines

Dynamic address lines refer to lines that are populated via free form entry by the user; while static lines refer to elements that are hardcoded, and therefore unchangeable by the user. In other words, dynamic address line values will vary each time the same base address is sent while hardcoded lines will remain static. In theory any address element can be either dynamic or static; however, there are some prevailing practices among JAGGAER's clients. Common dynamic elements include the attention line and room, floor, or suite number. Common static lines tend to include building and physical street number.

Dynamic lines are deployed by clients in an effort to reduce the overall number of ShipTo addresses that must be maintained and updated, and to better control the locations users can ship to. JAGGAER clients may choose to control addresses at the department, building, or street number level; meaning that only one ShipTo address is required per department, building, or street number. The addition of dynamic lines to handle the attention name and room/floor number elements introduces flexibility while still providing adequate control.

Due to overwhelming prevalence of use, suppliers must be able to accept dynamic address lines for successful cXML purchase order delivery. For those suppliers that can systematically accept dynamic address lines, the key factor is the understanding that dynamic lines will be passed exactly as the user enters it. This provides some element of unpredictability that suppliers should plan for and consider when developing the purchase order integration.

Address Code Lookup

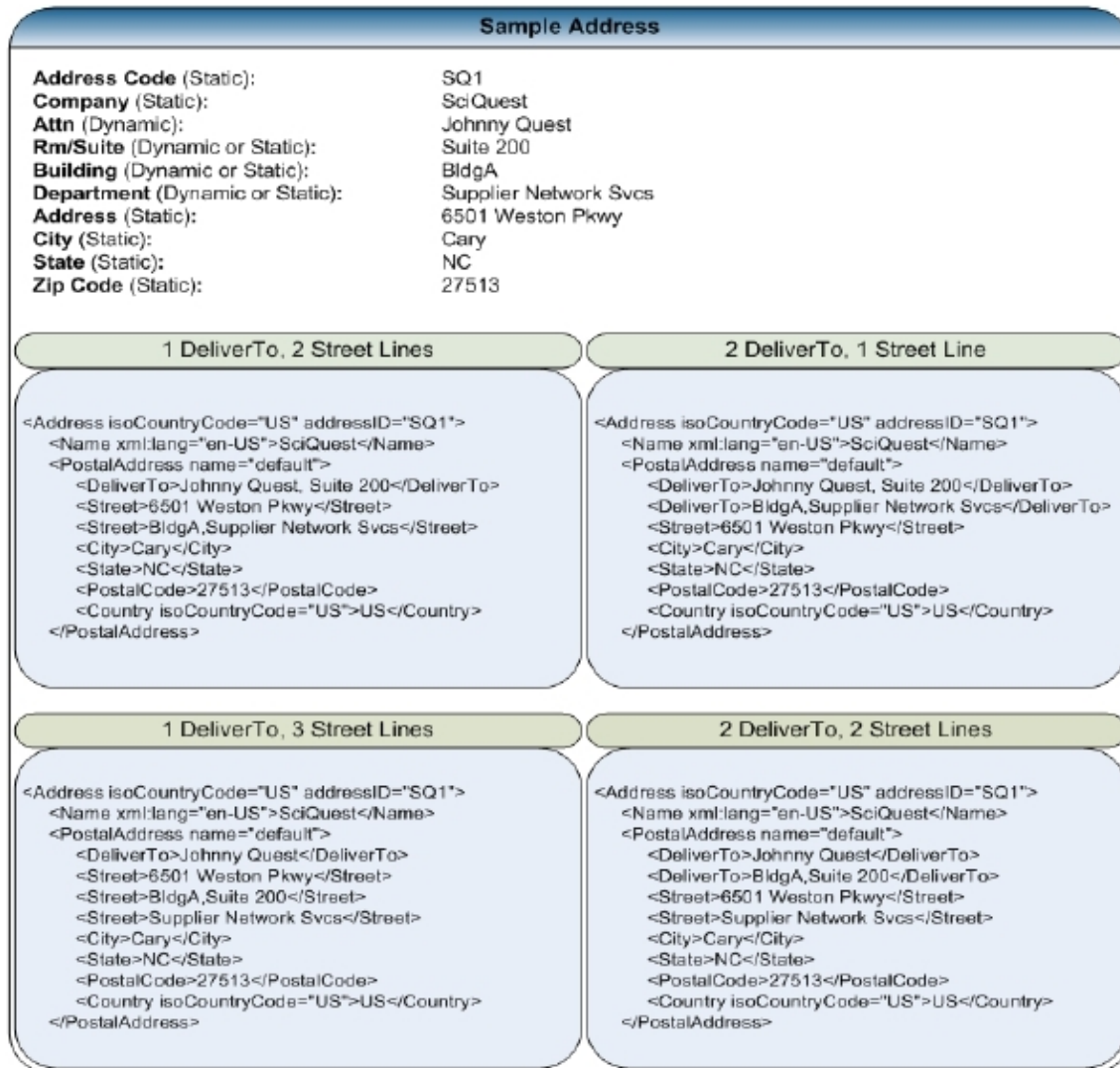
Address code lookup refers to the supplier practice of storing ShipTo/BillTo addresses and executing a system lookup based upon the address code received on the cXML purchase order document. In many cases, suppliers associate an account number for each ShipTo address code. This practice introduces challenges, particularly when new ShipTo addresses are added/deleted by the client and when considering the prevalence of dynamic address lines. Address elements that are dynamic should never be stored in the supplier system as these values change with each order. Only static fields should be stored as part of the address to avoid shipment errors.

JAGGAER recommends that suppliers parse the ShipTo address as received on the inbound cXML purchase order document. If the supplier systematically requires address code lookup, JAGGAER encourages suppliers to create a master account that can encompass any pre-existing client account numbers and can be associated to the "From Identity" as sent in the OrderRequest message. Additionally, suppliers must have a process in place to handle new, unrecognized ShipTo address codes on the the cXML purchase order document per integration requirements. If the supplier is unable to return a 200 series success response due to unrecognized address codes, JAGGAER will request the supplier receive orders via fax, email, or via the Supplier Network Portal.

JAGGAER will facilitate supplier receipt of a current ShipTo/BillTo addresses list at the time of implementation; however, address list maintenance and updates to the supplier are the sole responsibility of the client. JAGGAER does not provide updated address lists to suppliers.

Address Formatting

JAGGAER has the ability to send the supplier up to three DeliverTo and three Street lines in addition to the City, State, and Zip code fields. Client ShipTo addresses are mapped based upon available data and supplier capabilities. The supplier should advise the JAGGAER implementation Project Manager the number of DeliverTo and Street lines that can be systematically accepted. The following diagram illustrates a few of the formatting options:



Address Variances

Each JAGGAER client maintains ShipTo addresses in different formats and styles whether they are stored in the JAGGAER application or on the client ERP system. JAGGAER has the ability to standardize some address format differences through mapping; however, variances in how address information is provided to JAGGAER may limit mapping capabilities. The following diagram illustrates some common examples of address variances:

Sample Address Variances			
Company Name:	SciQuest		SciQuest
Address Code:	SQ1 (static)		SQ2 (static)
Contact Line 1:	Johnny Quest (dynamic)		STE 200/Front Desk (dynamic)
Contact Line 2:	Suite 200 (dynamic)		Supplier Network Svcs (static)
Contact Line 3:	Supplier Network Services (dynamic)		
Address Line 1:	6501 Weston Pkwy (static)		Building A (static)
Address Line 2:	BldgA (static)		
City:	Cary (static)		Cary (static)
State:	NC (static)		NC (static)
Zip Code:	27513 (static)		27513 (static)
1 DeliverTo, 2 Street Lines		1 DeliverTo, 2 Street Lines	
<pre> <Address isoCountryCode="US" addressID="SQ1"> <Name xml:lang="en-US">SciQuest</Name> <PostalAddress name="default"> <DeliverTo>Johnny Quest, Suite 200</DeliverTo> <Street>6501 Weston Pkwy</Street> <Street>BldgA, Supplier Network Services</Street> <City>Cary</City> <State>NC</State> <PostalCode>27513</PostalCode> <Country isoCountryCode="US">US</Country> </PostalAddress> </pre>		<pre> <Address isoCountryCode="US" addressID="SQ2"> <Name xml:lang="en-US">SciQuest</Name> <PostalAddress name="default"> <DeliverTo>STE 200/Front Desk, Supplier Network Svcs</DeliverTo> <Street>Building A</Street> <Street /> <City>Cary</City> <State>NC</State> <PostalCode>27513</PostalCode> <Country isoCountryCode="US">US</Country> </PostalAddress> </pre>	

Both of the above sample addresses are for the same client, with the same map applied. Maps are configured and set at the supplier level; and as a result, there are two very different outcomes when factoring in variances such as dynamic address lines, the absence of a physical street number (common with educational institutions where street numbers do not apply), and the way each address is stored or provided to JAGGAER.

While JAGGAER's clients are encouraged to remain consistent with address formats, it is impossible for JAGGAER to guarantee that the physical street address will always be provided in the first <Street> tag, that the attention line will always start with or contain the user's name, that every element will have an associated value, or that supplier address line character limitations will never be exceeded. JAGGAER will coordinate with the supplier and code for these inconsistencies wherever possible; however, suppliers should be prepared to handle some address variances.

Because of variances, JAGGAER strongly recommends that suppliers parse the ShipTo address as received on the inbound cXML purchase order document versus storing the addresses and executing a lookup based on address code. If the supplier is systematically required to execute address lookup, only static fields should be stored as part of the address to avoid shipment errors.

Payment Method

There are two primary ways in which clients approach payment with their supplier base; Purchase Order (Direct Billing) and Credit Card. In some cases, clients will choose a single payment method for all orders with suppliers; while others may employ both options. The following sections outline each payment method in further detail.

Purchase Order (Direct Billing)

The most common method of payment is direct billing, which is the standard practice of sending a bill or invoice to the client against the purchase order that was received. Client invoicing can be done via a traditional paper process directly between the client and supplier (outside of the JAGGAER application), or executed electronically via cXML invoicing in cases where the client has licensed an Accounts Payable product. The Supplier Portal offers suppliers additional options for invoicing in cases where cXML is not an option.

The purchase order document will contain a BillTo address for supplier billing; however, the supplier may have alternate methods or processes for client billing. During integration implementation, it is important that all parties have an understanding of how purchase orders will be billed and where the bill or invoice will be sent. The supplier setup for handling this process is typically done in one of the following three ways:

- **Internal Lookup:** Similar to ShipTo address code lookup, the supplier reads the BillTo address code from the inbound purchase order document and executes an internal system lookup for the billing address. In some cases, suppliers may internally tie a billing address to each client ShipTo address and bill based upon the ShipTo address code received. Either of these lookup methods are most commonly used in cases where the client has a decentralized purchasing authority (multiple billing addresses).
- **Address on PO:** The supplier reads the BillTo address exactly as sent on the inbound purchase order document and bills accordingly. This is the most efficient process, particularly in cases where the client has a decentralized purchasing authority (multiple billing addresses).
- **Address on File:** The supplier ignores the BillTo address on the inbound purchase order document in favor of a master billing address on file against the client account. This is most common in cases where the client has a centralized purchasing authority (single billing address).

During implementation, the billing process the supplier has in place will be tested thoroughly to ensure that the process is handled effectively and efficiently with the electronic purchase order integration.

BillTo Addresses

JAGGAER's clients utilize both centralized and decentralized purchasing authorities; meaning that clients can have either a single billing address for all users, or multiple billing addresses.

During implementation, all parties will discuss client billing/payment processes and how it can be efficiently handled with the electronic purchase order integration.

For those clients using a decentralized purchasing authority, JAGGAER will facilitate supplier receipt of a current BillTo addresses list at the time of implementation; however, address list maintenance and updates to the supplier are the sole responsibility of the client. JAGGAER does not provide updated address lists to suppliers.

Credit Card

Payment via credit card is often referred to as PCard, Ghost Card, or single use credit card. In any of these cases, the credit card number, expiration date, and associated cardholder name is passed by default to the supplier on the purchase order document for payment processing. While JAGGAER will provide the card information to the supplier, any credit card processing or payment dispute issues must be handled directly between the client and the supplier. The following sections outline card type definitions and additional considerations in further detail.

Card Type Definitions

PCard

A PCard (Purchasing Card) is simply another term for a company credit card used to make business related purchases. Each PCard has a unique number and cardholder name, and the user/shopper is in possession of the card itself.

Ghost Card

A Ghost card is a company credit card used to make business related purchases; however, the user/shopper is unaware of the card number and is not in possession of the card itself. The JAGGAER application stores the card information and passes it to the supplier on the outbound purchase order document for supplier payment processing. Typically, the same credit card number is used for all purchases with a single supplier. In these cases, the supplier may elect to forgo the receipt of the card number on the purchase order document in favor of storing the card number within the client account.

Single Use Card

Single use cards are credit cards that are issued by a bank authority and are valid for a single purchase with a specific, pre-authorized amount. GE vPayment and JPMorgan Chase are two examples of bank authorities offering single use card services to mid-size client organizations. Given that the card is pre-authorized for a specific amount, a process to handle additional order charges such as shipping and tax, as well as any supplier practices that may include card pre-authorization must be addressed during implementation.

Card Reporting

Clients may inquire about supplier credit card reporting practices before determining the payment method that will be used. The following table provides an overview of the card

reporting levels for supplier review and consideration:

Credit Card Reporting Level Definitions			
	Level 1	Level 2	Level 3
Merchant Name	X	X	X
Transaction Amount	X	X	X
Data	X	X	X
Tax Amount		X	X
Customer Code		X	X
Merchant Postal Code		X	X
Tax Identification		X	X
Merchant Minority Code		X	X
Merchant State Code		X	X
Item Product Code			X
Item Description			X
Item Quantity			X
Item Unit of Measure			X
Item Extended Amount			X
Item Net/Gross Indicator			X
Item Tax Amount			X
Item Tax Rate			X
Item Tax Identifier			X
Item Discount Indicator			X
Ship From Postal Code			X
Freight Amount			X
Duty Amount			X
Destination Postal Code			X
Destination Country Code			X
Alternate Tax Amount			X

Credit Card Considerations

The following is a list of additional considerations when dealing with credit card purchases:

- The CV2 code is not typically passed to the supplier on outbound purchase order documents. If the supplier requires the CV2 code in order to successfully process payment, further discussions between the client and supplier may be required. Clients typically do not include the CV2 code for security reasons and in most cases is not required to successfully do business with credit cards.
- If purchase order delivery will be via fax or email, extra caution should be exercised when credit cards are in use for security reasons. An alternate payment method may be required for non-catalog orders that will be routed to the supplier via a method other than cXML.
- During implementation, the client and supplier should discuss a process for handling any credit card authorization failures. Any credit card processing or payment dispute issues must be handled directly between the client and the supplier.
- Many clients require that the supplier refrain from pre-authorizing the credit card as it can cause authorization failures when the final charges are processed. This is particularly relevant for clients using single use credit cards.
- During implementation, the client and supplier should discuss any additional fees that may be charged to the credit card. Additional fees such as special handling, standard shipping, expedited shipping, and tax charges should be discussed in full before processing credit card orders in a production environment.

Non-Catalog Order Delivery

A non-catalog order is an "ad hoc" order that is manually entered, and is typically created when the shopper is unable to locate a product in the supplier catalog. Other uses for non-

catalog ordering include special supplier quotes, bulk orders, and services that cannot be included in standard catalog content.

Product details such as part number, description, unit of measure, and pricing are captured via free form entry; and as a result, may contain some inaccuracies that make electronic processing challenging for suppliers.

The JAGGAER application has functionality to identify non-catalog items and distribute them via a different, separate channel than catalog items if necessary. If the supplier will fail order delivery via cXML due to an unrecognized part number or other invalid field data (any response code other than in the 200 series), non-catalog orders will not be eligible for delivery via cXML. Supplier options for non-catalog order delivery are via fax, email, and delivery to the Supplier Network Portal.

Order Failure Management

JAGGAER monitors purchase order delivery to suppliers and will address failures directly with the client and/or supplier as applicable. With each order failure a Support Incident is generated and managed by the JAGGAER Customer Support Team based upon the following guidelines:

- JAGGAER manages order failures that occur as a result of a JAGGAER-sanctioned distribution. Orders delivered by the client outside the JAGGAER system or that are delivered to an address/phone number not sanctioned by JAGGAER are managed by the client.
- The JAGGAER Support Team will initiate communication within 2 hours of the failure for those orders with JAGGAER-sanctioned distribution.
- If failure resolution cannot be achieved by 4PM Eastern the same day, JAGGAER will alert the client of the potential delay in order fulfillment.
- If the JAGGAER Support Team is unable to reach the supplier to resolve the failure within 24 hours, JAGGAER will escalate the issue internally as well as to any other known supplier contacts.

Alternatively, suppliers have the option to self-manage order failures through the Supplier Portal if preferred. The supplier should notify the JAGGAER Project Manager of this preference during implementation such that the portal can be configured accordingly.

Contact Roles

Clients may elect to define the various purchasing roles in use by the organization on outbound purchase order documents. Contact roles are optional and customizable by the client; however, the most common examples are "User" and "Buyer". Roles are used to differentiate to the supplier the organization contact to be used for order inquiry, issue resolution, and email order confirmation delivery. During implementation, the various contact role definitions will be communicated to the supplier with instructions for their use.

For example, the “User” may be the original requestor and eventual receiver of the order; however, the client organization would like for suppliers to contact the “Buyer” should there be any order issues. If multiple contact roles are in use and passed to the supplier on the OrderRequest message, the supplier will be asked to parse each data set and act accordingly. Please see the sample OrderRequest message later in this chapter for a look at how this data is passed.

If the supplier is able to consume only one contact role, or will only look to the ShipTo section of the document for contact information; JAGGAER may be able to accommodate via data mapping. Prior to production deployment, all processes surrounding contact roles should be fully tested and resolved.

Email Confirmations

JAGGAER client organizations typically expect that suppliers will provide individual users with an email order confirmation for every delivered order. In some cases, suppliers may also provide an additional shipping confirmation email notification. To facilitate either or both of these notifications, JAGGAER will pass a dynamic email address to the supplier on the OrderRequest message when available.

Email order confirmation notices are sent by the supplier directly to the email address provided and do not pass through the JAGGAER system. During implementation, the location of the email address within the cXML OrderRequest message will be discussed (ShipTo section is most common), and the supplier should be prepared to test notification delivery in full.

The client may also request the cXML ConfirmationRequest and ShipNoticeRequest integrations with suppliers who support this functionality. The deployment of either of these optional integrations is not intended to replace the email confirmations users have grown accustomed to receiving.

OrderRequest Message

The supplier will need to provide JAGGAER with both the test and production URL for message post (HTTP over SSL, port 443) as well as coordinate message authentication credentials. If the URL or credential values change post integration go-live, the supplier must notify the JAGGAER Support Team (support@JAGGAER.com) for update.

Successful integration testing per the provided PO Test Plan is required before promotion to production can be considered.

Sample Message (Order Request)

The following is a sample OrderRequest message. For details about message elements and attributes, please see the Message Elements table.

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

```

<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/cXML.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"
version="1.2.011">
<Header>
<From>
<Credential domain="[From Domain]">
<Identity>[From Identity]</Identity>
</Credential>
</From>
<To>
<Credential domain="[To Domain]">
<Identity>[To Identity]</Identity>
</Credential>
</To>
<Sender>
<Credential domain="[Sender Domain]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request deploymentMode="[test or production]">
<OrderRequest>
<OrderRequestHeader orderDate="YYYY-MM-DD'T'HH:mm:ssZ" orderID="[PO Number]"
type="new" orderType="regular">
<Total>
<Money currency="USD">[PO Total]</Money>
</Total>
<ShipTo>
<Address isoCountryCode="[Country Code]" addressID="[Ship To Code]">
<Name xml:lang="en-US">[Organization Name]</Name>
<PostalAddress name="default">
<DeliverTo>[Recipient Data]</DeliverTo>
<DeliverTo>[Recipient Data]</DeliverTo>
<DeliverTo>[Recipient Data]</DeliverTo>
<Street>[Street Address]</Street>
<Street>[Street Address]</Street>

```

```

<Street>[Street Address]</Street>
<City>[City]</City>
<State>[State Abbreviation]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="[Code]">[Country]</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="[Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
<Fax name="default">
<TelephoneNumber>
<CountryCode isoCountryCode="[Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Fax>
</Address>
</ShipTo>
<BillTo>
<Address isoCountryCode="[Country Code]" addressID="[Bill To Code]">
<Name xml:lang="en-US">[Organization Name]</Name>
<PostalAddress name="default">
<DeliverTo>[AP Recipient]</DeliverTo>
<Street>[Street Address]</Street>
<City>[City]</City>
<State>[State Abbreviation]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="[Code]">[Country]</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>

```

```

<CountryCode isoCountryCode="[Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
<Fax name="default">
<TelephoneNumber>
<CountryCode isoCountryCode="[Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Fax>
</Address>
</BillTo>
<Shipping>
<Money currency="USD">[Total Shipping]</Money>
<Description xml:lang="en">[Shipping Description]</Description>
</Shipping>
<Tax>
<Money currency="USD">[Total Tax]</Money>
<Description xml:lang="en">[Tax Description]</Description>
</Tax>
<Payment>
<PCard number="[Number]" expiration="YYYY-MM-DDTHH:MM:SS"/>
</Payment>
<Contact role="[Role Title]">
<Name xml:lang="en-US">[Name]</Name>
<PostalAddress name="default">
<Street>[Street Address]</Street>
<City>[City]</City>
<State>[State Abbreviation]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="[Code]">[Country]</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>

```

```

<CountryCode isoCountryCode="[Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Comments>[Header Level Comment]</Comments>
<Extrinsic name="[Name]">[Value for Header Level Extrinsic]</Extrinsic>
</OrderRequestHeader>
<ItemOut quantity="[Qty]" lineNumber="[Line Number]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en-US">[Product Description]</Description>
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<Classification domain ="UNSPSC">[UNSPSC Code]</Classification>
<ManufacturerPartID>[Manufacturer Part Number]</ManufacturerPartID>
<ManufacturerName>[Manufacturer Name]</ManufacturerName>
<Extrinsic name="[Name]">[Misc Customer Code]</Extrinsic>
</ItemDetail>
<Comments>[Line Level Comment]</Comments>
</ItemOut>
<ItemOut quantity="[Qty]" lineNumber="[Line Number]">
<ItemID>
<SupplierPartID>[Part ID]</SupplierPartID>
<SupplierPartAuxiliaryID>[Auxiliary Part ID]</SupplierPartAuxiliaryID>
</ItemID>
<ItemDetail>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<Description xml:lang="en-US">[Product Description]</Description>

```

```

<UnitOfMeasure>[UOM]</UnitOfMeasure>
<Classification domain ="UNSPSC">[UNSPSC Code]</Classification>
<ManufacturerPartID>[Manufacturer Part Number]</ManufacturerPartID>
<ManufacturerName>[Manufacturer Name]</ManufacturerName>
<Extrinsic name="[Name]">[Misc Customer Code]</Extrinsic>
</ItemDetail>
<Comments>[Line Level Comment]</Comments>
</ItemOut>
</OrderRequest>
</Request>
</cXML>

```

Message Elements - Order Request

Element/Attribute	Status	Description
<cXML>		
payloadID	Always Passed	This represents the JAGGAER OrderRequest message identifier. This value is always returned by the supplier to JAGGAER in the ConfirmationRequest-ShipNoticeRequest and Invoice integrations.
<Header>		
From domain	Always Passed	Default=NetworkID; DUNS accepted
From Identity	Always Passed	Typically the Institution Abbreviated Name or DUNS#
To domain	Always Passed	Default=NetworkID; DUNS accepted
To Identity	Always Passed	NetworkID or DUNS (supplier preference)
Sender domain	Always Passed	Default=NetworkID
Sender Identity	Always Passed	Default-JAGGAER
SharedSecret	Always Passed	Specified by the supplier
UserAgent	Always Passed	Default=JAGGAER
<Request>		
deploymentMode	Always passed	Either "test" or "production" will be passed dependent upon the client environment the order was generated in.
<OrderRequestHeader>		
orderDate	Always Passed	<p>This attribute value represents the date of purchase order creation and is presented in the ISO-8601 standard format: YYYY-MM-DDT'HH:mm:ssZ</p> <p>This value is returned by the supplier to JAGGAER in the ConfirmationRequest-ShipNoticeRequest and</p>

Element/Attribute	Status	Description
		Invoice integrations.
orderID	Always Passed	This attribute value will always represent the client purchase order number and may contain both alpha and numeric characters. This value is returned by the supplier to JAGGAER in the ConfirmationRequest-ShipNoticeRequest and Invoice integrations.
type	Always Passed	This attribute value will always be "new". JAGGAER does not currently support the "update" or "delete" types.
orderType	Always Passed	This attribute value will always be "regular". JAGGAER does not currently support the "release" or "blanket" order types.
releaseRequired effectiveDate expirationDate agreementID agreementPayloadID parentAgreementID parentAgreementPayloadID	Not Supported via cXML	<p>These element attributes apply to blanket orders only and are not currently supported by JAGGAER. Blanket orders are not routed via cXML to the supplier; they are typically provided via an alternative delivery method, and in some cases are sent directly by the client where applicable.</p> <p>Any applicable contract numbers may be passed as extrinsics where applicable, but is atypical. This passing of contract numbers requires the client to have licensed the ContractManager module.</p>
orderVersion	Not Supported	This element attribute is currently not supported by JAGGAER. Because electronic order delivery results in timely shipments, change orders are not routed by JAGGAER to the supplier.
isInternalVersion	Not Supported	This element attribute is not currently supported by JAGGAER. Because electronic order delivery results in timely shipments, the client's purchase order revisions are never routed to the supplier.
shipComplete	Not Supported	This element attribute is not currently supported by JAGGAER, as JAGGAER clients typically accept partial shipments.
<Total>		
Money currency	Always Passed	This value is the sum of [line item extended prices + shipping/handling + tax]. Please see the Numerical Formats & Calculations section for more information.
<ShipTo>		
Address isoCountryCode addressID	Always Passed	The default isoCountryCode is "US", but will vary for international buying organizations. The addressID attribute value represents the client organization address identifier for the provided ShipTo address, and can contain both alpha and numeric characters. Also

Element/Attribute	Status	Description
		referred to as the ShipTo address code. Code length is typically 2 to 10 chars in length, but will vary by client.
Name xml:lang	Passed when available	<p>The Name element value is most often the buying Organization Name, but can be any client-specified value. If a value is not available, a null value will be passed.</p> <p><Name xml:lang="en-US" /></p> <p>The default language attribute value is "en-US".</p>
PostalAddress name DeliverTo Street City State PostalCode Country isoCountryCode	Always Passed	<p>The PostalAddress element values are passed per client/supplier capabilities. Please see the ShipTo Address section listed earlier in this chapter for further requirements and additional information. The PostalAddress name attribute value is typically "default", but may vary by client.</p> <p>The Country element value is typically "UnitedStates" or "US" by default, but may vary for international buying organizations.</p> <p>The isoCountryCode attribute value is "US" by default, but may vary for international buying organizations.</p> <p><CountryCode isoCountryCode="US">UnitedStates</CountryCode></p>
Email name	Passed when available	<p>Typically the user/shopper email address, but may represent the buyer or other contact address per client/supplier requirements. The name attribute default value is "default". Please see the Contact Roles and Email Confirmations sections for more information.</p> <p>If the user/shopper has not populated their email address within their JAGGAER profile, a null value will be passed.</p> <p><Email name="" /></p>
Phone name CountryCode isoCountryCode AreaOrCityCode Number	Passed when available	<p>Typically the user/shopper phone number, but may represent the buyer or other contact phone per client/supplier requirements. The name attribute value is typically "work" but may vary by buying organization.</p> <p>The Country Code element default value is "1", but will vary for international buying organizations. The isoCountryCode attribute default value is "US", but will vary for international buying organizations.</p> <p><CountryCode isoCountryCode="US">1</CountryCode></p> <p>If a phone extension is applicable (rare), it will be concatenated to the phone number (e.g. "5551212</p>

Element/Attribute	Status	Description
		<p>ext.111").</p> <p>If the user/shopper has not populated the phone number within their JAGGAER profile, a null value may be passed.</p> <pre> <TelephoneNumber> <Country isoCountryCode="" /> <AreaOrCityCode /> <Number /> </TelephoneNumber> </pre>
Fax name CountryCode isoCountryCode AreaOrCityCode Number	Rarely passed	<p>Fax elements are rarely passed. These element tags will be present but will contain null values if values are unavailable.</p> <pre> <Fax name="" /> <TelephoneNumber> <Country isoCountryCode="" /> <AreaOrCityCode /> <Number /> </TelephoneNumber> </pre>
<BillTo>		
Address isoCountryCode addressID	Always Passed	<p>The default isoCountryCode is "US", but will vary for international buying organizations. The addressID attribute value represents the client organization address identifier for the provided BillTo address, and can contain both alpha and numeric characters. Also referred to as the BillTo address code. Code length is typically 2 to 10 chars in length, but will vary by client.</p>
Name xml:lang	Passed when available	<p>The Name element value is most often the buying Organization Name, but can be any client-specified value. If a value is not available, a null value will be passed.</p> <pre> <Name xml:lang="en-US" /> </pre> <p>The default language attribute value is "en-US".</p>
PostalAddress name DeliverTo Street City State PostalCode Country isoCountryCode	Always Passed	<p>The PostalAddress element values are passed per client/supplier capabilities. Please see the BillTo Address section listed earlier in this chapter for further requirements and additional information.</p> <p>The PostalAddress name attribute value is typically "default", but may vary by client.</p> <p>The Country element value is typically "UnitedStates" or "US" by default, but may vary for international buying organizations.</p> <p>The isoCountryCode attribute value is "US" by default, but may vary for international buying organizations.</p> <pre> <CountryCode </pre>

Element/Attribute	Status	Description
		isoCountryCode="US">UnitedStates</CountryCode>
Email name	Passed when available	<p>An email address within the BillTo section of the OrderRequest document is rare. If a value is available it will be passed; otherwise, it will be passed with a null value.</p> <p><Email name="" /></p>
Phone name CountryCode isoCountryCode AreaOrCityCode Number	Passed when available	<p>Typically the central purchasing phone number, but may represent a hotline or administrator number per client requirements. The name attribute value is typically "work" but may vary by client.</p> <p>The Country Code element default value is "1", but will vary for international buying organizations.</p> <p><CountryCode isoCountryCode="US">1</CountryCode></p> <p>If a phone extension is applicable (rare), it will be concatenated to the phone number (e.g. "5551212 ext.111").</p> <p>In cases where the client does not provide a BillTo address phone number, the elements will be passed with a null value.</p> <p><TelephoneNumber> <Country isoCountryCode="" /> <AreaOrCityCode /> <Number /> </TelephoneNumber></p>
Fax name CountryCode isoCountryCode AreaOrCityCode Number	Rarely passed	<p>Fax elements are rarely passed. These element tags will be present but will contain null values if values are unavailable.</p> <p><Fax name="" /> <TelephoneNumber> <Country isoCountryCode="" /> <AreaOrCityCode /> <Number /> </TelephoneNumber></p>
<Shipping>		
Money currency	Rarely passed	<p>The passing of shipping charges is extremely rare as clients most often expect suppliers to provide the actual, final amounts on the invoice.</p> <p>This element is omitted in most cases; however, will be passed with a null value if the shipping description element is utilized.</p> <p><Money currency="" /></p>
Description xml:lang	Rarely passed	The description element value can be used to convert

Element/Attribute	Status	Description
		<p>the shopper's desired shipping method for the order. Two examples are "Best Carrier – Best Way" or "NextDay", but can be any value. Most JAGGAER clients expect that the supplier will ship ordered products per the contract or via the best, most cost-effective way – in these cases the element is omitted.</p> <p>If the client is expecting the supplier to parse and process this element, possible values and supplier capabilities should be discussed in detail during implementation.</p>
<Tax>		
Money currency	Rarely passed	<p>The passing of tax charges is extremely rare as clients most often expect suppliers to provide the actual, final amounts on the invoice.</p> <p>This element is omitted in most cases.</p>
Description xml:lang	Rarely passed	<p>The description element value will typically be passed as null if tax totals are included; however, is most often omitted.</p>
<Payment>		
PCard number expiration	Passed for Credit Card orders only	<p>This element and its associated attributes will only be present for credit card orders; otherwise, they will be omitted. The card number is passed without delimitation and the expiration date is presented in the ISO-8601 standard format of: YYYY-MM-DDTHH:mm:ss</p> <p>Note that the CV2 code is not passed by default. The supplier must discuss this with the client should this attribute be required.</p>
<PaymentTerm>		
payInNumberOfDays Discount DiscountPercent percent	Never Passed	<p>These elements are not currently supported by JAGGAER for OrderRequest and are never passed on the outbound document. JAGGAER clients typically expect these values to be present on the associated invoice (per contract terms) instead.</p>
<Contact>		
Role addressID Name Postal Address Email Phone	Sometimes Passed	<p>One or more contact roles and associated elements may be present. Please see the Contact Roles section for more information.</p> <p>If client is not utilizing contact roles, these elements will be omitted.</p> <p>The addressID attribute will never be passed within the contact section.</p>
Comments	Sometimes	Header level comments (100 char limit) are

Element/Attribute	Status	Description
	passed	<p>sometimes passed by the client and apply to the entire order. If comments are not present, the element will either be omitted or passed with a null value.</p> <p><Comments:/></p> <p>During integration implementation, the supplier must advise both JAGGAER and the Client how hheader comments via cXML will be handled, if at all. Buying organizations understand that comments are not ideal for electronic order delivery and are rarely supplier-supported; and as a result, typically do not pass them.</p>
Comments - Attachment URL	Never passed	JAGGAER does not currently pass attachments via cXML.
SupplierOrderInfo orderID	Not Supported	The SupplierOrderInfo element and attribute are not currently supported by JAGGAER as suppliers must not generate an order until the OrderRequest document is received. PunchOut suppliers must not create an order as a result of a punch-out shopping session.
Extrinsic name	Sometimes passed	<p>Extrinsics at the header level indicate additional information that applies to the entire order. None or an unlimited number of extrinsics are possible.</p> <p>There are various uses for extrinsics, but is most often used for account codes and other internal references.</p> <p>In the majority of cases, the supplier can ignore these element values unless otherwise specified by the client.</p> <p>In cases where the buying organization is not fully tax exempt, an extrinsic can be used as a flag to convey order exemption status to the supplier. In this use case, the extrinsic attribute name and element value is typically configurable. Most clients are either fully exempt or fully non-exempt; therefore, a tax flag is rarely passed.</p> <p><Extrinsic name="Taxable">"Exempt/NonExempt", or "Yes/No"</Extrinsic></p>
<ItemOut>		
quantity	Always Passed	Represents the ordered quantity of the associated item, and is typically a whole number. In some rare cases, decimal quantities may be passed where applicable.
lineNumber	Always Passed	Represents the item position within the OrderRequest message. If the supplier will provide electronic ConfirmationRequests, ShipNoticeRequests or electronic Invoices, this value must be maintained by the supplier for return to JAGGAER.

Element/Attribute	Status	Description
requestedDeliveryDate	Rarely Passed	<p>This element value can be used to convey the shopper's desired delivery date for the item. Most JAGGAER clients expect that the supplier will ship ordered products per the contract or via the best, most cost effective way – in these cases the element is omitted.</p> <p>If the client is expecting the supplier to parse and process this element, supplier capabilities should be discussed in detail during implementation.</p>
requisitionID	Never Passed	This element attribute is not currently in use by JAGGAER clients.
agreementItemNumber	Never Passed	This element attribute is not currently in use by JAGGAER clients.
requisitionID	Never Passed	This element attribute is not currently supported by JAGGAER.
<ItemID>		
SupplierPartID	Always Passed	<p>The part number value the supplier has provided for the associated item in the catalog will be passed. The syntax will be identical to the catalog value, including any delimiters such as dashes.</p> <p>Should the supplier require a non-delimited value on the OrderRequest message, JAGGAER can optionally remove delimiters at supplier request.</p>
SupplierPartAuxiliaryID	Passed when Available	<p>The SupplierPartAuxiliaryID value provided by the supplier for the associated item as part of the catalog is always passed if present. If the supplier has not provided a value in the catalog, the element will be omitted from the OrderRequest message.</p> <p>Please see the section on SupplierPartAuxiliaryID for more information about field use.</p>
<p>Note: JAGGAER does not currently support service detail elements via cXML such as SpendDetail, TravelDetail, AirDetail, CarRentalDetail, HotelDetail, RailDetail, FeeDetail or LaborDetail.</p> <p><ItemDetail></p>		
<UnitPrice> Money currency	Always Passed	<p>The UnitPrice the supplier has provided for the associated item in the catalog will be passed. The default currency is "USD".</p> <p>See the Numerical Formats and Supported Currencies sections within the Data Standards chapter for more information.</p>
Description xml:lang	Always Passed	<p>The product description the supplier has provided for the associated item in the catalog will be passed, up to 256 characters.</p> <p>The default language attribute value is "en-US".</p>

Element/Attribute	Status	Description
UnitOfMeasure	Always Passed	The UOM value the supplier has provided for the associated item in the catalog will be passed. For more information, please see the Units of Measure (UOM) section.
Classification domain: UNSPSC	Passed when Available	<p>The UNSPSC code the supplier has provided for the associated item in the catalog will be passed. For more information, please see the Product Categorization section within the Data Standards chapter.</p> <p>If the supplier has not provided a value in the catalog, the element will be passed with a null value.</p> <p><Classification domain="UNSPSC"/></p>
ManufacturerPartID	Passed when Available	<p>The ManufacturerPartID value the supplier has provided for the associated item in the catalog is always passed if present. If the supplier has not provided a value in the catalog, the element will be passed with a null value.</p> <p><ManufacturerPartID/></p>
ManufacturerName	Passed when Available	<p>The ManufacturerName value the supplier has provided for the associated item in the catalog is always passed if present. If the supplier has not provided a value in the catalog, the element will be passed with a null value.</p> <p><ManufacturerName/></p>
URL	Not Supported	JAGGAER does not currently support the URL element via cXML as it is not typically provided by the supplier in the catalog.
<Distribution>		
Accounting name AccountingSegment id	Never Passed	Accounting code segments, if passed at all, are passed as extrinsics. These element tags are omitted.
Comments	Passed when Available	<p>Any line level comments the user has provided (100 character limit) will be passed and apply only to the associated line. If no line level comments are present, the element will either be omitted or passed with a null value.</p> <p><Comments/></p> <p>During integration implementation, the supplier must advise both JAGGAER and the client how line comments will be handled, if at all. Buying organizations understand that comments are not ideal for electronic order delivery and are not often supplier-supported, and as a result, typically do not pass them.</p>
Extrinsic name	Passed when available	Extrinsics at the line level indication additional information that applies only to the referenced item.

Element/Attribute	Status	Description
		<p>None or an unlimited number of extrinsics are possible.</p> <p>There are various uses for extrinsics, but is most often used for account code segments and other internal references. In the majority of cases, the supplier can ignore these element values unless otherwise specified by the client.</p> <p>In cases where the buying organization is not fully tax exempt, an extrinsic can be used as a flag to convert exemption status by line to the supplier. In this use case, the extrinsic attribute name and element value is typically configurable. Most clients are either fully exempt or fully non-exempt; therefore, a tax flag is rarely passed.</p> <p><Extrinsic name="Taxable">"Exempt/NonExempt", or "Yes/No"</Extrinsic></p>
HTTP Header	Always passed	When posting the OrderRequestmessage, the HTTP header contains a Content Type of "text/xml"

The supplier can specify the variables listed below. If a variable is not provided, the default value will be used.

Response to an Order Request

JAGGAER expects to receive a synchronous 200 series status code for all orders delivered via cXML per "[General Integration Requirements](#)" on page 11. If an unrecognized ShipTo code, BillTo code, or part number will generate a supplier failure response, an alternate method of order delivery may be required for non-catalog and/or catalog orders.

JAGGAER does not redirect messages or handle 300 series responses, and any response outside the 200 range will be considered an order failure. Please see the Order Failure Management section outlined earlier in this chapter for more information.

Click here for a [Sample Message \(OrderMessage Request Response\)](#)

Sample Message (Response)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/cXML.dtd">
<cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
  <Response>
    <Status code="200" text="Success"/>
  </Response>
</cXML>
```


Confirmation Request

The ConfirmationRequest integration is often referred to as cXML Order Confirmation, Order Acknowledgement, or PO Acknowledgement. The integration offers suppliers the opportunity to provide clients with electronic, system-to-system order acknowledgements that often include additional order information such as estimated ship dates. Traditionally, suppliers route email order confirmations from the supplier system directly to the order requestor. This integration opportunity is not intended to replace that process, but instead is intended to enhance the client procurement system with electronic order data that can be referenced at any time.

DTD: Fulfill.dtd

Version: 1.2.009

Guides: [Confirmation Request Troubleshooting Guide](#), [JAGGAER Response Codes](#), [cXML User's Guide](#)

JAGGAER conforms to the cXML 1.2.009 Fulfill.dtd standard for ConfirmationRequest integrations; however, JAGGAER does not currently support all of the request types, statuses, operations, or element attributes. The StatusUpdateRequest integration is also not currently supported by JAGGAER – please refer to the Message Elements table and sample message for more information about support. If the supplier is utilizing another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, please see the [Supported cXML Versions](#) section.

ConfirmationRequest Message

There are two standard ConfirmationRequest message operations: “new” and “update”. A request operation of “new” indicates that no previous confirmation has been sent for the order, while an operation of “update” implies intent to modify a previously sent confirmation message. JAGGAER will accept a supplier message with either operation; however, the JAGGAER application will interpret all ConfirmationRequests as “new”.

For more information about JAGGAER handling of ConfirmationRequest:

Confirmation Request Integration Requirements

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the **ConfirmationRequest** integration:

- The supplier DUNS or Network Identity must be the same value as used in other supplier integrations with JAGGAER.
- JAGGAER currently only supports a header status of “detail” and requires confirmation status and details at the line level.

- JAGGAER will return a synchronous response message to the supplier via the same request path.
- Confirmation Request testing traffic must be posted to URL: <https://usertest-messages.scquest.com/apps/Router/CXMLReceive>
- Confirmation Request production traffic must be posted to URL: <https://integrations.scquest.com/apps/Router/CXMLReceive>
- The supplier may be required to download and install the JAGGAER SSL digital certificates.
- All other requirements as listed in the Message Elements Table.

Sample Message (Line Detail)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>
      </Credential>
      <UserAgent>[User Agent]</UserAgent>
    </Sender>
```

```

</Header>

<Request>

<ConfirmationRequest>

<ConfirmationHeader type="detail" operation="new" noticeDate="YYYY-MM-DD" confirmID="
[OrderID in Supplier System]">

<!--Header level order total, shipping, tax, and contact roles are ignored by JAGGAER. These
elements can be omitted; however, if included must follow cXML standards. Begin ignore
section-->

<Total>

<Money currency="[Currency]">[Order Total]</Money>

</Total>

<Shipping>

<Money currency="[Currency]">[Shipping Total]</Money>

<Description xml:lang="en">[Shipping Description]</Description>

</Shipping>

<Tax>

<Money currency="[Currency]">[Tax Total]</Money>

<Description xml:lang="en">[Tax Description]</Description>

</Tax>

<Contact role="ShipFrom">

<Name xml:lang="en">[Supplier Name]</Name>

<PostalAddress>

<Street>[Street]</Street>

<City>[City]</City>

<State>[State]</State>

<PostalCode>[Zip Code]</PostalCode>

<Country isoCountryCode="[Country Code]">[Country]</Country>

</PostalAddress>

<Phone>

```

```

<TelephoneNumber>
<CountryCode isoCountryCode="[Country Code]">[Country Code]</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<!-------end ignore section----->
<Comments>[Header Level Comment]</Comments>
</ConfirmationHeader>
<OrderReference orderID="[Customer PO Number]" orderDate="YYYY-MM-DD">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
<ConfirmationItem lineNumber="[lineNumber from Original PO]" quantity="[PO Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<ConfirmationStatus type="[Status]" quantity="[Status Qty]" shipmentDate="YYYY-MM-DD">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="[Currency]">[UnitPrice]</Money>
</UnitPrice>
<Tax>
<Money currency="[Currency]">[Tax Charge]</Money>
<Description xml:lang="en">[Description of Tax Charges]</Description>
</Tax>
<Shipping>
<Money currency="[Currency]">[Shipping Charge]</Money>
<Description xml:lang="en">[Description of Shipping Charges]</Description>
</Shipping>
<Comments>[Line Level Comment]</Comments>

```

```

</ConfirmationStatus>
</ConfirmationItem>
<ConfirmationItem lineNumber="[lineNumber from Original PO]" quantity="[PO Qty]">
  <UnitOfMeasure>[UOM]</UnitOfMeasure>
  <ConfirmationStatus type="[Status]" quantity="[Confirm Qty]" shipmentDate="YYYY-MM-DD">
    <UnitOfMeasure>[UOM]</UnitOfMeasure>
    <UnitPrice>
      <Money currency="[Currency]">[UnitPrice]</Money>
    </UnitPrice>
    <Tax>
      <Money currency="[Currency]">[Tax Charge]</Money>
      <Description xml:lang="en">[Description of Tax Charges]</Description>
    </Tax>
    <Shipping>
      <Money currency="[Currency]">[Shipping Charge]</Money>
      <Description xml:lang="en">[Description of Shipping Charges]</Description>
    </Shipping>
    <Comments>[Line Level Comment]</Comments>
  </ConfirmationStatus>
</ConfirmationItem>
</ConfirmationRequest>
</Request>
</cXML>

```

Message Elements - Confirmation Request

Element/Attribute	Description and Support Detail
<Header>	
From domain	DUNS or NetworkID. This element is required, but is not used in message authentication for this integration
From Identity	Supplier DUNS or NetworkID. This element is required, but is not used in message authentication for this integration. Typically the same value as Sender identity.
To domain	DUNS or NetworkID. This element is required, but is not used in message authentication for this integration.
To Identity	Client DUNS or NetworkID. This element is required, but is not used in message authentication for this integration.
Sender domain	DUNS or NetworkID. This element is required but is not used in message authentication for this integration; however, should be the same value as used by the supplier in other integrations with JAGGAER.
Sender Identity	Supplier DUNS or NetworkID. This element is required and authenticated, and should be the same value as used by the supplier in other integrations with JAGGAER.
SharedSecret	Supplier specified SharedSecret or password. Used in message authentication.
UserAgent	Supplier's UserAgent, typically represented by the sending application and version. This element is required but is not used in message authentication.
<ConfirmationHeader>	
type	JAGGAER supports a header message type of "detail". If a message type of "accept", "reject", or "backordered" is sent at the header level, the message will be accepted but will be interpreted as if "detail" were passed. "allDetail", "requestToPay", "except", or any other header type will generate a failure response.
operation	JAGGAER currently supports only the "new" operation. If an operation of "update" is sent, the message will be accepted but will be interpreted as if "new" were passed. Any other operation sent will generate a failure response.
noticeDate	Required element that represents the date of confirmation creation. The date format is "YYYY-MM-DD". The timestamp can be appended if desired.
confirmID	JAGGAER supports this header attribute that represents the supplier systemorderID number. If the supplier does not pass confirmID, the JAGGAER application will generate a number in its place.
invoiceID	The invoiceID attribute is not currently supported by JAGGAER. If the attribute is passed, it will be ignored.
incoTerms	The incoTerms attribute is not currently supported by JAGGAER. If the attribute is passed, it will be ignored.
Total, Shipping & Tax	Any order totals, shipping or tax charges at the header level will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.

Element/Attribute	Description and Support Detail
Contact, Postal Addresses, Telephone Number	Any contact roles passed at the header level, including associated postal addresses and telephone information, will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
Comments	JAGGAER supports header level comments with a character limit of 100. This element can be used to provide buyers with additional information the entire order.
Extrinsics	Any extrinsics passed at the header level will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
Hazard Class	JAGGAER ignores hazard classes with the ConfirmationRequest integration. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
<OrderReference>	
orderID & orderDate	<p>These reference attributes are required only if the supplier is unable to provide the payloadID from the original orderRequest. The orderID & orderDate must match the values passed on the original OrderRequest to the supplier. The order date format is "YYYY-MM-DD". If the payloadID will be provided, these attributes can be omitted; however, the <OrderReference> element itself must be present.</p> <p>Only one orderID can be referenced in a single message. JAGGAER will accept multiple ConfirmationRequests for a single purchase order; however, suppliers should not confirm the same line item multiple times. JAGGAER does not accept updates to previously sent ConfirmationRequests.</p>
DocumentReference payloadID	<p>This element is required. If the orderID & orderDate are provided in the OrderReference tag, the payloadID can be null but not omitted (e.g. <DocumentReference payloadID="">). The payloadID must have the value from the original orderRequest if orderID & orderDate are not provided.</p> <p>Only one payloadID can be referenced in a single message. JAGGAER will accept multiple ConfirmationRequests for a single purchase order; however, suppliers should not confirm the same line item multiple times. JAGGAER does not accept updates to previously sent ConfirmationRequests.</p>
<ConfirmationItem>	
lineNumber	The ConfirmationItemlineNumber attribute is required, and the value must match the line number from the original orderRequest document. lineNumbers should be presented sequentially.
quantity	This attribute value represents the original number of ordered items.
UnitOfMeasure	The UOM value from the original order.
<ConfirmationStatus>	
type	<p>JAGGAER supports a ConfirmationStatus type of "detail", "accept", "reject", and "backordered". "allDetail", "requestToPay", "except", or any other type will generate a failure response.</p> <p>"detail" & "accept" both indicate the item on the orderRequest is accepted.</p> <p>"reject" will indicate a canceled status for the item to the buyer.</p>

Element/Attribute	Description and Support Detail
	"backordered" will indicate a backorder status for the item to the buyer.
quantity	References the number of items the ConfirmationStatus applies to.
shipmentDate	The estimated ship date for the ConfirmationStatus quantity referenced.
deliveryDate	JAGGAER does not currently support this attribute, and if passed, will be ignored. "deliveryDate" is not a substitute for "shipmentDate"
UnitOfMeasure	The UOM for the referenced item.
UnitPrice	The supplier confirmed price for the item(s). Through integration configuration, the client can elect to either accept or ignore pricing updates. If accepted, the purchase order in the procurement application will be updated with the new price. If the client elects to ignore pricing updates, a 304 response will be routed to the supplier notifying that the confirmation is accepted, but changes to the purchase order document were not executed.
Tax & Shipping	Tax and shipping charges that apply to the referenced item(s). Through integration configuration, the client can elect to either accept or ignore updates to tax & shipping. If the client elects to ignore updates to tax & shipping, a 304 response will be routed to the supplier notifying that the confirmation is accepted, but changes to the purchase order document were not executed.
incoTerms	JAGGAER does not currently support incoTerms attributes with ConfirmationRequest, and if passed, will generate a failure response.
Contact, Postal Addresses, Telephone Number	JAGGAER does not currently support any contact information at the line level, and if passed, will generate a failure response.
Comments	JAGGAER supports line level comments with a character limit of 100. This field can be used to provide buyers with additional information about the item.
Extrinsics	Any extrinsics passed at the line level will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
Hazard Class	JAGGAER ignores hazard classes with the ConfirmationRequest. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
ItemIn	JAGGAER does not currently support the <ItemIn> element (substitutions) in a ConfirmationRequest message, and if passed, will generate a failure response.

Response to Confirmation Request

JAGGAER will route a synchronous response to the supplier via the same request path. JAGGAER strongly recommends that suppliers monitor ConfirmationRequest message response codes from JAGGAER to ensure message success.

The following is an example of a JAGGAER response to a ConfirmationRequest message. Please see the [ConfirmationRequest Response Code appendix](#) for further examples and integration troubleshooting.

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

```
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd">
```

```
<cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
```

```
<Response>
```

```
<Status code="200" text="OK" />
```

```
</Response>
```

```
</cXML>
```

Ship Notice Request

The ShipNoticeRequest integration is often referred to as Advanced Ship Notice or ASN. The integration offers suppliers the opportunity to provide clients with electronic, system-to-system shipment notifications that often include additional information such as delivery dates and shipment tracking numbers. Traditionally, suppliers route email shipment confirmations from the supplier system directly to the order requestor. This integration opportunity is not intended to replace that process, but instead is intended to enhance the client procurement system with electronic shipment data that can be referenced at any time.

DTD: Fulfill.dtd

Version: 1.2.009

Guides: [ShipNotice Request Troubleshooting Guide](#), [JAGGAER Response Codes](#), [cXML User's Guide](#)

JAGGAER conforms to the cXML 1.2.009 Fulfill.dtd standard for ShipNoticeRequest integrations; however, JAGGAER does not currently support all of the request operations, elements, or element attributes. Please refer to the Message Elements table and sample messages for more information about support. If the supplier is utilizing another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, please see the [Supported cXML Versions](#) section.

Ship Notice Request Integration Requirements

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the **ShipNoticeRequest** integration:

- The supplier DUNS or Network Identity must be the same value as used in other supplier integrations with JAGGAER.
- JAGGAER currently only supports a header operation of "new" and requires shipment status and details at the line level.
- JAGGAER will return a synchronous response message to the supplier via the same request path.
- Confirmation Request testing traffic must be posted to URL: <https://usertest-messages.scquest.com/apps/Router/ASNReceive>
- Confirmation Request production traffic must be posted to URL: <https://integrations.scquest.com/apps/Router/ASNReceive>
- The supplier may be required to download and install the JAGGAER SSL digital certificates.
- All other requirements as listed in the Message Elements Table.

ShipNoticeRequest Message

There are three standard ShipNoticeRequest message operations: "new", "update", and "delete". A request operation of "new" indicates that no previous ShipNotice has been sent for the order, while an operation of "update" implies intent to modify a previously sent ShipNotice message. JAGGAER will accept a supplier message with either the "new" or "update" operation; however, the JAGGAER application will interpret both ShipNoticeRequests as "new". JAGGAER does not support the "delete" operation, and if passed, will generate a failure response.

Sample Message (Single order, multiple lines)

The below sample references a single purchase order with multiple line items; both with the same carrier and tracking number. Each referenced line item should be presented in sequential order and correspond to the line number as presented on the original OrderRequest document.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd">
<cXML timestamp="YYYY-MM-DD'TH:mm:ssZ" payloadID="[payloadID]">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>
      </Credential>
      <UserAgent>[User Agent]</UserAgent>
    </Sender>
  </Header>
  <Request>
    <ShipNoticeRequest>
```

```

<ShipNoticeHeader noticeDate="YYYY-MM-DD" operation="new deliveryDate="YYYY-MM-DD"
shipmentDate="YYYY-MM-DD" shipmentID="[Supplier's Shipment ID]">
  <!-- contact information and ServiceLevel are ignored by JAGGAER. These elements can be
omitted; however, if included must follow cXML standards. Total, shipping, and tax are not
accepted elements in a ShipNoticeRequest and can be passed only in ConfirmationRequests.
Begin ignore section----->
  <ServiceLevel xml:lang="en">[Shipping Type]</ServiceLevel>
  <Contact role="ShipFrom">
    <Name xml:lang="en">[Supplier Name]</Name>
    <PostalAddress>
      <Street>[Street]</Street>
      <City>[City]</City>
      <State>[State]</State>
      <PostalCode>[Zip Code]</PostalCode>
      <Country isoCountryCode="[Country Code]">[Country]</Country>
    </PostalAddress>
    <Phone>
      <TelephoneNumber>
        <CountryCode isoCountryCode="[Country Code]">[Country Code]</CountryCode>
        <AreaOrCityCode>[Area Code]</AreaOrCityCode>
        <Number>[Phone Number]</Number>
      </TelephoneNumber>
    </Phone>
  </Contact>
  <!-------end ignore section----->
  <Comments>[Header Level Comment]</Comments>
</ShipNoticeHeader>
<ShipControl>
  <CarrierIdentifier domain="companyName">[Carrier Name] </CarrierIdentifier>
  <ShipmentIdentifier>[Carrier TrackingNumber]</ShipmentIdentifier>
</ShipControl>
<ShipNoticePortion>
  <OrderReference orderDate="YYYY-MM-DD" orderID="[Customer PO Number]">
    <DocumentReference payloadID="[payloadID from Original PO]" />
  </OrderReference>
  <Comments>[Comment For Both Referenced Line Numbers]</Comments>
  <ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
    <UnitOfMeasure>[UOM]</UnitOfMeasure>

```

```

<Hazard>
<Description xml:lang="en">[Handling Description]</Description>
<Classification domain="[domain name]">[Hazard Description]</Classification>
</Hazard>
</ShipNoticeItem>
<ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
</ShipNoticeItem>
</ShipNoticePortion>
</ShipNoticeRequest>
</Request>
</cXML>

```

Sample Message (multiple tracking numbers)

The below sample references a single purchase order with multiple line items; referencing two carrier and tracking number possibilities. Each referenced line item should be presented in sequential order and correspond to the line number as presented on the original OrderRequest document.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]">
<Header>
<From>
<Credential domain="[From Domain]">
<Identity>[From Identity]</Identity>
</Credential>
</From>
<To>
<Credential domain="[To Domain]">
<Identity>[To Identity]</Identity>
</Credential>
</To>
<Sender>
<Credential domain="[Sender Domain]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>

```

```

<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request>
<ShipNoticeRequest>
<ShipNoticeHeader noticeDate="YYYY-MM-DD" operation="new" deliveryDate="YYYY-MM-DD"
shipmentDate="YYYY-MM-DD" shipmentID="[Supplier's Shipment ID]">
<Hazard>
<Description xml:lang="en">[Handling Description]</Description>
<Classification domain="[anyvalue]">[Hazard Description]</Classification>
</Hazard>
<Comments>[Header Level Comment]</Comments>
</ShipNoticeHeader>
<ShipControl>
<CarrierIdentifier domain="companyName">[Carrier Name]</CarrierIdentifier>
<ShipmentIdentifier>[Carrier TrackingNumber]</ShipmentIdentifier>
</ShipControl>
<ShipControl>
<CarrierIdentifier domain="companyName">[Carrier Name]</CarrierIdentifier>
<ShipmentIdentifier>[Carrier TrackingNumber]</ShipmentIdentifier>
</ShipControl>
<ShipNoticePortion>
<OrderReference orderDate="YYYY-MM-DD" orderID="[Customer PO Number]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
<Comments>[Comment For Both Referenced Line Numbers]</Comments>
<ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
</ShipNoticeItem>
<ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
</ShipNoticeItem>
</ShipNoticePortion>
</ShipNoticeRequest>
</Request>
</cXML>

```

Sample Message (consolidated shipment)

The below sample references two purchase orders; referencing a single carrier and tracking number for all items (consolidated shipment). Each referenced line item should be presented in sequential order and correspond to the line number as presented on the original OrderRequest document.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]">
  <Header>
    <From>
      <Credential domain="[From Domain]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[To Domain]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="[Sender Domain]">
        <Identity>[Sender Identity]</Identity>
        <SharedSecret>[Shared Secret]</SharedSecret>
      </Credential>
      <UserAgent>[User Agent]</UserAgent>
    </Sender>
  </Header>
  <Request>
    <ShipNoticeRequest>
      <ShipNoticeHeader noticeDate="YYYY-MM-DD" operation="new" deliveryDate="YYYY-MM-DD"
        shipmentDate="YYYY-MM-DD" shipmentID="[Supplier's Shipment ID]">
        <Comments>[Header Level Comment]</Comments>
      </ShipNoticeHeader>
      <ShipControl>
        <CarrierIdentifier domain="companyName">[Carrier Name]</CarrierIdentifier>
        <ShipmentIdentifier>[Carrier TrackingNumber]</ShipmentIdentifier>
      </ShipControl>
```

```

<ShipNoticePortion>
<OrderReference orderDate="YYYY-MM-DD" orderID="[Customer PO Number]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
<Comments>[Comment For Referenced Line Number]</Comments>
<ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
</ShipNoticeItem>
</ShipNoticePortion>
<ShipNoticePortion>
<OrderReference orderDate="YYYY-MM-DD" orderID="[Customer PO Number]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
<Comments>[Comment For Both Referenced Line Numbers]</Comments>
<ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
</ShipNoticeItem>
<ShipNoticeItem lineNumber="[lineNumber from Original PO]" quantity="[Ship Qty]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
</ShipNoticeItem>
</ShipNoticePortion>
</ShipNoticeRequest>
</Request>
</cXML>

```

Message Elements - ShipNotice Request

Element/Attribute	Description and Support Detail
<Header>	
From domain	DUNS or NetworkID. This element is required, but is not used in message authentication for this integration
From Identity	Supplier DUNS or NetworkID. This element is required, but is not used in message authentication for this integration. Typically the same value as Sender identity.
To domain	DUNS or NetworkID. This element is required, but is not used in message authentication for this integration.
To Identity	Client DUNS or NetworkID. This element is required, but is not used in message authentication for this integration.
Sender domain	DUNS or NetworkID. This element is required but is not used in message

Element/Attribute	Description and Support Detail
	authentication for this integration; however, should be the same value as used by the supplier in other integrations with JAGGAER.
Sender Identity	Supplier DUNS or NetworkID. This element is required and authenticated, and should be the same value as used by the supplier in other integrations with JAGGAER.
SharedSecret	Supplier specified SharedSecret or password. Used in message authentication.
UserAgent	Supplier's UserAgent, typically represented by the sending application and version. This element is required but is not used in message authentication.
<ShipNoticeHeader>	
noticeDate	Required attribute that represents the date of confirmation creation. The date format is "YYYY-MM-DD". The timestamp can be appended if desired.
operation	JAGGAER currently supports only the "new" operation. If an operation of "update" is sent, the message will be accepted but will be interpreted as if "new" were passed. Any other operation sent will generate a failure response.
deliveryDate	Required attribute that represents the date the client can expect to receive the items. The date format is "YYYY-MM-DD".
shipmentDate	Required attribute that represents the date the shipment departed the supplier location. The date format is "YYYY-MM-DD"
shipmentID	Required attribute that represents the supplier's reference number for the shipment. Both alpha and numeric characters are accepted.
ServiceLevel	This element describes the level of service (e.g. NextDayAir, Ground, etc.); however, JAGGAER will ignore this element.
Total, Shipping & Tax	Order totals, shipping or tax charges are not accepted in ShipNoticeRequests. The supplier has the option to provide this data with the ConfirmationRequest.
Contact, Postal Addresses, Telephone Number	Any contact roles passed at the header level, including associated postal addresses and telephone information, will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
<Hazard>	
Description	<p>Optional header level element that applies to the entire shipment. Each hazard must be listed separately with a description that outlines handling suggestions/requirements. The Description xml:lang attribute is required if hazards are provided.</p> <p>For example: <Description xml:lang="en">Handle with care! </Description></p> <p>If hazards are provided at the line level, they must not also be included at the header level to avoid a failure response. If the shipment references only a single line item, hazards should be provided at the header level.</p>
Classification domain	<p>Required only when hazards are provided. The classification value will be displayed to the user rather than the domain value.</p> <p>For example: <Classification domain="anyvalue">Toxic</p>

Element/Attribute	Description and Support Detail
	<p>Material</Classification></p> <p>"Toxic Material" will be displayed to the user rather than the domain value. For this reason, providing an element value of "true" would not convert the nature of the hazard.</p> <p>If hazards are provided at the line level, they must not also be included at the header level to avoid a failure response. If shipment references only a single line item, hazards should be provided at the header level.</p>
Comments	JAGGAER supports header level comments with a character limit of 100. This element can be used to provide buyers with additional information the entire shipment.
Extrinsics	Any extrinsics passed at the header level will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
<ShipControl>	
CarrierIdentifier	<p>At least one domain is required per cXML standards. If multiple carrier identifiers are provided, only the first domain value will be displayed to the user as the carrier name; all others will be ignored. However, if a domain of "companyName" is provided, it will take precedence regardless of the sequence it appears in.</p> <p>First example:</p> <pre><CarrierIdentifier domain="SCAC">UPS</CarrierIdentifier> <CarrierIdentifier domain="companyName">FedEx</CarrierIdentifier></pre> <p>FedEx will be displayed to the user as the Carrier Name because the domain value is "companyName".</p> <p>Second example:</p> <pre><CarrierIdentifier domain="SCAC">UPS</CarrierIdentifier> <CarrierIdentifier domain="DUNS">123456</CarrierIdentifier></pre> <p>UPS will be displayed to the user as the Carrier Name because it appears first in sequence and a domain of "companyName" was not included.</p>
ShipmentIdentifier	Required element. The value must be the carrier tracking number for the shipment. For help with multiple tracking numbers, please see the corresponding sample message.
PackageIdentification	The PackageIdentification element is ignored by JAGGAER. If the supplier elects to pass this element, it must meet cXML standards in order to avoid a failure response.
Route	The Route element is currently ignored by JAGGAER. If the supplier elects to pass this element, they must meet cXML standards in order to avoid a failure response.
Comments	Comments within the ShipControl element are ignored by JAGGAER.
Extrinsics	Any extrinsics passed in the ShipControl section will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
<ShipNotice Portion>	

Element/Attribute	Description and Support Detail
<OrderReference>	
orderID & orderDate	<p>These reference attributes are required only if the supplier is unable to provide the payloadID from the original orderRequest. The orderID & orderDate must match the values passed on the original OrderRequest to the supplier. The order date format is "YYYY-MM-DD". If the payloadID will be provided, these attributes can be omitted; however, the <OrderReference> element itself must be present.</p> <p>For help with consolidated shipments, please see the corresponding sample message.</p> <p>JAGGAER will accept multiple ShipNoticeRequests for a single purchase order; however, suppliers must not confirm shipment for the full PO line quantity multiple times. JAGGAER does not accept updates to previously sent ShipNoticeRequests.</p>
DocumentReference payloadID	<p>This element is required. If the orderID & orderDate are provided in the OrderReference tag, the payloadID can be null but not omitted (e.g. <DocumentReference payloadID="">). The payloadID must have the value from the original orderRequest if orderID & orderDate are not provided.</p> <p>For help with consolidated shipments, please see the corresponding sample message.</p> <p>JAGGAER will accept multiple ShipNoticeRequests for a single purchase order; however, suppliers must not confirm shipment for the full PO line quantity multiple times. JAGGAER does not accept updates to previously sent ShipNoticeRequests.</p>
Contact, Postal Addresses, Telephone Number	JAGGAER does not currently support any contact information at the ShipNoticePortion level, and if passed, will generate a failure response.
Comments	Optional, with 100 character limitation. Comments within the ShipNoticePortion apply to all lines referenced within the section, and only one comment per ShipNoticePortion allowed. This element can be used to provide buyers with additional information about the referenced line item(s) shipment.
Extrinsics	Any extrinsics passed in the ShipControl section will be ignored and unprocessed by JAGGAER. If the supplier elects to pass these elements, they must meet cXML standards in order to avoid a failure response.
<ShipNoticeItem>	
lineNumber	The ShipNoticeItemlineNumber attribute is required, and the value must match the line number from the original orderRequest document. Line Numbers should be presented sequentially.
quantity	This required attribute value represents the quantity of shipped items. This value may not be the full PO quantity for the line if the shipment is partial.
UnitOfMeasure	The UOM value for the referenced line item.
<Packaging>	
PackagingCode	The Packaging element is currently ignored by JAGGAER. If the supplier elects to pass this element, it must meet cXML standards in order to avoid a failure response.

Element/Attribute	Description and Support Detail
Dimension	The Dimension element is currently ignored by JAGGAER. If the supplier elects to pass this element, it must meet cXML standards in order to avoid a failure response.
UnitOfMeasure	The Packaging UnitOfMeasure element is currently ignored by JAGGAER. If the supplier elects to pass this element, it must meet cXML standards in order to avoid a failure response.
<Hazard>	
Description	<p>Optional element. Each hazard must be listed separately with a description that outlines handling suggestions/requirements. The Description xml:lang attribute is required if hazards are provided.</p> <p>For example: <Description xml:lang="en">Handle with care!</Description></p> <p>If hazards are provided at the header level, they must not also be included at the line level to avoid a failure response. If the shipment references only a single line item, hazards should be provided at the header level.</p>
Classification domain	<p>Required only when hazards are provided. The classification value will be displayed to the user rather than the domain value.</p> <p>For example: <Classification domain="anyvalue">Toxic Material</Classification></p> <p>"Toxic Material" will be displayed to the user rather than the domain value. For this reason, providing an element value of "true" would not convey the nature of the hazard.</p> <p>If hazards are provided at the line level, they must not also be included at the header level to avoid a failure response. If shipment references only a single line item, hazards should be provided at the header level.</p>

Response to ShipNoticeRequest

JAGGAER will route a synchronous response to the supplier via the same request path. JAGGAER strongly recommends that suppliers monitor ShipNoticeRequest message response codes to ensure message success.

The following is an example of a JAGGAER response to a ShipNoticeRequest message. Please see the [ShipNoticeRequest Response Code appendix](#) for further examples and integration troubleshooting.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd">
<cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ">
<Response>
<Status code="200" text="OK" />
</Response>
</cXML>
```

Invoices and Credit Memos

Clients with the Accounts Payable Director or AP Express product may request electronic invoicing via cXML. An invoice is a bill for goods and services that can be sent against some or all of the items on the original purchase order document.

DTD: InvoiceDetail.dtd

Version: 1.2.011

Guides: [Invoices/Credit Memos Troubleshooting Guide](#), [JAGGAER Response Codes](#), [cXML User's Guide](#)

JAGGAER conforms to the cXML 1.2.011 InvoiceDetail.dtd standard for invoice integrations; however, JAGGAER does not currently support all of the purposes, or element attributes. The StatusUpdateRequest integration and service invoice elements are also not currently supported by JAGGAER – please refer to the Message Elements table and sample messages for more information about support. If the supplier is utilizing another version of the standard, JAGGAER can work with the supplier to further investigate the possibility of integration. For more information about support and obtaining the DTD, please see the [Supported cXML Versions](#) section.

Integration Requirements for Invoice Integration

In addition to the general requirements outlined in the Introduction, the following is a list of requirements specific to the invoice integration:

- JAGGAER will provide an asynchronous process, so when a cXML invoice is submitted, JAGGAER will perform an initial validation on the submittal and respond with a "201" "Acknowledged" status code or "401" or "406" error status code. No other immediate status updates will be provided. Once the invoice has been processed, it will be available for viewing in the supplier network and will be added to the daily status email. Please see the Invoice/Credit Memo Response Code appendix for further examples and integration troubleshooting.
- The payloadID or the orderID & orderDate from the original OrderRequest document must be returned to JAGGAER on the inbound InvoiceDetailRequest message. This requirement may be waived by the client in cases where acceptance of non-po invoices is permitted.
- Suppliers must maintain the PO line number as sent in the <ItemOut> element of the original OrderRequest document for return on the inbound InvoiceDetailRequest message. While extremely rare, this requirement may be waived by the client where acceptance of header level invoices is permitted.
- For punch-out suppliers providing cXML invoices, the unitPrice format (decimal places) and SupplierPartID values must be consistent.
- The InvoiceDetailRequest message structure must meet with client approval and all integration Test Plan requirements must be successfully completed before promotion to production can be considered.

- Invoice testing traffic must be posted to URL: <https://usertest-messages.scquest.com/apps/Router/CXMLInvoiceImport>
- Invoice production traffic must be posted to URL: <https://integrations.scquest.com/apps/Router/CXMLInvoiceImport>
- The supplier may be required to download and install the JAGGAER SSL digital certificates.
- All other requirements as listed in the [Message Elements](#) table.
- If any of the above requirements cannot be met, the client and/or JAGGAER will request the supplier route client invoices and credits either manually or via the Supplier Portal.

Document Structure

The Client's settlement configuration and process will dictate the document structure the supplier is asked to provide. The structure of the settlement document consists of a category, level, purpose, and a method for handling additional fees such as shipping, handling, and tax. JAGGAER supports two options for each structure element that are used in combination. The following outlines the definition for each:

Category

- Individual: Represents a single settlement document that applies to an individual purchase order. This is the most commonly requested document category type. The presence of a single `<InvoiceDetailOrder>` element is what distinguishes an individual invoice/credit from a summary invoice/credit.
- Summary: Represents a single settlement document that applies to multiple purchase orders. This is sometimes referred to as a batch invoice. Most JAGGAER clients do not accept this settlement document category as it does not allow individual acceptance/rejection by purchase order. The presence of multiple `<InvoiceDetailOrder>` elements is what distinguishes a summary invoice/credit from an individual invoice/credit.

Level

- Detail Line Level: References individual line items and detail that were present on the original purchase order. This is the most commonly requested settlement document level. A null value for `<InvoiceDetailHeaderIndicator />` and the presence of the `<InvoiceDetailOrder>` element is what distinguishes a detail line level invoice/credit from a header level invoice/credit.
- Header Level: Provides a total bill or credit amount that is applied as a weighted average across all line items that were present on the original purchase order. This level does not provide line details. Most JAGGAER clients do not accept settlement documents at this level. The presence of `<InvoiceDetailHeaderIndicator isHeaderInvoice="yes">` within the document header as well as the `<InvoiceDetailHeaderOrder>` element is what distinguishes a header level invoice/credit from a detail line level invoice/credit.

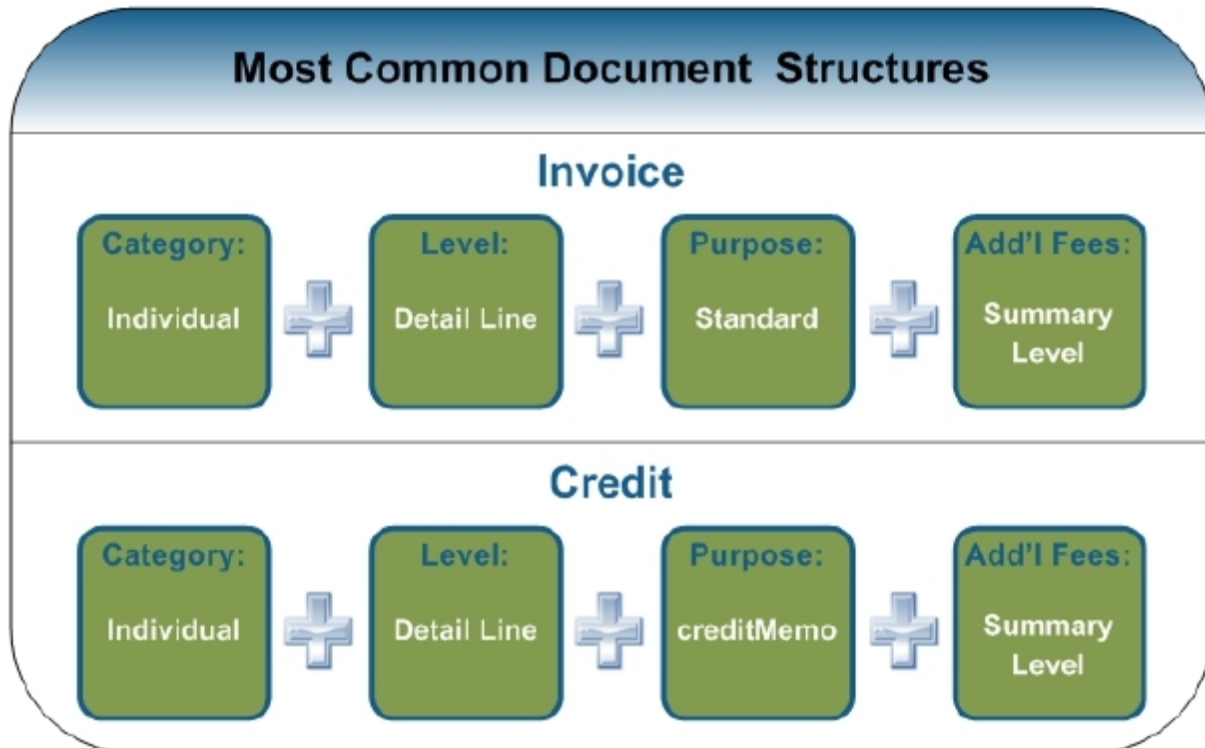
Purpose

- **Standard Invoice:** Represents a request for payment. The purpose attribute is present in <InvoiceDetailRequestHeader> and is set to "standard". Should the supplier provide a negative UnitPrice value for any line item, the negative amount will be processed as a credit while any positive UnitPrice values will be treated as a debit. Suppliers should not provide both positive and negative UnitPrice values in any single document. All quantity values must be positive.
- **Credit Memo:** Represents an account credit or amount due back to the buying organization. The purpose attribute is present in <InvoiceDetailRequestHeader> and is set to "creditMemo". Either positive or negative UnitPrice values can be provided; however, all quantity values must be positive.

Additional Fees

- **Summary Level:** Shipping, handling, and tax fees are summarized at the document level. This is the most commonly requested method. A null value for <InvoiceDetailLineIndicator /> and the presence of additional fee elements within the <InvoiceDetailSummary> section indicate provisions to apply total fees as a weighted average across all line items present on the invoice/credit document.
- **Line Level:** Shipping, handling, and tax fees are presented in detail, such that the buying organization can identify what specific portion of the total fees apply by line. The presence of <InvoiceDetailLineIndicator isTaxInLine="yes" isShippingInLine="yes" isSpecialHandlingInLine="yes"> within the document header as well as the elements for additional fees within the <InvoiceDetailItem> section indicate a provision for fees at the line level. While not as popular as summary level, some clients do request a line level fee structure.

When considering category, level, purpose, and additional fees, there are multiple settlement document structure options the client may request. Clients typically define one structure for standard invoices and one for credit memos that applies either organizationally or by supplier. In some cases, the client may request that credit memos continue to be processed manually due to ERP limitations. The below image illustrates the most commonly requested invoice and credit structures:



Suppliers interested in developing cXML invoice capabilities are encouraged to include the above illustrated structures, as well as the ability to provide additional fees at the line level.

Miscellaneous Fees

Customers can configure their invoice documents to include up to 15 different miscellaneous fee fields on a cXML or portal, single or bulk, invoice.

- JAGGAER pre-defined 5 miscellaneous fees, Suppliers must use the case sensitive Internal Names provided below
 - DryIceFee
 - EWasteFee
 - RentalFee
 - HazardousFee
 - InstallationFee
- The remaining 10 miscellaneous fee fields can be defined by each client to match their business processes. Suppliers must use the case sensitive Internal Names provided by each client.
- Suppliers can add other miscellaneous fees not configured by the customer if the invoice is passed electronically. However, a customer may reject the invoice or remove the fee value upon receipt.

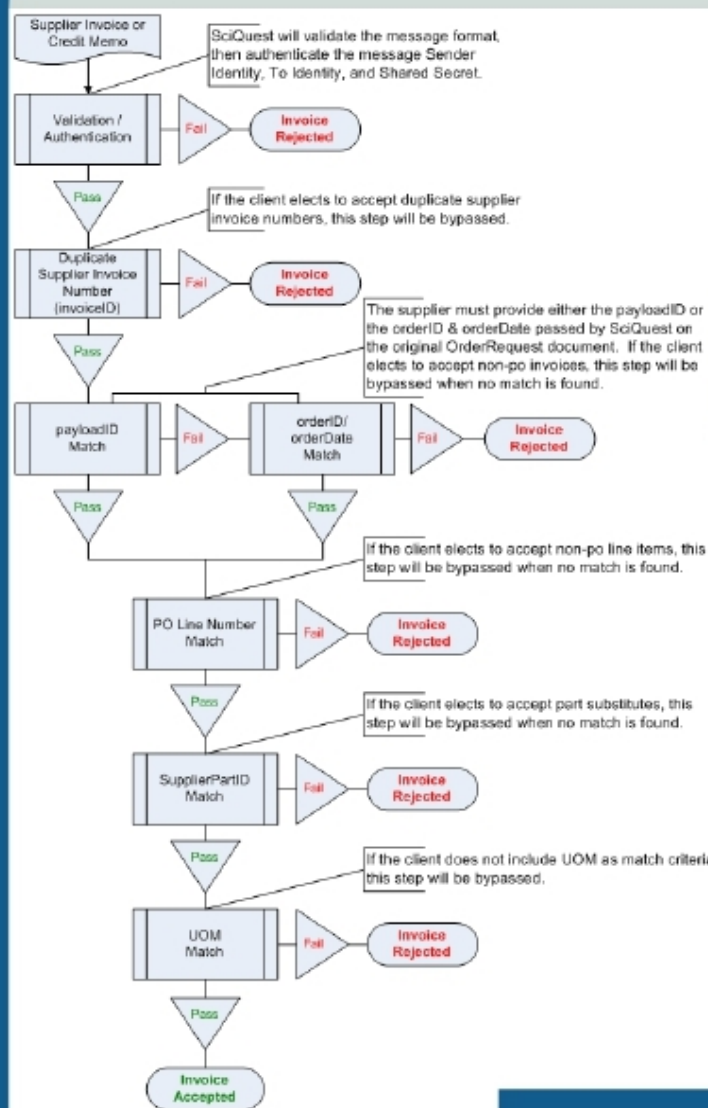
- Miscellaneous fees are listed within the <InvoiceDetailSummary> section of a cXML invoice or credit memo after the totals. If no fees are configured to show, the Miscellaneous Fees field does not display in the totals section.

Document Acceptance Process

The JAGGAER application performs an initial settlement document match to determine the first level of acceptance/rejection. The match criteria are based upon message validation, authentication, and the client's Accounts Payable product configuration. Once a settlement document is initially accepted by the JAGGAER system, it then moves through the client's defined workflow process where it will either be processed for payment or routed to an exception queue for issue resolution. Any and all issue resolution post-JAGGAER acceptance is handled directly between the client and supplier.

A settlement document that does not meet match criteria and is initially rejected by the JAGGAER system is not imported to the application; and is therefore not seen or reviewed by the client. Suppliers must monitor JAGGAER response messages and have a process in place to handle failures.

Invoice/Credit Memo Acceptance Process



Suppliers must have a process in place to handle invoice error response messages.

SciQuest does not accept or reject invoices/credits based on pricing. Any pricing issues will be addressed by the client directly with the supplier AFTER initial invoice acceptance.

Non-PO Definitions

Not all client purchase orders or purchase order line items may exist within the JAGGAER application. In some cases, clients may elect to generate and distribute a purchase order document directly from the ERP system, or users may call the supplier directly to place an order. In other cases, the purchase order may exist in the JAGGAER application, but a specific line item was added after document creation or requires part substitution by the supplier. Clients have the ability to configure non-po acceptance criteria via the Accounts Payable product in two areas:

- Non-PO Invoice/Credit: Represents an invoice or credit against a purchase order that does not exist in the JAGGAER application. If the client elects to reject non-po invoices/credits, suppliers will be required to provide these settlement documents via a manual or paper process.
- Non-PO Line Item: Represents an invoiced/credited line item that does not exist on the original purchase order document within the JAGGAER application. If the client elects to reject non-po line items, suppliers will be required to invoice/credit these lines via a manual or paper process. For more information about non-po line items and product substitution, please review the [Part Substitutions](#) section.

Part Substitutions

In some cases an item is either backordered or is no longer available from the supplier. Depending upon the client's Accounts Payable configuration, suppliers can either request the user/shopper generate a new order for an alternative item (substitutes not allowed), or substitute the ordered item with a comparable one on the invoice/credit document (substitutes allowed).

If the client elects to accept part substitutes, there are two possible ways to execute substitution on the invoice/credit document:

- Substitute on Same PO Line: Preferred. The supplier references the original purchase order line number in <InvoiceDetailItemReference lineNumber="Original PO Line Number" quantity="X">, and provides the substituted part number value in place of the original one in <SupplierPartID>.
- Substitute on New PO Line: The supplier does not invoice/credit the original purchase order line and instead creates/references a new line that does not exist on the original order document. Because this option requires the client to accept non-po line items, it is typically not permitted.

During integration implementation, the handling of part substitutes should be reviewed with all parties in detail. If the client elects to accept part substitutes, the method of substitution must be consistent with the client's Accounts Payable product configuration.

Settlement Extrinsic

Extrinsics are used to relay additional information about the document or item that is not covered in any other element or attribute, and can be defined by the client and/or supplier. There are two types of extrinsics that can be passed on a settlement document at either or both the header and line levels: standard and custom.

Standard Extrinsic

A standard extrinsic is defined by cXML standards and is mapped to either the header or line level notes section within the JAGGAER application based upon where it appears in the

InvoiceDetailRequest message. Standard extrinsics in combination with comments have a total character limitation of 100 each at the header and line levels respectively, including the extrinsic label and name. Since invoice comments appear before extrinsics in the document, only the remaining extrinsic characters after comments will be recorded, truncating any data that exceeds 100 chars.

Best practice is for suppliers to provide invoice extrinsic data only in cases where the client has requested it, or in cases where the supplier deems the data pertinent to client processing of the invoice and has discussed its purpose with the client. In general, data that has an element or attribute per cXML standards should not be passed in extrinsics (e.g. payment terms, remittance address, etc.) as invoice notes may not be reviewed or exported by some clients.

Usage: In rare cases a client may request that all or specific extrinsics passed to the supplier on the OrderRequestMessage be returned on the inbound settlement document for mapping to the respective notes section within the JAGGAER application.

Suppliers unable to provide the traditional <comments> element can also use standard extrinsics to pass notes at either or both the header and line levels.

In the above example, the extrinsic name can be defined by either the client (via the OrderRequestMessage) or by the supplier. "Extrinsic: Name=Value" is recorded to invoice notes.

Custom Extrinsics

A custom extrinsic is based upon cXML standards, but is further defined by JAGGAER via the extrinsic name. Custom extrinsics at either the header or line levels have a character limitation of 100 each and are stored individually rather than mapped to invoice notes within the JAGGAER application.

Suppliers will only provide custom extrinsics upon client request and must follow a specific syntax in order for intended processing to be successful. Any custom extrinsic not properly passed may generate an invoice error response or be mapped to invoice notes as standard extrinsics are.

Because each client can create and define their own unique custom fields, the supplier cannot expect that the custom extrinsic nomenclature for one client will work for all other clients with similar custom fields.

USAGE

In some cases a client may create a custom field within the JAGGAER application and request that the supplier return a corresponding value on the inbound settlement document. Custom fields are searchable within the application and therefore provide the client with additional value in reporting.

In very rare non-PO invoice cases, a client may request that account code values be passed as custom extrinsics on the inbound settlement document. Suppliers should discuss this scenario with the JAGGAER Project Manager before proceeding.

General Example: <Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">
[Extrinsic Value]</Extrinsic>

In the above example, CFValue stands for Custom Field Value and is what differentiates it from a standard extrinsic. The Custom Field Internal Name follows the underscore and is defined solely by the client. During implementation the supplier will be provided with the exact syntax and location (header or line) in order to ensure successful processing.

InvoiceDetailRequest Message

The following samples illustrate various accepted message formats; however, suppliers may elect to omit non-required elements. For more information about specific elements, refer to the [Message Elements](#) table.

Messages are posted via HTTP (over SSL, port 443) to either the test or production URL as applicable. JAGGAER ignores the message deployment mode passed by the supplier as two separate environments are maintained.

- Test: <https://usertest-messages.sciquest.com/apps/Router/CXMLInvoiceImport>
- Production: <https://integrations.sciquest.com/apps/Router/CXMLInvoiceImport>

JAGGAER calculates the extended price and invoice total using the UnitPrice and quantity elements for each line rather than parsing the sum amount(s) passed by the supplier (e.g. <SubtotalAmount>, <NetAmount>, <DueAmount>); however, the supplier must include these elements as they are required per cXML standards.

Invoice Total Calculation: =SUM(Line UnitPriceXQuantity) – Discount + Shipping + Handling + Tax

Sample Message (Most Common Structure)

The following sample illustrates the most commonly requested invoice or credit memo document structure.

Category: Individual

Level: Detail Line

Purpose: standard or creditMemo

Additional Fees: Summary Level

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

<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">

<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payload ID]" xml:lang="en-US" version="1.2.011">

```

<Header>
<From>
<Credential domain="[NetworkID or DUNS]">
<Identity>[From Identity]</Identity>
</Credential>
</From>
<To>
<Credential domain="[NetworkID or DUNS]">
<Identity>[To Identity]</Identity>
</Credential>
</To>
<Sender>
<Credential domain="[NetworkID or DUNS]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request deploymentMode="[production or test]">
<InvoiceDetailRequest>
<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="[standard or
creditMemo]" operation="new" invoiceDate="YYYY-MM-DD'T'HH:mm:ssZ">
<InvoiceDetailHeaderIndicator />
<InvoiceDetailLineIndicator />
<InvoicePartner>
<Contact role="billTo">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>

```

```

<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>
<Contact role="remitTo">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>

```

<!--JAGGAER ignores data within the InvoiceDetailShipping section. Suppliers have the option to omit the entire element when additional fees are provided at the summary level. If passed, the supplier must provide two contact roles. Providing only a single contact role will generate a failure

response. Begin ignore section----- >

```

<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[Supplier AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="[Supplier Address Name]">
<Street>[Street]</Street>

```

```

<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Supplier Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[ShipTo AddressID from PO]">
<Name xml:lang="en-US">[ShipTo Name from PO]</Name>
<PostalAddress name="[Address Name from PO]">
<DeliverTo>[DeliverTo Name from PO]</DeliverTo>
<Street>[Street from PO]</Street>
<City>[City from PO]</City>
<State>[State from PO]</State>
<PostalCode>[Postal Code from PO]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>

```



```

<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----->
<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>

```

```

<Comments xml:lang="en-US">[Line Level Comment]</Comments>

<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic
Value]</Extrinsic>

</InvoiceDetailItem>

<InvoiceDetailItem invoiceLineNumber="2" quantity="[Quantity]">

<UnitOfMeasure>[UOM]</UnitOfMeasure>

<UnitPrice>

<Money currency="USD">[Unit Price]</Money>

</UnitPrice>

<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">

<ItemID>

<SupplierPartID>[Part Number]</SupplierPartID>

</ItemID>

<Description xml:lang="en-US">[Part Description]</Description>

</InvoiceDetailItemReference>

<Extrinsic name="[Extrinsic Name]">[Line Extrinsic Value]</Extrinsic>

</InvoiceDetailItem>

</InvoiceDetailOrder>

<InvoiceDetailSummary>

<SubtotalAmount>

<Money currency="USD">[Subtotal]</Money>

</SubtotalAmount>

<Tax>

<Money currency="USD">[Total Tax Amount]</Money>

<Description xml:lang="en-US">[Tax Description]</Description>

</Tax>

<SpecialHandlingAmount>

<Money currency="USD">[Total Handling Amount]</Money>

</SpecialHandlingAmount>

```

```

<ShippingAmount>
<Money currency="USD">[Total Shipping Amount]</Money>
</ShippingAmount>
<GrossAmount>
<Money currency="USD">[Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Net Amount]</Money>
</NetAmount>
<DueAmount>
<Money currency="USD">[Amount Due]</Money>
</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Line Level Additional Fees)

The following sample illustrates an individual invoice or credit memo with provisions for additional fees at the line level.

Category: Individual

Level: Detail Line

Purpose: standard or creditMemo

Additional Fees: Summary Level

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

```
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">
```

<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payload ID]" xml:lang="en-US" version="1.2.011">

<Header>

<From>

<Credential domain="[NetworkID or DUNS]">

<Identity>[From Identity]</Identity>

</Credential>

</From>

<To>

<Credential domain="[NetworkID or DUNS]">

<Identity>[To Identity]</Identity>

</Credential>

</To>

<Sender>

<Credential domain="[NetworkID or DUNS]">

<Identity>[Sender Identity]</Identity>

<SharedSecret>[Shared Secret]</SharedSecret>

</Credential>

<UserAgent>[User Agent]</UserAgent>

</Sender>

</Header>

<Request deploymentMode="[production or test]">

<InvoiceDetailRequest>

<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="[standard or creditMemo]" operation="new" invoiceDate="YYYY-MM-DD'T'HH:mm:ssZ">

<InvoiceDetailHeaderIndicator />

<InvoiceDetailLineIndicator isTaxInLine="yes" isSpecialHandlingInLine="yes" isShippingInLine="yes" isDiscountInLine="yes"/>

<InvoicePartner>

<Contact role="billTo" >

```

<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>
<Contact role="remitTo" >
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>

```

```

<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<SubtotalAmount>
<Money currency="USD">[Line Subtotal Amount]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Line Tax Amount]</Money>
<Description xml:lang="en-US">[Line Tax Description]</Description>
</Tax>
<InvoiceDetailLineSpecialHandling>
<Money currency="USD">[Line Handling Amount]</Money>
</InvoiceDetailLineSpecialHandling>
<!--JAGGAER ignores data within the InvoiceDetailShipping section; however it is required per
cXML standards when providing additional fees at the line level. Suppliers must provide two
contact roles - providing only a single contact role will generate a failure response. Begin
ignore section----->

```

```

<InvoiceDetailLineShipping>
<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[Supplier AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="[Supplier Address Name]">
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Supplier Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[ShipTo AddressID from PO]">
<Name xml:lang="en-US">[ShipTo Name from PO]</Name>
<PostalAddress name="[Address Name from PO]">
<DeliverTo>[DeliverTo Name from PO]</DeliverTo>
<Street>[Street from PO]</Street>
<City>[City from PO]</City>
<State>[State from PO]</State>
<PostalCode>[Postal Code from PO]</PostalCode>
<Country isoCountryCode="US">United States</Country>

```

```

</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >
<Money currency="USD">[Line Shipping Total]</Money>
</InvoiceDetailLineShipping>
<GrossAmount>
<Money currency="USD">[Line Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Line Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Line Net Amount]</Money>
</NetAmount>
<Comments xml:lang="en-US">[Line Level Comment]</Comments>
<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic
Value]</Extrinsic>
</InvoiceDetailItem>
<InvoiceDetailItem invoiceLineNumber="2" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>

```



```

<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<SubtotalAmount>
<Money currency="USD">[Line Subtotal Amount]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Line Tax Amount]</Money>
<Description xml:lang="en-US">[Line Tax Description]</Description>
</Tax>
<InvoiceDetailLineSpecialHandling>
<Money currency="USD">[Line Handling Amount]</Money>
</InvoiceDetailLineSpecialHandling>
<!--JAGGAER ignores data within the InvoiceDetailShipping section; however, it is required
per cXML standards when providing additional fees at the line level. Suppliers must provide
two contact roles - providing only a single contact role will generate a failure response. Begin
ignore section----->
<InvoiceDetailLineShipping>
<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[Supplier AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="[Supplier Address Name]">
<Street>[Street]</Street>
<City>[City]</City>

```

```

<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Supplier Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[ShipTo AddressID from PO]">
<Name xml:lang="en-US">[ShipTo Name from PO]</Name>
<PostalAddress name="[Address Name from PO]">
<DeliverTo>[DeliverTo Name from PO]</DeliverTo>
<Street>[Street from PO]</Street>
<City>[City from PO]</City>
<State>[State from PO]</State>
<PostalCode>[Postal Code from PO]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>

```

```

</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >
<Money currency="USD">[Line Shipping Amount]</Money>
</InvoiceDetailLineShipping>
<GrossAmount>
<Money currency="USD">[Line Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Line Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Line Net Amount]</Money>
</NetAmount>
<Extrinsic name="[Extrinsic Name]">[Line Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
</InvoiceDetailOrder>
<InvoiceDetailSummary>
<SubtotalAmount>
<Money currency="USD">[Invoice Subtotal]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Invoice Total Tax Amount]</Money>
<Description xml:lang="en-US">[Tax Description]</Description>
</Tax>
<SpecialHandlingAmount>
<Money currency="USD">[Invoice Total Handling Amount]</Money>

```

```

</SpecialHandlingAmount>
<ShippingAmount>
<Money currency="USD">[Invoice Total Shipping Amount]</Money>
</ShippingAmount>
<GrossAmount>
<Money currency="USD">[Invoice Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Invoice Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Invoice Net Amount]</Money>
</NetAmount>
<DueAmount>
<Money currency="USD">[Invoice Amount Due]</Money>
</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Header Invoice, Summary Level Fees)

The following sample illustrates a header level invoice against a single purchase order with provisions for additional fees at the summary level. Note that most JAGGAER clients do not accept this invoice structure and should only be sent upon client request.

Category: Individual

Level: Header

Purpose: standard or creditMemo

Additional Fees: Summary Level

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

<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">

<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payload ID]" xml:lang="en-US" version="1.2.011">

<Header>

<From>

<Credential domain="[NetworkID or DUNS]">

<Identity>[From Identity]</Identity>

</Credential>

</From>

<To>

<Credential domain="[NetworkID or DUNS]">

<Identity>[To Identity]</Identity>

</Credential>

</To>

<Sender>

<Credential domain="[Same as From Domain]">

<Identity>[Same as From Identity]</Identity>

<SharedSecret>[Shared Secret]</SharedSecret>

</Credential>

<UserAgent>[User Agent]</UserAgent>

</Sender>

</Header>

<Request deploymentMode="[production or test]">

<InvoiceDetailRequest>

<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="[standard or creditMemo]" operation="new" invoiceDate="[Invoice Date]">

<InvoiceDetailHeaderIndicator isHeaderInvoice="yes"/>

```

<InvoiceDetailLineIndicator />
<InvoicePartner>
<Contact role="billTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>
<Contact role="remitTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<!--JAGGAER ignores data within the InvoiceDetailShipping section. Suppliers have the option
to omit the entire element when additional fees are provided at the summary level. If passed,
the supplier must provide two contact roles. Providing only a single contact role will generate a
failure response. Begin ignore section-----
----->

```

```

<InvoiceDetailShipping>
  <Contact role="shipFrom" addressID="[AddressID]">
    <Name xml:lang="en-US">[Supplier Name]</Name>
    <PostalAddress name="default">
      <Street>[Street]</Street>
      <City>[City]</City>
      <State>[State]</State>
      <PostalCode>[Postal Code]</PostalCode>
      <Country isoCountryCode="US">United States</Country>
    </PostalAddress>
    <Email name="default">[Email Address]</Email>
    <Phone name="work">
      <TelephoneNumber>
        <CountryCode isoCountryCode="US">1</CountryCode>
        <AreaOrCityCode>[Area Code]</AreaOrCityCode>
        <Number>[Phone Number]</Number>
      </TelephoneNumber>
    </Phone>
  </Contact>
  <Contact role="shipTo" addressID="[AddressID]">
    <Name xml:lang="en-US">[Client Name]</Name>
    <PostalAddress name="[Address Name from PO]">
      <DeliverTo>[DeliverTo Name]</DeliverTo>
      <Street>[Street]</Street>
      <City>[City]</City>
      <State>[State]</State>
      <PostalCode>[Postal Code]</PostalCode>
      <Country isoCountryCode="US">United States</Country>
    </PostalAddress>

```

```

<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >
<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailHeaderOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailOrderSummary invoiceLineNumber="1" >
<SubtotalAmount>
<Money currency="USD">[Invoice Subtotal]</Money>
</SubtotalAmount>
<Comments xml:lang="en-US">[Sample Line Comment]</Comments>

```



```

<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic
Value]</Extrinsic>

</InvoiceDetailOrderSummary>

</InvoiceDetailHeaderOrder>

<InvoiceDetailSummary>

<SubtotalAmount>

<Money currency="USD">[Invoice Subtotal]</Money>

</SubtotalAmount>

<Tax>

<Money currency="USD">[Invoice Total Tax Amount]</Money>

<Description xml:lang="en-US">[Tax Description]</Description>

</Tax>

<SpecialHandlingAmount>

<Money currency="USD">[Invoice Total Handling Amount]</Money>

</SpecialHandlingAmount>

<ShippingAmount>

<Money currency="USD">[Invoice Total Shipping Amount]</Money>

</ShippingAmount>

<GrossAmount>

<Money currency="USD">[Invoice Gross Amount]</Money>

</GrossAmount>

<InvoiceDetailDiscount>

<Money currency="USD">[Invoice Discount Amount]</Money>

</InvoiceDetailDiscount>

<NetAmount>

<Money currency="USD">[Invoice Net Amount]</Money>

</NetAmount>

<DueAmount>

<Money currency="USD">[Invoice Amount Due]</Money>

```

```

</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Header Invoice, Line Level Fees)

The following sample illustrates a header level invoice against a single purchase order with provisions for additional fees at the line level. Note that most JAGGAER clients do not accept this invoice structure and should only be sent upon client request.

Category: Individual

Level: Header

Purpose: standard or creditMemo

Additional Fees: Line Level

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[supplier payloadID]">
  <Header>
    <From>
      <Credential domain="[NetworkID or DUNS]">
        <Identity>[From Identity]</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="[NetworkID or DUNS]">
        <Identity>[To Identity]</Identity>
      </Credential>
    </To>
    <Sender>

```

```

<Credential domain="[NetworkID or DUNS]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request deploymentMode="[test or production]">
<InvoiceDetailRequest>
<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="[standard or
creditMemo]" operation="new" invoiceDate="[Invoice Date]">
<InvoiceDetailHeaderIndicator isHeaderInvoice="yes" />
<InvoiceDetailLineIndicator isTaxInLine="yes" isShippingInLine="yes"
isSpecialHandlingInLine="yes" isDiscountInLine="yes" />
<InvoicePartner>
<Contact role="billTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>
<Contact role="remitTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>

```

```

<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailHeaderOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailOrderSummary invoiceLineNumber="1">
<SubtotalAmount>
<Money currency="USD">[Invoice Subtotal]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Invoice Total Tax Amount]</Money>
<Description xml:lang="en-US">[Tax Description]</Description>
</Tax>

```

```

<InvoiceDetailLineSpecialHandling>
<Money currency="USD">[Invoice Total Handling Amount]</Money>
</InvoiceDetailLineSpecialHandling>

<InvoiceDetailLineShipping>

<!--JAGGAER ignores data within the InvoiceDetailShipping section; however, it is required
per cXML standards when providing additional fees at the line level. Suppliers must provide
two contact roles - providing only a single contact role will generate a failure response. Begin
ignore section----->

<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="default">
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress name="default">

```

```

<DeliverTo>[DeliverTo Name]</DeliverTo>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >
<Money currency="USD">[Invoice Total Shipping Amount]</Money>
</InvoiceDetailLineShipping>
<GrossAmount>
<Money currency="USD">[Invoice Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Invoice Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Invoice Net Amount]</Money>
</NetAmount>

```

```

<Comments xml:lang="en-US">[Sample Line Comment]</Comments>

<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic
Value]</Extrinsic>

</InvoiceDetailOrderSummary>

</InvoiceDetailHeaderOrder>

<InvoiceDetailSummary>

<SubtotalAmount>

<Money currency="USD">[Invoice Subtotal]</Money>

</SubtotalAmount>

<Tax>

<Money currency="USD">[Invoice Total Tax Amount]</Money>

<Description xml:lang="en-US">[Tax Description]</Description>

</Tax>

<SpecialHandlingAmount>

<Money currency="USD">[Invoice Total Handling Amount]</Money>

</SpecialHandlingAmount>

<ShippingAmount>

<Money currency="USD">[Invoice Total Shipping Amount]</Money>

</ShippingAmount>

<GrossAmount>

<Money currency="USD">[Invoice Gross Amount]</Money>

</GrossAmount>

<InvoiceDetailDiscount>

<Money currency="USD">[Invoice Discount Amount]</Money>

</InvoiceDetailDiscount>

<NetAmount>

<Money currency="USD">[Invoice Net Amount]</Money>

</NetAmount>

<DueAmount>

```

```

<Money currency="USD">[Invoice Due Amount]</Money>
</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Summary Invoice, Line Level Fees)

The following sample illustrates a summary category invoice with provisions for additional fees at the line level. A summary invoice with additional fees at the summary level is not appropriate as multiple purchase orders are referenced in a single document. Note that most JAGGAER clients do not accept this invoice structure and should only be sent upon client request.

Category: Summary

Level: Detail Line

Purpose: standard or creditMemo

Additional Fees: Line Level

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[Supplier payloadID]">
<Header>
<From>
<Credential domain="[NetworkID or DUNS]">
<Identity>[From Identity]</Identity>
</Credential>
</From>
<To>
<Credential domain="[NetworkID or DUNS]">
<Identity>[To Identity]</Identity>
</Credential>

```



```

</To>

<Sender>

<Credential domain="[NetworkID or DUNS]">

<Identity>[Sender Identity]</Identity>

<SharedSecret>[Shared Secret]</SharedSecret>

</Credential>

<UserAgent>[User Agent]</UserAgent>

</Sender>

</Header>

<Request deploymentMode="[test or production]">

<InvoiceDetailRequest>

<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="[standard or
creditMemo]" operation="new" invoiceDate="[Invoice Date]">

<InvoiceDetailHeaderIndicator />

<InvoiceDetailLineIndicator isTaxInLine="yes" isShippingInLine="yes"
isSpecialHandlingInLine="yes" isDiscountInLine="yes" />

<InvoicePartner>

<Contact role="billTo" addressID="[AddressID]">

<Name xml:lang="en-US">[Client Name]</Name>

<PostalAddress>

<Street>[Street]</Street>

<City>[City]</City>

<State>[State]</State>

<PostalCode>[Postal Code]</PostalCode>

<Country isoCountryCode="US">United States</Country>

</PostalAddress>

</Contact>

</InvoicePartner>

<InvoicePartner>

```

```

<Contact role="remitTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<PaymentTerm payInNumberOfDays="30" />
<Comments xml:lang="en-US">[Header Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]"> <DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>

```

```

</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<SubtotalAmount>
<Money currency="USD">[Line Subtotal Amount]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Line Tax Amount]</Money>
<Description xml:lang="en-US">[Line Tax Description]</Description>
<TaxDetail category="[Tax Category]">
<TaxableAmount>
<Money currency="USD">[Taxable Amount]</Money>
</TaxableAmount>
<TaxAmount>
<Money currency="USD">[Tax Amount]</Money>
</TaxAmount>
</TaxDetail>
</Tax>
<InvoiceDetailLineSpecialHandling>
<Money currency="USD">[Invoice Line Handling Amount]</Money>
</InvoiceDetailLineSpecialHandling>
<InvoiceDetailLineShipping>
<!--JAGGAER ignores data within the InvoiceDetailShipping section; however, it is required
per cXML standards when providing additional fees at the line level. Suppliers must provide
two contact roles - providing only a single contact role will generate a failure response. Begin
ignore section----->
<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>

```

```

<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
<Contact role="shipTo" addressID="[AddressID]">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress name="default">
<DeliverTo>[Deliver To Name]</DeliverTo>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----->
<Money currency="USD">[Line Shipping Amount]</Money>
</InvoiceDetailLineShipping>
<GrossAmount>
<Money currency="USD">[Line Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Line Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>

```

```

<Money currency="USD">[Line Net Amount]</Money>
</NetAmount>
<Comments xml:lang="en-US">[Line Level Comment]</Comments>
<Extrinsic name="CFVALUE_ [Client Custom Field Internal Name]">[Extrinsic
Value]</Extrinsic>
</InvoiceDetailItem>
</InvoiceDetailOrder>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original
PO]"> <DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<SubtotalAmount>
<Money currency="USD">[Line Subtotal Amount]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Line Tax Amount]</Money>

```

```

<Description xml:lang="en-US">[Line Tax Description]</Description>
<TaxDetail category="[Tax Category]">
<TaxableAmount>
<Money currency="USD">[Taxable Amount]</Money>
</TaxableAmount>
<TaxAmount>
<Money currency="USD">[Tax Amount]</Money>
</TaxAmount>
</TaxDetail>
</Tax>
<InvoiceDetailLineSpecialHandling>
<Money currency="USD">[Invoice Line Handling Amount]</Money>
</InvoiceDetailLineSpecialHandling>
<InvoiceDetailLineShipping>
<!--JAGGAER ignores data within the InvoiceDetailShipping section; however, it is required
per cXML standards when providing additional fees at the line level. Suppliers must provide
two contact roles - providing only a single contact role will generate a failure response. Begin
ignore section----->
<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
<Contact role="shipTo" addressID="[AddressID]">

```

```

<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress name="default">
<DeliverTo>[Deliver To Name]</DeliverTo>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >
<Money currency="USD">[Line Shipping Amount]</Money>
</InvoiceDetailLineShipping>
<GrossAmount>
<Money currency="USD">[Line Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Line Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Line Net Amount]</Money>
</NetAmount>
<Comments xml:lang="en-US">[Line Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
</InvoiceDetailOrder>
<InvoiceDetailSummary>
<SubtotalAmount>

```

```

<Money currency="USD">[Invoice Subtotal]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Invoice Total Tax Amount]</Money>
<Description xml:lang="en-US">[Tax Description]</Description>
<TaxDetail category="sales">
<TaxableAmount>
<Money currency="USD">[Taxable Amount]</Money>
</TaxableAmount>
<TaxAmount>
<Money currency="USD">[Tax Amount]</Money>
</TaxAmount>
</TaxDetail>
</Tax>
<SpecialHandlingAmount>
<Money currency="USD">[Invoice Total Handling Amount]</Money>
</SpecialHandlingAmount>
<ShippingAmount>
<Money currency="USD">[Invoice Total Shipping Amount]</Money>
</ShippingAmount>
<GrossAmount>
<Money currency="USD">[Invoice Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Invoice Discount Amount]</Money>
</InvoiceDetailDiscount>
<NetAmount>
<Money currency="USD">[Invoice Net Amount]</Money>
</NetAmount>

```



```

<DueAmount>
<Money currency="USD">[Invoice Due Amount]</Money>
</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Miscellaneous Fees)

The following sample illustrates the most commonly requested invoice or credit memo document structure with Miscellaneous fees added to the total section within the <InvoiceDetailSummary> element.

Category: Individual

Level: Detail Line

Purpose: standard or creditMemo

Additional Fees: Summary Level

Miscellaneous Fees: Summary Level

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payload ID]" xml:lang="en-US" version="1.2.011">
<Header>
<From>
<Credential domain="[NetworkID or DUNS]">
<Identity>[From Identity]</Identity>
</Credential>
</From>
<To>
<Credential domain="[NetworkID or DUNS]">
<Identity>[To Identity]</Identity>

```

```

</Credential>
</To>
<Sender>
<Credential domain="[NetworkID or DUNS]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request deploymentMode="[production or test]">
<InvoiceDetailRequest>
<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="[standard or
creditMemo]" operation="new" invoiceDate="YYYY-MM-DD'T'HH:mm:ssZ">
<InvoiceDetailHeaderIndicator />
<InvoiceDetailLineIndicator />
<InvoicePartner>
<Contact role="billTo">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>

```

```

<Contact role="remitTo">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>

```

<!--JAGGAER ignores data within the InvoiceDetailShipping section. Suppliers have the option to omit the entire element when additional fees are provided at the summary level. If passed, the supplier must provide two contact roles. Providing only a single contact role will generate a failure

response. Begin ignore section----- >

```

<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[Supplier AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="[Supplier Address Name]">
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Supplier Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>

```

```

<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[ShipTo AddressID from PO]">
<Name xml:lang="en-US">[ShipTo Name from PO]</Name>
<PostalAddress name="[Address Name from PO]">
<DeliverTo>[DeliverTo Name from PO]</DeliverTo>
<Street>[Street from PO]</Street>
<City>[City from PO]</City>
<State>[State from PO]</State>
<PostalCode>[Postal Code from PO]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >
<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>

```

```

<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<Comments xml:lang="en-US">[Line Level Comment]</Comments>
<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
<InvoiceDetailItem invoiceLineNumber="2" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>

```

```

<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<Extrinsic name="[Extrinsic Name]">[Line Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
</InvoiceDetailOrder>
<InvoiceDetailSummary>
<SubtotalAmount>
<Money currency="USD">[Subtotal]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Total Tax Amount]</Money>
<Description xml:lang="en-US">[Tax Description]</Description>
</Tax>
<SpecialHandlingAmount>
<Money currency="USD">[Total Handling Amount]</Money>
</SpecialHandlingAmount>
<ShippingAmount>
<Money currency="USD">[Total Shipping Amount]</Money>
</ShippingAmount>
<GrossAmount>
<Money currency="USD">[Gross Amount]</Money>
</GrossAmount>
<InvoiceDetailDiscount>
<Money currency="USD">[Discount Amount]</Money>
</InvoiceDetailDiscount>

```

```

<NetAmount>
<Money currency="USD">[Net Amount]</Money>
</NetAmount>

<DueAmount>
<Money currency="USD">[Amount Due]</Money>
</DueAmount>

<Extrinsic name="MiscFees">
<MiscFee name="MiscFee8">
<Money currency="USD">10.15</Money>
</MiscFee>
<MiscFee name="MiscFee7">
<Money currency="USD">20.98</Money>
</MiscFee>
<MiscFee name="MiscFee1">
<Money currency="USD">30.15</Money>
</MiscFee>
<MiscFee name="MiscFee3">
<Money currency="USD">40.98</Money>
</MiscFee>
<MiscFee name="MiscFee4">
<Money currency="USD">50.15</Money>
</MiscFee>
<MiscFee name="MiscFee9">
<Money currency="USD">60.98</Money>
</MiscFee>
<MiscFee name="MiscFee6">
<Money currency="USD">70.15</Money>
</MiscFee>
<MiscFee name="MiscFee2">

```

```

<Money currency="USD">80.98</Money>
</MiscFee>
<MiscFee name="MiscFee5">
<Money currency="USD">90.15</Money>
</MiscFee>
<MiscFee name="MiscFee10">
<Money currency="USD">100.98</Money>
</MiscFee>
</Extrinsic>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Credit, Most Common Structure)

The following sample illustrates the most commonly requested credit memo document structure. Not all clients accept credit memos electronically. Note that the only difference between a standard invoice and credit memo is the purpose attribute of "creditMemo".

Category: Individual

Level: Detail Line

Purpose: standard or credit memo

Additional Fees: Summary Level

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

```
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">
```

```
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payload ID]" xml:lang="en-US" version="1.2.011">
```

```
<Header>
```

```
<From>
```

```
<Credential domain="[NetworkID or DUNS]">
```

```
<Identity>[From Identity]</Identity>
```



```

</Credential>
</From>
<To>
<Credential domain="[NetworkID or DUNS]">
<Identity>[To Identity]</Identity>
</Credential>
</To>
<Sender>
<Credential domain="[NetworkID or DUNS]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request deploymentMode="[production or test]">
<InvoiceDetailRequest>
<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="creditMemo"
operation="new" invoiceDate="YYYY-MM-DD'T'HH:mm:ssZ">
<InvoiceDetailHeaderIndicator />
<InvoiceDetailLineIndicator />
<InvoicePartner>
<Contact role="billTo">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>

```

```

<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>
<Contact role="remitTo">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>

```

<!--JAGGAER ignores data within the InvoiceDetailShipping section. Suppliers have the option to omit the entire element when additional fees are provided at the summary level. If passed, the supplier must provide two contact roles. Providing only a single contact role will generate a failure response. Begin ignore section----- >

```

<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[Supplier AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="[Supplier Address Name]">
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>

```

```

<Email name="default">[Supplier Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[ShipTo AddressID from PO]">
<Name xml:lang="en-US">[ShipTo Name from PO]</Name>
<PostalAddress name="[Address Name from PO]">
<DeliverTo>[DeliverTo Name from PO]</DeliverTo>
<Street>[Street from PO]</Street>
<City>[City from PO]</City>
<State>[State from PO]</State>
<PostalCode>[Postal Code from PO]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>

```

```

<!-------End Ignore Section----- >
<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<Comments xml:lang="en-US">[Line Level Comment]</Comments>
<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
<InvoiceDetailItem invoiceLineNumber="2" quantity="[Quantity]">

```

```

<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<Extrinsic name="[Extrinsic Name]">[Line Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
</InvoiceDetailOrder>
<InvoiceDetailSummary>
<SubtotalAmount>
<Money currency="USD">[Subtotal]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Total Tax Amount]</Money>
<Description xml:lang="en-US">[Tax Description]</Description>
</Tax>
<SpecialHandlingAmount>
<Money currency="USD">[Total Handling Amount]</Money>
</SpecialHandlingAmount>
<ShippingAmount>
<Money currency="USD">[Total Shipping Amount]</Money>
</ShippingAmount>
<GrossAmount>
<Money currency="USD">[Gross Amount]</Money>

```

```

</GrossAmount>
<NetAmount>
<Money currency="USD">[Net Amount]</Money>
</NetAmount>
<DueAmount>
<Money currency="USD">[Amount Due]</Money>
</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Sample Message (Negative Invoice, Credit Alternative)

The following sample illustrates an alternative means to post a credit in situations where the supplier is unable to pass a value of "creditMemo" for the purpose attribute. Not all clients accept credit memos electronically. Note that all quantities are positive while unitPrices and totals are negative. Suppliers should not post both debit and credit amounts in any single document of this type.

Category: Individual

Level: Detail Line

Purpose: standard with intent to credit

Additional Fees: Summary Level

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

```
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">
```

```
<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payload ID]" xml:lang="en-US" version="1.2.011">
```

```
<Header>
```

```
<From>
```

```
<Credential domain="[NetworkID or DUNS]">
```

```
<Identity>[From Identity]</Identity>
```

```
</Credential>
```

```

</From>
<To>
<Credential domain="[NetworkID or DUNS]">
<Identity>[To Identity]</Identity>
</Credential>
</To>
<Sender>
<Credential domain="[NetworkID or DUNS]">
<Identity>[Sender Identity]</Identity>
<SharedSecret>[Shared Secret]</SharedSecret>
</Credential>
<UserAgent>[User Agent]</UserAgent>
</Sender>
</Header>
<Request deploymentMode="[production or test]">
<InvoiceDetailRequest>
<InvoiceDetailRequestHeader invoiceID="[Supplier Invoice Number]" purpose="standard"
operation="new" invoiceDate="YYYY-MM-DD'T'HH:mm:ssZ">
<InvoiceDetailHeaderIndicator />
<InvoiceDetailLineIndicator />
<InvoicePartner>
<Contact role="billTo">
<Name xml:lang="en-US">[Client Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>

```

```

</PostalAddress>
</Contact>
</InvoicePartner>
<InvoicePartner>
<Contact role="remitTo">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress>
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
</Contact>
</InvoicePartner>

```

<!--JAGGAER ignores data within the InvoiceDetailShipping section. Suppliers have the option to omit the entire element when additional fees are provided at the summary level. If passed, the supplier must provide two contact roles. Providing only a single contact role will generate a failure response. Begin ignore section----- >

```

<InvoiceDetailShipping>
<Contact role="shipFrom" addressID="[Supplier AddressID]">
<Name xml:lang="en-US">[Supplier Name]</Name>
<PostalAddress name="[Supplier Address Name]">
<Street>[Street]</Street>
<City>[City]</City>
<State>[State]</State>
<PostalCode>[Postal Code]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Supplier Email Address]</Email>

```



```

<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
<Contact role="shipTo" addressID="[ShipTo AddressID from PO]">
<Name xml:lang="en-US">[ShipTo Name from PO]</Name>
<PostalAddress name="[Address Name from PO]">
<DeliverTo>[DeliverTo Name from PO]</DeliverTo>
<Street>[Street from PO]</Street>
<City>[City from PO]</City>
<State>[State from PO]</State>
<PostalCode>[Postal Code from PO]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
<Email name="default">[Email Address]</Email>
<Phone name="work">
<TelephoneNumber>
<CountryCode isoCountryCode="US">1</CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
</Contact>
</InvoiceDetailShipping>
<!-------End Ignore Section----- >

```

```

<InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" />
<InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" />
<Comments xml:lang="en-US">[Header Level Comment]</Comments>
<Extrinsic name="[Extrinsic Name]">[Header Extrinsic Value]</Extrinsic>
</InvoiceDetailRequestHeader>
<InvoiceDetailOrder>
<InvoiceDetailOrderInfo>
<OrderReference orderID="[orderID from Original PO]" orderDate="[orderDate from Original PO]">
<DocumentReference payloadID="[payloadID from Original PO]" />
</OrderReference>
</InvoiceDetailOrderInfo>
<InvoiceDetailItem invoiceLineNumber="1" quantity="[Positive Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>
<UnitPrice>
<Money currency="USD">[Negative Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<Comments xml:lang="en-US">[Line Level Comment]</Comments>
<Extrinsic name="CFVALUE_[Client Custom Field Internal Name]">[Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
<InvoiceDetailItem invoiceLineNumber="2" quantity="[Positive Quantity]">
<UnitOfMeasure>[UOM]</UnitOfMeasure>

```

```

<UnitPrice>
<Money currency="USD">[Negative Unit Price]</Money>
</UnitPrice>
<InvoiceDetailItemReference lineNumber="[Line Number from Original PO]">
<ItemID>
<SupplierPartID>[Part Number]</SupplierPartID>
</ItemID>
<Description xml:lang="en-US">[Part Description]</Description>
</InvoiceDetailItemReference>
<Extrinsic name="[Extrinsic Name]">[Line Extrinsic Value]</Extrinsic>
</InvoiceDetailItem>
</InvoiceDetailOrder>
<InvoiceDetailSummary>
<SubtotalAmount>
<Money currency="USD">[Negative Subtotal]</Money>
</SubtotalAmount>
<Tax>
<Money currency="USD">[Negative Total Tax Amount]</Money>
<Description xml:lang="en-US">[Tax Description]</Description>
</Tax>
<SpecialHandlingAmount>
<Money currency="USD">[Negative Total Handling Amount]</Money>
</SpecialHandlingAmount>
<ShippingAmount>
<Money currency="USD">[Negative Total Shipping Amount]</Money>
</ShippingAmount>
<GrossAmount>
<Money currency="USD">[Negative Gross Amount]</Money>
</GrossAmount>

```

```

<NetAmount>
<Money currency="USD">[Negative Net Amount]</Money>
</NetAmount>
<DueAmount>
<Money currency="USD">[Negative Amount Due]</Money>
</DueAmount>
</InvoiceDetailSummary>
</InvoiceDetailRequest>
</Request>
</cXML>

```

Message Elements - Invoice Detail Request

| Element/Attribute | Status | Description and Support Detail |
|-------------------|-----------|---|
| <Header> | | |
| From Domain | Mandatory | DUNS or NetworkID. This element is required, but is not used in message authentication for this integration |
| From Identity | Mandatory | Supplier DUNS or NetworkID. This element is required, but is not used in message authentication for this integration. Typically the same value as Sender identity. |
| To Domain | Mandatory | DUNS or NetworkID. This element is required, but is not used in message authentication for this integration. |
| To Identity | Mandatory | Client DUNS or NetworkID. This element is required, and is used in message authentication. The client identifying value should be the same value used in other integrations between the client and supplier. |
| Sender Domain | Mandatory | DUNS or NetworkID. This element is required but is not used in message authentication for this integration; however, should be the same value as used by the supplier in other invoice integrations with JAGGAER. |
| Sender Identity | Mandatory | Supplier DUNS or NetworkID. This element is required and authenticated, and should be the same value as used by the supplier in other invoice integrations with JAGGAER. |

| Element/Attribute | Status | Description and Support Detail |
|---|-----------------|---|
| SharedSecret | Mandatory | Supplier specified SharedSecret or password. Used in message authentication. |
| UserAgent | Optional | Supplier's UserAgent, typically represented by the sending application and version. This element is required but is not used in message authentication. |
| Request deploymentMode | Optional | Ignored by JAGGAER as the post URL varies from the test to production environment. If passed, accepted values are either "test" or "production". Values are case sensitive and must match syntax exactly to avoid failure response. |
| <InvoiceDetailRequestHeader> | | |
| invoiceID | Client Required | Represents the supplier invoice number. This value is used as criteria in the Document Acceptance Process in cases where the client configuration is set to check for duplicates (typical). Suppliers who support partial shipments must use a unique invoiceID each time unless the client will accept duplicate supplier invoice numbers (atypical). Recommended character length is 15 or less.

Note: A null value for invoiceID may pass the initial JAGGAER acceptance process, but is likely to cause an exception in the client ERP system upon export. |
| purpose | Mandatory | JAGGAER currently supports "standard" or "creditMemo".

Note: JAGGAER does not accept a purpose of "debitMemo" and if passed, will generate an error response. An invoice with a purpose of "standard" that contains negative unitPrices will be treated as a creditMemo. Suppliers must not provide both positive and negative unitPrices in the same settlement document. |
| operation | Mandatory | The single value option of "new" must be passed to avoid failure response.

Invoice status updates, cancellations, and deletions are not supported. |
| invoiceDate | Client Required | Represents the supplier value for the invoice transaction date. Date format is: YYYY-MM-DDT'HH:mm:ssZ or YYYY-MM-DD. Invoices posted with a null date value or in an unsupported date format will be assigned an invoice date of the current day (date posted).

Note: This date will be utilized in determining the |

| Element/Attribute | Status | Description and Support Detail |
|--|-----------------|--|
| | | invoice due date, based upon the terms as sent by the supplier in the settlement document or based upon terms set in the JAGGAER application or client ERP. |
| invoiceOrigin
isInformationOnly | Not Supported | These element attributes are not currently supported by JAGGAER and if passed will be ignored. |
| <InvoiceDetailLineIndicator> | | |
| isTaxInLine
isSpecialHandlingInLine
isShippingInLine
isDiscountInLine
isAccountingInLine | Mandatory | <p>Element is mandatory. The attributes must be provided when additional fees will be passed at the line level and where tax, handling, shipping, and discounts apply. Accepted value is "yes". Omit any single attribute if the respective fee will be provided at the summary level.</p> <p>If all additional fees will be provided at the summary level, the element must be passed without any attributes as:
<InvoiceDetailLineIndicator /></p> <p>Note: JAGGAER will look first at the line for those fees with a line indicator value of "yes", then the summary level for all other additional fees. If fees are provided at both the line and summary levels and the indicator attribute(s) are omitted, those at the summary level will take precedence.</p> |
| <InvoicePartner> | | |
| Contact role
remitTo
soldTo
billTo
issuerOfInvoice | Client required | <p>Either the entire section can be omitted, or at least one contact role should be provided.
<InvoicePartner /> is not acceptable. JAGGAER ignores all invoice partner contact roles except for remitTo and billTo. Typically remitTo is included with one other; however, suppliers should review client requirements.</p> |
| Name xml:lang
PostalAddress | Client required | <p>If contact role(s) are passed, the corresponding postal address section for each role must be complete to avoid failure response:</p> <pre> <Name xml:lang="en-US">[value]</Name> <PostalAddress name="[value]"> <DeliverTo>[value]</DeliverTo> <Street>[value]</Street> <City>[value]</City> <State>[value]</State> <PostalCode>[value]</PostalCode> <Country isoCountryCode="US">United </pre> |

| Element/Attribute | Status | Description and Support Detail |
|--|-------------------------|--|
| | | <p>States</Country></p> <p></PostalAddress></p> <p>Note: <PostalAddress /> is not acceptable.</p> |
| Email name
Phone name
TelephoneNumber
Fax name
URL | Optional | <p>If contact role(s) are passed, the supplier may elect to include contact email, telephone, fax, and site URL information. This data will be ignored by JAGGAER.</p> <p><Email name="[email name]">[Email Address]</Email></p> <p><Phone name="[phone name]"></p> <p><TelephoneNumber></p> <p><CountryCode
isoCountryCode="US">1</CountryCode></p> <p><AreaOrCityCode>[Area Code]</AreaOrCityCode></p> <p><Number>[Phone Number]</Number></p> <p></TelephoneNumber></p> <p></Phone></p> <p><Fax name="[phone name]"></p> <p><TelephoneNumber></p> <p><CountryCode
isoCountryCode="US">1</CountryCode></p> <p><AreaOrCityCode>[Area Code]</AreaOrCityCode></p> <p><Number>[Phone Number]</Number></p> <p></TelephoneNumber></p> <p><URL>[Web Address]</URL></p> |
| <IdReference> | | |
| identifier
domain
Creator | Not Supported | The IDReference elements and corresponding attributes not currently supported by JAGGAER. |
| <ItemDetailShipping> | | |
| shippingDate | Not Supported | The shippingDate attribute is not currently supported by JAGGAER, and if passed, will be ignored. |
| Contact role
shipFrom
shipTo | Conditional Requirement | The InvoiceDetailShipping section is ignored by JAGGAER in all cases and can optionally be omitted in its entirety when additional fees are provided only at the summary level. If additional fees will be |

| Element/Attribute | Status | Description and Support Detail |
|--|-------------------------|--|
| carrierCorporate | | provided at the line level, the InvoiceDetailShipping section is required per cXML standards. When the section is passed, it must include at least two contact roles. Providing only a single contact role will generate a failure response.
Note: <InvoiceDetailShipping/> is not acceptable. |
| Name xml:lang
PostalAddress | Conditional Requirement | If the InvoiceDetailShipping section and contact role(s) are passed, the corresponding postal address section for each role must be complete to avoid failure response.

<Name xml:lang="en-US">[value]</Name>
<PostalAddress name="[value]">
<DeliverTo>[value]</DeliverTo>
<Street>[value]</Street>
<City>[value]</City>
<State>[value]</State>
<PostalCode>[value]</PostalCode>
<Country isoCountryCode="US">United States</Country>
</PostalAddress>
Note: <PostalAddress /> is not acceptable. |
| Email name
Phone name
TelephoneNumber
Fax name
URL | Optional | If the InvoiceDetailShipping section and contact role(s) are passed, the supplier may elect to include contact email, telephone, fax, and site URL information. This data will be ignored by JAGGAER.

<Email name="[email name]">[Email Address]</Email>
<Phone name="[phone name]">
<TelephoneNumber>
<CountryCode
isoCountryCode="US">1 </CountryCode>
<AreaOrCityCode>[Area Code]</AreaOrCityCode>
<Number>[Phone Number]</Number>
</TelephoneNumber>
</Phone>
<Fax name="[phone name]">
<TelephoneNumber>
<CountryCode |

| Element/Attribute | Status | Description and Support Detail |
|---|-----------------|---|
| | | <p>isoCountryCode="US">1</CountryCode></p> <p><AreaOrCityCode>[Area Code]</AreaOrCityCode></p> <p><Number>[Phone Number]</Number></p> <p></TelephoneNumber></p> <p><URL>[Web Address]</URL></p> |
| CarrierIdentifier
ShipmentIdentifier | Not Supported | These two elements are not currently supported by JAGGAER in a settlement document, and if passed, will be ignored. |
| <InvoiceDetailPaymentTerm> | | |
| payInNumberOfDays | Client Required | <p>This attribute is required only when the supplier will send payment terms. The attribute value represents the number of days the invoice can be paid relative to the discount percentageRate.</p> <p>For example, the following represents a 5% discount if paid within 20 days and no discount if paid within net terms of 30 days:</p> <pre><InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" /> <InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" /></pre> <p>The payment term with no discount rate and the greatest number of days will be recorded as the net terms. The element with the smallest number of days and discount rate greater than zero will be recorded as the discount days/amount. JAGGAER supports a single discount only.</p> <p>Alternative: Suppliers that do not offer payment term discounts can alternatively provide only the following:</p> <pre><PaymentTerm payInNumberOfDays="30" /></pre> <p>Note: Clients may elect to ignore payment terms as sent by the supplier in favor of terms set in the JAGGAER application or the client ERP system per the contract.</p> |
| percentageRate | Client Required | This attribute is required only when the supplier will send payment terms and discounts apply. The attribute value represents the discount |

| Element/Attribute | Status | Description and Support Detail |
|-----------------------|-----------------|--|
| | | <p>percentage relative to the number of days. Whole numbers (e.g. "0" for no discount, "5" for 5% discount)</p> <p>For example, the following represents a 5% discount if paid within 20 days and no discount if paid within net terms of 30 days:</p> <pre><InvoiceDetailPaymentTerm payInNumberOfDays="20" percentageRate="5" /> <InvoiceDetailPaymentTerm payInNumberOfDays="30" percentageRate="0" /></pre> <p>The payment term with no discount rate and the greatest number of days will be recorded as the net terms. The element with the smallest number of days and discount rate greater than zero will be recorded as the discount days/amount. JAGGAER supports a single discount only.</p> <p>Alternative: Suppliers that do not offer payment term discounts can alternatively provide only the following:</p> <pre><PaymentTerm payInNumberOfDays="30" /></pre> <p>Note: Clients may elect to ignore payment terms as sent by the supplier in favor of terms set in the JAGGAER application or the client ERP system per the contract.</p> |
| <Period> | | |
| startDate
endDate | Not Supported | The Period element and corresponding attributes are not currently supported by JAGGAER. |
| Comments | Optional | A comment provided at the header level applies to the entire settlement document. All header comments and standard extrinsics are mapped to the header level settlement document notes section within the JAGGAER application and combined should not exceed 100 characters to avoid truncation. |
| Extrinsic name | Client Required | An extrinsic provided at the header level applies to the entire settlement document. All header comments and standard extrinsics are mapped to the settlement document notes section within the JAGGAER application and combined should not exceed 100 characters to avoid truncation. |

| Element/Attribute | Status | Description and Support Detail |
|---|-------------------------|--|
| | | <p>Standard Extrinsic: < Extrinsic name="name">value</Extrinsic></p> <p>Custom Extrinsic: < Extrinsic name="CFValue_name">value</Extrinsic></p> <p>For more detailed information about extrinsic definition and usage, please see the Settlement Extrinsic section.</p> |
| <InvoiceDetailOrderInfo> | | |
| <OrderReference> | | |
| orderID | Conditional Requirement | <p>Attribute value represents the client purchase order number as sent within the <OrderRequestHeader> element of the corresponding OrderRequestMessage. This attribute is required along with orderDate in cases where the supplier is unable to return the payloadID value.</p> <p>This value is evaluated in the Document Acceptance Process. If JAGGAER is unable to match on both orderID and orderDate, JAGGAER will attempt to match on payloadID.</p> |
| orderDate | Conditional Requirement | <p>Attribute value represents the client purchase order date as sent within the <OrderRequestHeader> element of the corresponding OrderRequestMessage. This attribute is required along with orderID in cases where the supplier is unable to return the payloadID value. Acceptable date formats are: YYYY-MM-DD'T'HH:mm:ssZ or YYYY-MM-DD.</p> <p>This value is evaluated in the Document Acceptance Process. If JAGGAER is unable to match on both orderID and orderDate, JAGGAER will attempt to match on payloadID.</p> |
| <DocumentReference> | | |
| payloadID | Conditional Requirement | <p>Attribute value represents the payloadID as sent within the opening <cXML> element of the corresponding OrderRequestMessage. This attribute is required in cases where the supplier is unable to return both the orderID and orderDate . JAGGAER prefers this value be passed where possible.</p> <p>This value is evaluated in the Document Acceptance Process. If JAGGAER is unable to match payloadID, JAGGAER will attempt to match on both orderID and orderDate .</p> |
| <MasterAgreementReference> | | |

| Element/Attribute | Status | Description and Support Detail |
|--|------------------|--|
| <DocumentReference
payloadID=> | Not
Supported | This element and corresponding attribute are not currently supported by JAGGAER, and if passed, will be ignored. |
| <SupplierOrderInfo> | | |
| orderID | Not
Supported | This element and corresponding attribute are not currently supported by JAGGAER, and if passed, will be ignored. |
| <InvoiceDetailServiceItem> - service level invoices are not currently supported by JAGGAER and must not be passed | | |
| <InvoiceDetailItem> | | |
| invoiceLineNumber | Mandatory | This attribute value represents the supplier invoice line number. Line numbers should be presented in sequential order, starting at "0" or "01". |
| Quantity | Mandatory | <p>This attribute value represents the number of referenced items shipped, and must be a positive integer. In some rare cases, decimal quantities may be passed where applicable. Suppliers that support partial shipments may invoice at a lesser quantity than that on the original purchase order document.</p> <p>Quantity is not used as match criteria in the Document Acceptance Process. An invoice document with quantities higher than that on the original purchase order document will be accepted by JAGGAER, but will move to a client exception queue for resolution between the client and supplier.</p> |
| UnitOfMeasure | Mandatory | <p>This element value represents the item unit of measure. This value may be used as match criteria in the Document Acceptance Process in cases where the client elects to match on UOM (atypical).</p> <p>Unmapped UOM values will present match exceptions and error responses for those clients that elect to match on UOM. See the UOM Mapping section for more information.</p> |
| <UnitPrice> | | |
| Money currency | Mandatory | <p>Default currency is "USD". Value represents the UnitPrice in relation to the provided unit of measure.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard or</p> |

| Element/Attribute | Status | Description and Support Detail |
|---|-----------|--|
| | | <p>creditMemo). Suppliers should not pass both positive and negative unitPrices in any single settlement document – the document should either be entirely a debit or credit.</p> <p>See the Numerical Formats and Supported Currencies sections within the Data Standards chapter for more information.</p> |
| <InvoiceDetailItemReference> | | |
| lineNumber | Mandatory | <p>This attribute value represents the corresponding item line number passed to the supplier on the original OrderRequestMessage within the <ItemOut> element.</p> <p>This value is used as match criteria in the Document Acceptance Process. Suppliers that are unable to maintain and return the corresponding purchase order line number will be unable to successfully route settlement documents via cXML unless the client is willing to accept header level invoices (atypical).</p> |
| <ItemID>
SupplierPartID | Mandatory | <p>Represents the part number shipped by the supplier. Typically matches the value and syntax listed in the supplier catalog and on the corresponding OrderRequest message; however, may represent a substituted value in cases where the client configuration supports it.</p> <p>This value is used as match criteria in the Document Acceptance Process.</p> |
| Description xml:lang | Optional | <p>Element value represents the item description as listed in the catalog, up to 256 characters.</p> <p>The default language attribute value is "en-US".</p> |
| SerialNumber | Optional | <p>Represents the item serial number. Multiple serial numbers per line can be passed, up to 100 characters. Please see below for an example:</p> <pre> <SerialNumbers> <SerialNumber>2843356235662</SerialNumber> <SerialNumber>5123623572357257</SerialNumber> <SerialNumber>28329922929</SerialNumber> <SerialNumber>235729292</SerialNumber> <SerialNumber>2356283276237</SerialNumber> </SerialNumbers> </pre> |

| Element/Attribute | Status | Description and Support Detail |
|---|-----------------|--|
| <Line Level Additional Fees> | | |
| SubtotalAmount | Client Required | <p>Represents the line level subtotal and is passed in this location only when suppliers are providing additional fees at the line level. This element is required per cXML standards; however, JAGGAER will ignore the values passed by the supplier.</p> <p>JAGGAER calculates the extended price and invoice total using the UnitPrice and quantity elements for each line rather than parsing the sum amount(s) passed by the supplier.</p> <p>Invoice Total Calculation: =SUM(Line UnitPriceXQuantity) – Discount + Shipping + Handling + Tax</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| Tax
Description xml:lang | Client Required | <p>Represents the line level tax amount for the item and is only passed in this location when the supplier is providing additional fees at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes". Value can be passed as "0.00" in cases where the client is exempt from tax.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard, or creditMemo). Suppliers should not pass both positive and negative values in any single settlement document – the document should either be entirely a debit or credit.</p> <p>If tax is listed at both the line and summary levels and the isTaxInLine attribute is not present in the <InvoiceDetailLineIndicator> element, JAGGAER will look to the summary level for tax fees.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> <p>Note: Tax Detail (purpose, category, percentageRate, location) is not currently supported by JAGGAER, and if passed, will be ignored.</p> |
| InvoiceDetailLineSpecialHandling | Client Required | <p>Represents the line level handling amount for the item and is only passed in this location when the supplier is providing additional fees at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes". Value can be passed as "0.00" in cases where handling charges do not</p> |

| Element/Attribute | Status | Description and Support Detail |
|---------------------------|-----------------|---|
| | | <p>apply.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard, or creditMemo). Suppliers should not pass both positive and negative values in any single settlement document – the document should either be entirely a debit or credit.</p> <p>If handling is listed at both the line and summary levels and the isSpecialHandlingInLine attribute is not present in the <InvoiceDetailLineIndicator> element, JAGGAER will look to the summary level for handling fees.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| InvoiceDetailLineShipping | Client Required | <p>Represents the line level shipping amount for the item and is only passed in this location when the supplier is providing additional fees at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes". Value can be passed as "0.00" in cases where shipping charges do not apply.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard, or creditMemo). Suppliers should not pass both positive and negative values in any single settlement document – the document should either be entirely a debit or credit.</p> <p>If shipping is listed at both the line and summary levels and the isShippingInLine attribute is not present in the <InvoiceDetailLineIndicator> element, JAGGAER will look to the summary level for shipping fees.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| GrossAmount
NetAmount | Client Required | <p>Represents the line level gross and net amounts for the item and is only passed in this location when the supplier is providing additional fees at the line level. These elements are required per cXML standards; however, JAGGAER will ignore the values passed by the supplier.</p> <p>JAGGAER calculates the extended price and invoice total using the UnitPrice and quantity elements for each line rather than parsing the sum amount(s) passed by the supplier.</p> |

| Element/Attribute | Status | Description and Support Detail |
|--|---------------|---|
| | | <p>Invoice Total Calculation: =SUM(Line UnitPriceXQuantity) – Discount + Shipping + Handling + Tax</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| InvoiceDetailDiscount | Optional | <p>Represents the line level discount amount for the item and is only passed in this location when the supplier is providing additional fees at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes". Value can be passed as "0.00" or omitted in cases where a discount does not apply.</p> <p>If the discount is listed at both the line and summary levels and the isDiscountInLine attribute is not present in the <InvoiceDetailLineIndicator> element, JAGGAER will look to the summary level for the discount.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| <Distribution> | | |
| Accounting name
AccountingSegment id
Name xml:lang
Description xml:lang
Charge | Not Supported | <p>Account codes and distribution are pulled from the corresponding matched purchase order upon settlement document receipt; therefore, distribution elements and corresponding attributes should not be passed on the settlement document. If the client has a specific requirement, JAGGAER will coordinate with the supplier to evaluate possibilities.</p> <p>In the rare case of non-PO invoicing, account codes can be passed as Custom Extrinsics. For more information about extrinsics, please see the Settlement Extrinsics section for more information.</p> |
| Comments | Optional | A comment provided at the line level applies only to the corresponding item. All line comments and standard extrinsics are mapped to the line level settlement document notes section within the JAGGAER application and combined should not exceed 100 characters to avoid truncation. |
| Extrinsic name | Optional | An extrinsic provided at the line level applies only to the corresponding item. All line comments and standard extrinsics are mapped to the line level settlement document notes section within the JAGGAER application and combined should not |

| Element/Attribute | Status | Description and Support Detail |
|-------------------------------------|-----------------|--|
| | | <p>exceed 100 characters to avoid truncation.</p> <p>Standard Extrinsic: <Extrinsic name="name">value</Extrinsic></p> <p>Custom Extrinsic: <Extrinsic name="CFValue_name">value</Extrinsic></p> <p>For more detailed information about extrinsic definition and usage, please see the Settlement Extrinsics section included within this chapter.</p> |
| <InvoiceDetailSummary> | | |
| SubtotalAmount | Mandatory | <p>This element is required per cXML standards; however, JAGGAER will ignore the values passed by the supplier. JAGGAER calculates the extended price and invoice total using the UnitPrice and quantity elements for each line rather than parsing the sum amount(s) passed by the supplier.</p> <p>Invoice Total Calculation: =SUM(Line UnitPriceXQuantity) – Discount + Shipping + Handling + Tax</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| Tax
Description xml:lang | Client Required | <p>Represents the total tax amount for the entire settlement document and will be applied as a weighted average across all line items present on the invoice/credit document. Value can be passed as "0.00" in cases where the client is exempt from tax.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard, or creditMemo). Suppliers should not pass both positive and negative values in any single settlement document – the document should either be entirely a debit or credit.</p> <p>If tax is also provided at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes", the amount listed in the summary will be ignored.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> <p>Note: Tax Detail (purpose, category, percentageRate, location) is not currently supported by JAGGAER, and if passed, will be ignored.</p> |

| Element/Attribute | Status | Description and Support Detail |
|---------------------------------------|-----------------|---|
| SpecialHandlingAmount | Client Required | <p>Represents the total handling amount for the entire settlement document and will be applied as a weighted average across all line items present on the invoice/credit document. Value can be passed as "0.00" in cases where handling charges do not apply.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard, or creditMemo). Suppliers should not pass both positive and negative values in any single settlement document – the document should either be entirely a debit or credit.</p> <p>If handling is also provided at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes", the amount listed in the summary will be ignored.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| ShippingAmount | Client Required | <p>Represents the total shipping amount for the entire settlement document and will be applied as a weighted average across all line items present on the invoice/credit document. Value can be passed as "0.00" in cases where shipping charges do not apply.</p> <p>A negative value can be passed where there is intent to credit regardless of value passed in the document purpose attribute (standard, or creditMemo). Suppliers should not pass both positive and negative values in any single settlement document – the document should either be entirely a debit or credit.</p> <p>If shipping is also provided at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes", the amount listed in the summary will be ignored.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| GrossAmount
NetAmount
DueAmount | Mandatory | <p>These elements are required per cXML standards; however, JAGGAER will ignore the values passed by the supplier.</p> <p>JAGGAER calculates the extended price and invoice total using the UnitPrice and quantity elements for each line rather than parsing the sum amount(s) passed by the supplier.</p> |

| Element/Attribute | Status | Description and Support Detail |
|-----------------------|----------|--|
| | | <p>Invoice Total Calculation: =SUM(Line UnitPriceXQuantity) – Discount + Shipping + Handling + Tax</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |
| InvoiceDetailDiscount | Optional | <p>Represents the total discount amount for the entire settlement document and will be applied as a weighted average across all line items present on the invoice/credit document. Value can be passed as "0.00" or omitted in cases where a discount does not apply.</p> <p>If a discount is also provided at the line level and the <InvoiceDetailLineIndicator> attribute value is set to "yes", the amount listed in the summary will be ignored.</p> <p>See the Numerical Formats and Supported Currencies sections for more information.</p> |

Response to InvoiceDetailRequest Message

JAGGAER will route an asynchronous response to the supplier via the same request path. A settlement document that does not meet match criteria and is initially rejected by the JAGGAER system is not imported to the application; and is therefore not seen or reviewed by the client. As a result, the supplier is required to monitor invoice responses and have a process in place to handle failures. Suppliers unable to monitor or take action on invoice response messages from JAGGAER will not qualify for invoice posting via cXML.

For the asynchronous process, when a cXML invoice is submitted, JAGGAER will validate only the header information of the submittal and respond with a "201" "Accepted" status code or "401" or "406" error status code. No other immediate status updates will be provided. Once the invoice has been processed, it will be available for viewing in the supplier portal and will be added to the daily status email. Please see the Invoice/Credit Memo Response Code appendix for further examples and integration troubleshooting.

The following is an example of a JAGGAER response to an InvoiceDetailRequest message.

```
<?xml version="1.0" ?>

<!DOCTYPE cXML SYSTEM "https://solutions.scquest.com/app_
docs/dtd/cxml/InvoiceDetail.dtd">

<cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]">

<Response>
```

```
<Status code="201" text="Accepted">Success</Status>
```

```
</Response>
```

```
</cXML>
```

Appendix

JAGGAER Response Codes

For those integration points where a message/document is posted via HTTP (over SSL, port 443) to JAGGAER, the supplier will receive a synchronous response to the request. These integration points include ConfirmationRequest, ShipNoticeRequest, and cXML Invoice/Credit Memos. Suppliers must monitor JAGGAER response messages and take the appropriate action before attempting to repost. The response message itself is the supplier notification of success/failure – no further notification will be provided.

Code Definitions

JAGGAER generally adheres to the following definitions by response code series:

- 100 Series: A 100 series response is not currently in use by JAGGAER.
- 200 Series: A 200 series response represents a successful post attempt and the requested action was received, accepted, and processed. No further supplier action is required.
- 300 Series: A 300 series response is used by JAGGAER on a limited basis. Traditionally this series indicates the requestor must take further action on the request; however, JAGGAER may provide this response where no further supplier action is required. For this series, the tables below will indicate any specific action the supplier must take, if any, where applicable.
- 400 Series: A 400 series response represents a failed post attempt; and while the requested action was received, the message is not acceptable and will not be processed. The response text will indicate the nature of the error, and the supplier must revise the message before reposting.
- 500 Series: A 500 series response represents an internal server error and as a result, the requested action cannot be performed. The response text is general in nature, and the supplier should review/validate the message content before attempting to repost. If the supplier receives two consecutive responses in this series to the same request, please contact JAGGAER Support at support@JAGGAER.com for further help.

Codes by Integration Type

The tables within this appendix are separated by integration point; and while not fully inclusive, represent a significant sampling of the JAGGAER response messages. Suppliers can use the response code listing for both informational and troubleshooting purposes.

NOTE: As JAGGAER functionality is introduced or changed, response messages may change without notice.

Invoices/Credit Memos

| Invoice/Credit Memo Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| 200 | <p>Success, No Exceptions.
Invoice accepted. Success.</p> <pre> <?xml version="1.0" ?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'TH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="200" text="OK">Success</Status> </Response> </cXML> </pre> |
| 201 | <p>Success, with exception: Payment Terms Ignored
The client has elected to ignore the payment terms as sent by the supplier in favor of terms set on the PO or in the client ERP.</p> <pre> <?xml version="1.0" ?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'TH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="201" text="OK">Completed import with following warnings: 1. Invoice no [InvoiceNumber]: The payment terms were not used to create the invoice. The customer has opted to assign the payment terms from the PO or vendor master.</Status> </Response> </cXML> </pre> |
| 201 | <p>Success with exception: Remittance Address Ignored
The client has elected to ignore the remittance address as sent by the supplier in favor of the address stored in the client ERP.</p> <pre> <?xml version="1.0" ?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'TH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="201" text="OK">Completed import with following warnings: 1. Invoice no [InvoiceNumber]: The remit to address was not used to create the invoice. The customer has opted to assign the remit to address from their vendor master.</Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Invalid Quantity
The sender has passed a quantity value that is either negative or contains alpha characters. If the supplier is attempting to post a creditMemo, please see the Invoices & Credit Memos chapter within this document.</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="400" text="Bad Request">Unable to continue process OrderID [orderID Number]. Error found in OrderID [orderID Number] - Quantity is missing or invalid (non- numeric characters, negative, etc)</Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Invoice Purpose Not Supported</p> <p>The sender has sent an invoice purpose that is not supported.
 (<InvoiceDetailRequestHeader invoiceID="[InvoiceID]" purpose="debitMemo"
 operation="new" invoiceDate="[Invoice Date]">).
 Acceptable purposes are: "standard" or "creditMemo". The purpose "debitMemo" is not
 supported by JAGGAER.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="400" text="Bad Request">Invoice type: debitMemo is not handled by the invoice import application.</Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Invoice Operation Not Supported</p> <p>The sender has sent an invoice operation type that is not supported by JAGGAER.
 (<InvoiceDetailRequestHeader invoiceID="[InvoiceID]" purpose="standard"
 operation="delete" invoiceDate="[Invoice Date]">).
 The only accepted operation type is "new". Operation type "delete" is not supported by
 JAGGAER.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="400" text="Bad Request">Operation type: delete is not handled by the invoice import application.</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Invalid Authentication Credentials</p> <p>One or more credential elements are incorrect.</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="401" text="Unauthorized">Unable to determine identity for the given SenderIdentity, ToIdentity and Shared Secret.</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Missing Authentication Credential (Sender Identity)
 The Sender Identity value is absent. The Sender Identity cannot be blank or empty.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="401" text="Unauthorized">Sender Identity cannot be blank</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Missing Authentication Credential (To Identity)
 The To Identity value is absent. The To Identity cannot be blank or empty.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="401" text="Unauthorized">To Identity cannot be blank</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Missing Authentication Credential (Shared Secret)
 The shared secret value is absent. The shared secret cannot be blank or empty.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="401" text="Unauthorized">Sender Shared Secret cannot be blank</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Invoicing Turned Off for Supplier
 The client organization is not set to accept electronic invoices from the sender.</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.scquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="401" text="Unauthorized">The org has TURNED OFF the Receiving Invoice ability.</Status> </Response> </cXML> </pre> |
| 404 | <p>HTTP Response: Invalid URL</p> <p>The sender is posting to the incorrect URL. Repost to either accordingly:
 Test: https://usertest-messages.scquest.com/apps/Router/CXMLInvoiceImport
 Prod: https://integrations.scquest.com/apps/Router/CXMLInvoiceImport</p> <pre> 404: Invalid URL at com.scquest.components.http.AbstractHttpConnection.executeWithoutRelease(Abstract HttpConnection.java:176)<Status code="401" text="Unauthorized">The org has TURNED OFF the Receiving Invoice ability.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incorrect URL</p> <p>The sender has posted the invoice message to the JAGGAER ConfirmationRequest or ShipNoticeRequest in error. Repost to either accordingly:
 Test: https://usertest-messages.scquest.com/apps/Router/CXMLInvoiceImport
 Prod: https://integrations.scquest.com/apps/Router/CXMLInvoiceImport</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">Element type "InvoiceDetailRequest" must be declared.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Improperly Formatted Message</p> <p>The message is improperly formatted.
 Test: https://usertest-messages.scquest.com/apps/Router/CXMLInvoiceImport
 Prod: https://integrations.scquest.com/apps/Router/CXMLInvoiceImport</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.scquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Message Parse Failure</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Invalid Invoice Purpose</p> <p>The sender has sent an invalid invoice purpose. (<InvoiceDetailRequestHeader invoiceID="</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| | <p>[InvoiceID]" purpose="[INVALID VALUE]" operation="new" invoiceDate="[Invoice Date]">).</p> <p>Acceptable purposes are: "standard" or "creditMemo". The purpose "debitMemo" is not supported by JAGGAER.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Attribute "purpose" with value "[Supplier Value]" must have a value from the list "standard creditMemo debitMemo ".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Invalid Invoice Operation</p> <p>The sender has sent an invalid invoice operation type. (<InvoiceDetailRequestHeader invoiceID="[InvoiceID]" purpose="standard" operation="[INVALID VALUE]" invoiceDate="[Invoice Date]">).</p> <p>The only accepted operation type is "new". Operation type "delete" is not supported by JAGGAER.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Attribute "operation" with value "[Supplier Value]" must have a value from the list "new delete ".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Element(s) Not Properly Terminated</p> <p>One or more message element tags have not been properly terminated (e.g. <Identity> [Identity]</Identity>).</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">The end-tag for element type "[Element Name]" must end with a '&gt;' delimiter.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Element(s) Not Properly Opened or Terminated</p> <p>One or more message element tags have not been properly opened or terminated (e.g. <Header> open tag is present; however, </Header> is nabsent or vice versa.</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">The element type "[Element Name]" must be terminated by the matching end-tag "&lt;/[Element Name]&gt;".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Duplicate Supplier Invoice Number
 The client has elected to prevent duplicate supplier invoice numbers.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">This supplier invoice id already exists in system. Duplicates not allowed.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: No Match Found (PO Number)
 The system was unable to find a PO match based upon data sent by supplier. The client does not accept non-po invoices. The PO Number (orderId) match is not case sensitive with regards to any alpha characters that may be present in the orderId number.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Unable to find any match with OrderID = [OrderID Number], payloadID = [PayloadID Number] and CreateDate = [OrderDate]</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: No Match Found (Part Number)
 The system was unable to find a part number match based upon data sent by supplier. The client does not accept part substitutes. The part number match is case sensitive and must exactly match the syntax of the value that was passed on the PO (including any delimiters such as dashes), unless JAGGAER is directed by the supplier to ignore case sensitivity and/or strip delimiters on the inbound cXML invoice document.</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Unable to import orderID [OrderID Number]. No match found on Item SupplierPartID [Supplier Part Number] of Invoice Line Number [Invoice Line Number].</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: No Match Found (PO Line Number)</p> <p>The invoice references a PO line number that was not present on the original PO. The client does not accept non-po line items. The line number match will ignore any leading zeros (e.g. "01" vs. "1").</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Unable to create invoice line based on the PO Line number provided. OrderID is [OrderID Number]. Invoice Line Number is [Invoice Line Number], PO Line Number is [PO Line Number]</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: No Match Found (UOM)</p> <p>The invoice references an item UOM value that varies from the value on the original PO. The client has elected to match on UOM. The UOM match is case sensitive.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Unable to import orderID [OrderID Number]. No match found on Item UnitOfMeasure [Supplier UOM Value] of Invoice Line Number [Invoice Line Number].</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Invalid DeploymentMode</p> <p>The deploymentMode on the invoice is incorrect. The deploymentMode attribute is case sensitive and has only two options: "test" or "production".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Attribute "deploymentMode" with value "[Value]" must have a value from the list "production test".</Status> </Response> </cXML> </pre> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| 406 | <p>Unacceptable: Incomplete/Empty Postal Address</p> <p>The postal address is either incomplete or empty in either the InvoicePartner or InvoiceDetailShipping section. <PostalAddress/> is not acceptable.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">The content of element type "PostalAddress" is incomplete, it must match "(DeliverTo*,Street+,City,State?,PostalCode?,Country)".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incomplete Invoice Partner</p> <p>The <InvoicePartner> section is either incomplete or empty. Either the entire section can be removed or at least one complete contact role should be provided. <InvoicePartner /> is not acceptable. Typically "remitTo" is included with one of the following: "soldTo", "billTo", "issuerOfInvoice"; however, suppliers should review client requirements. JAGGAER ignores all InvoicePartner information except for "remitTo" and "billTo".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">The content of element type "InvoicePartner" is incomplete, it must match "(Contact,IdReference*)".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incomplete InvoiceDetailShipping</p> <p>The <InvoiceDetailShipping> section is incomplete. If contact roles are passed, at least two must be present (e.g. "ShipTo" & "shipFrom"; otherwise the entire element can be removed. <InvoiceDetailShipping /> is not acceptable.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">The content of element type "InvoiceDetailShipping" is incomplete, it must match "(Contact,Contact+,(CarrierIdentifier+,ShipmentIdentifier)?,DocumentReference?" .</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Missing DOCTYPE</p> <p>The DOCTYPE element is absent - <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.011/InvoiceDetail.dtd">.</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Document root element "cXML", must match DOCTYPE root "null".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incomplete Element (Country)
 The Country element must specify the country code (e.g. <Country isoCountryCode='US'>[Country]</Country>)</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Attribute "isoCountryCode" is required and must be specified for element type "Country".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incomplete Element (Name)
 The Country element must specify language (e.g. <Name xml:lang="en">[Name]</Name>)</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'THH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Attribute "xml:lang" is required and must be specified for element type "Name".</Status> </Response> </cXML> </pre> |
| 500 | <p>Internal Server Error: Various
 There are several reasons an internal error may occur:</p> <ol style="list-style-type: none"> 1. The value for a variable is too large (exceeds character limits) 2. The system received significant traffic (a large number of simultaneous invoice posts). 3. Other JAGGAER system issues. <p>For numbers 2 & 3 above, repost the invoice. If the same error is received a second time, contact JAGGAER Support (support@JAGGAER.com)</p> |

| Invoice/Credit Memo Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="500" text="Internal server error">Internal Server Error</Status> </Response> </cXML> </pre> |
| 503 | <p>Internal Error: Planned Maintenance
JAGGAER has stopped service temporarily for planned maintenance.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="503" text="Internal Error">Planned Maintenance Underway</Status> </Response> </cXML> </pre> |

ConfirmationRequest

| ConfirmationRequest Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| 200 | <p>Success, No Exceptions.
Confirmation accepted. Success.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="200" text="OK" /> </Response> </cXML> </pre> |
| 304 | <p>Success, with exception: Updated Pricing Ignored
The confirmation request has been accepted; however, the client has elected to ignore any updates to pricing, tax, or shipping as sent by the supplier. No further supplier action is required – do not repost.</p> |

| ConfirmationRequest Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="304" text="OK">Order acknowledgement successful, but Purchase Order was not updated with new pricing/tax/shipping charges </Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Sender Identity Not Provided
 The sender has not provided a value for Sender Identity. Sender Identity cannot be blank or empty.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">Supplier identity is not available with the Confirmation Request message </Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Shared Secret Not Provided
 The sender has not provided a value for Shared secret. Shared secret cannot be blank or empty.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">Supplier shared secret is not available with the Confirmation Request message </Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Invalid Client Document Reference
 The sender has provided a PO Number, orderDate, or payloadID that is not valid for the client organization.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">The PO informations given by the supplier are incorrect</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Element(s) Improperly Terminated
 One or more message element tags have not been properly terminated (e.g. <Identity>[Identity]</Identity).</p> |

| ConfirmationRequest Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">The end-tag for element type "[Element Name]" must end with a '&gt;' delimiter.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Element(s) Improperly Opened or Terminated
 One or more message element tags have not been properly opened or terminated (e.g. <Header> open tag is present; however, </Header> is nabsent or vice versa.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">The element type "[Element Name]" must be terminated by the matching end-tag "&lt;/[Element Name]&gt;".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incorrect URL
 The sender has posted the ConfirmationRequest message to the JAGGAER invoice URL in error. Repost to either accordingly:
 Test: https://usertest-messages.scquest.com/apps/Router/CXMLReceive
 Prod: https://integrations.scquest.com/apps/Router/CXMLReceive</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.scquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Element type "ConfirmationRequest" must be declared.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incorrect URL
 The sender has posted the ConfirmationRequest message to the JAGGAER ASN URL in error. Repost to either accordingly:
 Test: https://usertest-messages.scquest.com/apps/Router/CXMLReceive
 Prod: https://integrations.scquest.com/apps/Router/CXMLReceive</p> |

| ConfirmationRequest Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">Element type "ConfirmationHeader" must be followed by either attribute specifications, "&gt;" or "/&gt;".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Invalid Header or Status Type Attribute</p> <p>The sender has sent an invalid ConfirmationHeader or ConfirmationStatus type attribute. Acceptable types are: "accept", "detail", "reject" or "backordered". JAGGAER does NOT accept a status type of "allDetail", "requestToPay", or "unknown".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">Attribute "type" with value "[SupplierValue]" must have a value from the list "accept allDetail detail backordered reject unknown requestToPay ".</Status> </Response> </cXML> </pre> |
| 450 | <p>Not Implemented: Status Type "allDetail" Not Supported</p> <p>The sender has sent a status type of "allDetail". This value is not supported by JAGGAER. Acceptable status types are: "accept", "detail", "reject" or "backordered".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="450" text="Not Implemented">ConfirmationStatus "type" attribute value "allDetail" is not supported. Supported values are: detail, accept, reject, backordered</Status> </Response> </cXML> </pre> |
| 450 | <p>Not Implemented: Status Type "requestToPay" Not Supported</p> <p>The sender has sent a status type of "requestToPay". This value is not supported by JAGGAER. Acceptable status types are: "accept", "detail", "reject" or "backordered".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="450" text="Not Implemented">ConfirmationStatus "type" attribute value "requestToPay" is not supported. Supported values are: detail, accept, reject, backordered</Status> </Response> </cXML> </pre> |

| ConfirmationRequest Response Codes | |
|------------------------------------|---|
| Response Code | Response Message |
| 450 | <p>Not Implemented: Status Type "unknown" Not Supported</p> <p>The sender has sent a status type of "unknown". This value is not supported by JAGGAER. Acceptable status types are: "accept", "detail", "reject" or "backordered".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="450" text="Not Implemented">ConfirmationStatus "type" attribute value "unknown" is not supported. Supported values are: detail, accept, reject, backordered</Status> </Response> </cXML> </pre> |
| 500 | <p>Internal Server Error: Invalid Authentication Credentials (Sender Identity)</p> <p>The sender has provided a Sender Identity value that does not match the expected supplier value.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="500" text="Internal server error">Supplier identity and the sender identity are not equal</Status> </Response> </cXML> </pre> |
| 500 | <p>Internal Server Error: Invalid Authentication Credentials (Shared Secret)</p> <p>The sender has provided a Shared Secret value that does not match the expected supplier value.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="500" text="Internal server error">Supplier shared secret and the sender shared secret are not equal</Status> </Response> </cXML> </pre> |
| 500 | <p>Internal Server Error: Null Element Attribute</p> <p>One or more element attributes is null (e.g. <ShipNoticeItem lineNumber="" quantity="">)</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="500" text="Internal server error">null</Status> </Response> </cXML> </pre> |

| ConfirmationRequest Response Codes | |
|------------------------------------|--|
| Response Code | Response Message |
| 500 | <p>Internal Server Error: Various</p> <p>There are several reasons an internal error may occur:</p> <ol style="list-style-type: none"> 1. The system received significant traffic (a large number of simultaneous ConfirmationRequest posts). 2. Other JAGGAER system issues. <p>For either of the above, repost the ConfirmationRequest. If the same error is received a second time, contact JAGGAER Support (support@JAGGAER.com).</p> <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="500" text="Internal server error">Internal Server Error</Status> </Response> </cXML></pre> |
| 503 | <p>Internal Error: Planned Maintenance</p> <p>JAGGAER has stopped service temporarily for planned maintenance.</p> <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="503" text="Internal Error">Planned Maintenance Underway</Status> </Response> </cXML></pre> |

ShipNoticeRequest

| ShipNoticeRequest Response Codes | |
|----------------------------------|--|
| Response Code | Response Message |
| 200 | <p>Success, No Exceptions.</p> <p>ShipNoticeRequest accepted. Success.</p> <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="200" text="OK" /> </Response> </cXML></pre> |
| 400 | <p>Bad Request: Sender Identity Not Provided</p> <p>The sender has not provided a value for Sender Identity. Sender Identity cannot be blank or empty.</p> |

| ShipNoticeRequest Response Codes | |
|----------------------------------|---|
| Response Code | Response Message |
| | <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">Supplier identity is not available with the ASN message </Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Shared Secret Not Provided</p> <p>The sender has not provided a value for Shared secret. Shared secret cannot be blank or empty.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">Supplier shared secret is not available with the ASN message</Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Invalid Client Document Reference</p> <p>The sender has provided a PO Number, orderDate, or payloadID that is not valid for the client organization.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">The PO informations given by the supplier are incorrect</Status> </Response> </cXML> </pre> |
| 400 | <p>Bad Request: Invalid Shipment Reference</p> <p>The sender is attempting to update or delete a previously recorded shipment. The shipmentID is either not recognized and/or the original message payloadID is invalid or not referenced.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="400" text="Bad Request">Not able to find a shipment for the given payloadid [original ASN Message payloadID] and supplier shipment id [Supplier ShipmentID]</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Invalid Authentication Credentials (Sender Identity)</p> |

| ShipNoticeRequest Response Codes | |
|----------------------------------|--|
| Response Code | Response Message |
| | <p>The sender has provided a Sender Identity value that does not match the expected supplier value.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="401" text="Unauthorized">Supplier identity and the sender identity are not equal</Status> </Response> </cXML> </pre> |
| 401 | <p>Unauthorized: Invalid Authentication Credentials (Shared Secret)</p> <p>The sender has provided a Shared Secret value that does not match the expected supplier value.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="401" text="Unauthorized">Supplier shared secret and the sender shared secret are not equal</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Element(s) Improperly Terminated</p> <p>One or more message element tags have not been properly terminated (e.g. <Identity>[Identity]</Identity>).</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">The end-tag for element type "[Element Name]" must end with a '&gt;,' delimiter.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Element(s) Improperly Opened or Terminated</p> <p>One or more message element tags have not been properly opened or terminated (e.g. <Header> open tag is present; however, </Header> is nabsent or vice versa.</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">The element type "[Element Name]" must be terminated by the matching end-tag "&lt;/[Element Name]&gt;".</Status> </Response> </cXML> </pre> |

| ShipNoticeRequest Response Codes | |
|----------------------------------|--|
| Response Code | Response Message |
| 406 | <p>Unacceptable: Incorrect URL</p> <p>The sender has posted the ShipNoticeRequest message to the JAGGAER invoice URL in error. Repost to either accordingly:
 Test: https://usertest-messages.sciquest.com/apps/Router/ASNReceive
 Prod: https://integrations.sciquest.com/apps/Router/ASNReceive</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "https://solutions.sciquest.com/app_docs/dtd/cxml/InvoiceDetail.dtd"> <cXML timestamp="YYYY-MM-DD'T'HH:mm:ssZ" payloadID="[payloadID]"> <Response> <Status code="406" text="Not Acceptable">Element type " ShipNoticeRequest" must be declared.</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Incorrect URL</p> <p>The sender has posted the ShipNoticeRequest message to the JAGGAER ConfirmationRequest URL in error. Repost to either accordingly:
 Test: https://usertest-messages.sciquest.com/apps/Router/ASNReceive
 Prod: https://integrations.sciquest.com/apps/Router/ASNReceive</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">The content of element type "DocumentReference" must match "EMPTY".</Status> </Response> </cXML> </pre> |
| 406 | <p>Unacceptable: Invalid Header Operation</p> <p>The sender has sent an invalid Header operation. Acceptable types are: "new", "update", or "delete".</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">Attribute "operation" with value "[Supplier Value]" must have a value from the list "new update delete".</Status> </Response> </cXML> </pre> |
| 406 | <p>Not Implemented: Incomplete Element (ServiceLevel)</p> <p>The ServiceLevel element must specify language (e.g. <ServiceLevel xml:lang="en">[Service Level]</ServiceLevel></p> |

| ShipNoticeRequest Response Codes | |
|----------------------------------|--|
| Response Code | Response Message |
| | <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="406" text="Not Acceptable">Attribute "xml:lang" is required and must be specified for element type "ServiceLevel".</Status> </Response> </cXML></pre> |
| 500 | <p>Internal Server Error: Various</p> <p>There are several reasons an internal error may occur:</p> <ol style="list-style-type: none"> 1. The system received significant traffic (a large number of simultaneous ShipNoticeRequest posts). 2. Other JAGGAER system issues. <p>For either of the above, repost the ShipNoticeRequest. If the same error is received a second time, contact JAGGAER Support (support@JAGGAER.com).</p> <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="500" text="Internal server error">Internal Server Error</Status> </Response> </cXML></pre> |
| 503 | <p>Internal Error: Planned Maintenance</p> <p>JAGGAER has stopped service temporarily for planned maintenance.</p> <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/Fulfill.dtd"> <cXML payloadID="[payloadID]" timestamp="YYYY-MM-DD'T'HH:mm:ssZ"> <Response> <Status code="503" text="Internal Error">Planned Maintenance Underway</Status> </Response> </cXML></pre> |

Troubleshooting Guides

The following guides provide possible causes and recommended solutions to common issues by integration point. For some integration types, the JAGGAER response code tables provide the best source for troubleshooting. If the proposed solutions do not correct the problem and the supplier is in initial implementation, please contact the JAGGAER Project Manager responsible for the active project. If the supplier is already in production with integrations and requires

further assistance, please contact JAGGAER Support (Support@JAGGAER.com) for additional help.

PunchOut Troubleshooting Guide

Following are some common problems with punch-out, with possible causes and recommended solutions.

Failed Connection

There are many possible reasons user(s) may be unable to connect.

1. **Failed Authentication** - Verify the accuracy of the header credentials in the PunchOutSetupRequest Message. Credentials are case sensitive.
 - a. From/Do Domains
 - b. Sender/To Domains
 - c. From/To Identities
 - d. Shared Secret
2. **Invalid URL** - Verify the transactive URL is accurate in the PunchOutSetupRequest Message.
3. **Firewall Port** - Ensure firewall access has been granted and port 443 is open to traffic from <https://usertest-messages.scquest.com> (test) or <https://integrations.scquest.com> (production).
4. **SSL Digital Certificate**
 - a. Verify the SSL digital certificate is issued against the transactive URL.
 - b. If the SSL certificate authority is either self-signed, or issued by authorities other than Verisign or Thawte, provide the certificate to JAGGAER for installation.
5. **Invalid/Absent Data Fields** - Verify all expected values have been passed in the PunchOutSetupRequest Message. Note: fields such as email address, phone number, organization name, organizationID, department, and ShipTo address code may not be available to pass. PunchOut connectivity should not rely upon populated values in these fields. For more information, see the Elements table within the PunchOutSetupRequest section of this document.
6. **PunchOutSetupResponse Message**
 - a. Validate response message formatting against the format outlined in this specification.
 - b. Verify the response message is being posted (https, port 443) to the dynamic URL provided in the PunchOutSetupRequest Message. This URL varies by customer and can change at any time without notice.
 - c. Verify the response message contains the correct start page URL.
 - d. Verify the response message does not contain byte order marks.
7. **P3P Compliance** - Verify compact policy headers in place for all content. Default Internet Explorer (v.6 and higher) settings block 3rd party cookies for content without compact policy headers. PunchOut sessions from eProcurement applications such as JAGGAER display content in frames – IE treats all content displayed in frames as 3rd party content. Note: the punch-out site is required to P3P compliant. Users are often restricted

by IT/Security departments and do not have the ability to override default browser settings. Please refer to the [P3P policy](#) for more information.

8. **Unrecognized New User** - Verify if there is an automated process in place to handle new, unrecognized users at the time of PunchOutSetupRequest Message authentication. Access to the punch-out site should never fail or be dependent upon the supplier to setup the user prior to accessing.
9. **Supplier Site Outage** - Verify the supplier punch-out site is not experiencing network issues or down time.
10. **Multiple, Concurrent Users Not Support** - The supplier punch-out site is not able to handle multiple, concurrent users. Site redevelopment will be required per punch-out requirements.

Shopper asked to login

The shopper is asked to login upon initial access.

1. **Blocked Cookies** - Verify compact policy headers in place for all content. Default Internet Explorer (v.6 and higher) settings block 3rd party cookies for content without compact policy headers. PunchOut sessions from eProcurement applications such as JAGGAER display content in frames – IE treats all content displayed in frames as 3rd party content. Note: the punch-out site is required to be P3P compliant. Users are often restricted by IT/Security departments and do not have the ability to override default browser settings. Please refer to the P3P policy within this document for more information.
2. **Failure to Create User Session**
 - a. Verify the user session is created when the PunchOutSetupRequest Message is authenticated. The user should land on the supplier start page already logged in and ready to shop.
 - b. Verify there is an automated process in place to handle new, unrecognized users at the time of PunchOutSetupRequest Message authentication. Access to the punch-out site should never fail or require additional login as users will perceive the site is malfunctioning.
 - c. Verify the accuracy of the start page URL sent in the supplier PunchOutSetupResponse Message to JAGGAER.

Images do not display

A user reports that some or all images do not display.

1. **Images Lack Compacy Policy Headers**
 - a. The user's browser may have displayed a yes/no popup upon connection regarding the display of nonsecure items. If the user selected "no" ("yes" in IE 8+), images without compact policy headers will not display. Default Internet Explorer (v. 6 and higher) settings block 3rd party cookies for content without compact policy headers. PunchOut sessions from eProcurement applications such as JAGGAER display content in frames – IE treats all content displayed in frames as 3rd party content. Note: the punch-out site is required to be P3P compliant. Users are often restricted by IT/Security departments and do not have the ability to override default browser settings. Please refer to the P3P policy within this document for more information.
 - b. Verify images on alternate servers or locations have compact policy headers.

2. **Invalid Image URLs or Reference File Names** - Verify the page source code is referencing accurate image names and locations.

Cart contents from previous sessions still present

Shopper reports that cart contents from previous shopping sessions are still present.

1. **Shared Cookie with eCommerce Site** - PunchOut sites that share source code with the supplier eCommerce site should have a separate and unique cookie. A user may have added cart contents on the eCommerce site prior to logging in to the punch-out site.
2. **Failure to Create New User Session** - Each PunchOutSetupRequest Message sent by JAGGAER should initiate a new user session.

Session expired immediately upon login

The user reports a punch-out site notification on the landing page that the session has expired.

1. **Blocked Cookies** - Verify compact policy headers in place for all content. Default Internet Explorer (v.6 and higher) settings block 3rd party cookies for content without compact policy headers. PunchOut sessions from eProcurement applications such as JAGGAER display content in frames – IE treats all content displayed in frames as 3rd party content. Note: the punch-out site is required to be P3P compliant. Users are often restricted by IT/Security departments and do not have the ability to override default browser settings. Please refer to the P3P policy within this document for more information.
2. **Failure to Create New User Session** - Each PunchOutSetupRequest Message sent by JAGGAER should initiate a new user session. The shopper may have cancelled the previous session or requested a new session after prior sessions have expired.

Frame breaks

The user reports that the frame disappears upon initial connection, or when shopping a particular page.

1. **Site Code Forcing Frame Breaks**
 - a. Review the site source code for scripts that force break a frame session. For example, a script that designates the supplier page as the top window will break a frame session.
 - b. Review any links within the site that may have code forcing frame breaks. If links are required, set the site code such that the link opens in a new window.

Shopper unable to use punch-out favorites feature

A user(s) reports inability to add items to favorites within the punch-out site.

1. **User Email Address Not Available to Pass** - The user may not have their email address populated in their profile within the JAGGAER application, and is therefore unavailable to pass to the supplier at the time of punch-out. If the supplier is using the email address to key favorites, this may prevent favorites functionality from operating as expected. Consider keying favorites from a combination of the From Identity in the message credentials with the shopper's username – these two fields will be passed without fail to the supplier each time. The combination of these two fields ensures that each user across multiple customers will have a unique value that favorites can be keyed on.

2. **Unrecognized New User** - Verify there is an automated process in place to handle new, unrecognized users at the time of PunchOutSetupRequest Message authentication. Access to the punch-out site should never fail or be dependent upon the supplier to setup the user prior to accessing.

Shoppers seeing other user favorites

A user reports that favorites exist that were not added by the user.

1. **Multiple, Concurrent Users Not Supported** - The supplier punch-out site is not able to handle multiple, concurrent users; and as a result, shoppers are seeing favorites as added by all users. Site redevelopment will be required per punch-out requirements.
2. **User Viewing Organization Favorites** - The user is reviewing organizational favorites, which can be populated by all users, or by client administrators.

Shopper directed away from site

A user reports that normal punch-out site navigation has directed the browser outside of the punch-out.

1. **External Site Links Present** - Any links to external sites, or other internal sites outside of the punch-out should be disabled. If links are available that provide additional product information, the link should open in a new window but should not provide the user with new pathways to purchase products.

Incorrect pricing is displayed

The user/administrator reports that the incorrect pricing is displayed on the punch-out.

1. **Pricing Structure Not Associated With "From Identity"** - Verify that the "From Identity" passed in the PunchOutSetupRequest Message is associated with the correct client account.
2. **Contracted Pricing Not Loaded to Client Account** - Verify that the negotiated contract pricing has been loaded against the client account.
3. **Other Account Setup Issues** - Review client account setup and verify all required supplier steps/processes have been completed.
4. **Invalid/Absent Data Fields** - Review the PunchOutSetupRequest Message to ensure all necessary user data is being passed. Note: fields such as email address, phone number, organization name, organizationID, department, and ShipTo address code may not be available to pass. PunchOut pricing should not rely upon populated values in these fields. For more information, see the Elements table within the PunchoutSetupRequest section of this document.

User unable to add items to cart

The user reports successful punch-out connection, but is not able to add items to the cart.

1. **Blocked Cookies** - Verify compact policy headers in place for all content. Default Internet Explorer (v.6 and higher) settings block 3rd party cookies for content without compact policy headers. PunchOut sessions from eProcurement applications such as JAGGAER display content in frames – IE treats all content displayed in frames as 3rd party content. Note: the punch-out site is required to P3P compliant. Users are often restricted by IT/Security departments and do not have the ability to override default browser

settings. Please refer to the P3P policy within this document for more information.

2. **X-Frame-Options Enabled** - Microsoft introduced X-Frame-Options with IE8 that is designed to prevent content from displaying within a frame. eProcurement applications operate punch-out sites within frames; therefore, if the supplier site has X-Frame-Options enabled, the punch-out cart return will fail. Disable X-Frame-Options.

Failed cart return

A user reports an error when attempting to return a cart from the punch-out site to JAGGAER.

1. **Invalid Post Method / URL** - Verify the PunchOutOrderMessage is being posted via cxml-urlencoded form submit to the URL provided in the initial PunchOutSetupRequest Message.
2. **Invalid Message Format** - Verify the format of the POOM against format outlined in this specification.
3. **Cross Site Scripting (XSS)** - Microsoft introduced their new XSS filter with IE8 that is designed to help mitigate cross site scripting vulnerabilities. Cross site scripting can enable an internet attacker to monitor websites for user data or hijack account data via cookie theft. If IE8 determines a cross site scripting threat in URL or HTTP Post requests, the browser will automatically block the script from executing by cleaning it, thereby modifying the message being sent. This can impact the punch-out cart return to JAGGAER for shoppers using IE8 with the browser XSS filter is enabled.
4. **X-Frame-Options Enabled** - Microsoft introduced X-Frame-Options with IE8 that is designed to prevent content from displaying within a frame. eProcurement applications operate punch-out sites within frames; therefore, if the supplier site has X-Frame-Options enabled, the punch-out cart return will fail. Disable X-Frame-Options.
5. **Content Contains Special Characters**
 - a. Verify the part number, product description, and manufacturer part number does not contain any non UTF-8 characters or characters used in cXML standards (e.g. <, &).
 - b. Use CDATA tags for product descriptions.
6. **PunchOut Session Expired** - The user's idle time exceeded thresholds and was logged out.
7. **SPAID Exceeds Character Limitations** - Verify the SupplierPartAuxiliaryID passed on the PunchOutOrderMessage does not exceed 100 characters.

Pricing displayed on punch-out site changes when cart is returned

The user reports that the pricing displayed on the punch-out site is not the same pricing received on cart return.

1. **Incorrect UnitPrice / Qty per UOM** - Review the PunchOutOrderMessage (POOM) and confirm the UnitPrice and Qty correlates to the UOM. For example: if the shopper ordered qty 1 Box of 30 gloves at a total of \$9.99, but the POOM passes qty 30 EA of gloves at \$9.99, JAGGAER will calculate the extended price as [Qty=30] X [UnitPrice-e=\$9.99] = \$299.70. For more information, see the Numerical Formats & Calculations section within this document.
2. **UnitPrice has more than 4 Decimal places** - JAGGAER accepts up to 4 decimal places for UnitPrice. This means that rounding will occur when the supplier passes more

than 4 decimal places. For more information, see the Numerical Formats & Calculations section within this document.

Product flags or UNSPSC not received

The user reports that product flags and/or UNSPSC codes are not being passed.

1. **Incomplete PunchOutOrderMessage** - Review the PunchOutOrderMessage (POOM) and verify product flags and/or UNSPSC codes are being returned for all products.
2. **Invalid Product Flag / UNSPSC Data Field Formatting** - Review the POOM and verify that the product flags and/or UNSPSC code are being passed as <Classification domains>. For more information about the passing of these fields, please see the PunchOutOrderMessage section within this document.
3. **Product Flags Not Configured in JAGGAER** - If the product flags are being passed to JAGGAER per the format outlined in this document (and verified with the previous step), contact Support to review the supplier configuration for these flags.

CAS Number not received

The user reports that chemical products are not being returned with a CAS number.

1. **Incomplete PunchOutOrderMessage** - Review the PunchOutOrderMessage (POOM) and verify the CAS Number is being returned for all chemical products.
2. **Invalid CAS Number Data Field Formatting** - Review the POOM and verify that the CAS Number is being passed as an extrinsic <Extrinsic name="CAS">. For more information about the passing of these fields, please see the PunchOutOrderMessage section within this document.
3. **CAS Number Not Configured in JAGGAER** - If the CAS Number is being passed to JAGGAER per the format outlined in this document (and verified with the previous step), contact Support to review the supplier configuration for this field.

Tax/shipping estimates not received

The user reports that the tax and shipping charges listed in the cart summary on the punch-out site are not being received upon cart return to JAGGAER.

1. **Invalid PunchOutOrderMessage Format** - Review the PunchOutOrderMessage (POOM) and validate the charges are being passed per client requirements, and as outlined in the PunchOutOrderMessage section within this document.
2. **Charges Not Supported by Client** - The client's application configuration may not support the return of summary level shipping or tax charges on the POOM. Discuss alternatives with both JAGGAER and the client.
3. **Shipping as Line Item in Cart** - The client configuration may not support the return of shipping charges at the summary level, but may support the return of charges as a line item in the cart. For more information about shipping and tax charges on the POOM, please visit the PunchOutOrderMessage section within this document.

Handling estimates not received

The user reports that the tax and shipping charges are being returned successfully, but handling charges are not.

1. **Invalid Data Field** - Handling estimates cannot be passed on a PunchOutOrder-Message. These charges can be added to the shipping charges upon client approval.

Full description not received

The user reports that the full product description is not being received.

1. **Character Limitations Exceeded** - JAGGAER can consume no more than 256 characters for the product description. Verify the description is within character limitations. If not, consider revising the description to within limitations.

Edit/Inspect Troubleshooting Guide

Following are some common problems with edit/inspect functionality, with possible causes and recommended solutions.

User unable to reinitiate prior sessions

The user reports that they do not have link to access within JAGGAER to reinitiate the punch-out session to make changes.

1. **operationAllowed is "create"** - Review the PunchOutOrderMessage returned from the initial session to ensure that the operation Allowed is "edit". If the operation is set to "create", the user will not have the ability to reinitiate the session. Please see the chapter on edit/inspect for more information.
2. **operationAllowed is "inspect"** - The user may be looking to make changes to the initial cart return. Review the PunchOutOrderMessage returned from the initial session to ensure that the operation Allowed is "edit". If the operation is set to "inspect", the user can only reinitiate the session to view contents, not make changes. Please see the chapter on edit/inspect for more information.
3. **Operation of "edit" or "inspect" Not Supported** - Verify that the supplier can support an operationAllowed other than "create" and that the punch-out will operate with the expected behavior.
4. **Integration Not Enabled** - Request that JAGGAER confirm the edit/inspect integration is turned on for the supplier within the client organization application.

Level 2 PunchOut Troubleshooting Guide

Following are some common problems with Level 2 PunchOut functionality, with possible causes and recommended solutions.

Failed Connection

The user reports that connection to the level 2 punch-out is unsuccessful. Also see the [PunchOut Troubleshooting](#) section.

1. **Unrecognized / Invalid Part Number** - Review the part number passed in the PunchOutSetupRequest Message and confirm its validity. If invalid or obsolete, ensure there is a process in place to either land the user on the home page, or display a page

informing the user that the part is no longer available. Please see the Level 2 section within this document for more information.

2. **Invalid Data in <SupplierPartID> field** - Review the PunchOutSetupRequest Message and confirm that the expected value is present in the <SupplierPartID> tag of the message. For a list of the value options, please see the Level 2 section within this document.
3. **Other PunchOut Issues** - Review the standard PunchOut troubleshooting guide for possible punch-out issues not related to Level 2.

Navigation not as expected

The user reports that upon initial connection, navigation to the selected product or category was not achieved.

1. **Unrecognized / Invalid Part Number or Category** - Review the value passed in the <SupplierPartID> tag of the PunchOutSetupRequest Message and confirm its validity. If invalid or obsolete, ensure there is a process in place to either land the user on the home page, or display a page informing the user that the part is no longer available. Please see the Level 2 section within this document for more information.
2. **Store Level Support Only** - Verify supplier ability to parse data passed in within the <SelectedItem> tags. Supplier may support Store Level only, which will land the user on the standard home page. Please see the Level 2 section within this document for more information.
3. **Other PunchOut Issues** - Review the standard [PunchOut Troubleshooting Guide](#) for possible punch-out issues not related to Level 2.

Supplier products do not appear in user search results

The user reports that the supplier product offering does not appear in JAGGAER search results.

1. **Absent Proxy Catalog Accessibility (PCA) File**
 - a. Verify the supplier has loaded the PCA file via the Supplier Portal and has provided general product accessibility to the specific requesting customer
 - b. Verify the accessibility within the PCA file is "True" for the specific requested product or category of products.
2. **Proxy Catalog Has Not Been Successfully Loaded** - Verify the supplier has successfully loaded proxy catalog content via the Supplier Portal.

Product in search results has no link to supplier punch-out

The user reports that supplier products appear in search results, but the link to the punch-out is not present.

1. **Static Contract / List Pricing Still Active:**
 - a. Ensure the supplier has removed any static contracted pricing that was in place prior to Level 2 implementation. Contracted, hosted pricing takes precedence over punch-out pricing.
 - b. Verify the client organization has turned off the viewing of List Price within the JAGGAER supplier profile. Available list pricing that was in place prior to Level 2 implementation takes precedence over punch-out pricing .

2. **User Does Not Have PunchOut Access** - Verify through the client administrator that the user has been provided punch-out accessibility

LivePrice Troubleshooting Guide

Following are some common problems with LivePrice functionality, with possible causes and recommended solutions.

Price retrieval error

The user reports that an error is displayed when attempting to retrieve the latest pricing for item(s).

1. **Failed Authentication** - Confirm the URL, customer code, authentication identity, and shared secret credentials are accurate. Credentials are case sensitive.
2. **Invalid / Absent Fallback Location Code**
 - a. The user does not have a default ShipTo address listed in their JAGGAER profile, or has selected a new unrecognized address code. Verify the validity and presence of the fallback location code and obtain an updated ShipTo address list from the client.
 - b. Consider utilizing zip code for the location code type, as most new ShipTo address codes created are within an already recognized area.
 - c. If the entire buying organization is geographically located in the same area and will be serviced from a single supplier distribution center, consider using a single location for all users. JAGGAER has the option to pass the fallback location code for every availability call.
3. **Invalid Supplier Response Message**
 - a. Review the supplier outbound response message and verify validity. See the [LivePrice Message Elements](#) table for more information about message format.
 - b. Verify the supplier response message is routed to JAGGAER via the same request path.
 - c. Verify the response message contains the MessageId value from the associated request message in the RelatedMessageId field.
4. **Supplier System at Capacity**
 - a. The supplier system is receiving a significant number of request messages and is unable to process at current volumes. The number of items in a single message may be too low. Consider increasing the number of items per message to reduce the overall number of incoming requests. See the [Items per LivePrice Call](#) section of the [LivePrice](#) chapter for more information.
 - b. The supplier system response time is delayed due to a high number of items per message. JAGGAER resends the request when a response is not received within two seconds and will display an error after three failed attempts. Consider decreasing the number of items in a single message to facilitate quicker response times. See the [Items per LivePrice Call](#) section of the [LivePrice](#) chapter for more information.
 - c. Review the LivePrice cache lifespan to ensure that it is appropriate. If pricing typically does not change from week to week, consider increasing the lifespan to reduce the overall number of incoming messages. See the [LivePrice Cache Lifespan](#) section within the [LivePrice](#) chapter for more information.

5. **Unrecognized / Invalid Part Number** - Review the part number passed in the LivePrice Request message and confirm its validity. If invalid or obsolete, update the catalog content via the Supplier Portal.
6. **SSL Digital Certificate**
 - a. Verify the SSL digital certificate is issued against the transactive URL.
 - b. If the SSL certificate authority is either self-signed, or issued by authorities other than Verisign or Thawte, provide the certificate to JAGGAER for installation.

Supplier products do not appear in user search results

The user reports that the supplier product offering does not appear in JAGGAER search results.

1. **Absent Proxy Catalog Accessibility (PCA) File**
 - a. Verify the supplier has loaded the PCA file via the Supplier Portal and has provided general product accessibility to the specific requesting customer
 - b. Verify the accessibility within the PCA file is "True" for the specific requested product or category of products.
2. **Proxy Catalog Has Not Been Successfully Loaded** - Verify the supplier has successfully loaded proxy catalog content via the Supplier Portal.

Static pricing is displayed

The user reports that static pricing is displayed rather than LivePrice.

1. **Static Contract / List Pricing Still Active**
 - a. Ensure the supplier has removed any static contracted pricing that was in place prior to LivePrice implementation. Contracted, hosted pricing takes precedence over LivePrice.
 - b. Verify the client organization has turned off the viewing of List Price within the JAGGAER supplier profile. Available list pricing that was in place prior to implementation takes precedence over LivePrice .

Invalid pricing is displayed

The user reports that stale or old pricing is displayed.

1. **LivePrice Cache Requires Reset**
 - a. If the supplier has recently updated pricing, visit the Supplier Portal to clear the LivePrice cache. This ensures that a new request message is routed to the supplier for each item rather than the retrieval of the old price from the cache.
 - b. Evaluate the LivePrice cache lifespan to ensure the allotted time is appropriate.
2. **Updated Pricing Not Associated With Customer** - Verify the latest pricing is associated with the customer account and/or customer code.

Availability Call Troubleshooting Guide

Following are some common problems with Availability Call functionality, with possible causes and recommended solutions.

Failed connection

The user reports that an error is displayed when attempting to retrieve product availability.

1. **Failed Authentication** - Confirm the URL, customer code, authentication identity, and shared secret credentials are accurate. Credentials are case sensitive.
2. **Invalid / Absent Fallback Location Code**
 - a. The user does not have a default ShipTo address listed in their JAGGAERr profile, or has selected a new unrecognized address code. Verify the validity and presence of the fallback location code and obtain an updated ShipTo address list from the client.
 - b. Consider utilizing zip code for the location code type, as most new ShipTo address codes created are within an already recognized area.
 - c. If the entire buying organization is geographically located in the same area and will be serviced from a single supplier distribution center, consider using a single location for all users. JAGGAER has the option to pass the fallback location code for every availability call.
3. **Invalid Supplier Response Message**
 - a. Review the supplier outbound response message and verify validity. See the LivePrice Message Elements table for more information about message format.
 - b. Verify the supplier response message is routed to JAGGAER via the same request path.
 - c. Verify the response message contains the MessageId value from the associated request message in the RelatedMessageId field.
4. **Unrecognized / Invalid Part Number** - Review the part number passed in the Availability Call message and confirm its validity. If invalid or obsolete, update the catalog content via the Supplier Portal.
5. **SSL Digital Certificate**
 - a. Verify the SSL digital certificate is issued against the transactive URL.
 - b. If the SSL certificate authority is either self-signed, or issued by authorities other than Verisign or Thawte, provide the certificate to JAGGAER for installation.

Updated price is not consumed by JAGGAER

The user/shopper reports that the updated price sent by the supplier is not available or consumed by JAGGAER.

1. **LivePrice is Not Applicable**
 - a. The client organization is not setup for the LivePrice integration. The shopper is attempting to retrieve product availability only, and the pricing available to the shopper is the contracted price that was reviewed/approved by the client. The Availability Call integration ignores pricing in the supplier response message. See the [Availability Call](#) chapter for more information.
 - b. Verify with JAGGAER that the "Supplier Updated Price" field is turned off for the buying organization to avoid confusion.
 - c. Ensure that the pricing returned for the organization is the identical pricing that was loaded against the hosted catalog. If pricing is stale, update the client's pricing via the Supplier Portal.

Purchase Order (OrderRequest) Troubleshooting Guide

Following are some common problems with OrderRequest functionality, with possible causes and recommended solutions.

Failed connection / Failure response

There are many possible reasons for a failed connection / failure response message receipt.

1. **Failed Authentication** - Verify the accuracy of the header credentials in the Order-Request Message. Credentials are case sensitive.
 - a. From/To Domains
 - b. Sender/To Domains
 - c. From/To Identities
 - d. Shared Secret
2. **Invalid URL** - Verify the transactive URL is accurate.
3. **Firewall Port** - Ensure firewall access has been granted and port 443 is open to traffic from <https://usertest-messages.scquest.com> (test) or <https://integrations.scquest.com> (production).
4. **SSL Digital Certificate**
 - a. Verify the SSL digital certificate is issued against the transactive URL.
 - b. If the SSL certificate authority is either self-signed, or issued by authorities other than Verisign or Thawte, provide the certificate to JAGGAER for installation.
5. **Unrecognized/Absent Data Fields** - Review the values passed in the OrderRequest message and evaluate their impact/contribution to the failure.
 - a. ShipTo Address code
 - b. BillTo Address code
 - c. Unexpected Address Format
 - d. Invalid/Unrecognized SupplierPartID
 - e. Contact Information (Phone Number, Email)
 - f. Payment Method
 - g. Product Availability

Note: Any of the above referenced field values should not generate a failure response due to unexpected or invalid data. If a faxed order is illegible or contains ambiguous data, customer service typically contacts the shopper to clarify order intent in order to complete the sale. Similar to that, suppliers will benefit from an electronic exception process that accepts orders with issues, moving them to an internal queue for manual corrections before processing. See the [Purchase Order](#) chapter for more information.

6. **Duplicate Client Purchase Order Number** - Review the orderID to determine whether it is a duplicate purchase order number. If yes, contact the customer to verify intent and/or request a new order.
7. **System Outage**

- a. Verify the supplier system is not experiencing network issues or down time.
- b. Contact JAGGAER to determine system availability.

Confirmation Request Troubleshooting Guide

Below are the best sources for troubleshooting the ConfirmationRequest integration.

Troubleshooting Sources

1. **ConfirmationRequest Chapter** - The ConfirmationRequest chapter within this document provides an overview of the integration and outlines the requirements.
2. **ConfirmationRequest Sample Message** - Within the ConfirmationRequest chapter is a sample message for supplier review.
3. **ConfirmationRequest Message Elements Table** - The [ConfirmationRequest Message Elements](#) table within the integration chapter outlines element and attribute formatting, usage, options, and a description of JAGGAER support.
4. **ConfirmationRequest Response Codes** - One of the best sources of troubleshooting this integration is the listing of [JAGGAER response codes](#) within the associated appendix.
5. **cXML.org** - The cXML Users Guide available at www.cxml.org is a comprehensive document that outlines this and other cXML based integrations.

ShipNotice Request Troubleshooting Guide

Below are the best sources for troubleshooting the ShipNoticeRequest integration.

Troubleshooting Sources

1. **ShipNoticeRequest Chapter** - The [ShipNoticeRequest](#) chapter provides an overview of the integration and outlines the requirements.
2. **ShipNoticeRequest Sample Message** - Within the ShipNoticeRequest chapter are [sample messages](#) for supplier review.
3. **ShipNoticeRequest Message Elements Table** - The [ShipNoticeRequest Message Elements](#) table within the integration chapter outlines element and attribute formatting, usage, options, and a description of JAGGAER support.
4. **ShipNoticeRequest Response Codes** - One of the best sources of troubleshooting this integration is the listing of [JAGGAER response codes](#) within the associated appendix.
5. **cXML.org** - The cXML Users Guide available at www.cxml.org is a comprehensive document that outlines this and other cXML based integrations.

Invoices/Credit Memos Request Troubleshooting Guide

Below are the best sources for troubleshooting the Invoices/Credit Memo integration.

Troubleshooting Sources

1. **Invoices / Credit Memos Chapter** - The [Invoices / Credit Memos](#) chapter provides an overview of the integration and outlines the requirements.
2. **Invoices / Credit Memo Sample Message** - Within the Invoices / Credit Memos chapter are [sample messages](#) for supplier review.
3. **Invoices/Credit Memo Message Elements Table** - The [Invoices/Credit Memo Message Elements](#) table within the integration chapter outlines element and attribute formatting, usage, options, and a description of JAGGAER support.
4. **Invoices/Credit Memo Response Codes** - One of the best sources of troubleshooting this integration is the listing of [JAGGAER response codes](#) within the associated appendix.
5. **The cXML Users Guide** available at cXML.org is a comprehensive document that outlines this and other cXML based integrations.