

Cost of Resource Exchange (CORE) Protocol

Trial Use Period: April 1, 2009 to March 31, 2010

Abstract: CORE defines an XML schema to facilitate the exchange of financial information related to the acquisition of library resources between systems. The two systems may be within the same organization, e.g., an ILS and an ERMS, or from two different organizations, e.g., a subscription agent and a library.

A (proposed) American National Standard
Developed by the
National Information Standards Organization



Published by:
NISO, Baltimore, Maryland, U.S.A.

DRAFT STANDARD FOR TRIAL USE

About NISO Standards

This is a draft standard and subject to change.

NISO standards are developed by Working Groups and Topic Committees of the National Information Standards Organization. The development process is a strenuous one that includes a rigorous peer review of proposed standards open to each NISO Voting Member and any other interested party. Final approval of the standard involves verification by the American National Standards Institute that its requirements for due process, consensus, and other approval criteria have been met by NISO. Once verified and approved, NISO Standards also become American National Standards.

This standard may be revised or withdrawn at any time. For current information on the status of this standard, contact the NISO office or visit the NISO website at:

www.niso.org/standards

Published by

NISO

One North Charles Street

Suite 1905

Baltimore, MD 21201

www.niso.org

Copyright © 2009 by the National Information Standards Organization

All rights reserved under International and Pan-American Copyright Conventions. For noncommercial purposes only, this publication may be reproduced or transmitted in any form or by any means without prior permission in writing from the publisher, provided it is reproduced accurately, the source of the material is identified, and the NISO copyright status is acknowledged. [To be added to final publication: For permission to photocopy or use material electronically from NISO Z39.95-200x, *Cost of Resource Exchange (CORE) Protocol*, ISSN 1041-5653, please access www.copyright.com or contact Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users.]

Printed in the United States of America

ISSN: 1041-5653 (National information standards series)

ISBN: xxxx [NISO will add to final publication]

DRAFT STANDARD FOR TRIAL USE

Contents

Foreword	v
1 Purpose	1
2 Scope	1
2.1 CORE Transactions	1
2.2 Transport Mechanism	1
2.3 Out of Scope	1
3 References	2
4 Definitions	2
5 CORE Schema Synopsis	5
5.1 General	5
5.2 CORE Namespace	5
5.3 CORE Transactions	5
5.4 Request Design	6
5.5 Reply Design	6
6 Data Dictionary Reference Guide	7
6.1 Element Listings	7
6.2 Graphic Representations	7
6.3 Data Types	8
6.4 Attributes	9
7 CORE Data Dictionary	10
7.1 COREDocument	10
7.2 Request	11
7.2.1 DocumentId	12
7.2.2 Query	14
7.3 Response	19
7.3.1 DocumentId	20
7.3.2 Query	20
7.3.3 RequestProblem	20
7.3.4 Query Reply	21
8 Transaction Failure	31
8.1 Failures	31
8.2 Unknown Version Problem	32
8.3 Resubmitted Requests	32

9 Profiles	32
Appendix A: (normative) CORE Schema	33
Appendix B: (informative) CORE Use Cases and Scenarios	40
Appendix C: (informative) Security Considerations	44
Appendix D: (informative) Summary of CORE Elements	46
Appendix E: (informative) CORE Document Examples.....	49

Figures

Figure 1: CORE protocol interoperability.....	5
Figure 2: COREDocument element.....	11
Figure 3: Request element.....	11
Figure 4: DocumentId element	12
Figure 5: Query element.....	14
Figure 6: payment-type attribute.....	17
Figure 7: DatePair element.....	18
Figure 8: Response element	20
Figure 9: RequestProblem element.....	21
Figure 10: QueryReply element	22
Figure 11: ErrorRecord object.....	23
Figure 12: AcqRecord element	24
Figure 13: Medium element	25
Figure 14: PaymentDetailsRecord element.....	28
Figure 15: PaymentAmount element.....	31

Foreword

(This foreword is not part of the *Cost of Resource Exchange (CORE) Protocol*, NISO Z39.95-200x. It is included for information only.)

About This Standard

The NISO Cost of Resource Exchange (CORE) project was first proposed by Jeff Aipperspach (Product Manager, Serials Solutions), Ted Koppel (AGent Verso (ILS) Product Manager, Auto-Graphics, Inc.), and Ed Riding (Technical Product Manager, SirsiDynix). They felt that the ability to request financial data (whether for display or for writing to populate an ERM) from an ILS acquisitions system would enable both real-time lookups and cost-per-click and other cost-related reports in the ERMS all the more possible, without the work of manually entering the same data in two different systems. Using defined XML data schemas, they anticipated, a standard would provide a common method of requesting cost-related information from an ILS for a specific electronic resource, within the boundaries of a subscription period. Once defined, implemented and successful, it is likely that this standard could be expanded to include other elements for purposes not yet envisioned.

The CORE Working Group was approved in June 2007 with a three-goal charter:

- 1) To develop and refine the list of data elements exchanged between an Electronic Resource Management System (ERMS), and Integrated Library System (ILS), Business Systems, and other interested parties holding acquisitions metadata to support the population of the ERMS with financial and vendor information in the automated system.
- 2) To create a transport protocol useful in moving these data elements from one system to another.
- 3) To write a small number of use cases that will help all parties understand the capabilities of the protocol.

Previously, in late 2006, as a next step in the progress of the Digital Library Federation's (DLF) Electronic Resource Management Initiative, a subcommittee of the DLF Electronic Resource Management Initiative, Phase II, constituted by Norm Medeiros (Haverford College), Linda Miller (Library of Congress), Adam Chandler (Cornell), and Angela Riggio (UCLA), queried a number of librarians, ERMS suppliers and ILS vendors to discover which elements from an ILS would most likely be required to facilitate cost management within the context of the ERMS. They published their findings in the *White Paper on Interoperability between Acquisitions Modules of Integrated Library Systems and Electronic Resource Management Systems*¹. In that white paper, the authors summarized the data elements critical to exchange in acquisitions as follows:

¹ Medeiros, Norm, et al. *White Paper on Interoperability between Acquisitions Modules of Integrated Library Systems and Electronic Resource Management Systems*. Prepared by a Subcommittee of the Digital Library Federation's Electronic Resource Management Initiative, Phase II. Washington, D.C.: Digital Library Federation, January 2008. Available at: http://www.diglib.org/standards/ERMI_Interop_Report_20080108.pdf

Critical Acquisitions Elements

Element Name	UCLA	Cornell University	Tri-College Consortium	Library of Congress
purchase order number	X	X	X	X
price	X	X	X	X
start/end dates	X	X	X	X
vendor	X	X	X	X
vendor ID	X	X	X	X
invoice number	X	X	X	X
fund code	X	X	X	
invoice date		X	X	X
selector		X	X	
vendor contact information				X
purchase order note		X		
line item note		X		
invoice note		X		

The CORE Working Group built on this work in assembling the Data Dictionary portion of this standard, defining and validating the data elements to be exchanged, specifying which are required and which are optional, as well as specifying the protocol to be used to transport the data from one system to another.

Trademarks, Service Marks

Wherever used in this standard, all terms that are trademarks or service marks are and remain the property of their respective owners.

NISO Voting Members

[added by NISO after final ballot]

NISO Board of Directors

[added by NISO after final ballot]

At the time NISO approved this standard, the following individuals served on its Board of Directors:

, Chair	Directors:
, Vice Chair and Chair-Elect	
, Immediate Past Chair	
, Treasurer	
, Managing Director/Secretary NISO	

Business Information Topic Committee

At the time this Draft Standard was approved for trial use, the following were members of the Business Information Topic Committee:

Kathleen Folger, Co-Chair
University of Michigan Library

Helen Szigeti, Co-Chair
HighWire Press, Stanford University

Ivy Anderson
California Digital Library

John Carlo Bertot
Information Use Management Policy Institute

Johan Bollen
Los Alamos National Laboratory

Denise M. Davis
American Library Association

Herbert Gruttemeier
INIST Institut de l'Information Scientifique et
Technique

Karla Hahn
Association of Research Libraries (ARL)

Helen Henderson
Ringgold

Steve Hiller
University of Washington Libraries

Tim Jewell
University of Washington Libraries

Heather Reid
Copyright Clearance Center

CORE Working Group Members

The following members of the CORE Working Group developed this standard:

Jeff Aipperspach
Serials Solutions, Inc.

Rafal Kasprowski
Rice University

Kathy Klemperer
EDItEUR

Ted Koppel (Co-Chair)
Auto-Graphics, Inc.

Nettie Lagace
Ex Libris, Inc.

Debbie Logan
EBSCO Information Services

Joyce McDonough
Columbia University

Bob McQuillan
Innovative Interfaces, Inc.

Rose Nelson
Colorado Alliance of Research Libraries

Ed Riding (Co-Chair)
SirsiDynix

Dani Roach
University of St. Thomas

Brian Rosmaita
VTLS, Inc.

Clara Ruttenberg
Georgetown University

Gracemary Smulewitz
Rutgers University Libraries

Mary Walker
Wichita State University

Kelvin Watson
The Library Corporation (TLC)

Candy Zemon
Polaris Library Systems

Acknowledgments

The CORE Working Group gratefully acknowledges the substantial contributions made by Mark Wilson in creating the schema files for the protocol.

Cost of Resource Exchange (CORE) Protocol

1 Purpose

The purpose of the Cost of Resource Exchange (CORE) specification is to facilitate the transfer of cost and related library acquisitions information from one automated system to another. This transfer may be from: 1) an Integrated Library System (ILS) acquisitions module (the data source and CORE responder) to an Electronic Resource Management System (ERMS) (the data recipient and CORE requester), both belonging to the same library; 2) a book or serials vendor to a library's ERMS; 3) a transfer of cost and transaction data among members of a consortium; or 4) any transaction partner to another that can benefit from the sharing of cost and library acquisitions-related data.

Using the defined CORE XML data schema, this standard provides a common method of requesting cost-related information by a client application (an ERMS, for example) for a specific order transaction, a specific resource, or all resources that the library owns, within the boundaries of a payment period or access period. The client requester must supply sufficient request information (e.g., a unique order identifier, a date range) in its request, so that the responding system (an ILS, for example) can interpret the request, identify the appropriate financial record(s), and respond with the appropriate financial and/or resource data elements.

2 Scope

2.1 CORE Transactions

The CORE protocol has been generalized in order to be useful for a variety of transaction partners by identifying and defining data elements that are generally supported by ILS, ERMS, subscription agents, and materials vendors.

The CORE protocol defines the semantics for a set of transactions implied by three defined CORE use cases (see [Appendix B](#)). While not every CORE implementation may require the full set of use cases, the design supports all three.

An XML schema conveys CORE protocol transactions (see [Appendix A](#)). The schema, using CORE-specific derived data types (see [Section 6.3](#)), frequently restricts the form and value of data components. The schema presented supports single-query requests with possible multi-part replies.

2.2 Transport Mechanism

The CORE protocol describes a data structure and not the transport mechanism. While the NISO CORE Working Group recommends SOAP be used as the Web Services transport mechanism, the specifics of the SOAP configuration are beyond the scope of the CORE standard and are left to trading partners to devise. Implementers of the CORE protocol may also use other transport mechanisms that are already in place, such as FTP or SMTP, or even physical transfer media.

2.3 Out of Scope

It is not the intent of this standard to specify or restrict whether the request shall be initiated automatically or manually, nor whether the data provided in the response should be used for

purposes of display or data population. The standard is only intended to specify the data elements and a schema used in such an exchange.

CORE, by definition, is transmitting financial information and, therefore, data transferred in the CORE message should be considered confidential between the two parties. CORE, being the payload, does not include methods for secure transmission. Some suggested methods for securing the CORE payload are provided in [Appendix C](#).

3 References

This standard references the following documents. When cited in the text of the standard, the standard may be referred to by its number only or an abbreviated title. Where no date is supplied, the most current version of the standard should be used.

ISO 4217, *Codes for the representation of currencies and funds*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

4 Definitions

The following terms, as used in this standard, have the meanings indicated.

<u>Term</u>	<u>Definition</u>
access period	<p>The range of dates for which access to a Product (resource) is provided.</p> <p>Example: A 2009 invoice payment was for the access period of August 1, 2009 until July 31, 2010.</p> <p>Also referred to as subscription period.</p> <p>Compare to payment period.</p>
attribute	<p>In XML, a name-value pair within a tagged XML element that modifies certain features of the element.</p> <p>NOTE: The allowable attributes for the elements described in the CORE protocol are listed and defined in Table 4.</p>
Boolean	<p>An element that uses the logical value of either “true” or “false”.</p>
client	<p>In client/server software architecture, the system that makes requests.</p> <p>NOTE: The actual requesting application may physically reside on a server, even though it referred to as the client.</p>
cost	<p>Amount paid.</p> <p>NOTE 1: In CORE, the term payment is used to refer to cost.</p> <p>NOTE 2: Cost as a data element does not stand alone; it must be qualified as “cost for this transaction” or “cost for this subscription period.”</p>
currency	<p>The system of money used, such as the U.S. dollar or the EURO.</p>

<u>Term</u>	<u>Definition</u>
data type	<p>The type of data that can be represented in the value of the data element.</p> <p>NOTE: The allowable data types for the elements described in the CORE protocol are listed and defined in Table 3. The character set used is UTF-8, the default for XML.</p>
date range	<p>The series of dates as specified by a start date and an end date.</p> <p>NOTE: See the sections on BeginDate (7.2.2.5.1.1.1) and EndDate (7.2.2.5.1.1.2) for guidelines on date range entry, particularly when either start or end date is indefinite.</p>
element	<p>A block of text or data in an XML document that is made up of a start and end tag, and the content between the tags.</p> <p>NOTE: The Data Dictionary, Section 7, of this standard defines the CORE XML schema elements.</p>
ERM (Electronic Resource Management [system])	<p>A database and software system dedicated to handling the cataloging, licensing, accessing, and other aspects of electronic resource use.</p>
fiscal year	<p>The budget year of the organization acquiring a resource.</p> <p>NOTE: In the CORE schema, the PaymentPeriod element (7.2.2.5.2) generally corresponds to fiscal year.</p>
FTP (File Transfer Protocol)	<p>An Internet protocol used for transferring files from one computer to another through a network.</p>
fund code	<p>A code used by a library to identify which fund or budget is associated with a payment.</p> <p>NOTE: See the CORE element, LibraryFundCode (7.3.4.2.16).</p>
identifier	<p>A unique “name” for an object to differentiate it from other objects.</p> <p>NOTE 1: The identifier syntax is usually defined by a national or international standard and assigned through some type of registration agency.</p> <p>NOTE 2: An object may have more than one unique identifier, each of which identifies it for a different purpose.</p>
ILS (Integrated Library System)	<p>A set of automated library services—such as an online catalog, circulation management, and serial tracking—that share a common database.</p>
number	<p>A numerical (or alphanumeric) identifier that may or may not be unique, and is often only relevant or unique within the organization that assigned it.</p> <p>NOTE: See for example PONumber (7.3.4.2.7) and InvoiceNumber (7.3.4.2.11) in the CORE schema.</p>
payment	<p>The amount paid for a resource.</p> <p>NOTE: CORE is designed to transmit payment information, not prices or estimates.</p>

<u>Term</u>	<u>Definition</u>
payment period	<p>The range of dates in which the product (resource) is paid. Often synonymous with fiscal year.</p> <p>Example: Provide a list of all invoices paid during the Payment Period of January 1, 2009 through December 31, 2009.</p> <p>Compare to access period.</p>
product	<p>The specific information object or collection of objects, in any format, that is acquired or licensed. Also called a resource.</p>
request	<p>An XML-formatted message sent from one computer application (client) to another computer application (server), containing a query for which it expects to receive a response.</p>
resource	<p>An information object or collection of objects, in any format, that is acquired or licensed.</p> <p>Examples: books, journals, e-journals, databases.</p> <p>NOTE: In the CORE protocol, the resource is what is being paid for in the transaction and is referred to as the product.</p>
response	<p>An XML-formatted message sent from one computer application (server) to another computer application (client) containing the information specified in a request.</p>
server	<p>In a client/server software architecture, the system that responds to requests.</p>
SMTP (Simple Mail Transfer Protocol)	<p>A protocol to allow mail servers to exchange messages with each other.</p>
SOAP (Simple Object Access Protocol)	<p>A web service protocol that specifies a self-contained messaging system used to exchange data and access services across a network (commonly the Internet).</p>
subscription period	<p>The period of time (expressed as a start date and end date) that is covered by a payment for a subscription.</p> <p>NOTE: In CORE, the element AccessPeriod reflects the subscription period.</p>
transaction	<p>A complete CORE exchange, comprised of a Request and a Response.</p>
transaction partner	<p>One of two entities exchanging CORE messages.</p>
XML (Extensible Markup Language)	<p>A specification of the World Wide Web Consortium (W3C) for creating custom mark-up languages that enable information systems to share structured information.</p>

5 CORE Schema Synopsis

5.1 General

The CORE protocol acts as a standardized interface between two systems that exchange financial information, such as an ILS and an ERMS. A system on either end of the exchange needs to create a one-time interface to the CORE protocol and can then exchange data with any other CORE-compliant system, as illustrated in [Figure 1](#).

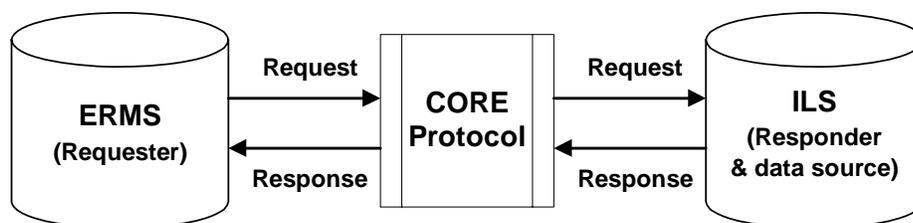


Figure 1: CORE protocol interoperability

The CORE protocol supports single-query requests with possible multi-part replies in either a client-server or peer-to-peer architecture. The basic element of the protocol is the COREDocument; all elements within the schema are children of the COREDocument element.

The schema uses an object-oriented approach and is deliberately terse to encourage rapid implementation and lightweight profiles. Placing each major unit in its own sub-schema promotes schema reading and editing while enhancing reusability. Implementers may unit-test components with a validating parser by building an XML document from any of the sub-schemas. The entire schema has been kept as flat as possible: there are no unnecessary elements.

The individual elements in the schema are defined in Section 7, [CORE Data Dictionary](#). The data types and attributes used in the elements are defined in Section 6, [Data Dictionary Reference Guide](#). The CORE XML schema is in [Appendix A](#) and is available in an xsd file. All CORE xsd files referenced in this standard are available on the NISO website (www.niso.org/schemas/core). A summary of the CORE elements is found in [Appendix D](#).

5.2 CORE Namespace

The CORE namespace is:

`http://www.niso.org/schemas/core/[version no.]`

The version element, [version no.], following the namespace designates the version of the CORE schema in use.

5.3 CORE Transactions

A COREDocument presents either a Request, containing a Query from a Requester to a Responder, or presents a Response, containing a QueryReply, from the Responder to the Requester. In the event of a transaction problem, the responder may substitute a RequestProblem in place of the QueryReply. In the event that the responder is unable to supply the requested information, an ErrorRecord is sent in the QueryReply.

The schema is generalized to allow information exchange between different types of systems or for different purposes. Many of the actual values used, e.g., OrderId or ProductId, are determined through agreements of the transaction partners.

Example:

In a CORE exchange between an Integrated Library System (ILS) and an Electronic Resource Management system (ERM), the ProductId may be a local ERMS control number, while in a CORE Exchange between an ILS and a serials vendor, the ProductId may be a publisher's or an agent's product number.

5.4 Request Design

The use cases (see [Appendix B](#)) require precisely two transaction partners: the requester who presents a query, and the responder who replies to that query. In the case where the exchange is between an ILS and ERMS, the same organization may be both parties to the transaction.

The CORE single-query model substantially reduces transaction and implementation complexity by limiting requests to a single query. If information on multiple orders or multiple products is needed, individual requests for each order/product must be sent. The initiating system should be able to easily create such multiple request records, and the responding system should be able to receive and queue multiple incoming requests.

The CORE protocol supports a single request, intended to retrieve a record that contains fields related to the product information, order transaction, and payment details. The requester may supply either a specific OrderId, a specific ProductId, or request all records, qualified by either access period (subscription period) or payment period (e.g., fiscal year) date range and a CustomerId.

5.5 Reply Design

CORE's design permits retrieval of single transactions, multiple transactions that meet the query qualifiers, or the transfer of a complete "dump" of transaction data. The degree of inclusion in the data returned is governed by what is requested in the query.

The response will return a CORE Response record with payment detail, order transaction, and product fields. For any given requested OrderId or ProductId, there may be multiple PaymentDetailsRecords contained within the CORE Response, depending on how many payments were made during the date periods specified in the query. A query request for all records, whether date qualified or not, can result in a response with multiple transaction records (AcqRecords), each of which could contain multiple PaymentDetailsRecords.

The RequestProblem exception is designed to let the responder signal the requester that the reply has failed. In such instances, one of six **problem** attributes is provided to explain the problem.

When there is no overall problem with the entire transaction, the responder must return a QueryReply. The responder must construct all multi-part replies in a single Response transaction. Some of the data elements defined in the standard may not be available in all responding applications (servers). This standard requires that a responding server deliver the most complete CORE data structure as is possible.

If the responder is unable to return any of the requested data (for example, the OrderId is invalid), an ErrorRecord is sent as the QueryReply with one of four **error** attributes. An ErrorRecord may not appear in a multi-part reply.

6 Data Dictionary Reference Guide

6.1 Element Listings

The Data Dictionary section of this standard (Section 7) defines the XML schema elements of the CORE protocol. For each element, a table is provided that describes and defines the element, as shown in [Table 1](#).

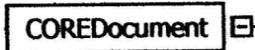
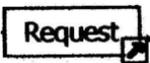
Table 1: Description of Element tables

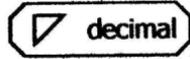
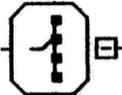
Child Elements	Elements that are sub-elements in the schema to the element described in that section. "None" indicates there are no children elements.
Description	A description of the element's purpose.
Requirement	An indication of whether the element is: Mandatory Mandatory if applicable [applicability is explained in the usage notes] Optional
Repeatability	Indicates whether the element can be repeated within the scope of its parent element and how many times it may be repeated. Values used are: 0 - 1 = may either not occur at all or occur once 1 = must occur one and only one time 0 – n = may either not occur at all or occur multiple times 1 – n = must occur at least once and may occur multiple times
Data Type	Specifies the type of data that is allowed for this element. See Section 6.3 for further information on data types.
Attribute Used	Indicates which attribute, if any, is associated with this element. "N/A" indicates that attributes are not applicable to the element.
Example(s)	Provides one or more examples of valid element content.
Usage Notes	Provides additional information on how this element is intended to be used.

6.2 Graphic Representations

The elements used in the CORE protocol and their relationships are graphically depicted in the Data Dictionary section of this standard (Section 7). The shapes and symbols used and their meanings are shown in [Table 2](#).

Table 2: Graphic representations

Shape / Symbol	Meaning
	An element with the name that is listed in the box.
	An element with the name that is listed in the box and that has additional children. Refer to the diagram for that element for the additional information.

Shape / Symbol	Meaning
	An attribute (see Section 6.4) with the name that is listed in the box.
	A data type (see Section 6.3) with the name that is listed in the box.
	A connector indicating the element is an XML "complex type", i.e., it is generally a container for other elements.
	A connector indicating that a choice must be made between the elements to the right of the connector.
	A connector indicating a group of elements in sequence to the right of the connector. Any or all of the elements may be present depending on the occurrences noted.
0..∞	An occurrence and repeatability indicator. In this example, the element may occur from zero to an infinite number of times.

6.3 Data Types

The allowable data types for the elements described in the CORE protocol are listed and defined in [Table 3](#). The character set used is UTF-8, the default for XML.

Table 3: CORE protocol data types

Data Type	Description
Boolean	An element that uses the logical value of either "true" or "false".
container	An element whose role is to be the parent of other elements but contains no data of its own. In some instances, a container may have an attribute.
core-string	A sequence of alphanumeric characters. It cannot be empty and must contain at least one character. Derived from the XML <i>string</i> data type. Although XML allows a blank character, use of a single blank is not recommended in CORE. The <i>core-string</i> data type is used in a number of elements, chiefly names or identifiers.
decimal	Real numbers, which can be represented by decimal numerals.
enumerated	Strings that are restricted to a particular list of values.
iso-currency	The data type used for all currency data in the schema. It is restricted to precisely three uppercase letters to conform to the ISO currency code standard (ISO 4217).

Data Type	Description
iso-date	The data type used for all dates in the schema. Values are restricted to ISO formatted dates (see ISO 8601) between January 1, 1900 and December 31, 2099 using the format CCYYMMDD. <i>NOTE: The date of February 29 will validate in XML for non-leap years, even though it is not a real date.</i>
positive integer	A positive integer that ranges from the numeral "1" upward.

6.4 Attributes

The CORE schema uses the attributes, defined in [Table 4](#), that may be applicable to one or more elements. The element tables in the Data Dictionary (Section [7](#)) indicate if an element uses an attribute.

Table 4: CORE protocol attributes

Attribute	Description	Data Type
currency-code	Signifies the currency type for monetary data.	iso-currency
error	Restricted to one of four values: <ul style="list-style-type: none"> - i nval i d-peri od - unknown-order-i d - unknown-product-i d - record-not-avai l abl e 	enumerated
form	Restricted to one of three values: <ul style="list-style-type: none"> - physi cal - onl i ne - both 	<i>enumerated</i>
payment-type	Restricted to one of three values: <ul style="list-style-type: none"> - unknown - conti nuous - oneti me 	enumerated
problem	Restricted to one of six values: <ul style="list-style-type: none"> - system-probl em - unknown-versi on - servi ce-refused - unknown-requester - unknown-responder - transacti on-i d-probl em 	enumerated

Attribute	Description	Data Type
transaction-id	A unique value assigned by the requester for each transaction. Binding the transaction-id to the Requester assures globally unique Requester/ transaction-id pairs across all transactions. A well-behaved implementation might simply increment the value of the transaction-id for each new transaction.	core-string
version	Identifies the version of the protocol invoked for a transaction. NOTE 1: The version number is unspecified in the schema to promote version resolution by providing parsers an opportunity to present the attribute to an implementation. Were the version number specified in the schema itself, a parser might not be able to pass the attribute to the implementation. NOTE 2: A proposed behavior for version resolution is discussed under Profiles (see Section 8).	decimal

7 CORE Data Dictionary

This Data Dictionary defines all the elements found in the CORE protocol. The XML schema for the CORE protocol that uses these elements is found in [Appendix A](#) and on the NISO website at: www.niso.org/schemas/core.

7.1 COREDocument

Child Elements	Request (7.2) Response (7.3)
Description	The COREDocument element wraps every transaction, thus providing for early recognition of non-protocol documents and version conflicts. The COREDocument element consists of a version attribute and a choice of either a Request element or a Response element, as illustrated in Figure 2 . The Request element initiates a transaction by presenting a query. The Response element completes the transaction by replying to the query. The version attribute identifies the version of the protocol invoked.
Requirement	Mandatory
Repeatability	1
Data Type	decimal
Attribute Used	version
Example(s)	version="0.1"

<p>Usage Notes</p>	<p>Only one of the child elements (Request or Response) may be present and one must be present.</p> <p>A COREDocument either presents a Request (7.2), containing a Query (7.2.2), to a responder, or presents a Response (7.3), containing a QueryReply (7.3.4), to a requester. In the event of a transaction problem, the responder may substitute a RequestProblem (7.3.3) in place of the QueryReply.</p> <p>See also Section 8, Transaction Failure.</p> <p>A sample of CORE Documents is in Appendix E.</p>
---------------------------	--

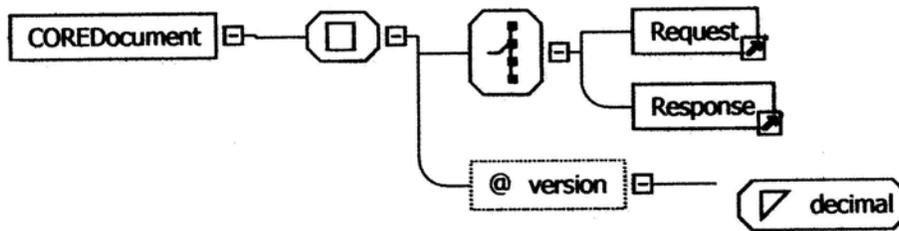


Figure 2: COREDocument element

7.2 Request

<p>Child Elements</p>	<p>DocumentId (7.2.1) Query (7.2.2)</p>
<p>Description</p>	<p>The choice of a Request element in the COREDocument initiates a transaction. See Figure 3.</p> <p>Identifying information and the actual query are carried in the two child elements.</p>
<p>Requirement</p>	<p>Mandatory if applicable. (Applicable if initiating a Request as opposed to a Response.)</p>
<p>Repeatability</p>	<p>1</p>
<p>Data Type</p>	<p>Container</p>
<p>Attribute Used</p>	<p>N/A</p>
<p>Usage Notes</p>	<p>Both child elements must be present.</p> <p>See also Section 8, <i>Transaction Failure</i>, which describes when a Request may need to be resubmitted.</p>

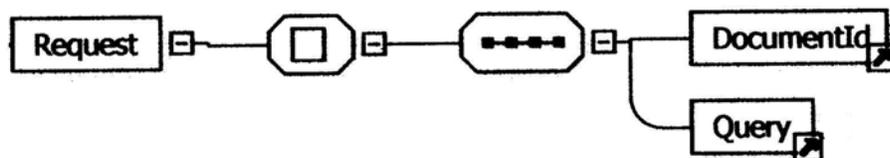


Figure 3: Request element

The complete hierarchy of elements for the Request element can be found in the summary table in [Appendix D](#).

7.2.1 DocumentId

Child Elements	Requester (7.2.1.1) Responder (7.2.1.2)
Description	The DocumentId element ties together a transaction's Request and Response. It also distinguishes between a series of transactions involving the same parties by associating each transaction with a unique transaction identifier and identifies the expected responder of the request. See Figure 4 .
Requirement	Mandatory
Repeatability	1
Data Type	Container
Attribute Used	transaction-id
Example(s)	transaction-id="12345"
Usage Notes	Both child elements, Requester and Responder, must be present. Requesters must assign a unique value to the transaction-id attribute for each transaction. A well-behaved implementation might simply increment the value of the transaction-id for each new transaction. Binding the transaction-id to the Requester assures globally unique Requester/ transaction-id pairs across all transactions.

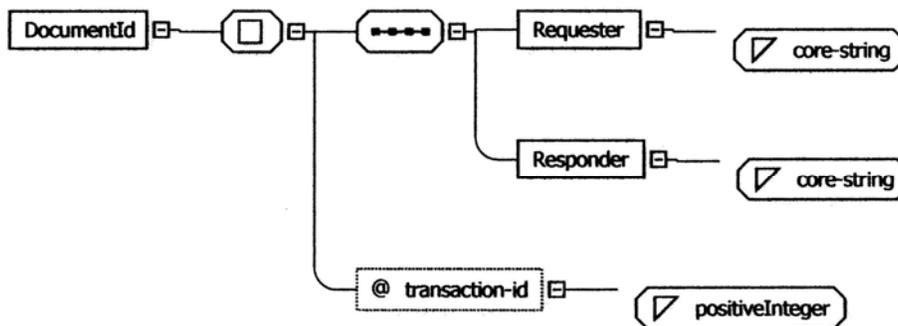


Figure 4: DocumentId element

7.2.1.1 Requester

Child Elements	None
Description	A unique name or identifier for the party requesting data. See Figure 4 .
Requirement	Mandatory

Repeatability	1
Data Type	core-string
Attribute Used	N/A
Example(s)	Scottsdale Public Library UPgh12345 library@university.edu American Serials Agent Services Inc.
Usage Notes	<p>The specific type of identifier used should be agreed upon between transaction partners.</p> <p>When exchanging CORE schema between two different organizations, the Requester may be used as a security element, i.e., that a valid Requester is asking for the data.</p> <p>The Requester element might, or might not, coincide with the transport protocol's address. For instance, if the transport protocol used were SMTP, the transport protocol's addressing mechanism would be e-mail addresses.</p> <p>A profile could choose to use the same type of identifier for the Responder and the Requester, or could choose to use entirely different kinds of identifiers (see Section 9 for more on profiles).</p>

7.2.1.2 Responder

Child Elements	None
Description	A unique name or identifier for the party receiving the request who is expected to return the requested data. See Figure 4 .
Requirement	Mandatory
Repeatability	1
Data Type	core-string
Attribute Used	N/A
Example(s)	Provider12345 http://www.domain.com provider@domain.com Library Consortium of East Florida
Usage Notes	<p>The specific type of identifier used for this element should be agreed upon between transaction partners.</p> <p>The Responder element might, or might not, coincide with the transport protocol's addresses. For instance, if the transport protocol used were SMTP, the transport protocol's addressing mechanism would be e-mail addresses.</p> <p>A profile could choose to use the same type of identifier for the Responder and the Requester, or could choose to use entirely different kinds of identifiers (see Section 9 for more on profiles).</p>

7.2.2 Query

Child Elements	<p>OrderId (7.2.2.1) ProductId (7.2.2.2) RequestAll (7.2.2.3) CustomerId (7.2.2.4) Period (7.2.2.5)</p>
Description	<p>The Query element specified the data that is to be returned. It identifies the specific order or product for which information is desired (or all data), and the contextual date-related information, if applicable. See Figure 5.</p>
Requirement	Mandatory
Repeatability	1
Data Type	Container
Attribute Used	N/A
Usage Notes	<p>The Query must use one and only one of these child elements: OrderId (7.2.2.1), ProductId (7.2.2.2), or RequestAll (7.2.2.3). Use of OrderId or ProductId indicates a request for data specific to that order or product. Either may be qualified by Period (7.2.2.5) dates. Use of RequestAll indicates a request for all data, optionally qualified by Period (7.2.2.5) dates.</p> <p>If data for more than one order or product is desired (but not all records), each OrderId or ProductId must be submitted in a separate Request.</p>

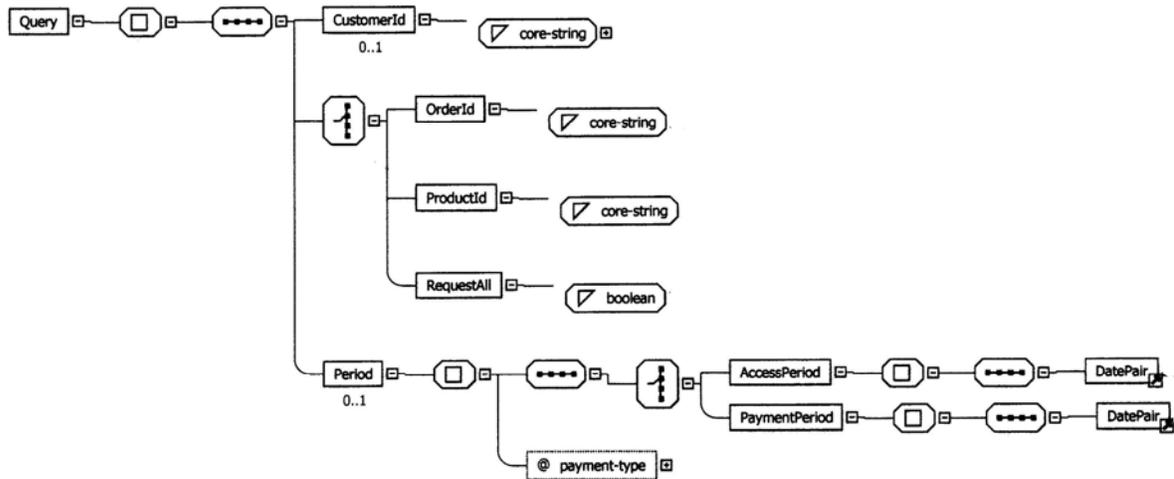


Figure 5: Query element

7.2.2.1 OrderId

Child Elements	None
Description	A unique identifier for the particular order for which information is requested. See Figure 5 .
Requirement	Mandatory if applicable. One of either OrderId, ProductId, or RequestAll must be used.
Repeatability	0-1
Data Type	core-string
Attribute Used	N/A
Example(s)	PQ-234
Usage Notes	<p>The identifier to be used as OrderId should be agreed upon between the transaction partners. The OrderId could, for example, be the customer PO number, a vendor assigned order number, or a publishers' order id.</p> <p>Use OrderId when data is needed related to a specific order transaction. If OrderId is used, then neither ProductId (7.2.2.2), nor RequestAll (7.2.2.3) can be used in the same Query.</p> <p>The OrderId may be qualified by the Period (7.2.2.5) element.</p> <p>Only one OrderId may be supplied per Request. If data for more than one order is desired (but not all orders), each OrderId must be submitted in a separate Request. Use RequestAll instead of OrderId if <u>all</u> order records are desired.</p>

7.2.2.2 ProductId

Child Elements	None
Description	A unique identifier for the particular product for which information is requested. See Figure 5 .
Requirement	Mandatory if applicable. One of either OrderId, ProductId, or RequestAll must be used.
Repeatability	0-1
Data Type	core-string
Attribute Used	N/A
Example(s)	12345/444443 88-43265 ISSN 0393-6643

Usage Notes	<p>The identifier to be used as ProductId should be agreed upon between the transaction partners.</p> <p>Use ProductId when data is needed related to a specific product. If ProductId is used, then neither OrderId (7.2.2.1) nor RequestAll (7.2.2.3) can be used in the same Query.</p> <p>The ProductId may be qualified by the Period (7.2.2.5) element.</p> <p>Only one ProductId may be supplied per Request. If data for more than one product is desired (but not all products), each ProductId must be submitted in a separate Request. Use RequestAll instead of ProductId if all product records are desired.</p>
--------------------	---

7.2.2.3 RequestAll

Child Elements	None
Description	Signifies that all transaction records (as optionally date qualified) are being requested. See Figure 5 .
Requirement	Mandatory if applicable. One of either OrderId, ProductId, or RequestAll must be used.
Repeatability	0-1
Data Type	Boolean
Attribute Used	N/A
Usage Notes	<p>Use RequestAll when all data is being requested that matches the optional Period (7.2.2.5) dates.</p> <p>When RequestAll is used it has the fixed Boolean value of "true". If RequestAll is used, then neither OrderId (7.2.2.1) nor ProductId (7.2.2.3) can be used in the same Query.</p> <p>RequestAll may be qualified by the Period (7.2.2.5) element.</p>

7.2.2.4 CustomerId

Child Elements	None
Description	A unique identifier for the particular customer for whom information is requested. See Figure 5 .
Requirement	Optional
Repeatability	0-1
Data Type	core-string
Attribute Used	N/A
Example(s)	76-333 Customer 4875644

Usage Notes	<p>The identifier to be used as CustomerId should be agreed upon between transaction partners. It is recommended that this be an identifier and not a name.</p> <p>The CustomerId may or may not be the same as the Requester. Use of the CustomerId is a method by which, for example, consortia could request data for a particular member or a library system with multiple libraries could request data for a particular library.</p> <p>Only one CustomerId may be supplied per Request. If data for more than one customer is desired, each CustomerId must be submitted in a separate Request.</p>
--------------------	---

7.2.2.5 Period

Child Elements	<p>AccessPeriod (7.2.2.5.1) PaymentPeriod (7.2.2.5.2)</p>
Description	<p>The subscription/availability or payment period context. See Figure 5.</p>
Requirement	<p>Optional</p>
Repeatability	<p>0-1</p>
Data Type	<p>Container</p>
Attribute Used	<p>payment-type (See 6.4.)</p>
Example(s)	<p>payment-type="onetime" payment-type="continuous" payment type="unknown"</p>
Usage Notes	<p>The payment-type attribute is restricted to three values: as described in Section 6.4, and illustrated in Figure 6. The default value is "unknown".</p> <p>The use of Period is recommended when large data sets are expected. Multiple requests can be sent with different period qualifiers to reduce the size and processing time of the individual responses.</p>

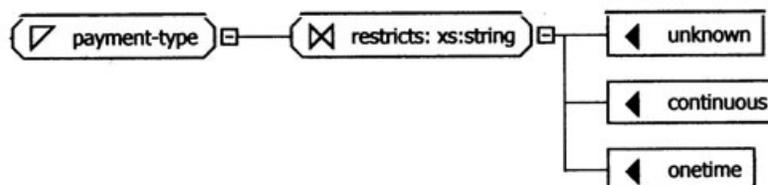


Figure 6: payment-type attribute

7.2.2.5.1 AccessPeriod

Child Elements	<p>DatePair (7.2.2.5.1.1)</p>
Description	<p>Time period of the subscription for or the availability of the product covered by the Query. See Figure 5.</p>

Requirement	Mandatory if applicable.
Repeatability	1
Data Type	Container
Attribute Used	N/A
Usage Notes	If the optional Period element is used, then one and only one of either AccessPeriod or PaymentPeriod must be present.

7.2.2.5.1.1 DatePair

Child Elements	BeginDate (7.2.2.5.1.1.1) EndDate (7.2.2.5.1.1.2)
Description	The beginning and ending dates of the date range requested. See Figure 7 .
Requirement	Mandatory
Repeatability	1
Data Type	Container
Attribute Used	N/A
Usage Notes	Both the beginning and ending date child elements must be present.

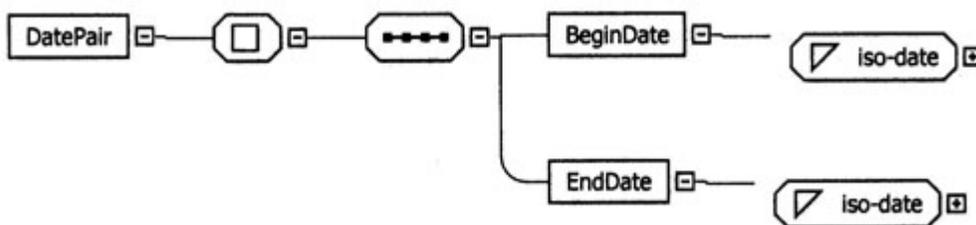


Figure 7: DatePair element

7.2.2.5.1.1.1 BeginDate

Child Elements	None
Description	The beginning date of the requested period. See Figure 7 .
Requirement	Mandatory
Repeatability	1
Data Type	iso-date
Attribute Used	N/A
Example(s)	20080101
Usage Notes	Values are restricted to dates between January 1, 1900 and December 31, 2099 using the format CCYYMMDD. If the beginning date is unknown or unspecified, use "19000101", the earliest possible begin date.

7.2.2.5.1.1.2 EndDate

Child Elements	None
Description	The ending date of the requested period. See Figure 7 .
Requirement	Mandatory
Repeatability	1
Data Type	iso-date
Attribute Used	N/A
Example(s)	20080131
Usage Notes	Values are restricted to dates between January 1, 1900 and December 31, 2099 using the format CCYYMMDD. For open-ended access periods, or when the end date is unknown or unspecified, use "20991231", the latest possible end date.

7.2.2.5.2 PaymentPeriod

Child Elements	DatePair (7.2.2.5.2.1)
Description	Time period of the payment(s) that is being requested. See Figure 5 .
Requirement	Mandatory if applicable. Either AccessPeriod or PaymentPeriod must be used if Period is present.
Repeatability	No
Data Type	Container
Attribute Used	N/A
Usage Notes	If the optional Period element is used, then one and only one of either AccessPeriod or PaymentPeriod must be present. See DatePair (7.2.2.5.1.1), BeginDate (7.2.2.5.1.1.1), and EndDate (7.2.2.5.1.1.2) for how to use these elements in the PaymentPeriod.

7.3 Response

Child Elements	DocumentId (7.2.1) Query (7.2.2) RequestProblem (7.3.3) QueryReply (7.3.4)
Description	The choice of a Response element in the COREDocument completes a transaction initiated by a Request. A repeat of the query and the reply are carried within the Response container. See Figure 8 .

Requirement	Mandatory if applicable (Applicable if sending a Response as opposed to a Request.)
Repeatability	1
Data Type	Container
Attributes Used	N/A
Usage Notes	The DocumentId (7.2.1) and the Query (7.2.2) must be present. Additionally, one and only one of either RequestProblem (7.3.3) or QueryReply (7.3.4) must be present. See Section 8, <i>Transaction Failure</i> , for further information on when and how RequestProblem is used.

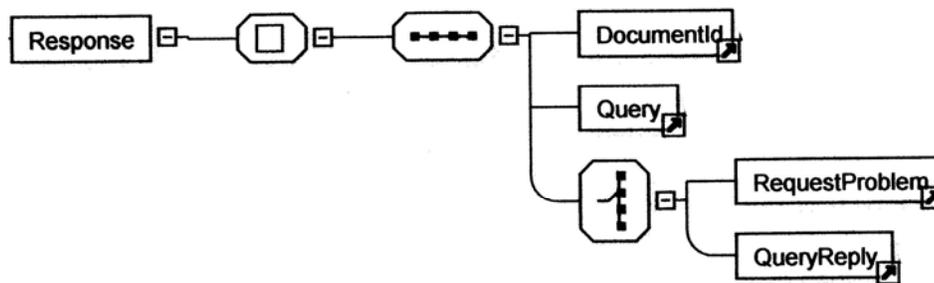


Figure 8: Response element

The complete hierarchy of elements for the Request element can be found in the summary table in [Appendix D](#).

7.3.1 DocumentId

The Response's DocumentId element, an exact copy of the DocumentId element from the transaction's corresponding Request, ties the transaction's Response to a Query.

Refer to Section [7.2.1](#) for the details on this element.

7.3.2 Query

This portion of the Response must repeat the entire Request Query that was sent to provide context for the Response.

7.3.3 RequestProblem

Child Elements	None
Description	Indicates conditions that impact the ability of the responder to process the transaction. See Figure 9 .
Requirement	Mandatory if applicable.
Repeatability	1
Data Type	contains an enumerated attribute
Attribute Used	problem
Example(s)	problem="unknown-requester" problem="transaction-id-problem"

<p>Usage Notes</p>	<p>One and only one of QueryReply or RequestProblem is returned. If for any reason the responder is unable to respond, considers the entire transaction a problem, or finds a problem in the Request's DocumentId (7.2.1) element, the RequestProblem element is returned. Otherwise, a QueryReply (7.3.4) element is returned. Problems in responding to the particular query criteria are addressed by the ErrorRecord (7.3.4.1).</p> <p>The problem attribute is selected from one of six values as defined in Section 6.4 and illustrated in Figure 9.</p> <p>The value unknown-version is returned when the responder cannot construct a COREDocument of the appropriate version. While the protocol itself cannot address behavior upon receiving an unknown version, a profile may suggest the appropriate behavior.</p> <p>A responder may return the value service-refused to a requester with whom no agreement to exchange records is in force.</p> <p>If within the DocumentId element a problem is detected, there are several options available. If the implementation does not recognize the Requester, an unknown-requester should be returned. If the implementation does not recognize the Responder (e.g., the request was sent to the wrong responder), an unknown-responder should be returned. If the transaction-id duplicates one already active, a transaction-id-problem value should be returned to alert the requester to transaction ambiguities.</p> <p>The value system-problem is returned if a reply cannot be constructed because of any system problems, such as when a database is temporarily unavailable for a CORE request. The requester should resubmit the query at a later (unspecified) time. When and how often resubmissions are done is a profile issue. A well-behaved requester implementation must provide a new transaction-id for the resubmitted Request.</p>
---------------------------	---

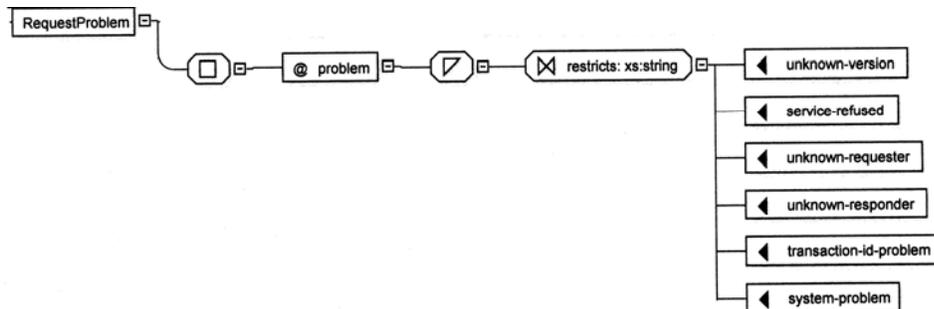


Figure 9: RequestProblem element

7.3.4 Query Reply

<p>Child Elements</p>	<p>ErrorRecord (7.3.4.1) AcqRecord (7.3.4.2)</p>
<p>Description</p>	<p>The QueryReply element contains the specific report data that corresponds to the transaction's query criteria. See Figure 10.</p>

Requirement	Mandatory if applicable.
Repeatability	1
Data Type	Container
Attribute Used	N/A
Usage Notes	<p>One and only one of QueryReply or RequestProblem is returned. QueryReply is returned unless there is a problem with the entire transaction.</p> <p>If there is no problem with the transaction but there is a problem in responding to the specific Query criteria, then an ErrorRecord is returned within the QueryReply.</p> <p>If there is no error, then one or more AcqRecords are returned.</p>

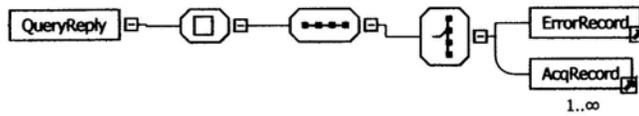


Figure 10: QueryReply element

7.3.4.1 ErrorRecord

Child Elements	None
Description	<p>If there is no problem with the transaction as a whole but the particular query request cannot be fulfilled, then the ErrorRecord substitutes for payload records that cannot be returned.</p> <p>See Figure 11.</p>
Requirement	Mandatory if applicable.
Repeatability	0-1
Data Type	contains an enumerated attribute
Attribute Used	error
Example(s)	<p>error="service-not-supported"</p> <p>error="unknown-order-id"</p>

<p>Usage Notes</p>	<p>The ErrorRecord address issues connected with the Query itself rather than with the whole transaction. Transaction problems are addressed by the RequestProblem element (7.3.3).</p> <p>If an ErrorRecord appears, then no AcqRecord is present in the QueryReply.</p> <p>An ErrorRecord's error attribute contains one of four values as defined in 6.4.</p> <p>The invalid-period, the unknown-order-id, and the unknown-product-id errors are sent to report invalid data in a Query element. The invalid-period error would generally indicate a problem with the syntax of the dates, i.e., the correct date format was not used, or when the begin and end dates conflict, e.g., the end date is before the begin date.</p> <p>The record-not-available error is returned when there are no available records that match the query. If the Request had specified a Period, the requester may consider resubmitting a query with a different Period.</p>
---------------------------	---

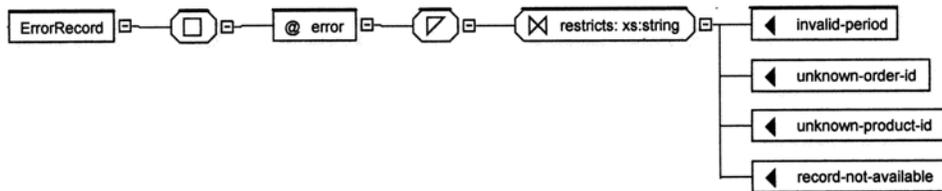


Figure 11: ErrorRecord object

7.3.4.2 AcqRecord

<p>Child Elements</p>	<p>ProductId (7.3.4.2.1) ProductName (7.3.4.2.2) Publisher (7.3.4.2.3) Medium (7.3.4.2.4) Platform (7.3.4.2.5) Vendor (7.3.4.2.6) PONumber (7.3.4.2.7) CustomerId (7.3.4.2.8) ConsortiumId (7.3.4.2.9) PaymentDetailsRecord (7.3.4.2.10)</p>
<p>Description</p>	<p>The full record containing all relevant order, product, and payment information. See Figure 12.</p>
<p>Requirement</p>	<p>Mandatory if applicable.</p>
<p>Repeatability</p>	<p>0-n</p>
<p>Data Type</p>	<p>Container</p>
<p>Attribute Used</p>	<p>N/A</p>

Usage Notes	The AcqRecord is the report payload that must be returned unless an ErrorRecord is returned.
--------------------	--

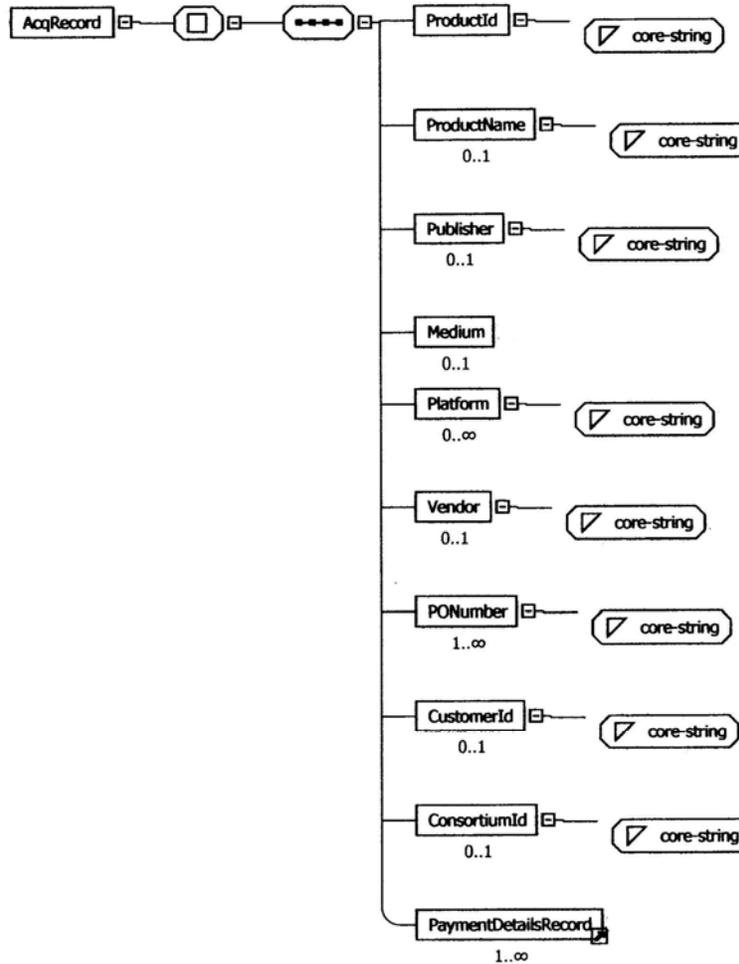


Figure 12: AcqRecord element

7.3.4.2.1 ProductId

The ProductId is a unique identifier for the particular product for which information is requested. See [Figure 12](#).

The ProductId in the Response is mandatory and should match the ProductId in the Query, if one was supplied. See Section [7.2.2.2](#) for the details on this element.

7.3.4.2.2 ProductName

Child Elements	None
Description	The full name of the product. See Figure 12 .
Requirement	Optional
Repeatability	0-1
Data Type	core-string

Attribute Used	N/A
Example(s)	Journal of Information Science Historical Abstracts
Usage Notes	While optional, the use of this element is recommended as it is more descriptive than the ProductId.

7.3.4.2.3 Publisher

Child Elements	None
Description	The name of the organization that published the product. See Figure 12 .
Requirement	Optional
Repeatability	0-1
Data Type	core-string
Attribute Used	N/A
Example(s)	University of Chicago Index to Jewish Periodicals
Usage Notes	See also Platform (7.3.4.2.5) and Vendor (7.3.4.2.6) to correctly distinguish the data that belongs in each of these three related elements.

7.3.4.2.4 Medium

Child Elements	None
Description	An indicator of the product's medium or format. See Figure 12 .
Requirement	Optional
Repeatability	0-1
Data Type	Contains an enumerated attribute.
Attribute Used	form
Example(s)	form="online"
Usage Notes	The form attribute can be one of three values as described in Section 6.4 and illustrated in Figure 13 .

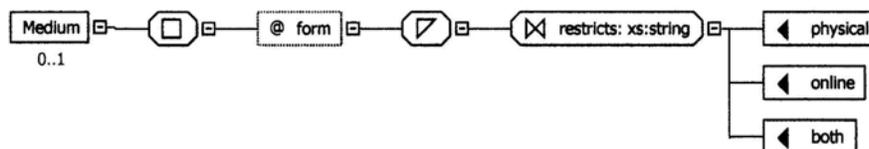


Figure 13: Medium element

7.3.4.2.5 Platform

Child Elements	None
Description	The name of the access provider of the product. See Figure 12 .
Requirement	Optional
Repeatability	0-n
Data Type	core-string
Attribute Used	N/A
Example(s)	HighWire EbscoHost
Usage Notes	This may or may not be the same as the publisher or the vendor. See also Publisher (7.3.4.2.3) and Vendor (7.3.4.2.6) to correctly distinguish the data that belongs in each of these three related elements.

7.3.4.2.6 Vendor

Child Elements	None
Description	Name of the vendor with whom the financial transaction to acquire the product was conducted. See Figure 12 .
Requirement	Optional
Repeatability	0-1
Data Type	core-string
Attribute Used	N/A
Example(s)	EBSCO Information Services Swets Information Services
Usage Notes	This may or may not be the same as the publisher or the platform. See also Publisher (7.3.4.2.3) and Platform (7.3.4.2.5) to correctly distinguish the data that belongs in each of these three related elements.

7.3.4.2.7 PONumber

Child Elements	None
Description	Purchase order number(s) for the transaction. See Figure 12 .
Requirement	Mandatory
Repeatability	0-n
Data Type	core-string
Attribute Used	N/A
Example(s)	2008-0212-09647

Usage Notes	The information supplied should be the purchase order number(s) assigned to the transaction by the party that placed the original order.
--------------------	--

7.3.4.2.8 CustomerId

This element is identical to the CustomerId element as defined in Section [7.2.2.4](#).

7.3.4.2.9 ConsortiumId

Child Elements	None
Description	An identifier for the consortium related to the transaction.
Requirement	Optional
Repeatability	0-1
Data Type	core-string
Attribute Used	N/A
Example(s)	76-312 SDLIN Northeast Consortium of Libraries
Usage Notes	The ConsortiumId element can be used as needed by the transaction partners. For instance, it may signal that the CORE transaction was purchased under a consortial contract, or it may signal that a consortium actually made the purchase on behalf of the library. CORE does not describe or recommend how any transaction partners might decide on the use of this element.

7.3.4.2.10 PaymentDetailsRecord

Child Elements	OrderId (7.2.2.1) InvoiceNumber (7.3.4.2.11) InvoiceDate (7.3.4.2.13) LineItemNumber (7.3.4.2.14) Quantity (7.3.4.2.15) Period (7.2.2.5) LibraryFundCode (7.3.4.2.16) PaymentDate (7.3.4.2.18) PaymentAmount (7.3.4.2.15)
Description	Each PaymentDetailsRecord represents a single payment record. See Figure 14 .
Requirement	Mandatory
Repeatability	1-n
Data Type	Container
Attribute Used	payment-type
Example(s)	payment-type="onetime"

<p>Usage Notes</p>	<p>Multiple records are returned if there is more than one payment transaction in the specified Query Period.</p> <p>The payment-type attribute is restricted to three values: as described in Section 6.4, and illustrated in Figure 6.</p>
---------------------------	---

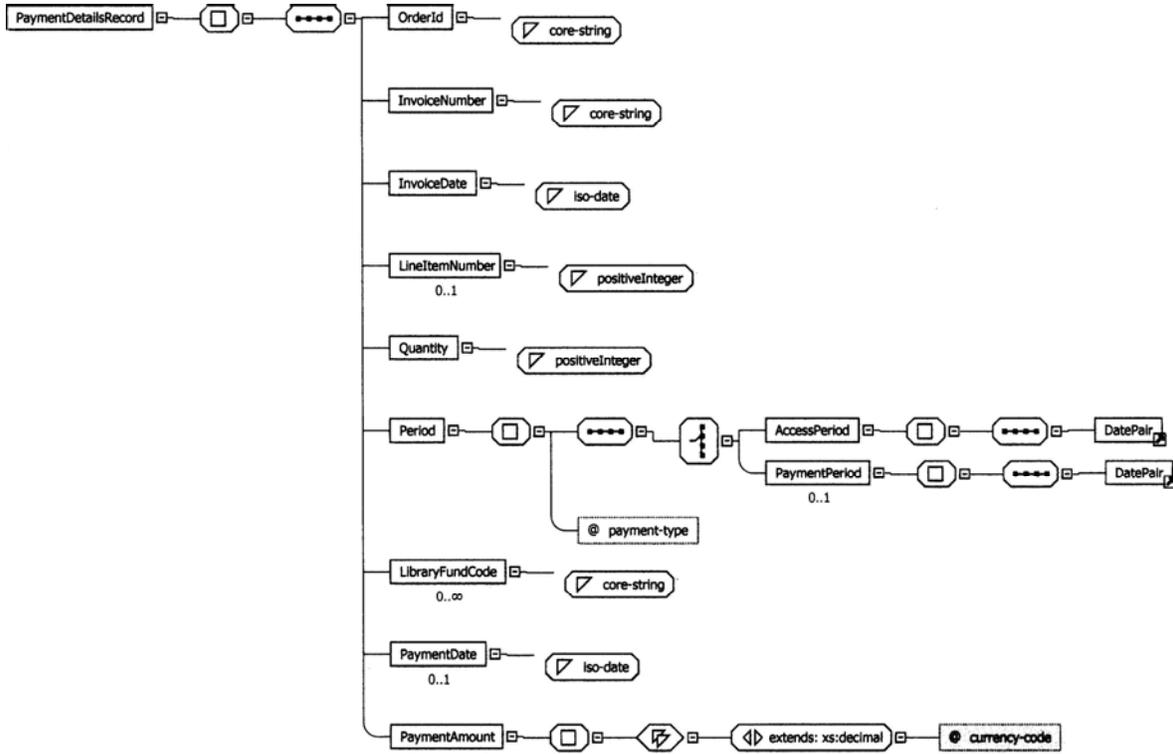


Figure 14: PaymentDetailsRecord element

7.3.4.2.11 OrderId

The OrderId in the Response should match the OrderId in the Query, if one was supplied. This element is mandatory in the Response even if it was not used in the Query. See Section 7.2.2.2 for the details on this element.

7.3.4.2.12 InvoiceNumber

<p>Child Elements</p>	<p>None</p>
<p>Description</p>	<p>Invoice number for this payment transaction. See Figure 14.</p>
<p>Requirement</p>	<p>Mandatory</p>
<p>Repeatability</p>	<p>1</p>
<p>Data Type</p>	<p>core-string</p>
<p>Attribute Used</p>	<p>N/A</p>
<p>Example(s)</p>	<p>Q-003</p>

Usage Notes	Exactly one InvoiceNumber is provided for each PaymentDetailsRecord.
--------------------	--

7.3.4.2.13 Invoice Date

Child Elements	None
Description	The date on the invoice. See Figure 14 .
Requirement	Mandatory
Repeatability	1
Data Type	iso-date
Attribute Used	N/A
Example(s)	20080215
Usage Notes	Exactly one InvoiceDate is provided for each PaymentDetailsRecord. Values are restricted to dates between January 1, 1900 and December 31, 2099 using the format CCYYMMDD.

7.3.4.2.14 LineItemNumber

Child Elements	None
Description	The number of the line item on the invoice that applies to this payment transaction. See Figure 14 .
Requirement	Optional
Repeatability	0-1
Data Type	positive integer
Attribute Used	N/A
Example(s)	5 45
Usage Notes	The line item is always used in conjunction with the InvoiceNumber (7.3.4.2.12) element. Example: Invoice M47344; Line Item: 45. On many invoices, each line item corresponds to a particular ProductId (7.3.4.2.1).

7.3.4.2.15 Quantity

Child Elements	None
Description	Indicates how many of the products are included in this transaction. See Figure 14 .
Requirement	Mandatory
Repeatability	1
Data Type	positive integer

Attribute Used	N/A
Example(s)	1 2
Usage Notes	For electronic resources, Quantity will usually have the value "1".

7.3.4.2.16 LibraryFundCode

Child Elements	None
Description	Library's internal fund code that pertains to this transaction. See Figure 14 .
Requirement	Optional
Repeatability	0-n
Data Type	core-string
Attribute Used	N/A
Example(s)	837-9878-001-B
Usage Notes	The response will indicate all the LibraryFundCodes associated with the total PaymentAmount for this record. The breakdown of amounts for each code is not provided.

7.3.4.2.17 Period

The AccessPeriod element is required in the Response while the PaymentPeriod is optional. Refer to section 7.2.2.5 for all other aspects of the Period element.

7.3.4.2.18 PaymentDate

Child Elements	None
Description	Date the payment transaction was paid. See Figure 14 .
Requirement	Optional
Repeatability	0-1
Data Type	iso-date
Attribute Used	N/A
Example(s)	20080524
Usage Notes	Values are restricted to dates between January 1, 1900 and December 31, 2099 using the format CCYYMMDD.

7.3.4.2.19 PaymentAmount

Child Elements	None
Description	The monetary amount paid for the specified InvoiceNumber and LineItemNumber. See Figure 14 . Also known as cost.
Requirement	Mandatory
Repeatability	1
Data Type	decimal
Attribute Used	currency-code
Example(s)	25.00 Example of full element with currency-code attribute: <PaymentAmount currency-code="USD">25.00</PaymentAmount>
Usage Notes	A currency-code attribute, as defined in Section 6.4, is required to be provided with the PaymentAmount, as illustrated in Figure 15 .

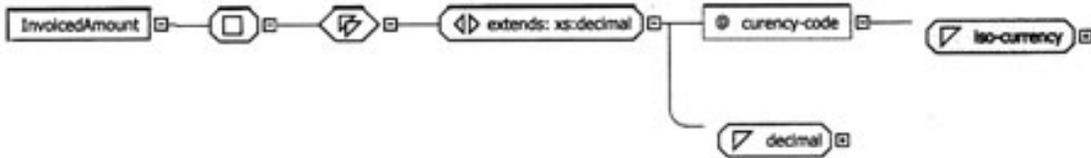


Figure 15: PaymentAmount element

8 Transaction Failure

Failed transactions are protocol errors that prevent replies to a Request. If there is no problem with the transaction as a whole, but the particular query request cannot be fulfilled, then the transaction does not fail. (However, in that instance an ErrorRecord [7.3.4.1] must be returned.) This section refers only to those instances where the transaction has failed.

8.1 Failures

A transaction will fail under any of the following conditions:

- A non-COREDocument is received.
- An invalid or not-well-formed COREDocument is received.
- An unknown value is received in the **version** attribute and the responding implementation cannot determine the **version** of a COREDocument or is unable to do a **version** resolution. (See Section 8.2 for more on handling version problems.)
- There are unknown values in the DocumentId (i.e., the Requester or the Responder).

The appropriate RequestProblem response (see Section [7.3.3](#)) must be returned when a transaction fails.

8.2 Unknown Version Problem

The following scenario proposes a solution to versioning problems that may be encountered during the exchange of COREDocuments.

If a Request within a COREDocument bearing an unknown version number is received, the responder constructs a Response COREDocument bearing the value of the constructor's own version number, adds a RequestProblem element containing the error unknown-version, and returns the constructed document to the submitter of the Request.

If a Response within a COREDocument bearing an unknown version number is received, and contains a RequestProblem element with the error value unknown-version, the requester is made aware of the corresponding implementation's version and may act accordingly (either resubmit a new Request element within a COREDocument bearing the appropriate version number or quit the exchange). In any other circumstance, the recipient of a Response element within a COREDocument bearing a version number different from the submitted COREDocument's Request must discard the Response; the transaction has failed.

This scenario is implementation specific. An incoming message validated against a different version of the schema may be rejected by an XML parser before it is seen by the implementation. In this case, implementations would be unable to conform to the above version resolution suggestion. The absence of a value for the **version** attribute in the schema is an attempt to circumvent this possibility.

8.3 Resubmitted Requests

When the RequestProblem element's **problem** attribute has the value system-problem, the requester may choose to submit a new request for the same query, but is not required to do so. A well-behaved requester implementation must provide a new **transaction-id** for the later request.

9 Profiles

Profiles are used to define data content and form, to resolve ambiguities the protocol cannot, and to narrow (if necessary) the broader behaviors implicit in the protocol to a particular community of implementers. Any developed profiles should indicate the namespace of the original CORE schema.

The types of issues that might be addressed in a profile include (but are not limited to):

- Using specific standards for identifiers, e.g., the ISIL (ISO 15511) or SAN (ANSI/NISO Z39.43) for Requester or Responder.
- Timeout limits
- Retry times

Appendix A (normative) CORE Schema

This appendix reproduces the XML schema (xsd file) defined by the CORE standard. The schema file can be found on the NISO website at www.niso.org/schemas/core. The version at the time of this Draft Standard for Trial Use is 0.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.niso.org/schemas/core-v0.1"
xmlns:core="http://www.niso.org/schemas/core-v0.1">

  <xs:element name="COREDocument">
    <xs:complexType>
      <xs:choice>
        <xs:element name="Request">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="core:DocumentId"/>
              <xs:element ref="core:Query"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Response">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="core:DocumentId"/>
              <xs:element ref="core:Query"/>
            <xs:choice>
              <xs:element name="RequestProblem">
                <xs:complexType>
                  <xs:attribute name="problem" use="required">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="unknown-version"/>

```

```

        <xs:enumeration value="service-refused" />
        <xs:enumeration value="unknown-requester" />
        <xs:enumeration value="unknown-responder" />
        <xs:enumeration value="transaction-id-problem" />
        <xs:enumeration value="system-problem" />
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="QueryReply">
    <xs:complexType>
        <xs:sequence>
            <xs:choice>
                <xs:element name="ErrorRecord">
                    <xs:complexType>
                        <xs:attribute name="error" use="required">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="invalid-period" />
                                    <xs:enumeration value="unknown-order-id" />
                                    <xs:enumeration value="unknown-product-id" />
                                    <xs:enumeration value="record-not-available" />
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                    </xs:complexType>
                </xs:element>
                <xs:element name="AcqRecord" minOccurs="1" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="ProductId" type="core:core-string" />
                            <xs:element name="ProductName" type="core:core-string" minOccurs="0"
maxOccurs="1" />
                            <xs:element name="Publisher" type="core:core-string" minOccurs="0"
maxOccurs="1" />
                            <xs:element name="Medium" minOccurs="0" maxOccurs="1">
                                <xs:complexType>
                                    <xs:attribute name="form" use="required">
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="physical" />
        <xs:enumeration value="online" />
        <xs:enumeration value="both" />
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
maxOccurs="unbounded" />

<xs:element name="Platform" type="core:core-string" minOccurs="0"

maxOccurs="unbounded" />
maxOccurs="1" />
maxOccurs="1" />

<xs:element name="Vendor" type="core:core-string" minOccurs="0" maxOccurs="1" />
<xs:element name="PONumber" type="core:core-string" minOccurs="1"

<xs:element name="CustomerId" type="core:core-string" minOccurs="0"

<xs:element name="Consortium" type="core:core-string" minOccurs="0"

<xs:element name="PaymentDetailsRecord" minOccurs="1" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="OrderId" type="core:core-string" />
      <xs:element name="InvoiceNumber" type="core:core-string" />
      <xs:element name="InvoiceDate" type="core:iso-date" />
      <xs:element name="LineNumber" minOccurs="0" maxOccurs="1"

      <xs:element name="Quantity" type="xs:positiveInteger"

    <xs:element name="Period">
      <xs:complexType>
        <xs:sequence>
          <xs:choice>
            <xs:element name="AccessPeriod">
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="core:DatePair" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:choice>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:element name="PaymentPeriod" minOccurs="0"

  <xs:complexType>
    <xs:sequence>
      <xs:element ref="core:DatePair" />
    </xs:sequence>
  </xs:complexType>

```

```

        </xs: sequence>
      </xs: compl exType>
    </xs: el ement>
  </xs: choi ce>
</xs: sequence>
<xs: attri bute name="payment-type" type="core: payment-type"
  </xs: compl exType>
</xs: el ement>
<xs: el ement name="Li braryFundCode" type="core: core-string"
<xs: el ement name="PaymentDate" type="core: i so-date" mi nOccurs="0"
<xs: el ement name="PaymentAmount">
  <xs: compl exType>
    <xs: si mpl eContent>
      <xs: extensi on base="xs: deci mal ">
        <xs: attri bute name="currency-code" type="core: i so-
          </xs: extensi on>
        </xs: si mpl eContent>
      </xs: compl exType>
    </xs: el ement>
  </xs: sequence>
</xs: compl exType>
</xs: el ement>
</xs: sequence>
</xs: compl exType>
</xs: el ement>
</xs: choi ce>
</xs: sequence>
</xs: compl exType>
</xs: el ement>
</xs: choi ce>
</xs: sequence>
</xs: compl exType>
</xs: el ement>
</xs: choi ce>
  <xs: attri bute name="versi on" type="xs: deci mal " defaul t="0.01"/>
</xs: compl exType>
</xs: el ement>

```

```

<xs:element name="DocumentId" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Requester" type="core:core-string"/>
      <xs:element name="Responder" type="core:core-string"/>
    </xs:sequence>
    <xs:attribute name="transaction-id" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="Query" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CustomerId" type="core:core-string" minOccurs="0" maxOccurs="1"/>
      <xs:choice>
        <xs:element name="OrderId" type="core:core-string"/>
        <xs:element name="ProductId" type="core:core-string"/>
        <xs:element name="RequestAll" type="xs:boolean" fixed="true"/>
      </xs:choice>
      <xs:element name="Period" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:choice>
              <xs:element name="AccessPeriod">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="core:DatePair"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="PaymentPeriod">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="core:DatePair"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:sequence>
          <xs:attribute name="payment-type" type="core:payment-type" default="unknown"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:simpleType name="core-string">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
    </xs:restriction>
</xs:simpleType>

<xs:element name="DatePair">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="BeginDate" type="core:iso-date"/>
            <xs:element name="EndDate" type="core:iso-date"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:simpleType name="iso-date">
    <xs:restriction base="xs:string">
        <!-- .....year.....feb.....30 day
months.....31 day months -->
        <xs:pattern value="(19|20)\d\d(((02((0[1-9])|(1|2)[0-9]))|(0[(4|6|9)])|11)((0[1-9])|(1|2)[0-9])|30))|(((1[(0|2)])|0[(1|3|5|7|8]))((0[1-9])|(1|2)[0-9])|3[0-1]))"/>
    </xs:restriction>
</xs:simpleType>

<!-- 02((0[1-9])|(1|2)[0-9])                                feb -->
<!-- ((0[(4|6|9)])|11)((0[1-9])|(1|2)[0-9])|30                30 day months -->
<!-- (((1[(0|2)])|0[(1|3|5|7|8]))((0[1-9])|(1|2)[0-9])|3[0-1])) 31 day months -->

<xs:simpleType name="iso-currency">
    <xs:restriction base="xs:string">
        <xs:pattern value="[A-Z]{3}"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="payment-type">
    <xs:restriction base="xs:string">

```

```
<xs:enumeration value="unknown"/>  
<xs:enumeration value="continuous"/>  
<xs:enumeration value="onetime"/>  
</xs:restriction>  
</xs:simpleType>  
  
</xs:schema>
```

Appendix B: (informative) CORE Use Cases and Scenarios

(This appendix is not part of the *Cost of Resource Exchange (CORE) Protocol*, NISO Z39.95-200x.
It is included for information only.)

This section describes and illustrates the three CORE use cases and scenarios—the examples which illustrate the use cases. These examples are illustrative and are in no way designed to limit CORE’s use to any given set of partners or data exchange environments.

B.1 Use Cases

The following use cases are offered as a means of describing basic information exchange. In any and all of these, the information requested may be qualified by customer ID and by the date range of the payment period or by the access (subscription) period. The date range terms “access period” and “subscription period” are used interchangeably.

- 1) Exchange of information for a single order – The request includes one order identifier and may also include a customer ID and/or a date range. The response includes details of all payments for the specified order, limited to the specified customer and date range (if any).
- 2) Exchange of information for a single product – The request includes one product identifier and may also include a customer ID and/or a date range. The response includes details of all payments for all orders for the specified product, limited to the specified customer and date range (if any).
- 3) Exchange of information for all orders for all products – The request contains no order or product identifier, but may include a customer ID and/or a date range. The response contains details of all payments for all orders, limited to the specified customer and date range (if any).

B.2 Data Exchange Scenarios

The following scenarios are offered to 1) show the steps of the CORE data transfer, 2) establish requirements for the data transfer mechanism, and 3) delineate expectations of the various systems, specifying who is responsible for what.

For brevity’s sake, the following terms or abbreviations shall be used in the Scenarios:

- Acquisitions System – Library System Acquisitions module
- ERMS – Electronic Resource Management System
- SAS – Subscription Agent System

Unless otherwise indicated, “library” refers to the party running the ERMS and the acquisitions system. “Customer” refers to a party that purchases products, both print and electronic. “Agent” refers to an intermediary that supplies the customer with its products, whether in print or electronic format.

B.2.1 Scenarios for Use Case 1

B.2.1.1 ERMS requests payment information for a particular order from the local acquisitions system, for the entire history of the order.

Assumptions: The ERMS stores or has access to a unique order identifier—such as a PO number, PO line number, or agent’s subscription number—that is also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing one order identifier.
- 2) Acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for the specified order.

B.2.1.2 ERMS requests payment information for a specific access (subscription) period for a particular order from the local acquisitions system.

Assumptions: The ERMS stores or has access to a unique order identifier—such as a PO number, PO line number, or subscription number—that is also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing one order identifier and access period date range.
- 2) Acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for the specified order for the specified access period.

B.2.1.3 ERMS requests payment information for a specific payment period for a particular order from the local acquisitions system.

Assumptions: The ERMS stores or has access to a unique order identifier—such as a PO number, PO line number, or agent’s subscription number—that is also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing one order identifier and payment period date range.
- 2) Acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for the specified order during the specified payment period.

B.2.1.4 ERMS requests payment information for a specific access (subscription) period for multiple subscriptions from a subscription agent.

Assumptions: The ERMS stores or has access to a unique order identifier—such as a PO number, PO line number, or agent’s subscription number—that is also known to the SAS. The ERMS also stores or has access to the customer ID that the agent system associates with all the library’s subscriptions.

Sequence of events:

- 1) ERMS sends a separate CORE request to the SAS containing an order identifier and access period date range for each order or subscription, and the customer ID.
- 2) The SAS returns a CORE response message containing product, order, and payment details for all payments made for each of the specified orders for the specified access period.

B.2.2 Scenarios for Use Case 2

B.2.2.1 ERMS requests payment information for a particular product from the local acquisitions system, for the entire subscription history of the product.

Assumptions: The ERMS stores or has access to a unique product identifier, such as ERMS e-resource number, that is also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing one product identifier.
- 2) Acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for the specified product.

B.2.2.2 ERMS requests payment information for a specific access (subscription) period for a particular product from the local acquisitions system.

Assumptions: The ERMS stores or has access to a unique product identifier, such as ERMS e-resource number, that is also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing one product identifier and access period date range.
- 2) Acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for the specified product for the specified access period.

B.2.2.3 ERMS requests payment information for multiple payment periods for multiple products from the local acquisitions system.

Assumptions: The ERMS stores or has access to the unique Product Identifier, such as ERMS e-resource number, that is also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing one product identifier and payment period date range for each product for which information is desired.
- 2) Acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for each specified product for the specified payment period.

B.2.2.4 ERMS asks the local acquisitions system how much money was spent for all E-Resource products provided by a specific vendor during a single payment period.

Assumptions: The ERMS is able to identify all the products on order from a particular subscription agent and can send product IDs for these that are also known to the acquisitions system.

Sequence of events:

- 1) ERMS sends a CORE request to the acquisitions system containing a product identifier and payment period date range for each agent's product.
- 2) The acquisitions system returns a CORE response message containing product, order, and payment details for all payments made for each of the specified products for the specified payment period.

B.2.2.5 ERMS requests payment information for a specific access (subscription) period for multiple products from a subscription agent.

Assumptions: The ERMS stores or has access to a unique product identifier, such as agent title number, that is also known to the SAS. The ERMS also stores or has access to the customer ID which the Agent system associates with all the library's subscriptions.

Sequence of events:

- 1) ERMS sends a CORE request to the SAS containing a product identifier for each product, access period date range, and customer identifier.
- 2) The SAS returns a CORE response message containing product, order, and payment details for all payments made for each of the specified products for the specified access period.

B.2.2.6 ERMS requests payment information from a subscription agent in behalf of its consortial members for a specific payment period for a specific product.

Assumptions: The ERMS stores or has access to a unique product identifier, such as agent title number, that is also known to the SAS. The ERMS also stores or access to the customer IDs which the agent system associates with each consortium member.

Sequence of events:

- 1) ERMS sends a different CORE request to the SAS for each consortium member, containing the product identifier, the payment period date range, and the consortium member's customer ID.
- 2) The SAS returns a CORE response messages (one for each consortium member) containing product, order, and payment details issued for the specified product for the specified payment period.

B.2.3 Scenarios for Use Case 3

B.2.3.1 ERMS requests payment information for a specific access (subscription) period for all products the library subscribes to from a subscription agent.

Assumptions: The ERMS stores or access to the customer ID that the agent system associates with all the library's subscriptions.

Sequence of events:

- 1) ERMS sends a CORE request to the SAS containing a request for "all" cost information with the access period date range, and the customer ID.
- 2) The SAS returns a CORE response message containing product, order, and payment details for all invoices issued for all customer-subscribed products for the specified access period.

Appendix C (informative) Security Considerations

(This appendix is not part of the *Cost of Resource Exchange (CORE) Protocol*, NISO Z39.95-200x.
It is included for information only.)

The CORE standard does not include an integrated security mechanism, however, security was a consideration in developing this standard. The Working Group recognizes that, for many institutions, financial data about their acquisitions are considered confidential. Where a CORE transfer takes place between an ILS and ERMS within the same organization, security of the protocol transmission may not be an issue.

However, for those instances when the CORE transaction occurs between two different parties and security is a concern, this appendix discusses three levels of security that can be implemented without building a security mechanism into the standard protocol. These are:

- 1) Securing the communications channel
- 2) Data integrity
- 3) Authenticating the requesting organization
- 4) Validating the rights of a requesting organization to access specific customer data

C.1 Securing the Communications Channel

The recommended transfer mechanism for CORE is SOAP (Simple Object Access Protocol), which in turn uses either HTTP or HTTPS for transmission between client and server applications. By using HTTPS, the communication between client and server is encrypted using SSL (Secure Sockets Layer), thereby preventing any third party from intercepting the transmission and discovering its content.

C.2 Data Integrity

Although the CORE protocol does not provide requirements for implementing protection from data tampering or spoofing, it is possible to use standard techniques in securing the data through use of certificates, data encryption, and information verification methods.

C.3 Authenticating the Requesting Organization

Using HTTPS, the communication channel is secure, however, there is still a possibility that an unauthorized software application could access a CORE-compliant server and request data. To prevent this, the data provider can implement a security layer within their CORE server. Following are three options:

- Requester validation
- IP authentication
- Username/password

The simplest form of authentication would be to validate the Requester. Unless the Requester is known to the server, the request would not be processed. This, however, is not very secure if the Requestor identifier is easy to guess—such as the name of the organization.

Using IP authentication adds a much stronger level of security. The requesting organizations would need to register the IP address of the computer running their CORE software with the data provider. The data provider would only process requests for recognized IP addresses.

A third and even more secure method of authentication would be to use WS-Security extensions to introduce username/password authentication. The data provider would issue unique usernames and passwords to each requesting organization. Without a valid username/password, the connection would be rejected at the SOAP level and processing of the request would not take place.

C.4 Validating Rights of a Requesting Organization to Access Specific Customer Data

Some data providers may be supplying CORE data for a number of institutions. The provider in such instances can introduce another security layer to restrict authorized requesting organizations to certain customer data. This method is especially useful if the requesting organization is not the same as the customer whose data is being requested.

Data providers can fairly easily set up a system that requires their customers to “authorize” the requesting organizations who can harvest their data. If the service provider registers the requesting organizations, then it can present their customers with a simple user interface that gives them the option to “activate” CORE retrieval, then identify the requesting organization(s) allowed to do the retrieval. The result is a mapping between CustomerId and Requestor, allowing the CORE server to verify that the data retrieval is permitted before processing is continued.

For service providers who are using IP authentication for the requesting organization, a simpler model could be implemented when the requestor and the customer are the same. The CORE server could verify that the IP address of the requestor is included in the IP range registered for the customer and, if so, processing of the request would continue.

C.5 Summary of Security Considerations

Even though security is not part of this standard, CORE can be a very secure protocol. Using HTTPS, the communication channel is secure. Data providers can authenticate the requesting organizations using IP addresses or username/password authentication. Data providers can allow customers to have their data further protected by first requiring them to opt in to CORE data retrieval, and then by identifying which requesting organizations are allowed to retrieve their data.

Appendix D: (informative) Summary of CORE Elements

(This appendix is not part of the *Cost of Resource Exchange (CORE) Protocol*, NISO Z39.95-200x. It is included for information only.)

This table provides a summary of the elements defined in this standard.

Note1: For the Requirement (Req) column, the following abbreviations are used:

M = mandatory (Note: Elements are only mandatory within their parent, i.e., they are only mandatory if the parent element is used.)

MA = mandatory if applicable

O = optional

Note 2: For the Repeatability (Repeat) column, the numbers indicate how many times a field may be repeated. "0" indicates the element is optional; a "1" as the only or beginning number indicates the element is mandatory.

D.1 Request Elements

Element	Req	Repeat	Type	Attributes
COREDocument	M	1	decimal	version
Request	MA	1	Container	N/A
DocumentId	M	1	Container	N/A
Requester	M	1	core-string	N/A
Responder	M	1	core-string	N/A
Query	M	1	Container	N/A
OrderId	MA	0-1	core-string	N/A
ProductId	MA	0-1	core-string	N/A
RequestAll	MA	0-1	Boolean	N/A
CustomerId	O	0-1	core-string	N/A
Period	O	0-1	Container	payment-type
AccessPeriod	MA	0-1	Container	N/A
DatePair	M	1	Container	N/A
BeginDate	M	1	iso-date	N/A
EndDate	M	1	iso-date	N/A
PaymentPeriod	MA	0-1	Container	N/A
DatePair	M	1	Container	N/A
BeginDate	M	1	iso-date	N/A
EndDate	M	1	iso-date	N/A

D.2 Response Elements

Element					Req	Repeat	Type	Attributes
COREDocument					M	1	decimal	version
	Response				MA	1	Container	N/A
		DocumentId			M	1	Container	N/A
			Requester		M	1	core-string	N/A
			Responder		M	1	core-string	N/A
		Query			M	1	Container	N/A
		RequestProblem			MA	1	contains enumerated attribute	problem
		QueryReply			MA	1	Container	N/A
			ErrorRecord		MA	1	contains enumerated attribute	error
			AcqRecord		MA	0-n	Container	N/A
				ProductId	M	1	core-string	N/A
				ProductName	O	0-1	core-string	N/A
				Publisher	O	0-1	core-string	N/A
				Medium	O	0-1	contains enumerated attribute	form
				Platform	O	0-n	core-string	N/A
				Vendor	O	0-1	core-string	N/A
				PONumber	O	0-n	core-string	N/A
				CustomerId	O	0-1	core-string	N/A
				ConsortiumId	O	0-1	core-string	N/A
				PaymentDetailsRecord	M	1-n	Container	N/A
				OrderId	M	1	core-string	N/A
				InvoiceNumber	M	1	core-string	N/A
				InvoiceDate	M	1	iso-date	N/A
				LineItemNumber	O	0-1	positive integer	N/A
				Quantity	M	1	positive integer	N/A
				Period	M	1	Container	payment-type
				AccessPeriod	M	1	Container	N/A
				DatePair	M	1	Container	N/A

Element								Req	Repeat	Type	Attributes
							BeginDate	M	1	iso-date	N/A
							EndDate	M	1	iso-date	N/A
							PaymentPeriod	O	0-1	Container	N/A
							DatePair	M	0-1	Container	N/A
							BeginDate	M	0-1	iso-date	N/A
							EndDate	M	0-1	iso-date	N/A
							LibraryFundCode	O	0-n	core-string	N/A
							PaymentDate	O	0-1	iso-date	N/A
							PaymentAmount	M	1	decimal	currency-code

**Appendix E:
(informative)
CORE Document Examples**

(This appendix is not part of the *Cost of Resource Exchange (CORE) Protocol*, NISO Z39.95-200x. It is included for information only.)

[Examples will be added after the trial when some actual use data is available.]