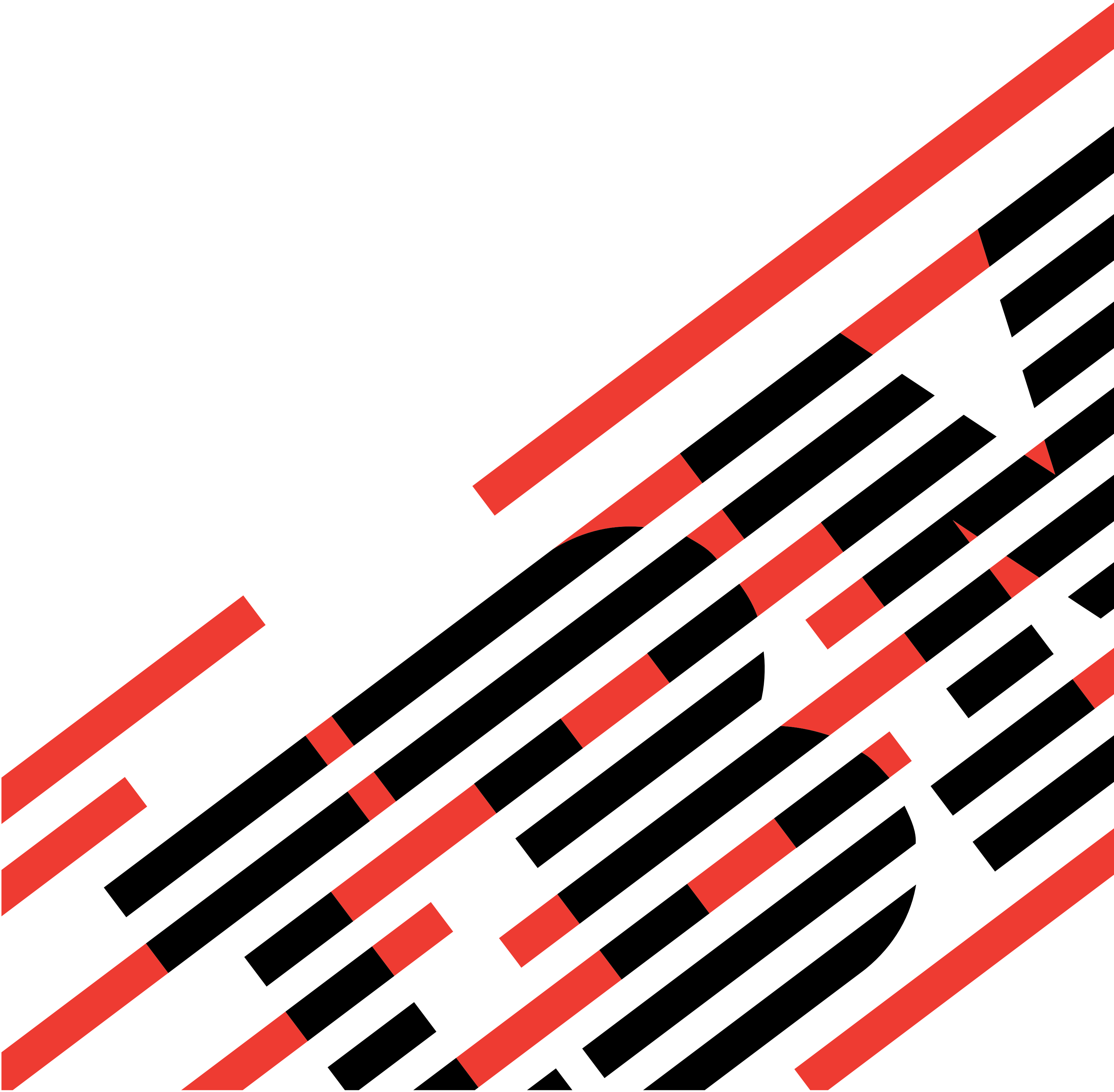
iSeries

Extensible Markup Language: XML

Version 5 Release 3



iSeries

Extensible Markup Language: XML

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 15.

Fourth Edition (August 2005)

This edition applies to version 5, release 3, modification 0 of Extensible Markup Language: XML (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

This edition replaces rzak-l000-01.

Contents

Extensible Markup Language (XML)

This information will help you use XML:

- “Using XML” on page 2

  Find out what XML is, what extensions and companion standards it uses, and what it can do for you.
- “XML tools for OS/400” on page 6

  Find out about the XML tools that are integrated in OS/400. This information also discusses other useful XML tools.
- “Samples” on page 12

  Working samples illustrate how XML can help you share information.
- “XML related information” on page 12

  Use these APIs to help you integrate XML into your Java$^{(TM)}$, C++, and procedural language (ILE C, RPG, and COBOL) programs. This page includes links to more information about the evolving XML standards.

> **Note:** The XML parsers for ILE C, C++, COBOL, and RPG that were introduced and integrated into the operating system in V5R1 have been replaced by the licensed program offering (LPO) XML Toolkit for iSeries$^{(TM)}$ (5733-XT1). Although service programs QXML4C310 and QXML4PR310 still exist in library QSYS, you are encouraged to use the newer parsers provided by the XML Toolkit for iSeries LPO.
>
> The XML and XSL services for Java also remain in the operating system. See the XML for iSeries Web site for further information about these services.

Note: Read the “Code disclaimer information” on page 18 for important legal information.

Print this topic

To view or download the PDF version, see rzakl.pdf (about 200 KB). Note that the PDF version of the XML information does not contain the API documentation.

To save a PDF on your workstation for viewing or printing:

1. Open the PDF in your browser (click the link).
2. In the menu of your browser, click **File**.
3. Click **Save Target As...** if you are using Internet Explorer.
   Click **Save Link As...** if you are using Netscape Communicator.
4. Navigate to the directory in which you would like to save the PDF.
5. Click **Save**.

If you need Adobe Acrobat Reader to view or print these PDFs, you can download a copy from the Adobe Web site.

Other information

You can also view or print any of these Redbook PDFs:

- The XML Files: Using XML for Business-to-Business and Business-to-Consumer Applications (about 3 MB)

- AS/400$^{(R)}$ XML in Action: PDML and PCML (about 3.5 MB)
- Integrating XML with DB2$^{(R)}$ XML Extender and DB2 Text Extender (about 4.5 MB)

Using XML

Extensible Markup Language (XML) allows you to describe and organize information in ways that are easily understandable by both humans and computers. You can then share that information and its description with others over the Internet, an extranet, network, or in other ways.

XML, like Standard Generalized Markup Language (SGML), is a metalanguage. A metalanguage allows you to define a document markup language and its structure. For example, both XML and Hypertext Markup Language (HTML) are derived from SGML.

You can use XML to create your own markup language that includes a set of rules and tags that describe information suited to your needs, for example, name, title, address, and zip code. You define this markup language in a document type definition (DTD) that functions as the standard way to describe your information. Using XML to share standardized information you are no longer required to write programs to focus on proprietary software or convert and translate different data formats.

You and others can use the DTD to tag information that you can then use in a variety of ways: printed on an address label, business card, or stationary; displayed in a Web page; or sorted in a list of data with similar attributes.

For example, you might want to create an efficient way to share information (say, purchase orders, shipping acknowledgments, order status, and stock status) with your partners and suppliers. You can use XML to share that information by creating and using XML documents that conform to your DTD, in which you specify the standard for the electronic exchange of information.

Although both XML and HTML use tags to describe content, they are also very different:

- HTML describes how to format information for display and is meant for computer-to-human interaction.
- XML describes what the information is and is meant for computer-to-computer interaction.

"Advantages of XML"
Take a quick look at how using XML offers advantages over using HTML to exchange information.

"XML standards and extensions" on page 4
Other standards and extensions to XML work together to make your information more portable and useful. You need to know about these standards and extensions in order to:

- Use XML with your Java, C++, RPG, and COBOL programs
- Perform complex data searches in XML documents
- Display XML data on different types of devices
- Provide your XML documents with orderly linking capabilities
- Produce standard structures for related DTDs

Advantages of XML

Using XML to exchange information offers many benefits, including the following:

- Uses human, not computer, language. XML is readable (and understandable, even by novices) and no more difficult to code than HTML.
- Completely compatible with Java and 100% portable. Any application that can process XML (on any operating system) can use your information.

- Extendable. Create your own tags (or use tags created by others) that use the basic language of your domain, have the attributes you need, and make sense to you and your users.

The following example illustrates, in a simplified way, the readability and extensibility of XML:

HTML example	XML example

`<HTML>`
`<H1 ID="MN">State</H1>`
`<H2 ID="12">City</H2>`
`<DL>`
`  <DT>Name</DT>`
`    <DD>Johnson</DD>`
`  <DT>Population</DT>`
`    <DD>5000</DD>`
`</DL>`
`<H2 ID="15">City</H2>`
`<DL>`
`  <DT>Name</DT>`
`    <DD>Pineville</DD>`
`  <DT>Population</DT>`
`    <DD>60000</DD>`
`</DL>`
`<H2 ID="20">City</H2>`
`<DL>`
`  <DT>Name</DT>`
`    <DD>Lake Bell</DD>`
`  <DT>Population</DT>`
`    <DD>20</DD>`
`</DL>`
`</HTML>`
	`<?XML VERSION="1.0" STANDALONE="yes" ?>`
	`<STATE STATEID="MN">`
	`  <CITY CITYID="12">`
	`    <NAME>Johnson</name>`
	`    <POPULATION>5000</POPULATION>`
	`  </CITY>`
	`  <CITY CITYID="15">`
	`    <NAME>Pineville</NAME>`
	`    <POPULATION>60000</POPULATION>`
	`  </CITY>`
	`  <CITY CITYID="20">`
	`    <NAME>Lake Bell</NAME>`
	`    <POPULATION>20</POPULATION>`
	`  </CITY>`
	`</STATE>`

HTML tag names reveal nothing about the meaning of their content. The previous example uses an HTML definition list, but the problems inherent in using HTML occur if the data were contained in a table or some other kind of HTML tags: For example:

- Many the HTML tags are acronyms, therefore they are not as readable as common language.
- HTML tags represent data (in this example, city names and populations) as items to display, for example, as definitions in a list or cells in a table. This makes it difficult to manipulate the data or or exchange it between applications.

The XML tag names are readable and convey the meaning of the data. Each XML tag immediately precedes the associated data, helping to make the information structure easily discerned by both humans and computers. The data structure follows a noticeable and useful pattern, making it easy to manipulate and exchange the data.

Uses of XML

XML has a variety of uses, including:

- **Web publishing**: XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications more intuitive. With XML, you store the data once and then render that content for different viewers or devices based on style sheet processing using an XSL/XSLT processor.
- **Web searching and automating Web tasks**: XML defines the type of information contained in a document, making it easier to return useful results when searching the Web:
  - For example, using HTML to search for books authored by Tom Wolf is likely to return instances of the term ’wolf’ outside of the context of author. Using XML restricts the search to the correct context (say, the information contained in the <author> tag) and returns only the required type of information. Using XML, Web agents and robots (programs that automate Web searches or other tasks) will be more efficient and produce more useful results.

- **General applications**: XML provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.
- **e-business applications**: XML implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions, and business-to-consumer transactions.
- **Metadata applications**: XML makes is easier to express metadata (Unified Modeling Language design models or user interface properties, for example) in a portable, reusable format.
- **Pervasive computing**: XML provides portable and structured information types for display on pervasive (wireless) computing devices such as PDAs, cellular phones, and others.
  - For example, WML (Wireless Markup Language) and VoiceXML are currently evolving standards for describing visual and speech-driven wireless device interfaces.

XML standards and extensions

XML is very good for describing information, but it cannot do everything. For example, XML documents do not contain the kind of information that current browsers and many other devices require to display it in a useful way. The same is true for linking to other information, transporting XML data so that it can be used in a meaningful way by the receiving application, and so on.

The XML community has and continues to develop standards and extensions to expand the capabilities of XML:
- "APIs"
- "XSL and XSLT" on page 5
- "XLink" on page 5
- "XPath and XPointer" on page 5
- "Namespaces and XML Schema" on page 5

APIs

Application programming interfaces (APIs) allow applications to work with XML information using a standard set of portable interfaces. For more information, see "XML tools integrated in OS/400" on page 6 for links to API documentation for a supported parser.

DOM and DOM Level 2

The Document Object Model (DOM) API enables you to build XML documents as well as parse them. With these interfaces you can access, manipulate, and create XML documents (and the data within) as programming objects that have methods and events. Your programs can construct or change a DOM tree in memory and then persist that DOM tree to a file or stream. DOM is best suited for instances where you will parse few XML documents but require extensive control over the contents.

SAX

The Simple API for XML (SAX) is a read-only, single-pass interface best suited for processing many documents or very large documents. You can use this API to extract information from the XML documents, but you cannot use it to add data to or change the content of the XML documents. The SAX API is event-driven, notifying your application when certain events happen as it parses your document. For example, your application might need to know when the parser encounters the start or end of an element node. Note that it is your application that must keep the necessary state information to determine the content and context of these XML events.

For links to more information about the DOM and SAX APIs, see the "XML related information" on page 12.

For the most current versions of these APIs, see the IBM alphaWorks Web site .

XSL and XSLT

Extensible Stylesheet Language (XSL) and Extensible Stylesheet Language Transformations (XSLT) work in combination for you to display XML data in a variety of ways, for example, in a browser or on a PDA, or printed in a brochure. With XSL and XSLT processing you can transform an XML message or document from one XML markup language to another, which has key applications in e-business.

A detailed explanation of the mechanics of this process is beyond this article. Briefly, however, the process has two basic components:

- Use XSL stylesheets to define a set of patterns and templates you want to use to replace XML elements. A pattern identifies the XML element, and the corresponding template is used by an XSL processor, like Xalan (which is included in OS/400), to actually replace the XML element. For example, you can transform data elements in an XML document to display appropriately, say, in a browser or mailing label.
- Use XSLT documents to transform the hierarchical tree of XML data into a different kind of tree, reordering elements as required. For example, you can add a table of contents or an index to a set of data that does not have one. You can also use XSLT to transform the grammar of XML documents. For example, you can transform the grammar for a set of incoming XML request documents to a different XML grammar required by the receiving application.

These technologies do more than format the display of an XML document, they change it so that it becomes a different kind of document. In conjunction with other XML tools and extensions, such as parsers and XLink, you can produce documents formats such as specific word processing formats, PDF, HTML, and more.

For links to more information about XSL and XSLT, see the “XML related information” on page 12.

XLink

XML Linking Language (XLink) enables you to link your XML document to other resources on the Web, including files of just about any format, database searches, and so on. Moreover, you can link to the structure of the resource, not a predetermined place holder, like an HTML <A NAME> anchor tag. Multiple links allow users to traverse the linked information in any order based on restrictions that you specify.

For links to more information about XLink, see the “XML related information” on page 12.

XPath and XPointer

XML Path Language (XPath) and XML Pointer Language (XPointer) enable you to search for and identify data in the hierarchical XML document structure.

XPath defines a syntax for locating data in an XML document. (Both XSLT and XPointer use XPath.) XPath defines an XML document as a hierarchy of nodes, with the top node being the root. Just like using a regular expression finds one or more patterns in text, using XPath finds patterns in data within the nodes of one or more XML documents.

XML Pointer Language (XPointer) extends XPath to enable locating specific portions of data (called fragments) based on XML attribute values, types, content, or relative position. These fragments can be seperate pieces of data, a range of information between two points, or a continuous series of elements.

For links to more information about XPath and XPointer, see the “XML related information” on page 12.

Namespaces and XML Schema

Namespaces are pointers (URIs) that enable you to differentiate between duplicate XML elements or attribute names, a situation that can occur when using XSLT stylesheets or more than a single DTD. For example, the <code> element from one DTD might mean something very different from a <code> element in another DTD. To avoid name collisions and ambiguity, giving each URI a unique local name makes it simple to distinguish between the different namespaces.

XML Schema Language defines the logical structure of an XML document, much like a DTD.

The significant difference between DTDs and XML Schemas are that schemas:

- Are written as XML markup language itself, making them extensible, unlike DTDs
- Focus on the problem of cardinality, enabling the enumeration of minimum and maximum allowed elements
- Allow constraints on values
- Allow additional data types and definitions of datatypes that can be inherited

All of these enhancements giving you more control over the allowable content of the XML document/message.

For example, you can add a different type of element to an existing schema as long as your addition does not break the original schema. Schemas also have many more available datatypes than do DTDs, making importing and exporting data somewhat easier.

At this time, the XML parsers included in OS/400 do not include support for XML Schema Language. For links to more information about Namespaces and XML Schema Language, see the "XML related information" on page 12.

XML tools for OS/400

OS/400 integrates three different XML parsers and an XSL processor for Java right into the operating system. Additionally, you can use a wide variety of "Additional XML tools and programs" on page 10 to help you use XML.

XML tools integrated in OS/400

Three XML parsers and an XSL processor are included with OS/400. This variety of XML support makes it easier for your application to use XML, especially when the application is written in different languages.

XML parsers

A parser is a tool for parsing, generating, manipulating, and validating XML documents. Each XML parser supports basic functions of one or more programming languages, so you can more easily program applications that use the data contained in the XML documents. Parsers also support one or more "APIs" on page 4, in whole or in part.

OS/400 includes three different parsers:

- "XML Parser for Java" on page 7
- "XML for C++ Parser" on page 7
- "XML Interface for RPG and Procedural Languages" on page 8

XSL Processor for Java

OS/400 includes the "XSL Processor for Java" on page 9, which provides a device for using XSL and XSLT to format and transform XML, either at the browser or on the server.

Additional XML tools

Some of the "Additional XML tools and programs" on page 10 that are available for free but not integrated into OS/400 include:

- "XML Enabler" on page 10 (for the "XSL Processor for Java" on page 9)
- "TaskGuide Viewer" on page 10

- “Xeena” on page 10

XML licensed programs

Some licensed programs that offer support for XML on iSeries servers are:

- “IBM Toolbox for Java (ReportWriter classes)” on page 11
- “DB2 XML Extender” on page 11
- “IBM WebSphere Host Publisher” on page 11

XML Parser for Java

The XML Parser for Java allows you to parse, create, manipulate, and validate XML documents. The XML Parser for Java, Version 3.0.1 is contained in a Java archive (JAR) file in the following directory on your iSeries Model 400 server:

```
/QIBM/proddata/OS400/xml/lib/xerces103.jar
```

As implied by the name of the JAR file, the XML Parser for Java Version 3.0.1 is equivalent to the Apache Xerces Java Parser Version 1.0.3.

Before you can use the XML Parser for Java in your programs, you need to add xerces103.jar to your CLASSPATH environment variable.

For detailed information about the XML Parser for Java, see the API documentation, which includes working samples.

Note: The XML Parser for Java is essentially a snapshot of XML W3C parser technology. As newer versions of this technology become available, you can find them on the IBM alphaWorks Web site or the Apache Web site .

XML for C++ Parser

Note: Starting in V5R3M0, the XML parser QSYS/QXML4C310 will no longer exist. Instead, use the XML Toolkit for iSeries (5733XT1) to provide the latest XML parsers.

The XML for C++ Parser allows you to parse, create, manipulate, and validate XML documents. The XML for C++ Parser, Version 3.1.0, is the following service program:

```
QSYS/QXML4C310
```

The compile header files (with an extension of .hpp) are provided under the following directory structure and use symbolic links to members (with shorter names) in QSYSINC/XML310. Note that the 310, 3_1_0, in the naming denotes the version of the XML parser. Because newer/future versions might result in another version of header files, these are uniquely identified for version 3.1.0.

Use the Create C++ Module (CRTCPPMOD) CL command for the ILE C++ compiler to compile your XML application source code. In order to compile with the XML for C++ header files, be sure to include the following directory by using the Include Directory (INCDIR) keyword:

```
/QIBM/include/xml3_1_0/xml/
```

The QXML4C310 service program exports both the “APIs” on page 4.

After compiling your C++ application code, send one of the following commands to create your application program or service program and bind it to the XML for C++ Parser:

```
CRTPGM PGM(<your_library>/<your_program>) MODULE(<your_library>/</your_modules>)
   BNDSRVPGM(QSYS/QXML4C310)
```

or

```
CRTSRVPGM SRVPGM(<your_library>/<your_program>) MODULE(<your_library>/</your_modules>)
   BNDSRVPGM(QSYS/QXML4C310)
```

For detailed information about the XML for C++ Parser, see the API documentation, which includes working samples.

> **Note:** The XML for C++ Parser is essentially a snapshot of XML W3C parser technology. As newer versions of this technology become available, you can find them on the IBM alphaWorks[(R)] Web site or the Apache Web site. Be sure to use the C++ header files that correspond to the correct version of the parser that you are using.

XML Interface for RPG and Procedural Languages

> » **Note:** Starting in V5R3M0, the XML parser QSYS/QXML4PR310 will no longer exist. Instead, use the XML Toolkit for iSeries (5733XT1) to provide the latest XML parsers. «

The XML Interface for RPG and Procedural Languages, Version 3.1.0 (XML4PR), is a wrapper interface to XML for C++ Parser (XML4C). This XML interface allows ILE C, RPG, and COBOL programs on the iSeries to parse XML documents in a procedural language rather than in an object-oriented language.

The XML Interface for RPG and Procedural Languages is the QXML4PR310 service program in library QSYS. This service program exports both the "APIs" on page 4.

Header files

A header file is your interface to the XML parser.

ILE C

For ILE C, the header file is library/file/member:

> QSYSINC/H/QXML4PR310

RPG

For RPG, the header file is library/file/member:

> QSYSINC/QRPGLESRC/QXML4PR310

COBOL

For COBOL, see the API specification in library/file/member:

> QSYSINC/QCBLLESRC/QXML4PR310

The connection file to facilitate your COBOL application code is library/file/member:

> QSYSINC/QCBLLESRC/QXML4PRLNK

> **Note:** Either qualify the header in your code or make sure that QSYSINC is on your library list.

Binding your program to the parser

Use the CL command for the compiler appropriate to the language of your application to compile your code:

- For the C ILE compiler, use the CL command Create C Module (CRTCMOD).
- For the RPG ILE compiler, use the CL command Create RPG Module (CRTRPGMOD).
- For the COBOL ILE compiler, use the CL command Create COBOL Module (CRTCBLMOD.

After compiling your application code, create your application program or service program and bind it to the XML Parser for ILE C, RPG, and COBOL.

Use one of the following commands, specifying the parser service program on the bind:

```
CRTPGM PGM(<your_library>/<your_program>) MODULE(<your_library>/<your_modules>)
  BNDSRVPGM(QSYS/QXML4PR310)
```

or

```
CRTSRVPGM SRVPGM(<your_library>/<your_service_program>) MODULE(<your_library>/<your_modules>)
  BNDSRVPGM(QSYS/QXML4PR310)
```

Notes:

- When using COBOL and the SAX APIs, pay particular attention to the Activation Group (ACTGRP) setting on the Create Program (CRGPGM) request. For an example of the ACTGRP setting, see the XML Interface for RPG and Procedural Languages documentation for building the COBOL samples.
- For detailed information about the XML Interface for RPG and Procedural Languages, see the API documentation, which includes working samples.
- As newer versions of this technology become available (for example, incorporating changes in the W3C standards for XML), you can find them on the IBM alphaWorks Web site .

XSL Processor for Java

The XSL Processor for Java allows you to use “XSL and XSLT” on page 5 to transform the data in your XML documents into a presentation language, such as HTML or WML, or into another XML document type.

You can run the XSL Processor for Java from the command line, in a Java program, or as a servlet. The default setting uses the “XML Parser for Java” on page 7, but it can interface to any XML parser that conforms to the “APIs” on page 4.

The XSL Processor for Java is a Java archive (JAR) file (xalan101.jar for XSL Version 1.0.1) is in the following OS/400 directory:

/QIBM/proddata/OS400/xml/lib

Before you can use the XSL processor in your programs or servlets, you need to add xalan101.jar to your CLASSPATH environment variable.

For detailed information about the XSL Processor for Java, see the API documentation, which includes working samples.

Note: The XSL Processor for Java is essentially a snapshot of XSL/XSLT W3C processor technology. As newer versions of this technology become available, you can find them on the IBM alphaWorks Web site or the Apache Web site .

Additional XML tools and programs

You can write your programs to take advantage of XML and its extensions and companion standards. Using additional tools that are not part of the OS/400 operating system enables you to do this with much less development time and effort.

The XML community provides a steady source of tools and helpful applications for using XML:

- "XML Enabler" (for the XSL Processor for Java)
- "TaskGuide Viewer"
- "Xeena"

IBM also offers licensed programs that will help you use XML with your iSeries servers:

- "IBM Toolbox for Java (ReportWriter classes)" on page 11
- "DB2 XML Extender" on page 11
- "IBM WebSphere Host Publisher" on page 11

XML Enabler

The XML Enabler is a servlet that developers can use with the XSL Processor for Java to put stylesheetsin effect in real time. When any browser sends a request to the servlet, it responds with data that it formats using XSL stylesheets configured for specific browser types. In this way, the XML Enabler allows any user of any browser to view and use XML data.

Any browser means just that. You do not need an XML-enabled browser because the servlet takes care of that by using XML and XSL technology combined with the information in the HTTP header. After the system administrator defines the mapping between browser types and XSL stylesheets, the servlet does the remaining.

When an HTTP request comes in to the XML Enabler, the following actions occur:

1. The XML Enabler gets the XML document requested by the client (the Web address of that document is passed as a parameter on the Web address).
2. The XML Enabler then looks at the client type, using the user-agent field of the HTTP header, and selects an XSL style sheet. The style sheet selected for each user-agent type is defined by the developer.
3. After the XML document and the XSL style sheet are selected, the XSL Processor for Java combines them, and the servlet returns the output to the client.

TaskGuide Viewer

TaskGuide Viewer is an XML-based tool for creating wizards that makes building and displaying wizards as easy as creating and viewing HTML files. After you create a wizard script, the TaskGuide Viewer displays the specified panels and follows your instructions.

The TaskGuide Viewer enables you to focus on task content rather than design elements. It offers usability-tested screen layout and navigation options that eliminate the most difficult parts of building wizards: screen layout, navigation, and data management.

For more information, see the TaskGuide Viewer information on the IBM alphaWorks Web site .

Xeena

Xeena is a visual XML editor that enables you to visually edit valid XML documents derived from a valid DTD. It is a Java application built on top of Swing and "XML Parser for Java" on page 7.

Use a hierarchical tree view to create, edit, and expand any document derived from a valid DTD. You can edit multiple XML documents and copy, cut and paste from one document into another.

A key feature of Xeena is its syntax-directed editing ability, which ensures that all documents generated are valid with respect to the given DTD. Xeena also helps you insert elements into the tree correctly (with respect to the DTD) by being sensitive to the current selected tree node and refusing to allow you to insert elements in an incorrect order.

For more information, see the Xeena information on the IBM alphaWorks Web site .

IBM Toolbox for Java (ReportWriter classes)

IBM Toolbox for Java is a set of Java(TM) classes that allow you to use Java programs to access data on your iSeries and AS/400e servers. You can use these classes to write client/server applications, applets, and servlets that work with data on your iSeries. The IBM Toolbox for Java includes a reportwriter package containing classes that enable your applications to work with XML data.

The ReportWriter classes allow your applications to create formatted documents from XML data sources. Your application defines the document format (layout) by using XSL stylesheets in combination with XSL Formatting Objects and the XML data sources. With ReportWriter classes you can generate documents in the Hewlett Packard Printer Control Language (PCL) format and the Adobe Portable Document Format (PDF).

For more information, see the ReportWriter classes in IBM Toolbox for Java.

DB2 XML Extender

The DB2 XML Extender is a licensed program that provides data types that you can store XML documents in DB2 UDB for iSeries databases and functions that assist you in working with these structured documents.

You can store entire XML documents in DB2 UDB databases as character data or store them as external files but still manage them by using DB2 UDB. Retrieval functions allow you to retrieve either the entire XML document or individual elements or attributes.

Other features include:

- The ability to extract XML elements and attributes into traditional SQL data types
- The ability to transform existing DB2 UDB data into XML documents
- Storage, retrieval, and updates of XML documents in a single column
- Storage of XML documents as a collection of DB2 UDB data in multiple columns and tables
- DTD management
- Support for international code pages

For more information, see the DB2 UDB XML Extender site .

IBM WebSphere Host Publisher

Through its XML gateway, WebSphere Host Publisher provides access to existing 3270 and 5250 applications in an XML format for use within e-business applications.

Additionally, the HTML mapper capability provides an install-and-go HTML entry-level emulator for 3270 or 5250 application access. Without customization, existing 3270 and 5250 applications can be extended as HTML to Web users. This capability is targeted at users who need occasional access to the host application and do not yet have desktops enabled for Java applications.

For more information, see the IBM WebSphere Host Publisher Web site .

Samples

Samples are provided in the included API documentation for each XML tool integrated into OS/400.

- XML Parser for Java

  To use the samples on your iSeries Model 400, see the included samples documentation.
- XSL Processor for Java

  To use the samples on your iSeries Model 400, see the included samples documentation.
- XML for C++ Parser

  > **Note:** Starting in V5R3M0, the XML parsers QSYS/QXML4C310 and QSYS/QXML4PR310 will no longer exist. Instead, use the XML Toolkit for iSeries (5733XT1) to provide the latest XML parsers.

  To use the samples on your iSeries Model 400, see the included samples documentation.
- XML Interface for RPG and Procedural Languages

  > **Note:** Starting in V5R3M0, the XML parsers QSYS/QXML4C310 and QSYS/QXML4PR310 will no longer exist. Instead, use the XML Toolkit for iSeries (5733XT1) to provide the latest XML parsers.

  The XML Interface for RPG and Procedural Languages includes samples for each supported language.

  To use the samples on your iSeries Model 400, see the included samples documentation.

Code example disclaimer

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you ″AS IS″ without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

XML related information

The XML community continually improves existing tools, adds tools, and helps evolve standards and extensions that increase the usefulness and flexibility of XML. Use the following links to view API documentation for parsers integrated into OS/400, find help and information for the most current versions of those parsers, and to keep up to date on developments in the XML community.

API documentation for tools integrated into OS/400

XML Parser for Java API documentation

XML for C++ Parser API documentation

XML Interface for ILE C, RPG, and COBOL API documentation

XSL Processor for Java

External links to tools integrated into OS/400

> **Note:** Starting in V5R3M0, the XML parsers QSYS/QXML4C310 and QSYS/QXML4PR310 will no longer exist. Instead, use the XML Toolkit for iSeries (5733XT1) to provide the latest XML parsers.

Use the following links to find the most current versions of the parsers and tools integrated in OS/400. All the links in this section connect to the IBM alphaWorks Web site .

XML Parser for Java (XML4J)

XML for C++ Parser (XML4C)

XML Interface for ILE C, RPG, and COBOL (XML4PR)

- If you have trouble installing or using the APIs after reading this documentation, try the previous link for more assistance. Use the discussion area to ask questions and read comments by other users.

XSL Processor for Java

Links to XML Web sites

Use the following links to see more information about XML, including introductory information, tutorials, advanced and related materials, evolving XML standards, and XML tools that you can download.

- IBM-sponsored sites
  - alphaWorks
  - developerWorks XML Zone
  - IBM eServer Solutions Enablement XML Technical Resources
- Other sites
  - W3C - XML
  - xml.apache.org
  - www.xml.org
  - www.xml.com

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

```
IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY  10594-1785
U.S.A.
```

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This Extensible Markup Language (XML) documents intended Programming Interfaces that allow the customer to write programs to obtain the services of XML.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
AS/400
e (logo)
IBM
iSeries
Operating System/400
OS/400
400

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

Code disclaimer information

This document contains programming examples.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you ″AS IS″ without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

Printed in USA