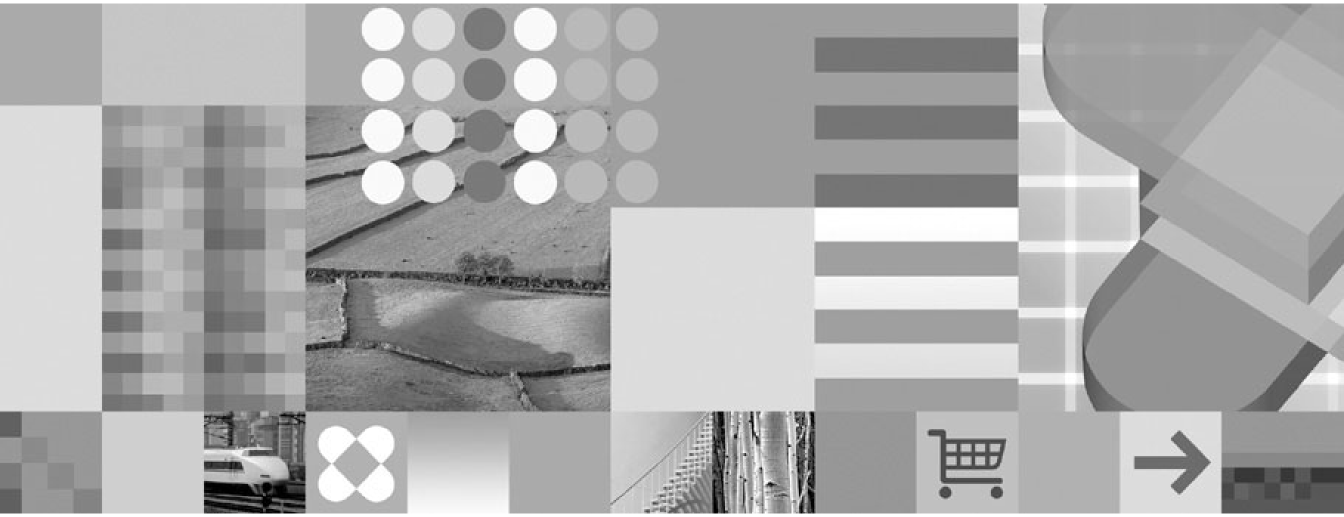
IBM® DB2® XML Extender for OS/390 and z/OS

Version 7

Implementation Hints and Tips

IBM® DB2® XML Extender for OS/390 and z/OS

Version 7



Implementation Hints and Tips

Third Edition (January 2005)

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

Contents

About this document

This document provides additional information (including hints and tips) to accompany the existing XML implementation documentation provided with DB2 for OS/390 Version 7 and other available IBM Redbook publications.

Available reference material

This section details the reference material available for implementation guidance:

- *XML Extender Administration and Programming for DB2 Version 7 (SC26-9949)* or *XML Extender Administration and Programming for DB2 Version 8 (SC27-1234)*

  Chapter 3: Preparing to use the XML Extender: Administration
- Red book: *DB2 for OS/390 and z/OS Powering the World's e-business. (SG24-6257)*

  Chapter 8 is split into two parts:
  - Part 1 for theory and reasons why XML. See page 193.
  - Part 2 for details on implementation and ″how-to″ instructions and tips. See page 215.

Chapter 1. XML Extender Prerequisites

UNIX Systems Services (USS) and Hierarchical File System (HFS)

Introduction

The XML Extender requires the UNIX System Services (USS) element of OS/390. USS was originally called OpenEdition MVS and was first made available with MVS/ESA V4.3. The product evolved to full SPEC 1170 (Universal UNIX) with MVS/ESA V5.2 and achieved XPG4 branding in 1995, making it a fully conforming UNIX operating system. The product has now been fully integrated into the OS/390 and z/OS code base with OS/390 and z/OS now openly considered to be a UNIX operating system.

Maintenance

Software servicing techniques such as ServicePac and SFS automatically ship fixes for the UNIX System Services (USS), including products that use the USS, such as WebSphere and DB2 Universal Database. The products store their code in the Hierarchical File System (HFS), a directory tree structure identical to that found on any UNIX environment. The USS must be active in order to access the HFS before applying the maintenance.

USS components

The USS comprises a number of components, which include:

- TSO shell (OMVS) and TSO commands
  - Is a full screen UNIX ″session″ running under TSO
  - Is invoked with TSO command OMVS from ISPF command shell (Option 6)
  - Functions as a native UNIX Environment, where you can enter commands such as **ls - l**, which lists all of the files in a particular directory.
- UNIX System Services ISPF Shell (ISHELL)
  - Acts as a dialog-based ISPF ″style″ interface to USS
  - Is invoked via TSO ISH command
  - Mounts Hierarchical File System (HFS) datasets
  - Edits, browses, and copies HFS members
- Other
  - Programming APIs

- dbx debugger

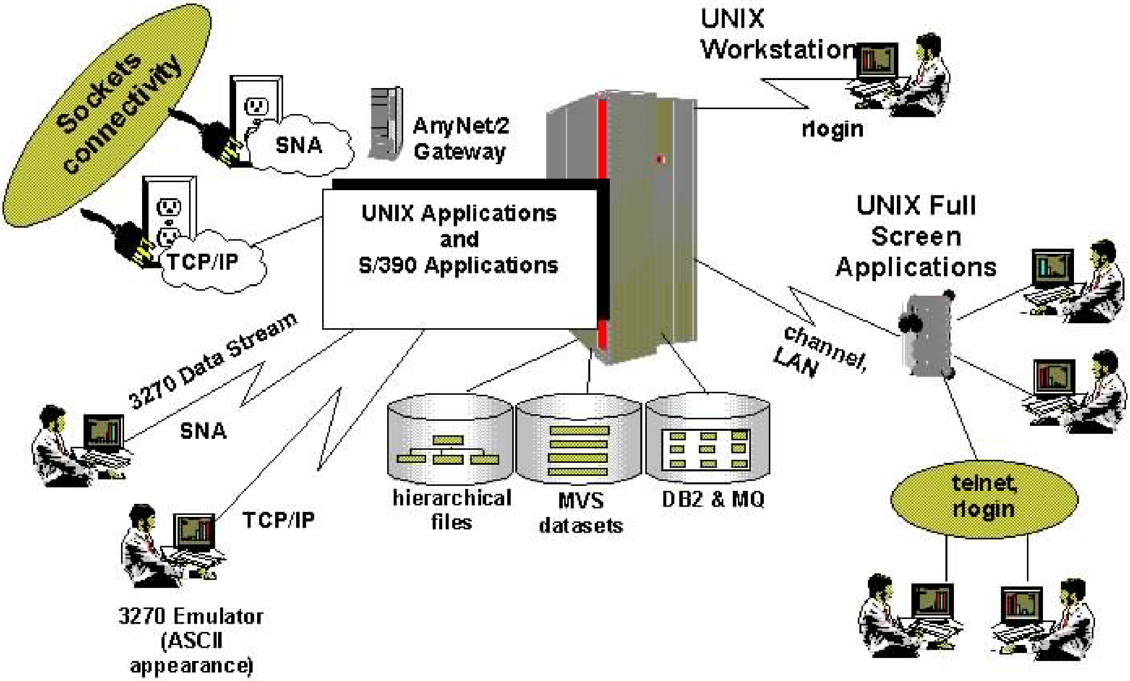


Figure 1. A Typical UNIX Systems Services (USS) Configuration

Resource Recovery Services (RRS)

RRS is the MVS system-level syncpoint manager and is required for XML Extender stored procedures and UDFs that run in WLM managed address spaces. For information on setting up WLM, RRS, and SPs, see the *DB2 for OS/390 V6 Management Tools Package (SG24-5759)* redbook.

RRS provides services that implement a two-phase commit protocol on the MVS platform and protects resources by invoking these services and providing exit routines. See *OS/390 MVS Programming: Resource Recovery (GC28-1739)* for an overview of how to plan for a resource manager, how to use RRS services, and how to code the exit routines. Details about each resource recovery service and each exit routine are also documented in this book.

Before you can start RRS, you must:

- Set up your log streams and ensure that system logger is active. (RRS requires system logger; if it is not active when RRS starts, RRS waits until system logger is active).
- Establish the priority of the RRS address space. The best way to control RRS priority is through the workload manager (WLM) and the recommendation is to use the system default for started tasks as the goal of the RRS address space.
- Define RRS as a subsystem by placing statement SUBSYS SUBNAME(RRS) in your IEFSSNxx parmlib member. Place the statement after the statement that defines the primary subsystem. You can replace RRS with a subsystem name of your choice, but do not supply any other parameters.

WLM Application Environment

Set up a WLM Application Environment to run UDFs and stored procedures such as DXXENABLEDB. WLM must be running in GOAL mode so that new stored procedure's address spaces are started automatically when required. Build the following procedure, and be sure to include the XML Toolkit library in the STEPLIB:

```
//ISC3XML  PROC RGN=0K,APPLENV=ISC3XML,DB2SSN=ISC3,NUMTCB=8
//IEFPROC EXEC PGM=DSNX9WLM,REGION=&RGN,TIME=NOLIMIT,
//        PARM='@DB2SSN, @NUMTCB, @APPLENV'
//STEPLIB  DD  DISP=SHR,DSN=ISC710P1.RUNLIB.LOAD
//         DD  DISP=SHR,DSN=SYS2.DB2.V710.SDSNLOAD   <======= DB2
//         DD  DISP=SHR,DSN=SYS2.DB2.V710.SDXXLOAD   <======= XML Extender
//         DD  DISP=SHR,DSN=SYS1.SIXMMOD1            <======= XML Toolkit
//SYSIN    DD  UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
//SYSPRINT DD  UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
//SYSTSPRT DD  UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
```

Multiple WLM environments can be configured for running XML UDFs. Specifying two WLM environments can improve performance because:

- All stored procedures will run in the first WLM environment
- All UDFs will run in the second WLM environment

Performance benefits of running in WLM GOAL mode

WLM will automatically start WLM established address spaces for UDFs to help meet your pre-set service class goals. In Compatibility mode, WLM cannot automatically start new address spaces to handle high-priority requests. Instead, performance of UDFs *must* be monitored to determine how many WLM address spaces are required. Address spaces must be manually started and stopped by an operator.

Warning message

During the APPLY of PTF UQ65515, you may receive a warning message labeled: MSGGIM23913W with RC = 8 even though the apply job ends with RC = 4. This is OK and you can proceed with the APPLY. The MSGGIM23901E error message may also occur with RC = 8. In this case, create a special SMP/E global zone option entry with a special link edit UTILITY entry to allow RC = 08 (one time only). This error will no longer occur after migrating to Version 7 Fixpak 8 (PQ73241/UQ76110).

XML Toolkit for OS/390 and z/OS Version 1.4 or 1.6

Requirements for DB2 Version 7: DB2 for OS/390, Version 7 FP10 (UQ79958) requires XML Toolkit Version 1.4 (FMID HXML140). Version 1.4 is no longer available to download. Instead, download the XML C++ Parser for the OS/390 V2.10 C++ Compiler for z/OS Web Deliverable at http://www14.software.ibm.com/webapp/download/product.jsp?id=WKOS-5DETKX.

Requirements for DB2 Version 8: DB2 for z/OS Version 8 FP7 (UQ93919) requires XML Toolkit Version 1.6 (FMID HXML160).

IBM XML Toolkit Version 1.6 is a no-charge product, which is available on tape or from the Internet to your workstation and then uploaded to the host. There are two download package options for XML Toolkit: SMP/E (orderable and downloadable) and non-SMP/E (downloadable).

The SMP/E install option is recommended for DB2 XML Toolkit. You can order the XML toolkit as a CBPDO or serverpac by calling 1-800-IBM-CALL. If you order the SMP/e install version of the XML Toolkit, you will be sent the current release and the two prior releases. Instructions will be provided to install each of the releases. The default install procedure will install previous releases as well as the latest release, if the prior releases were not installed.

The XML Toolkit includes:

- XML Parser for z/OS and OS/390, C++ Edition
- XML Parser for z/OS and OS/390, Java Edition
- Open-source code, run on multiple platforms with the benefit of input from a large community of developers

The code has been tested and packaged for deployment on z/OS or OS/390 and includes the enterprise servers' world class level of service.

Check that your host environment has the appropriate PTFs to support the new SMP/E feature. When running the jobs on the host, ensure that any SMP/E commands are in lowercase. It will *not* work otherwise.

For information about how to correctly implement the Toolkit for the current 1.6 (HXML160), see the *Program Directory for XML Toolkit for z/OS and OS/390 Version 1 Release 6, Modification Level 0*.

ODB2 command line tool (optional)

The ODB2 command line tool is an optional feature that allows you to issue SQL requests from the OMVS shell. This tool can be downloaded from the IBM OS/390 USS Tools and Toys Web site:

```
http://www.s390.ibm.com/products/oe
```

Select **Tools and Toys** from the menu panel, and then select **OS/390 UNIX Tools**. The list of downloadable UNIX tools should include the ODB2 tool. Make sure that you download or print the Readme file as well as downloading the tool itself, because the Readme file tells you how to install ODB2. After you install the tool, start the OMVS shell, change the directory path to one that is specified in your PATH environment variable, and define a symbolic link from DB2 to the odb2 executable, as in the following example:

```
cd /usr/lpp/bin
ln -si /usr/lpp/db2/odb2v1.3/odb2 db2
```

This allows you to run the sample command files provided with the XML Extender. If you are a frequent user of the OMVS shell, you will find ODB2 a useful tool for accessing DB2 data from the command line without having to exit the shell.

Summary of requirements

The following are general host requirements:

- DB2 Version 7 (HDB7710) with XML Extender (JDB771X (Compid 5740XYR06)), must both be installed in the same CSI.
- RRS is required for XML Extender stored procedures and UDFs that run in WLM managed address spaces.
- The ODB2 command line tool is required to allow SQL from the UNIX shell.
- The XML Parser for z/OS and OS/390, Java Edition requires that Java 1.3.1 or later be installed.

Running the XML Extender for z/OS from a workstation client

You can use the XML Extender administration wizard instead of the command line to run administration commands. You can install the wizard from:

```
http://www.ibm.com/software/data/db2/extenders/xmlext/downloads.html
```

The wizard requires the following software on the client desktop:

- DB2 UDB Connect Personal or Enterprise edition.
- JAVA Development Kit 1.3.1 or later.

Chapter 2. A USS view of the XML components

Introducing the USS Root directory

The internal directory structure of the Hierarchical File System looks exactly the same as any standard UNIX environment, with all files residing in a directory tree structure off the root directory. The XML files and directories shipped with the XML Extender product reside by default in your home directory.

Because the entire USS directory resides on the mainframe, the actual directory and file structure data must reside on a Direct Access Storage Device (DASD) supported by MVS. The data will reside in a series of MVS data sets as shown in the above diagram where you see the root directory housed in a data set called OMVS.HFS.ROOT on volume VOL123. The allocation for this data set would look something like the example JCL below:

```
//OMVSXX JOB
//STEP03  EXEC  PGM=IEFBR14
//MKFS    DD   DSNAME=OMVS.HFS.ROOT,
//             SPACE=(CYL,(40,1,1)),DCB=(DSORG=PO),
//             DSNTYPE=HFS,
//             DISP=(NEW,CATLG,DELETE),
//             STORCLAS=STANDARD
```

The root file system must be allocated by a user who has a UID of 0, indicating superuser authority. To create HFS file system data sets, a security product that supports the SAF calls made during the system processing must be running. **Note:** All newly allocated file systems have permission bits set at 700.

The example above shows a sample job that defines the root HFS data set in an MKFS DD statement. The allocation specified in this sample does not reflect the amount of space needed for the root file system for MVS. For exact size information, consult the OS/390 Program Directory. When you specify space for the HFS data set, either provide a nonzero value for the directory space parameter (which is not used) or specify DSORG=PO to create a data set with partitioned organization.

Later, during the customization process, the DSNAME of the root file system will be used in the ROOT statement in the BPXPRMxx parmlib member.

You can issue the following command from ISPF to dynamically mount the HFS after it is created and before the parmlib update takes place:

```
TSO MOUNT FILESYSTEM('OMVS.HFS.ROOT') MOUNTPOINT('/') TYPE(HFS) MODE(RDWR)
```

Installing the XML Extender into the HFS

The DB2 XML Extender will (by default) be installed into the directory $HOME/dxx/samples. When in USS, you view the entire UNIX environment as a tree structure starting from the ″/″ root directory. If you have the authority, you can issue a ″change directory″ cd command to any directory in the entire tree structure.

USS security

RACF (or other security packages including ACF2) can be used to house security for the UNIX Systems Services file structures. Security controls are defined in an OMVS segment which is required to access UNIX services and consists of a UID, GID, PROGRAM, and HOME directory reference.

- UID or User ID is the UNIX equivalent to a USER in RACF and is a value in the range of 0 to 2147483647. A user defined with a UID of 0 is a special Superuser who has access to all UNIX resources currently defined.
- GID or Group ID is UNIX equivalent to a GROUP in RACF and values range from 0 to 2147483647. There are *no* special values for GID.
- PROGRAM defines the shell program used for a given user and takes the form of *PROGRAM(′/bin/sh′)*.
- HOME defines the home directory for a given user and takes the form of *HOME(′/u/cooki′)* where *cooki* is the user name. The home directory for a given user is the starting point that the user is positioned at when they invoke USS.

XML Extender HFS contents

After DB2 XML Extender is used, the USS structure (by default) will look like the following diagram:

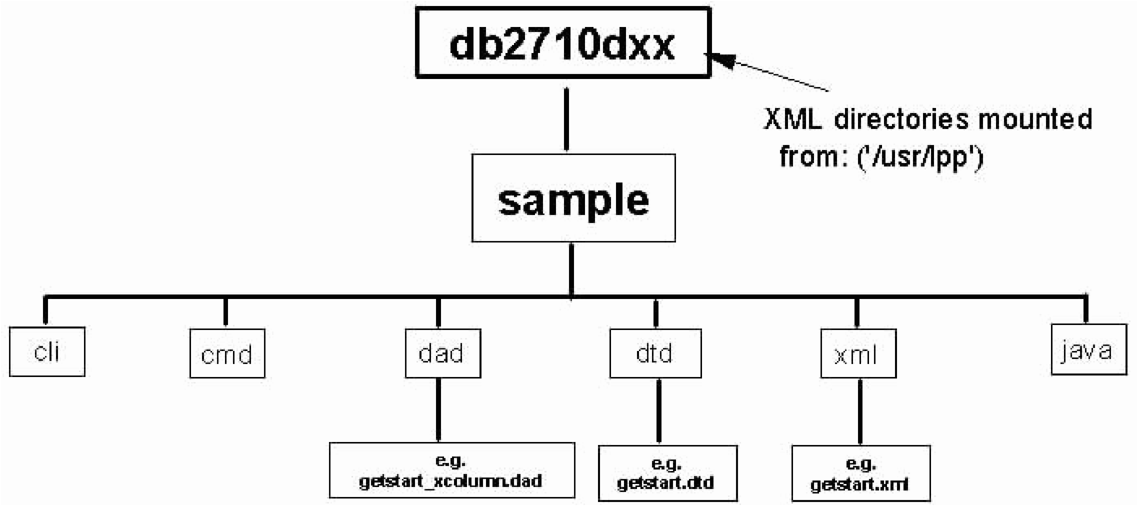


Figure 2. USS Structure

Chapter 3. Enabling the XML Extender

This chapter supplements the information found in *DB2 XML Extender Administration and Programming for z/OS*.

DXXGPREP

Binds and enables the server. This job includes the following:

- Frees/binds DB2 plans/packages
- Drops/creates DB2 XML procedures/LOBs
- Grants authorizations
- Copies XML scripts to USS
- Enables XML server

Hints and Tips

- Before running this batch job, make sure that you read the instructions in the prolog for any required customizations.
- The job contains a number of MKDIR statements that define the USS directories, and each has permission settings for each directory set as MODE (7,5,5). Depending upon the level of USS authorization that you have, you may need to change these to ″MODE(7,7,5)″ to provide the necessary permission required to create each subsequent directory in the path to the new directories. Another alternative could be to make your user ID the owner of each directory; however, this is not a valid option at the /usr or /lpp levels.
- DB2 Objects should be created with DB2XML as the owner.
- A DB2XML database has been created to house the DB2XML product objects.

DB2 XML Extender for OS/390 and z/OS no longer supports the usage of the **using** keyword while specifying tablespace names while enabling a server. In Chapter 3 of *DB2 XML Extender for OS/390 and z/OS Administration and Programming , Version 7 (SC26-9949-00)*, there is a section labeled ″Tasks that the DXXGPREP job performs.″ Part of this section is incorrect and should read:

```
Enables the DB2 subsystem to XML specifying the following options:
ENABLE_SERVER -a V71A wlm environment wlm_environment_name
```

There are other examples in *DB2 XML Extender Administration and Programming* such as :

```
dxxadm 'enable_server -a SUBSYS1 using tbspc1,tbspc2
wlm environment envir233' ASIS
```

where the keyword ″using″ should no longer be used.

Chapter 4. Getting started with XML Extender

This chapter contains only **supplemental** information on these batch jobs, and more complete definitions of these batch jobs can be found in *DB2 XML Extender Administration and Programming for z/OS*.

Before running these batch jobs, make sure to read the instructions in the prolog for any required customizations.

Store an XML document in a column

Run these batch jobs to store an XML document in a column.

DXXGDB

Creates and populates sample tables:

- Creates ORDER_TAB, PART_TAB, SHIP_TAB, RESULT_TAB
- Populates ORDER_TAB, PART_TAB, SHIP_TAB

Hints and Tips

It is best to leave the DDL unchanged (no schema) and to create the objects with your own prefix. Later, when running scripts in USS or sample batch jobs, the unqualified DB2 objects will default to your user ID as the schema.

DXXGIDTD

Inserts the DTD ″getstart.dtd″ into the DTD repository, the DB2 table: DB2XML.DTD_REF. See the SQL shown below:

```
INSERT INTO DB2XML.DTD_REF
   (DTDID,
   CONTENT,
   USAGE_COUNT,
   AUTHOR,
   CREATOR,
   UPDATOR)
    VALUES('getstart.dtd',DB2XML.XMLCLOBFROMFILE
   ('dxx/sample/dtd/getstart.dtd'),
    0, 'user1', 'user1', 'user1')
```

Hints and Tips

Make sure to set the correct path of the DTD file.

DXXGCTCO

Creates the DB2 Sales table SALES_TAB as follows, with a ″unique″ index:

```
CREATE TABLE SALES_TAB(INVOICE_NUM CHAR(6)
NOT NULL PRIMARY KEY, SALES_PERSON VARCHAR(20))  IN DATABASE DB2XML;

CREATE UNIQUE INDEX SALES_TABI ON SALES_TAB(INVOICE_NUM);
```

Hints and Tips

It is best to leave the DDL unchanged (no owner for objects) and to create the objects with your own user ID as the schema.

DXXGATCO

Adds the XML column ORDER to the DB2 Sales table SALES_TAB.

```
ALTER TABLE SALES_TAB ADD ORDER DB2XML.XMLVARCHAR;
```

DXXGECOL

Enables the XML ORDER column:

- This batch job first copies the DAD file to your home directory. The command does this because of the 100 character parameter limitation in the enable column step. Running the getstart_enableCol.cmd command script eliminates this problem and runs with the DAD file where job DXXGPREP installed it.
- The following side tables are created during the enable process:

  ```
  ORDER_SIDE_TAB
  PART_SIDE_TAB
  SHIP_SIDE_TAB
  ```

DXXGCRIN

Creates indexes on the side tables.

Hints and Tips

The following indexes were created:

```
CREATE INDEX KEY_IDX ON ORDER_SIDE_TAB(ORDER_KEY);
CREATE INDEX CUSTOMER_IDX ON ORDER_SIDE_TAB(CUSTOMER);
CREATE INDEX PRICE_IDX ON PART_SIDE_TAB(PRICE);
CREATE INDEX DATE_IDX ON SHIP_SIDE_TAB(DATE);
```

DXXGIXML

Inserts the XML document.

Hints and Tips

The following OPUT command copies the supplied XML data into the HFS, then the INSERT command inserts an XML document into the sales_tab XML column:

```
OPUT 'COOKI.DB2.V710.SDXXXML(DXXGXML)' -
'dxx/sample/xml/getstart.xml' BINARY
```

```
INSERT INTO SALES_TAB(INVOICE_NUM, SALES_PERSON, ORDER)
VALUES('123456', 'Sriram Srinivasan',
DB2XML.XMLVarcharFromFile
('dxx/sample/xml/getstart.xml'));
```

DXXGCQOL

Queries the XML document.

Hints and Tips

The following SELECT statement was issued:

```
SELECT DISTINCT SALES_PERSON FROM SALES_TAB S,
PART_SIDE_TAB P
WHERE PRICE > 2500.00
AND S.INVOICE_NUM = P.INVOICE_NUM;
```

Composing an XML document from DB2 data

Compose an XML document from existing DB2 data. Use a DAD to map source table structures to an XML document structure. Run the following batch jobs to compose an XML document.

DXXGDB
This batch job is described above in **Store an XML document in a Column**.

DXXGSTP
Composes an XML document and displays the results.

DXXGEXML
Exports an XML document. Check that you indicate the correct path to export the XML file.

Running DXXADM from USS

The most convenient way to run the DXXADM command (for example, to enable columns) is to execute the program from the OMVS shell. This avoids the 100 character limitation of the parameter in JCL and TSO.

1. Invoke the OMVS shell and change the directory to one in your PATH environment variable.
2. Create a dummy file called **dxxadm**. Make sure the user permissions for dxxadm (the administration program) are correct, and turn the ″sticky bit″ on. This causes USS to search STEPLIB, LPA and the link list concatenation for the program, instead of searching for an executable file stored in the HFS. To do this:

```
cd bin
touch dxxadm
chmod 1755 dxxadm
```

OS/390 V2R10.0 UNIX System Services Command Reference, SC28-1892 provides more information on the syntax and options of these commands. As they are typically executed only once, you can execute them from the OMVS command line, but you can also include them in a shell script. See *OS/390 V2R10.0 UNIX System Services User's Guide, SC28-1891* for more information on creating and using shell scripts.

3. Issue the following command either from the shell (specifying your own data set names) or by adding it to your .profile file to make sure that the XML and DB2 load libraries are part of STEPLIB:

   ```
   export STEPLIB=DSN710.SDSNLOAD:DSN710.SDXXLOAD:$STEPLIB
   ```

   This command must be executed once in every session before running dxxadm. You can use this method to enable running the XML Extender sample programs (insert, retrieve, retrievc, shred, tests2x and tests2xc) as well as dxxtrc, to turn the trace on or off from USS.

You can either issue the dxxadm command directly from the OMVS shell or by creating a simple shell script and executing the script. If you are typing a long command that will not fit on the command line, you can use the \ (backslash) continuation character at the end of the first line. When you then press **Enter**, the command line is cleared so that you can continue typing. The line you typed prior to the backslash is displayed in the output area, and the shell prompt changes to > beneath it to indicate that you are continuing a command.

The backslash is also the continuation character in a shell script. For example, the command to enable the ORDER column in the **Getting Started** samples could be entered as:

```
dxxadm enable_column -a DB2A SALES_TAB ORDER \
dxx/sample/dad/getstart_xcolumn.dad -v \
SALES_ORDER_VIEW -r INVOICE_NUM
```

Creating a shell script using the ISPF editor

You can create a shell script using the ISPF editor in OMVS (oedit) and save it as enable_col.scr.

1. Set the execute permission bits:

   ```
   chmod +x enable_col.scr
   ```

2. Execute enable_col.scr from the OMVS command line:

   ```
   enable_col.scr
   ```

Chapter 5. DB2 XML Extender Administration

You can perform administration functions for XML Extender in the following ways:

- Invoking the DXXADM command from MVS batch jobs.
- Invoking the DXXADM command from within the USS.
- Running batch jobs, based on samples found in the SDXXJCL library, as shipped with DB2/390 XML Extender product feature.
- Building custom built applications using the XML Extender Administration Stored Procedures.
- Invoking the XML Extender Administration wizard.

Introduction to the XML Administration wizard

The XML Administration wizard is a Java-based application that allows you to easily administer your XML Extender–enabled data sources. It provides an easy-to-use interface to perform these tasks:

- Enabling or disabling a server for XML Extender
- Working with XML columns - adding, enabling, or disabling an XML column
- Working with XML collections - enabling or disabling an XML collection
- Creating and editing DAD files that map the XML tree structure to relational tables to generate or shred XML documents (This function is being deprecated from the wizard, and an alternative DAD editor is provided in the Websphere Application Development (WSAD) tool. It is recommended that you use the WSAD tool.).
- Importing DTDs into the DTD repository

XML Administration wizard software and prerequisites

You can get the XML Extender Administration wizard software from either of the following sources:

- It is shipped as member DXXADMIN in the SDXXADM library on OS/390 and z/OS. This library comes standard when you install XML Extender with DB2 UDB for OS/390 and z/OS Version 7 or later.
- It can be downloaded from the XML Extender software web site at:

  `http://www.ibm.com/software/data/db2/extenders/xmlext/downloads.html`
- If you install DB2 XML Extender on your workstation, the Administration wizard is included in the installation.

Running the Administration wizard on the workstation requires:

- DB2 or Connect Personal Edition
- DB2 XML Extender

Installing XML Extender includes the Administration wizard and the XML Extender samples.

Setting up the Administration Wizard

To set up and invoke the XML Extender Administration wizard:

1. Download the SDXXADM member DXXADMIN in binary format to the workstation.
2. Unzip the file using PKUNZIP. Unzipping extracts a dxxadmin.jar file that executes the Administration wizard.
3. To use the XML Extender administration wizard, you must have your PATH and CLASSPATH environment variables set up properly.

   In this section, the following terms are used to indicate product installation directories. These are documentation conventions, not actual variables.

Version 7	**Version 8**
dxx_install	*dxx_install*

   The directory under which the XML Extender has been installed. For example, on Windows operating systems, the default is: c:\dxx

Version 7	**Version 8**
db2_install_directory	*c:\program\ibm\sqllib\bin*

   The directory under which DB2 has been installed. For example, on Windows operating systems, the default is:

Version 7	**Version 8**
c:\program files	*c:\program files\ibm*

   The PATH statement must contain the directories for the Java™ binary files and optionally the XML Extender administration directory. The following table describes the requirements for the PATH statement:

Table 1. CLASSPATH values

Software	Path Value Version 7	Path Value Version 8
XML Extender administration directory	*dxx_install/dxxadmin/*, except on Linux, where the `dxxadmin` directory is called `adm_g`	*SQLLIB/bin*
Java binary files	Depends on the Java package you use, which could be one of the following: • DB2–provided Java Developer's Kit (JDK) or Java Runtime Environment (JRE) in *SQLLIB/java/jdk/bin* or *SQLLIB/java/jre/bin* • Another JDK or JRE that exists on the system	Depends on the Java package you use, which could be one of the following: • DB2–provided Java Developer's Kit (JDK) or Java Runtime Environment (JRE) in *SQLLIB/java/jdk/bin* or *SQLLIB/java/jre/bin* • Another JDK or JRE that exists on the system

The CLASSPATH points to the administration wizard jar files, Java class libraries, DB2 JDBC™ driver file, and the Swingall jar file. It is used to invoke the wizard. See the *DB2 Universal Database XML Extender Administration and Programming, Version 7* documentation for more information about starting the wizard.

The JDK and JRE are provided with DB2 UDB Version 7, on certain operating systems. When available, the JRE is automatically installed, and the JDK is installed when you install the DB2 UDB Application Development selection. Using the JDK or JRE provided with DB2 is optional. See DB2 Universal Database: Java Enablement with DB2 Universal Database Version 7.1 Web site for detailed information about Java support in DB2 Universal Database Version 7.1.

If you are using an operating system in which Java support is not provided by DB2, you must install the JDK or JRE yourself. Use the link above to the DB2 Java site to find download packages for the various levels of Java that have been tested with DB2. You can also choose to install another version of the JDK or JRE even if you have the DB2–provided versions installed. In this case, you should point to the same JDK, or JRE, installation in both the PATH and in the CLASSPATH environment variables.

Table 2. Path Elements

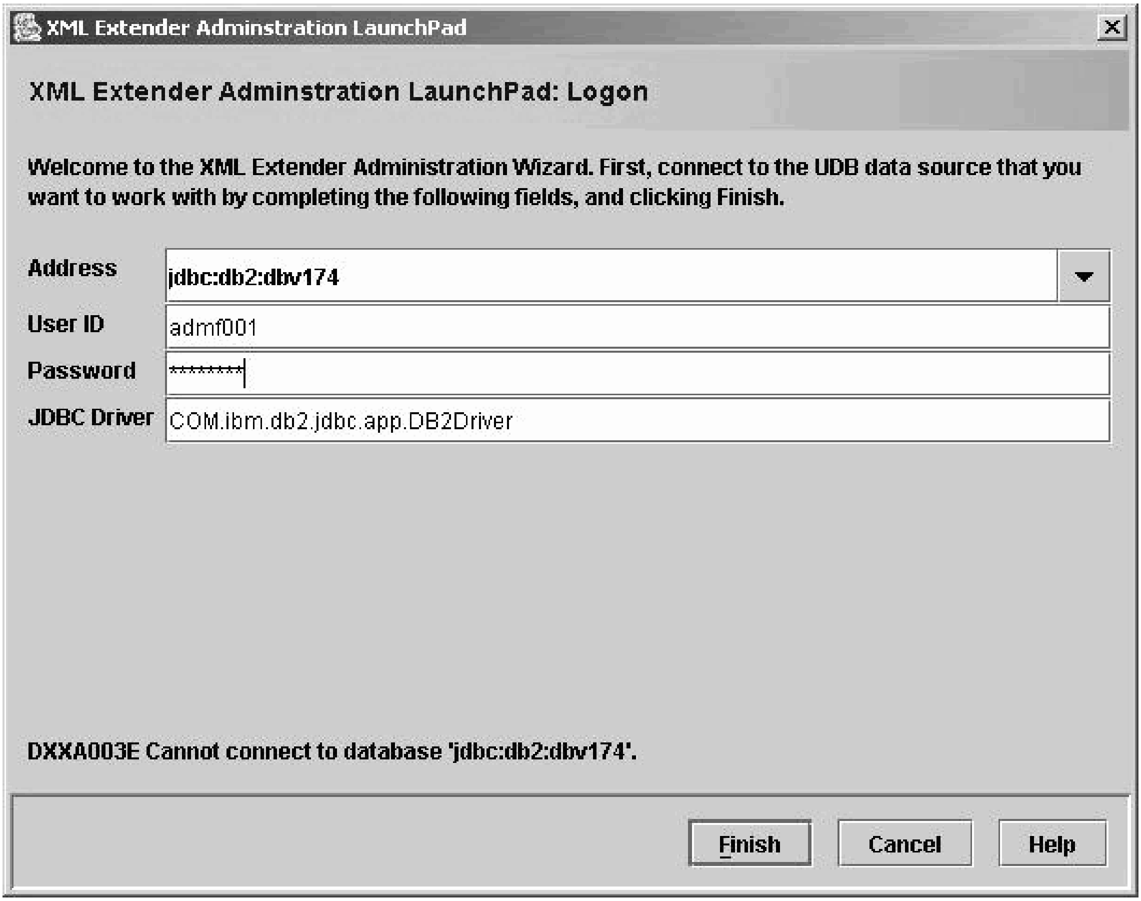
Software	CLASSPATH Value
XML Extender V7 jar files: • dxxadmin.jar • xml4j.jar	V7: • *dxx_install/dxxadmin/dxxadmin.jar* • *dxx_install/dxxadmin/xml4j.jar* On Linux, the `dxxadmin` directory is called `adm_g`.

Table 2. Path Elements (continued)

Software	**CLASSPATH Value**
XML Extender V8 jar files: • dxxadmin.jar • xerces.jar • xml-apis.jar	V8: • *SQLLIB/tools*
Java class libraries: determined by whether you are running the **java** or **jre** command.	**java command** Point to the `classes.zip` file. For example: *SQLLIB/java/jdk/lib/classes.zip***jre command**Point to the `rt.jar` file. For example: */db2_install_directory/SQLLIB/java/jre/lib/rt.jar*
Swingall.jar file	Available from three sources: • JDK 1.3.1 or higher • DB2 Connect Personal Edition V7 or higher • downloading from the Web separately For V7 using the DB2-provided swingall: *SQLLIB/java/swingall.jar* For V8 using the DB2-provided swingall: *SQLLIB/doc/prodcd/en_US/swingall.jar*
JDBC: db2java.zip	Available in DB2 Connect Personal Edition V7 or higher under: *SQLLIB/java/db2java.zip*

4. Test the installation by invoking the DB2 XML Extender Administration wizard as follows:
   a. Invoke the Administration wizard:
      - To invoke the Administration wizard from the **Start** menu, click the **Start** button and select **Programs** ⟶ **DB2 XML Extender** ⟶ **XML Extender Administration Wizard**. These steps only work for version 7.
      - To invoke the Administration wizard for version 8, enter the following command from the command line:

        ```
        dxxadmin
        ```
   b. After you start the Administration wizard, you see the logon panel:



XML Extender Adminstration LaunchPad

XML Extender Adminstration LaunchPad: Logon

Welcome to the XML Extender Administration Wizard. First, connect to the UDB data source that you want to work with by completing the following fields, and clicking Finish.

Address: jdbc:db2:dbv174

User ID: admf001

Password: ********

JDBC Driver: COM.ibm.db2.jdbc.app.DB2Driver

DXXA003E Cannot connect to database 'jdbc:db2:dbv174'.

Figure 3. Logon page of Administration wizard

c. On the Logon Window, type the information describing the server that is associated with the XML data that you want to work with. The address field contains the fully qualified JDBC URL to the data source you are connecting to. For example:

```
Jdbc:db2:ISC710P1
```

where ISC710P1 is the alias database name cataloged on the local machine.

The following commands allow SQL requests to be fowarded by the local DB2 system to the remote server:

```
-- Remove prior catalog entries
db2 uncatalog node isc710p1
db2 uncatalog database db710p1
db2 uncatalog dcs database isc710p1
```

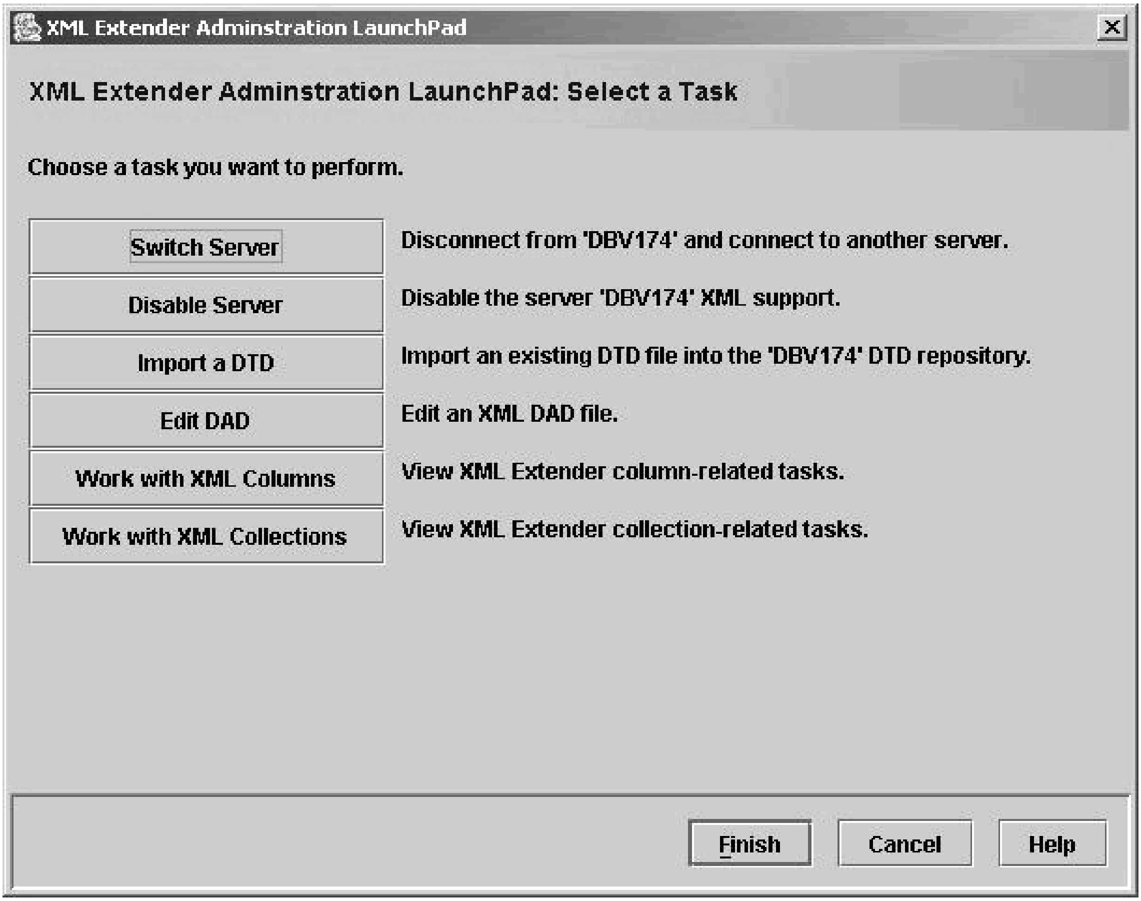
```
-- Add TCP/IP node entry to the node directory to access the remote node
db2 catalog tcpip node isc710p1 remote isc710p1.stl.ibm.com server 446
-- Store remote database location information in the local database
   directory
db2 catalog database isc710p1 as db710p1 at node isc710p1
    authentication dcs
-- Store remote database information in the Database Connection Services
   (DCS) directory
db2 catalog dcs database isc710p1 as stlec1
```

d. User ID and password are for a valid TSO user ID and password from the host.

e. Type the following for the JDBC Driver name:

```
COM.ibm.db2.jdbc.app.DB2Driver
```

The XML Extender Administration launchpad is invoked following a successful logon via the panel detailed above and the following panel appears, highlighting the various administration options available:



XML Extender Adminstration LaunchPad

XML Extender Adminstration LaunchPad: Select a Task

Choose a task you want to perform.

Disconnect from 'DBV174' and connect to another server.

Disable the server 'DBV174' XML support.

Import an existing DTD file into the 'DBV174' DTD repository.

Edit an XML DAD file.

View XML Extender column-related tasks.

View XML Extender collection-related tasks.

Figure 4. Task selection page of the Administration wizard

Chapter 6. Setting up XML Extender on a Windows client and a z/OS server

Software requirements

- Windows NT or Windows 2000
- DB2 UDB Version 7.2 or later
- Java Development Kit 1.3.1 or later

Setting up DB2 XML Extender on a Windows client to an z/OS server

When using DB2 XML Extender with a z/OS server, the following *getstart* commands cannot be run:

- getstart_prep.cmd
- getstart_clean.cmd
- getstart_enableCol.cmd
- getstart_stp.cmd
- getstart_exportXML.cmd
- getstart_xslt.cmd

The rest of the *getstart* commands can be run against a z/OS server in the following sequence **(with indicated modifications)**:

1. Catalog the node and database. For example:

   ```
   db2 catalog tcpip node node_name remote server_name server
   server_number

   db2 catalog database node_name as sales_db at node node_name
   authentication dcs

   db2 catalog dcs database database_name as stlec1
   ```

2. Run **getstart_db.cmd**. Before running getstart_db.cmd, you must:
   - remove the database creation statement within the command:

     ```
     db2 "create database SALES_DB"
     ```

   - modify the database connection statement within the command to include your user id and password.
3. Enable the server from the XML Extender administration wizard.
4. Run the **getstart_insertDTD.cmd**. Before running getstart_insertDTD.cmd, you must:
   - modify the database connection statement within the command to include your user id and password.

   - specify the correct path to find the getstart.dtd file in the *db2xml.XMLClobFromFile* statement.

5. Run the **getstart_createTabCol.cmd**. Before running the getstart_createTabCol.cmd, you must:
   - modify the database connection statement within the command to include your user id and password.
   - add a line to create a unique index for the primary key:

     ```
     db2 "create unique index invoice_num_index
     on sales_tab(invoice_num asc)"
     ```

6. Run the **getstart_alterTabCol.cmd**. Before running the getstart_alterTabCol.cmd, you must:
   - modify the database connection statement within the command to include your user id and password.
7. Enable the column from the XML Extender administration wizard. Make sure to FTP the getstart_xcolumn.dad file from MVS to your Windows client first, because the XML Extender administration wizard only recognizes files on your local machine.
8. Run the **getstart_createIndex.cmd**. Before running the getstart_createIndex.cmd, you must:
   - modify the database connection statement within the command to include your user id and password.
9. Run the **getstart_insertXML.cmd**. Before running the getstart_insertXML.cmd, you must:
   - modify the database connection statement within the command to include your user id and password.
   - specify the path of the getstart.xml file.
10. Run the **getstart_queryCol.cmd**. Before running the getstart_queryCol.cmd, you must:
    - modify the database connection statement within the command to include your user id and password.
11. Disable the XML column with the XML Extender Administration Wizard.
12. You can clean entries from the db2xml.dtd_ref table and the sales_tab table by entering the following SQL statements from a DB2 command line:

    ```
    db2 "connect to SALES_DB user nnn using xxx"
    db2 "delete from db2xml.dtd_ref where dtdid='getstart.dtd'"
    db2 "drop table sales_tab"
    ```

    where ″nnn″ is the database user id and ″xxx″ is the password.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

AIX
DB2
DB2 Universal Database
IBM
MQSeries
OS/400
OS/390

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

Java, JDK, and JDBC are registered trademarks of Sun Microsystems, Inc..

UNIX is a registered trademark of X/Open Company Limited.

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

Printed in USA