

# Welcome to the e-learning workshop "Using the DB2 XML Extender"

## Part 2: Administration

# Overview of education series

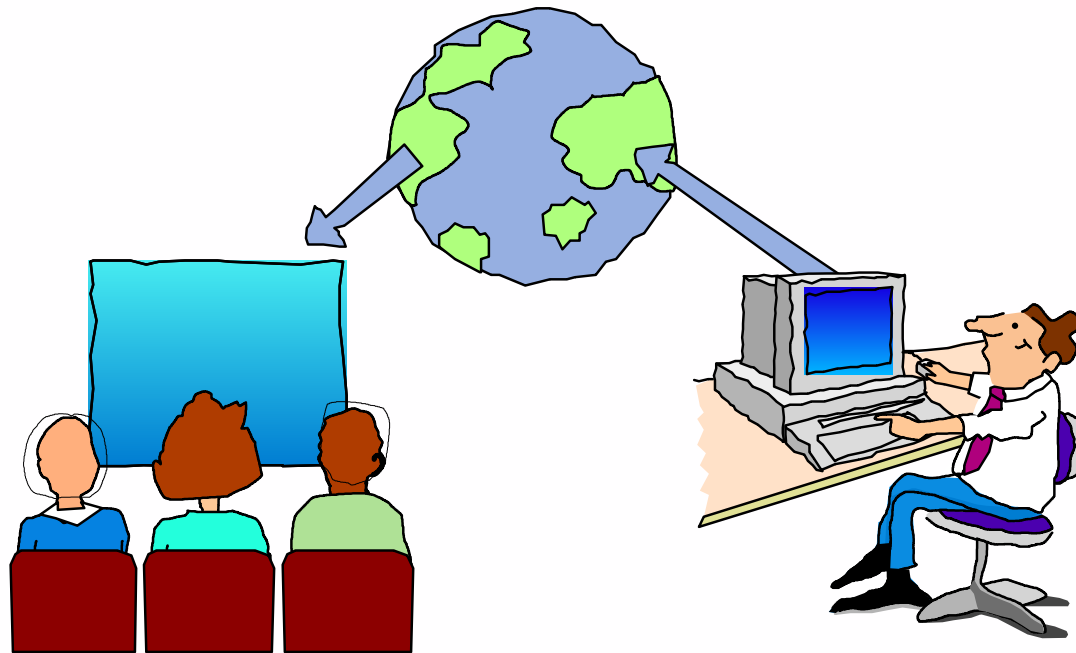
---

- Workshop 1 : DB2, XML and Web services
  - ▷ Part 1 : Intro to XML and DB2 XML Extender
  - ▷ Part 2 : DB2 and Webservices
  
- Workshop 2 : Using the DB2 XML Extender
  - ▷ Part 1 : Mapping XML to DB2 databases, Building DADs
  - ▷ **Part 2: Administration, Storing and Retrieving data**

# E-learning

---

- Please ask questions via Sametime messaging at the bottom part of the screen
- Phones will be muted



# Agenda

---

- Administration
  - Enabling databases, columns and collections
  - Tools and facilities
- Storing and retrieving XML data
- Practical advice
- Repository futures
- Hands on exercises
- OS/390 considerations

# XML Administration

Enabling databases, columns, collections  
Including tooling

IBM Software Group

# Enabling the database/subsystem

---

- For distributed DB2
  - ▷ Enable database via dxxadm enable\_db or wizard
- For z/OS
  - ▷ Enable subsystem
- This creates the following:
  - ▷ User-defined types eg XMLVarchar, XMLCLOB
  - ▷ User-defined functions eg extractInteger
  - ▷ Control tables and metadata eg DTD\_REF, XML\_USAGE
  - ▷ DB2XML schema and assigns appropriate privileges

# Administration tables

---

## □ XML\_USAGE

- ▷ Read-only - should only be updated via XML extender
- ▷ Stores information about each XML-enabled column or collection
- ▷ Key is Table\_name, Column\_name
  - Schema will be added in future
- ▷ DAD stored as CLOB

# Administration tables

---

## □ DTD\_REF

- ▷ Populate via Insert statement or Import option in wizard
- ▷ DTDID is the unique, primary key
- ▷ DTD stored as XMLCLOB
- ▷ One DTD can be used by many documents, columns, collections
  - Can only update the DTD when the usage count is zero
- ▷ The DTDID can be a full path file name or anything unique



# Setting up an XML collection

---

## □ **Step 1 : Insert DTD into DTD\_REF (optional)**

- ▷ Via DB2 XML Administration Wizard, Command Window, SPUFI etc
- ▷ Necessary only if validation is required NB performance impact
- ▷ **Note that if the DTD changes you will need to reinsert!**

## □ **Step 2 : Create DAD**

- ▷ Via Websphere Studio or text editor

# Setting up an XML collection

---

## □ **Step 3: Enable collection**

- ▷ Via Administration Wizard or dxxadm enable\_collection command
- ▷ Inserts a row in XML\_USAGE table
- ▷ Tables will be created if they do not already exist

## □ **Step 4 : Test using Stored Procedure Builder**

- ▷ Avoids writing calling programs
- ▷ Can cut and paste XML documents into parameters fields

# Enable collection parameters

---

**dxxadm enable\_collection dbname collname dadfile -t tname**

- **dbname** : database name (Windows) or -a subsystem name (z/OS)
- **collname** : collection name
  - ▷ Stored in XML\_USAGE and used by XML stored procedures
  - ▷ NB collection names are case sensitive
- **DAD file**
  - ▷ Filename on operating system (USS filename for z/OS)
- **tname**
  - ▷ Tablespace where tables will be created if they don't already exist
  - ▷ Recommend that you create any new tables in advance with required tablespace parameters and indexes

# Defining an XML column

---

- XMLVARCHAR

- ▷ Maximum length 3K

- XMLCLOB

- ▷ Define a ROWID column to support it
- ▷ Define as NOT LOGGED
- ▷ Maximum size 2GB

- XMLFile

- ▷ Allows you to store the XML column in an external file, stored in the HFS
- ▷ Only the external file name is stored in the XML column
- ▷ Maximum sizes 2GB as data goes via UDF as CLOB

# Setting up an XML column

---

- **Step 1 : Register DTD (optional)**

- ▷ Insert into DTD\_REF or use XML Wizard

- **Step 2 : Create column to be enabled for XML**

- ▷ Via CREATE TABLE, ALTER TABLE or XML Wizard

- **Step 3 : Create DAD**

- ▷ If validation and/or side tables are required

- ▷ Decide which elements and attributes are to be used for searching

- ▷ Via Websphere Studio or Editor

# Setting up an XML column

---

- **Step 4: Create tablespaces for side tables if any**
- **Step 5: Enable column**
  - ▷ dxxadm\_enable\_column via command or XML wizard
  - ▷ Required parameters : subsystem, table name, column names and DAD file
  - ▷ Performs the following:
    - ⇒ Creates the side tables defined in the DAD file
    - ⇒ Creates the triggers on the XML column to keep the side tables in sync
    - ⇒ Adds entry in XML\_USAGE to keep relationship between column and side tables
    - ⇒ Increments USAGE\_COUNT in DTD\_REF if validation performed
  - ▷ **YOU ARE RESPONSIBLE FOR CREATING INDEXES IF REQUIRED**

# Enable column optional parameters

---

- **-t tname**

- ▷ Specifies a particular tablespace for the side tables
- ▷ Recommended to avoid default database, parameters etc
- ▷ Future enhancement may allow one for each side table

- **-v viewname**

- ▷ Sets up a default view giving a single view of XML column and side tables
- ▷ Consider performance impact if many tables are to be joined

- **-r name**

- ▷ Name of the single primary key in the user table which is added to the side tables to tie the data together
- ▷ It is recommended you define your own primary key as root\_id
- ▷ If it is not specified and the table has no single column primary key defined, DXXROOTID is automatically added to the user table and side tables

# Enable column example

```
dxxadm enable_column SALES_DB  
sales_tab order getstart.dad -v  
sales_order_view -r invoice_num
```

## SALES\_TAB

INVOICE_NUM	CHAR(6)
SALES_PERSON	VARCHAR(20)
ORDER	XMLVARCHAR

Triggers

Side tables (as defined in DAD)

Each contain  
invoice\_num

## SALES\_ORDER\_VIEW

(default view - join of  
the 4 tables using  
invoice\_num)



# XML Column set-up considerations

---

- Create LOB columns as not logged
  - eg `Create table X`  
`(COL1 INT NOT NULL,`  
`LOB_COL DB2XML..XMLCLOB NOT LOGGED...`
- Consider creating your own views if there are many tables as may give better performance than default view
- Note that documents in the same XML column can be validated using different DTDs as long as they have similar structures
  - ▷ Specify DTD in the DOCTYPE of each XML document
  - ▷ At least one DTD must be registered in DTD\_REF
- You cannot enable a column without creating at least one side table - if you don't want one just store the data in a LOB (but no DTD validation)

# If you need to change the DAD...

---

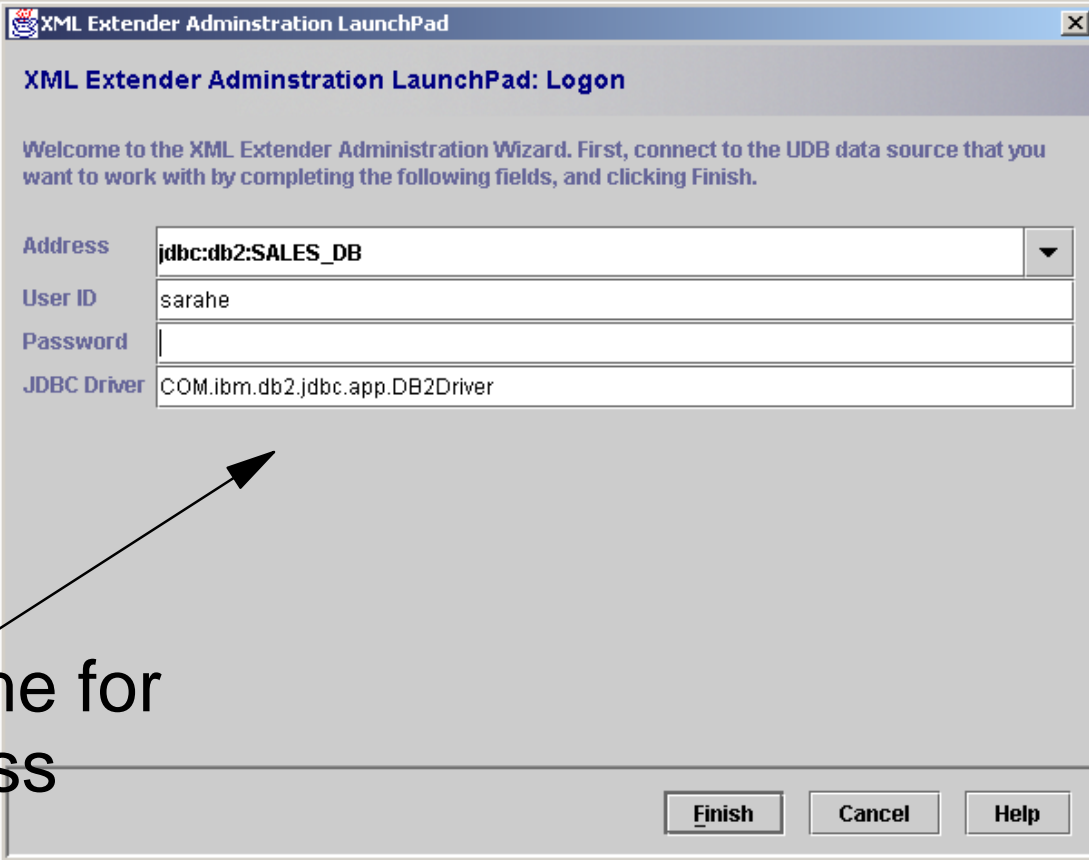
- You will need to disable the column or collection via wizard or command
  - ▷ Note that this includes situations where adding side tables
- Disable will
  - ▷ Delete entry in XML\_USAGE
  - ▷ Drop any side tables
  - ▷ Does NOT drop any data tables
- Note that if you disable the database or subsystem, the UDTs, UDFs and control tables will be dropped

# XML Extender Administration Wizard

---

- A Java based application providing a GUI that allows a user to easily administer their XML Extender enabled datasources
- Provides an easy-to-use interface to such functions as:
  - ▷ Enable or Disable a database for XML Extender
  - ▷ Work with XML Columns - Add / Enable / Disable a XML Column
  - ▷ Work with XML Collections - Enable / Disable a XML Collection
  - ▷ Import DTD's into the database
- **Note that the ability to create and edit DADs will soon be withdrawn**
  - ▷ Websphere Studio can be used to build DADs

# Launchpad - local database



The image shows a Windows-style dialog box titled "XML Extender Administration LaunchPad". Inside, the title "XML Extender Administration LaunchPad: Logon" is displayed. Below the title, a welcome message reads: "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." There are four input fields: "Address" with the value "jdbc:db2:SALES\_DB", "User ID" with the value "sarahe", "Password" which is empty, and "JDBC Driver" with the value "COM.ibm.db2.jdbc.app.DB2Driver". An arrow points from the text "Driver name for local access" to the "JDBC Driver" field. At the bottom right, there are three buttons: "Finish", "Cancel", and "Help".

XML Extender Administration 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:SALES\_DB

User ID: sarahe

Password:

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

Finish Cancel Help

Driver name for  
local access

# Launchpad - remote database

XML Extender Administration LaunchPad

**XML Extender Administration 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://cooki:6789/isc710p1

User ID: cooki

Password: \*\*\*\*\*

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

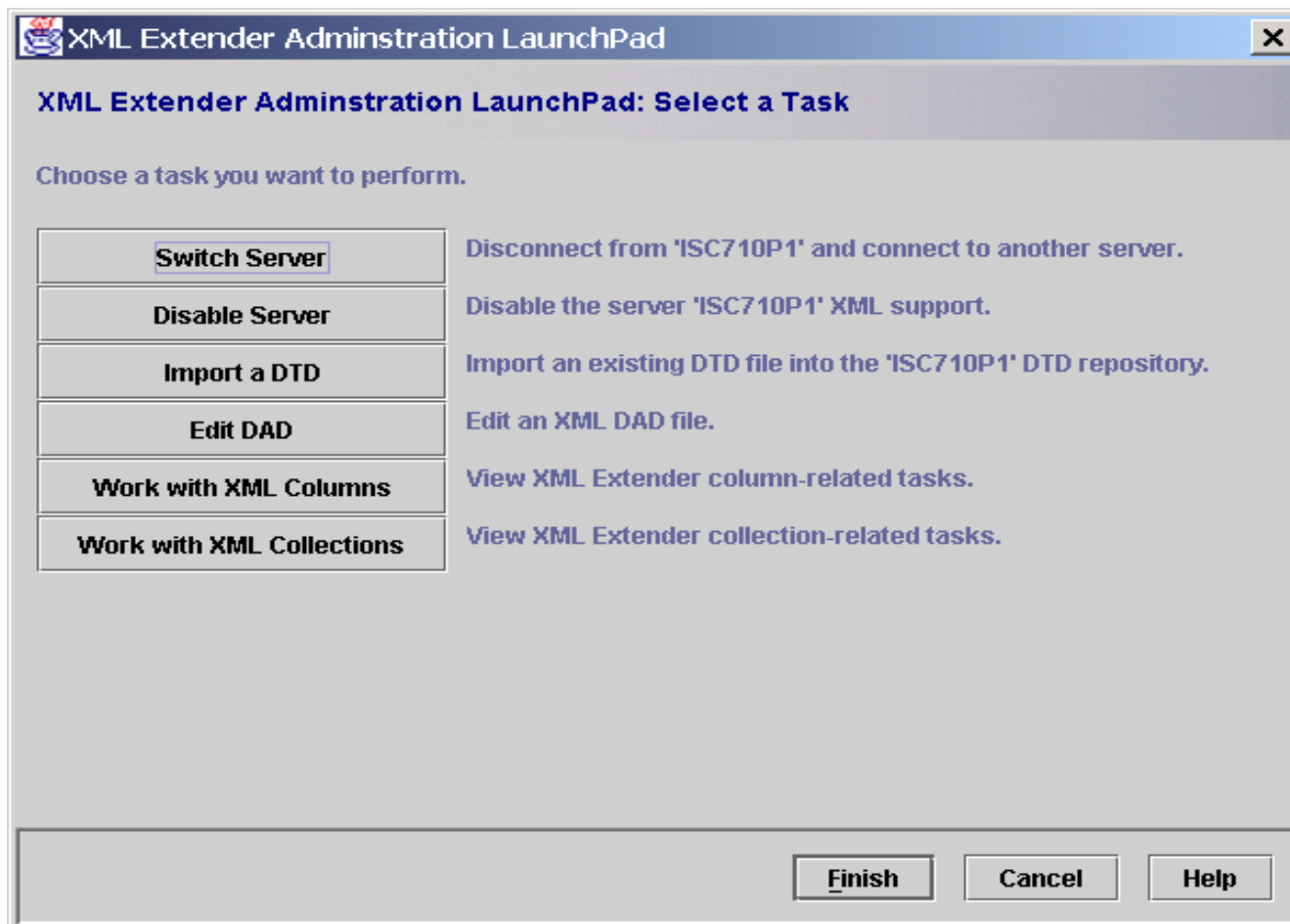
Finish Cancel Help

Driver name for  
remote access

cooki=workstation name  
6789 = workstation host tcpip port number\*  
isc710p1= database alias of target DB2  
valid userid and password for the target system

\*determined by issuing db2jstrt

# XML Extender Administration Launchpad



# Administration Wizard software and prerequisites

---

- Where can you find the software?
  - ▷ <http://www.ibm.com/software/data/db2/extenders/xmlext/downloads.html>
  - ▷ Also shipped as member DXXADMIN in the SDXXADM library on S/390
    - ⇒ ensure you download it in binary format
  
- Installation requires:
  - ▷ JAVA Development Kit (JDK) 1.1.7 or higher OR Java Runtime Environment (JRE) 1.1.x
  - ▷ XML Extender

# Setting up the Administration Wizard

---

- Ensure your workstation PATH environment variable includes Java  
eg c:\SQLLIB\JAVA\JDK\BIN
  - ▷ Note that the wizard requires a Path name without a space so if you have installed DB2 under Program Files directory copy Java code to a simpler path eg c:\JAVA

- Ensure your workstation CLASSPATH environment variable includes the following

C:\SQLLIB\java\db2java.zip;  
C:\SQLLIB\java\runtime.zip;  
C:\SQLLIB\bin;  
C:\dxx\dxxadmin\dxxadmin.jar;  
C:\dxx\dxxadmin\dxxadmin.cmd;  
C:\dxx\dxxadmin\html\dxxahelp\*.htm;  
C:\SQLLIB\java\jdk\lib\classes.zip;  
C:\SQLLIB\java\swingall.jar



## Storing and Retrieving XML data

# Extracting XML data using UDFs

---

- Alternative to using collections when you want to store and access parts of an XML document
- Extract UDFs are supplied with the DB2 XML Extender to extract particular element or attribute values from an XML document
  - ▷ eg `DB2XML.extractInteger`
- Advantages over collection approach
  - ▷ Quick and easy approach
  - ▷ No requirement for DADs and XML Extender administration
  - ▷ Useful when you just need to pull out a few pieces of data from an XML document

# Extracting UDFs

---

- Scalar extracting UDFs find an element or attribute
  - ▷ eg extractInteger, extractDate
  - ▷ `SELECT extractVarchar(pers_rec, '/personnelRec/person/email')`  
`from personnel_tab WHERE salary > 100000.00`
- Table extracting UDFs can find multi-occurrences and put them in a result set
  - ▷ eg extractIntegers, extractDates

# Collections: Composing XML documents

---

- Use the following **NEW composition stored procedures** (in fixpack 6)
  - ▷ They return the first composed document in a CLOB
  - ▷ Do not require a result table unlike their counterparts dxxGenXML and dxxRetrieveXML
- **dxxGenXMLDoc**
  - ▷ Composes XML document from tables using DAD
  - ▷ Input: DAD, override type, override, maxrows
  - ▷ Output: num rows, return code, return message
- **dxxRetrieveXMLDoc**
  - ▷ Similar but first parameter is character string identifying the DADs collection name as defined in XML\_USAGE
  - ▷ Should be used for production as DAD is stored in the database

# Composing multiple XML documents

---

- Use original stored procedures
  - ▷ dxxGenXML and dxxRetrieveXML
  
- Only difference is XML documents are placed in a **result table**
  - ▷ Table must be set up in advance and contain a single column defined as XMLCLOB or XMLVARCHAR
  - ▷ Recent versions of sample database SALES\_DB contains a RESULT\_TAB you can use
  - ▷ Table name passed as input parameter
  - ▷ Consider use of temporary tables

# Composition considerations

---

- Use override and override type parameter to compose XML documents meeting certain criteria
  - ▷ SQL\_OVERRIDE to override a condition in the DAD with an SQL statement
  - ▷ XML\_OVERRIDE to override a condition in the DAD file with an XPATH condition
- Use the GEN versions of the SPs in development
  - ▷ DAD passed as an input parameter
  - ▷ Does not require collection to be enabled
  - ▷ Avoids repeated enabling/disabling if DAD changes during testing
- Use the RETRIEVE versions in production
  - ▷ DAD must be in XML\_USAGE table

# Collections: Decomposing XML documents

---

- Two stored procedures : dxxshredxml and dxxinsertXML
- **dxxshredxml**
  - ▷ Should be used for testing as DAD is passed as a parameter
  - ▷ Input is DAD and XML object
  - ▷ Output is return code and return message
  - ▷ All tables in DAD file must exist and data types must match
- **dxxinsertxml**
  - ▷ For production - reads the DAD file from the database
  - ▷ Must enable the collection first
  - ▷ Insert DTD and DAD

# Updating collections

---

- As data is stored rather than documents, use UPDATE/DELETE statements
- Some guidelines:
  - ▷ To update an element or attribute value, write an update statement that contains the join condition specified in the DAD file and also include the required predicates
  - ▷ To update a document, do not delete a row containing the primary key of the table which is the FK of the other collection tables. When the primary key and foreign key row is deleted the document is deleted
  - ▷ To replace or delete elements and attribute values you can delete and insert rows in lower level tables without deleting the document
- Needs careful thought!



# Potential method to simulate update in decomposition

---

- Create a **temporary table** that is identical to the actual base table
  - ▷ Use same data types but define columns as nullable (except PK)
  - ▷ Note that you may not necessarily have a value for a particular column
- Create **two triggers** on the temp table, one to do an insert, one an update.
  - ▷ Only one trigger will fire depending on the when clause which determines if a row exists in the actual base table
- Issues
  - ▷ It is not possible to update a column value to a NULL value.
  - ▷ Once the decomposition stored procedure runs successfully, the rows in the temporary table need to be deleted. Application needs to know the primary key to do this so may need to add a unique key to the inserted data which it could refer to in the delete.
  - ▷ Could consider using a Global or Declared Temporary table

# Columns: Inserting XML document

---

- Use the following Supplied storage UDFs

- ▷ XMLvarcharfromfile
- ▷ XMLfilefromvarchar
- ▷ XMLclobfromfile
- ▷ XMLfilefromclob

- Example:

```
INSERT INTO SALES_TAB(ID,NAME,ORDER)
VALUES ('1234' , 'Jo Smith',
        XMLVarcharFromFile('dxx_install/samples/cmd/getstart.xml'))
```

- **Do NOT update side tables directly!!!!!!**

# Updating XML columns

---

- Method 1 : Use a default casting function
  - ▷ Varchar(XMLvarchar), CLOB(XMLCLOB), varchar(XMLFILE)
- Method 2: Use a storage UDF to update the whole XML doc
  - ▷ eg `set ORDER = XMLVarcharFromFile('filename')`
- Method 3: Use update UDF to update element or attribute
  - ▷ Update(xmlobj, path,value)
  - ▷ eg `UPDATE personnel_tab`  
`SET pers_rec =`  
`Update(pers_rec, '/personnelRec/person/email', 'bwallace@us.ibm.com')`  
`WHERE family = 'Wallace'`
  - ▷ If multi-occurrence, all values are updated

# Accessing XML columns - Options

---

- To retrieve entire document
  - ▷ Use a **default casting function** to convert UDT to DB2 base type
    - ═ eg **SELECT DB2XML.varchar(ORDER) FROM SALES\_TAB**
  - ▷ Or use the content overloaded UDF
    - ═ to retrieve document from external storage and put it in memory
    - ═ or to retrieve document from internal storage and export to an external file
- To retrieve elements or attributes
  - ▷ Use **extract UDFs**
- For structural or full text searching
  - ▷ Use **Text extender**
- To search XML columns by structure see next slide

# Searching XML columns by structure

---

- **Method 1 : Direct query on side tables (easiest!)**
- **Method 2: Searching from a joined view**
- **Method 3: Searching with extracting UDFs**
  - ▷ scalar extracting UDFs find an element or attribute
  - ▷ table extracting UDFs can find multi-occurrences and put them in a result set
  - ▷ eg `extractInteger(scalar)` `extractIntegers (table)`
- **Method 4: Searching on elements and attributes with multiple occurrence using distinct clause to prevent duplicate values**

```
SELECT SALES_PERSON FROM SALES_TAB
WHERE INVOICE_NUM IN
(SELECT DISTINCT(INVOICE_NUM FROM PART_SIDE_TAB
WHERE PRICE > 2500.00))
```

## Some practical information

Including stored procedures, LOBs

# A word on performance...

---

- The performance of the XML Extender is constantly being improved and a workgroup focusses on the issues
- Benchmark tests are currently being carried out and a white paper being produced
- There is no load utility at present - for large volumes, it is recommended you test inserting 10 to 20 documents at the design stage to check decomposition performance before production
- A future strategy is to move XML functionality into the DB2 engine

# XML Extender limits

---

- Maximum number of rows that can be inserted into a single table when decomposing is 10240 (NB documentation is wrong!)
- Can use up to 100 columns in a single table to generate an XML document
- Override string limited to 1024 until FP4 when extended to 32K
- Note that these figures are using the latest fixpack as limits are continually being lifted
- XML Admin Guide Appendix D gives SP and UDF limit information



# Large objects (LOBs)

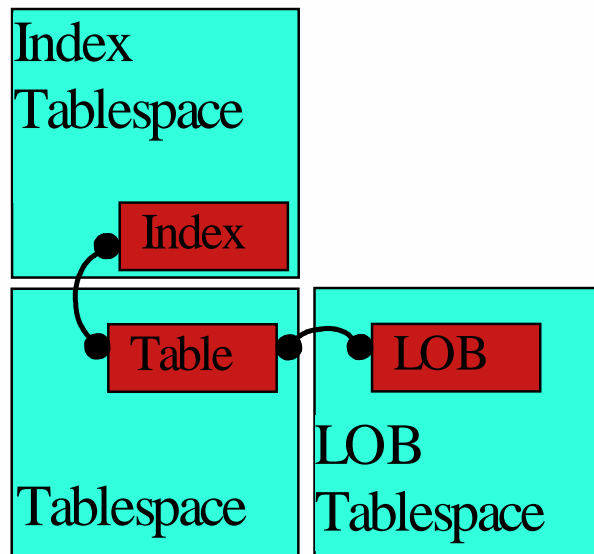
---

- If storing XML documents as columns you may need to use Large Objects
  - ▷ New area for many customers - introduced in V6
- Ability to store data objects up to 2 GB in size
  - ▷ Audio, documents, pictures
  - ▷ Can store data that will not fit into 32 KB maximum page size
  - ▷ BLOBs, CLOBs, DBCLOBs
- Programming with LOBs is challenging
  - ▷ Buffer requirements may exceed capacity
  - ▷ LOB locators introduced to process LOBs efficiently
- Data manipulation
  - ▷ Process as though (very long) strings
  - ▷ Delete is a logical delete, update is logical delete and insert

# LOB overview

## DB2 Distributed:

- Managed internally
- LOBS and their indexes can be put in their own tablespace with DMS



DB2 Data Management Software

## DB2 for OS/390:

- LOBS stored outside base table in auxiliary table
- ROWID stored in base table

### Base table space

Base table			
Key	ROWID	Column_2	LOB indicator
Key A	prt to LOB 1	user data A	LOB indicator 1
Key B	ptr to LOB 2	user data B	LOB indicator 2

Auxiliary index:  
based on ROWID  
used to navigate to LOB data

### LOB table space

Auxiliary table	
ROWID	LOB data
LOB 1 ROWID	LOB data for row user data A
LOB 2 ROWID	LOB data for row user data B

# General LOB usage notes

---

- Use locators to process them
- Expect Load utility overhead
- Some z/OS considerations
  - LOAD/Unload utility does not support LOBS greater than 32K
    - Sample program DSN8DLPL shows how to load lobs greater than 32K from a flat file using programming techniques
  - Note that Reorg LOB TS does not delete/redefine datasets
  - Put separate LOB data in its own bufferpools - set DWQT = 0
  - Look out for a LOB redbook

# Stored procedures, UDFs and UDTs

---

- They handle the XML parsing and processing
  - ▷ Makes application programming simpler and also provide better performance by performing this processing on the database server
- Set current function path to DB2XML to avoid fully qualifying them
- The DB2 XML Extender assumes application does commit or rollback and never performs these functions
- Explicitly issue a commit before issuing any SQL against a row updated by an XML UDF

# Calling the XML stored procedures

---

- Must specify host variables in the right order (including indicator variables)
- Use upper case name to enable calling programs to be run on many platforms
- You can increase the 1MB limit for a CLOB parameter being passed to a SP
  - ▷ By dropping the stored procedure and recreating it with an increased CLOB limit
- Do not use the exclamation point (!) when calling stored procedures from JDBC or CLI eg db2xml.dxxshredxml rather than db2xml!dxxshredXML
- Use OVERRIDETYPE and OVERRIDE parameters to overwrite values in DAD

## Code page considerations

This is an important area to avoid problems!

# Encodings on UNIX and Windows

---

- XML document encoding
  - ▷ The code page of an XML document
- Document encoding declaration
  - ▷ The name of the code page specified in the document declaration
- Consistent document
  - ▷ The document encoding and the document encoding declaration match
- DB2CODEPAGE registry (environment) variable
  - ▷ DB2 gets the client code page from the client operating system locale unless this variable is set

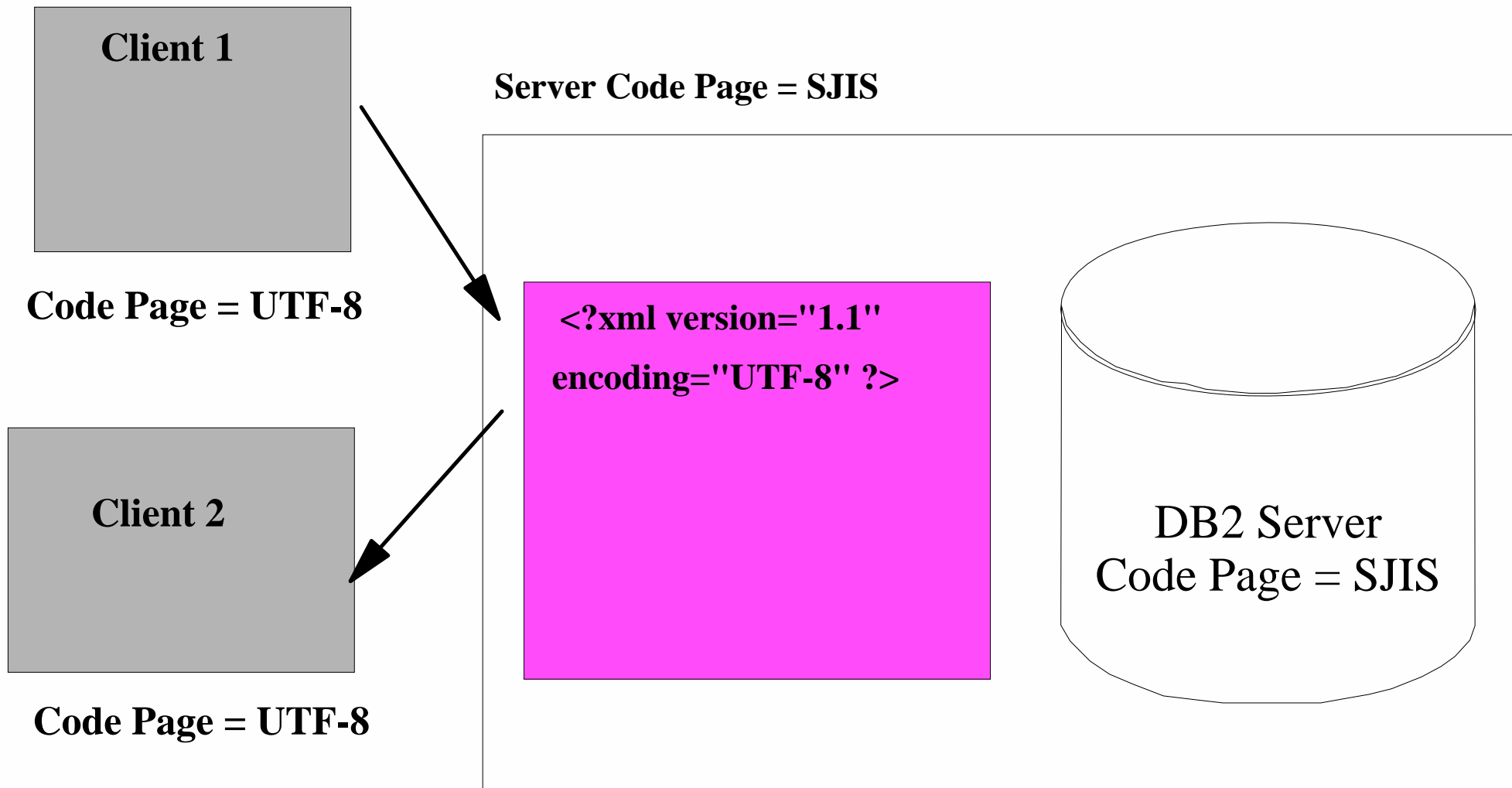
# Encodings on UNIX and Windows

---

- Client code page
  - ▷ The application code page. If the DB2CODEPAGE variable is set, the client code page is the value of DB2CODEPAGE. Otherwise, the client code page is the client's operating system locale.
- Server code page, or server operating system locale code page
  - ▷ The operating system locale on which the DB2 database is installed.
- Database code page
  - ▷ The encoding of the stored data, determined at database create time. If not explicitly specified with the USING CODESET clause, this value defaults to the server operating system locale.

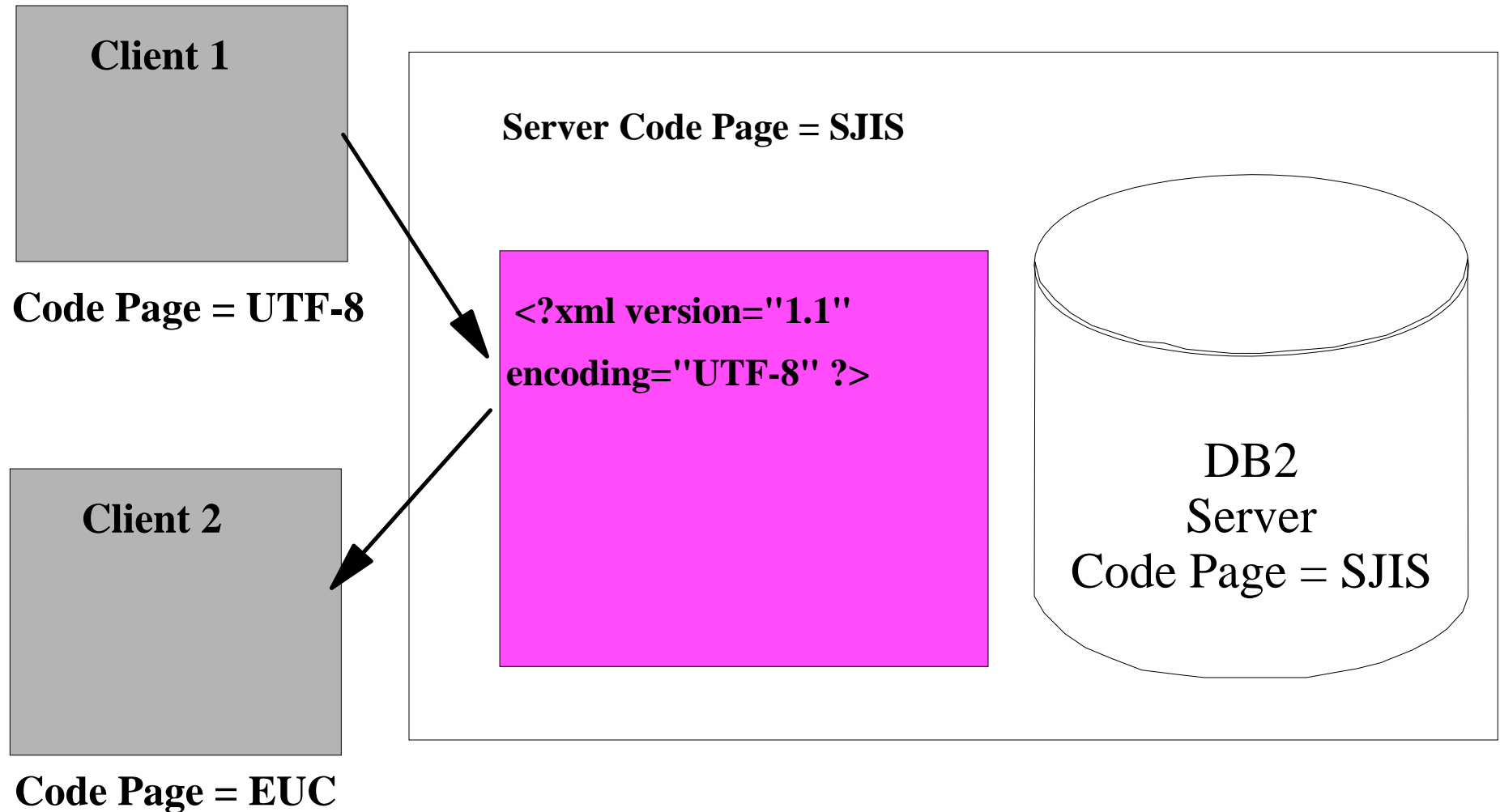


# Retrieving a Consistent Document



**Consistent document**

# Retrieving an Inconsistent Document



**Inconsistent document**

# Some zSeries Considerations

---

- The DB2 XML extender on zSeries can be used from:
  - ▷ Local or remote Java applications using JDBC or SQLJ
  - ▷ Local or remote C or Cobol applications using ODBC or SQL
  - ▷ Batch, IMS, CICS, servlet applications
  - ▷ DB2 stored procedures in Java, C, Cobol, SQL Procedures (PSM), or Rexx
- The DB2 XML extender requires the XML Toolkit. See:
  - ▷ <http://www.ibm.com/servers/eserver/zSeries/software/xml/>
  - ▷ The toolkit required can change in an XML Extender PTF
- EBCDIC encoding declaration example
  - ▷ `<?xml version="1.0" encoding="ibm037">`
- For information on OS390 line endings and encoding declarations see:
  - ▷ <http://www.ibm.com/servers/eserver/zSeries/software/xml/usage/>

# Line Ending Summary for zSeries

XML Creation Method	Line Ending
FTP from ASCII environment	[NEL]
JDBC/ODBC through DRDA from PC	[CR][LF]
JDBC/ODBC through DRDA from Unix	[LF]
JDBC/ODBC local or remote OS390	[NEL]
vi editor on USS	[NEL]
OS390 C program output \n	[NEL]
PC/Unix C program output \n produces ASCII	[LF]
OS390 C iconv conversion from ASCII to EBCDIC	[NEL]

# zSeries: EBCDIC and USS

---

- EBCDIC code pages have variant characters in areas that affect XML
  - ▷ In general cannot mix and match different EBCDIC code pages when using XML
- You can consume and generate XML in EBCDIC – just be aware of
  - ▷ XML encoding declarations and their impact
  - ▷ Line endings on zSeries
  - ▷ EBCDIC variant characters
- “Getting Started” XML Extender samples assume USS
  - ▷ DB2 XML applications are not themselves required to use USS
  - ▷ USS needed primarily for XML Extender internal use, in particular, the XML Extender uses the USS file system
  - ▷ The default code page for USS is ibm1047
- Ensure you have the right code page on the bind of DXXADM

# zSeries: EBCDIC and USS

---

- EBCDIC code pages have variant characters in areas that affect XML
  - ▷ In general cannot mix and match different EBCDIC code pages when using XML
- You can consume and generate XML in EBCDIC – just be aware of
  - ▷ XML encoding declarations and their impact
  - ▷ Line endings on zSeries
  - ▷ EBCDIC variant characters
- “Getting Started” XML Extender samples assume USS
  - ▷ DB2 XML applications are not themselves required to use USS
  - ▷ USS needed primarily for XML Extender internal use, in particular, the XML Extender uses the USS file system
  - ▷ The default code page for USS is ibm1047

# Encoding reference information

---

- EBCDIC Variant Characters

- ⇒ [http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/FL2CPE00/6.6.1](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2CPE00/6.6.1)
- ⇒ [ibm1047](#) is USS code page
- ⇒ <http://w3.enterlib.ibm.com:80/cgi-bin/bookmgr/books/cbcopg03/8.7.1.1>

- For information on OS390 line endings and encoding declarations see:

- ⇒ <http://www.ibm.com/servers/eserver/zseries/software/xml/usage/>
- ⇒ <http://www.w3.org/TR/newline>

- Can process ASCII documents

# XML Repository Futures

Bruce Benfield

IBM Software Group

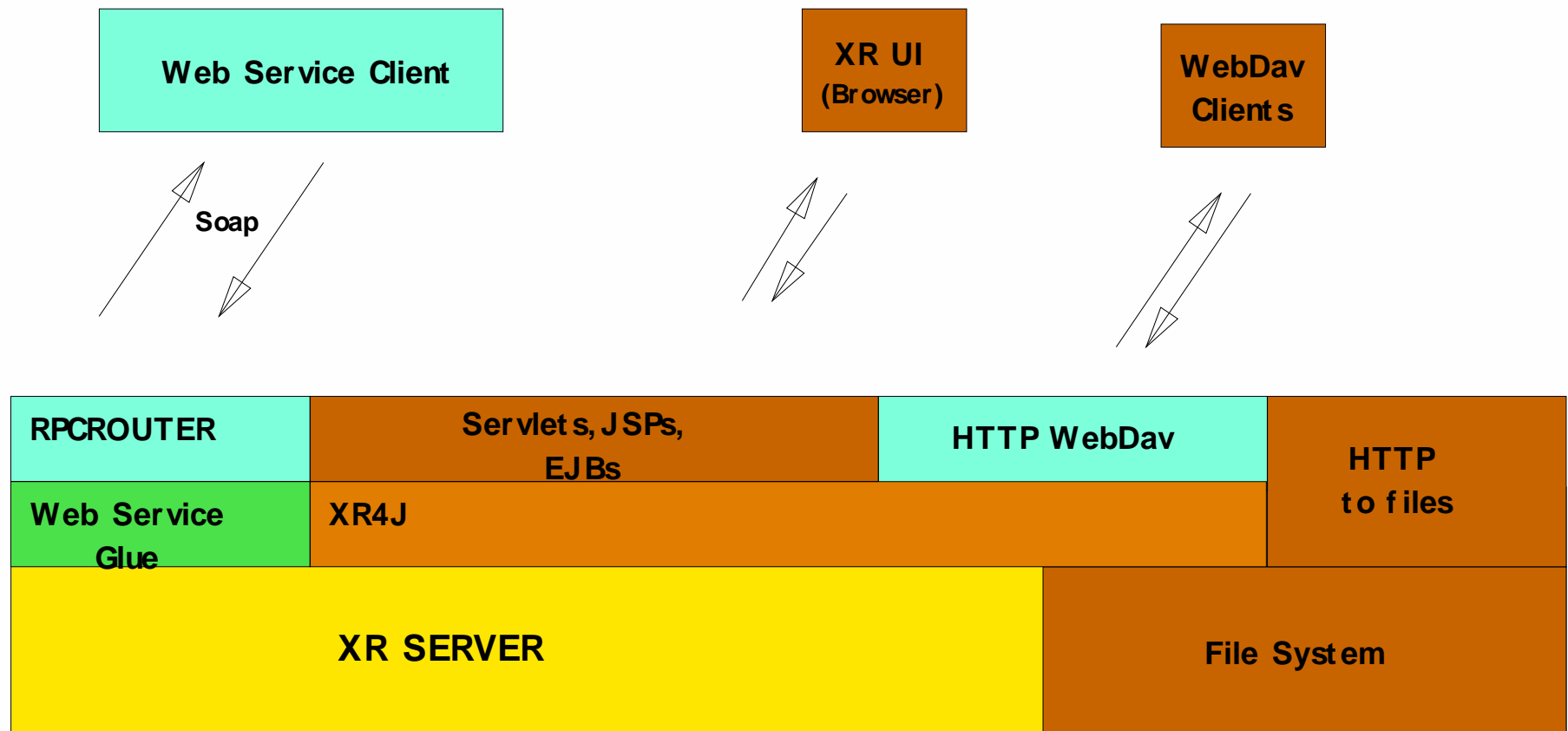


# XR Overview

---

- XML Repository for managing all types of XML artifacts
  - ▷ WSDL, XSD, DTD, XSL, CSS, Message formats...
  - ▷ URL Addressable
- Meta Data ( owner, refers to , is grouped with )
- XML content search using Xpath
- Browser Based & WebSphere Studio UIs
- Web Service, Java, and WebDav APIs
- Basic Versioning
- Audit trail, security
- UDDI business data integration

# XR Architecture



## Lab 2 : Exercise Overview

# Tasks - What you will do

---

- Use collection DADs produced in Lab 1 to compose XML documents
- Insert the DTD Myorder.DTD into DTD Reference table
- Run composition stored procedures via DB2 Stored Procedure Builder
  - ▷ Avoids need to write programs to call stored procedures
  - ▷ Define SPB project **Myproject**
- Compose document
  - ▷ Use stored procedure dxxGenXMLCLOB()
  - ▷ DADBUF variable input will be the entire DAD **Myorder.dad**
  - ▷ Note : Fixpack 5 users will use dxxGenXML()
- Enable a Collection using the XML Extender Administration Wizard
- Use dxxRetrieveXMLCLOB() to compose the XML document

# Storing an XML document in an XML Column scenario

---

## □ Your task

- ▷ Archive sales data for the service department into **SALES\_TAB**
  - Two columns hold data about each sale, the third column **ORDER** contains the XML document

## □ Service department:

- ▷ Will use this information when working with customer requests and complaints
- ▷ Specified which element data will be queried most frequently
- ▷ Want to be able to search the documents quickly

## □ What will you do?

- ▷ Create the SALES\_TAB and enable the ORDER column
- ▷ Insert a DTD for the XML document validation
- ▷ Store the document as an XMLVARCHAR data type
- ▷ Define side tables to be indexed for searching the document in the DAD

## DB2 Table used for XML Column

---

### SALES\_TAB

Column Name	Data type
INVOICE_NUM	CHAR(6) NOT NULL PRIMARY KEY
SALES_PERSON	VARCHAR(20)
<b><i>ORDER</i></b>	<b><i>XMLVARCHAR</i></b>

ORDER Column to be enabled for XML

## DB2 Side Tables created during XML Column exercise

### ORDER\_SIDE\_TAB

Column Name	Data type	Location Path	Multiple occur?
ORDER_KEY	INTEGER	/Order/@key	No
CUSTOMER	VARCHAR(16)	/Order/Customer/Name	No

### PART\_SIDE\_TAB

Column Name	Data type	Location Path	Multiple occur?
PRICE	DECIMAL(10,2)	/Order/Part/Extended/Price	Yes

### SHIP\_SIDE\_TAB

Column Name	Data type	Location Path	Multiple occur?
DATE	DATE	/Order/Part/Shipment/Shipdate	Yes

# Optional column enablement exercise

---

- Create SALES\_TAB with XML Column ORDER (XMLVARCHAR)
- DTD [Myorder.dtd](#) as used in previous scenarios
- DAD [Myorder\\_xcolumn.dad](#) as supplied
- Invoke Wizard to Enable the ORDER Column
  - ▷ Insert a Primary Key [INVOICE\\_NUM](#) used as the root id for the side tables
  - ▷ Provide a default view [SALES\\_ORDER\\_VIEW](#) that joins the XML column and side tables
- Optionally create indexes on side tables
- Insert the XML document [getstart.xml](#) into the ORDER column
  - ▷ C:\dxx\samples\xml
- Query the SALES\_TAB table



For those of you who will leave us...

**Thanks!** We hope you have found these sessions useful

Don't forget to complete your feedback form

# OS/390 Considerations

When using the DB2 XML Extender

# Agenda

---

- Software requirements to run the DB2 XML Extender on OS390 and z/OS
  - ▷ UNIX Systems Services, WLM, RRS
  - ▷ XML Toolkit and XML Extender Administration Wizard
- Security Considerations
- XML Implementation & Setup
- XML Administration options
- Using USS to administer XML Extender
- Codepage and encoding considerations
- Reference Materials

# Software requirements

---

- Unix Systems Services (USS) and Hierarchical File System (HFS)
- Workload Manager (WLM)
- Recoverable Resource Services (RRS)
- XML Toolkit for OS/390 and z/OS V1R2 and V1R3
- odb2 command (optional)
- XML Extender Administration Wizard
  - ▷ DB2 Connect Personal or Enterprise Edition
  - ▷ JDK 1.1.7 or higher

# Introduction to OS/390 UNIX (USS)

---

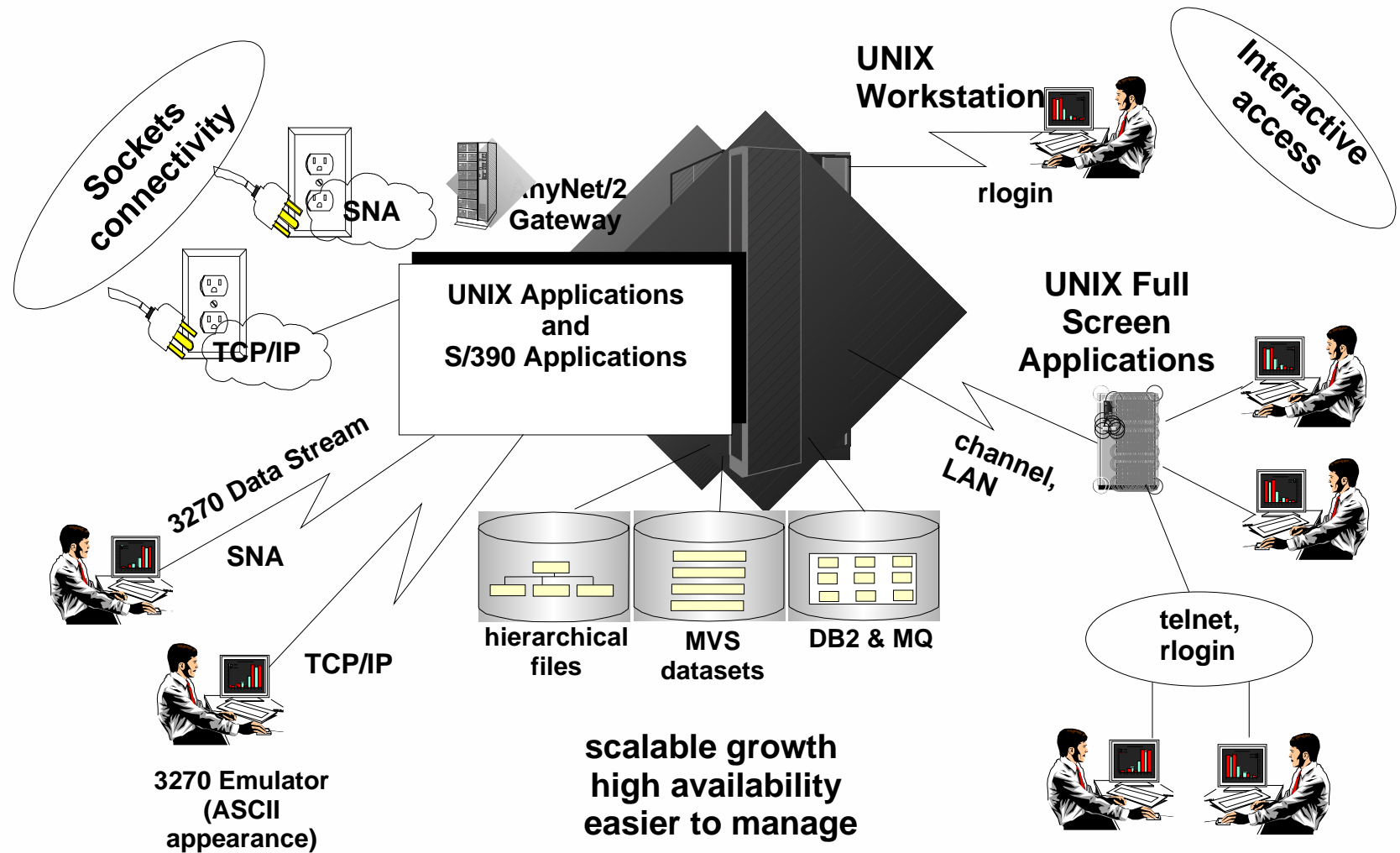
- OpenEdition MVS - Early Version Available for MVS/ESA v4.3
- Evolved to full SPEC 1170 (Universal Unix) with MVS/ESA 5.2
- XPG4 UNIX 1995 conforming operating system
- Fully Integrated into Base OS/390 Code
  - ▷ OS/390 and z/OS can be considered as a UNIX OS
- Software Servicing techniques such as ServicePac & SFS automatically ship fixes for USS
  - ▷ OE Code and Products that use USS often exist within HFS
- USS must be Active to Access HFS

# Introduction to OS/390 UNIX (USS)

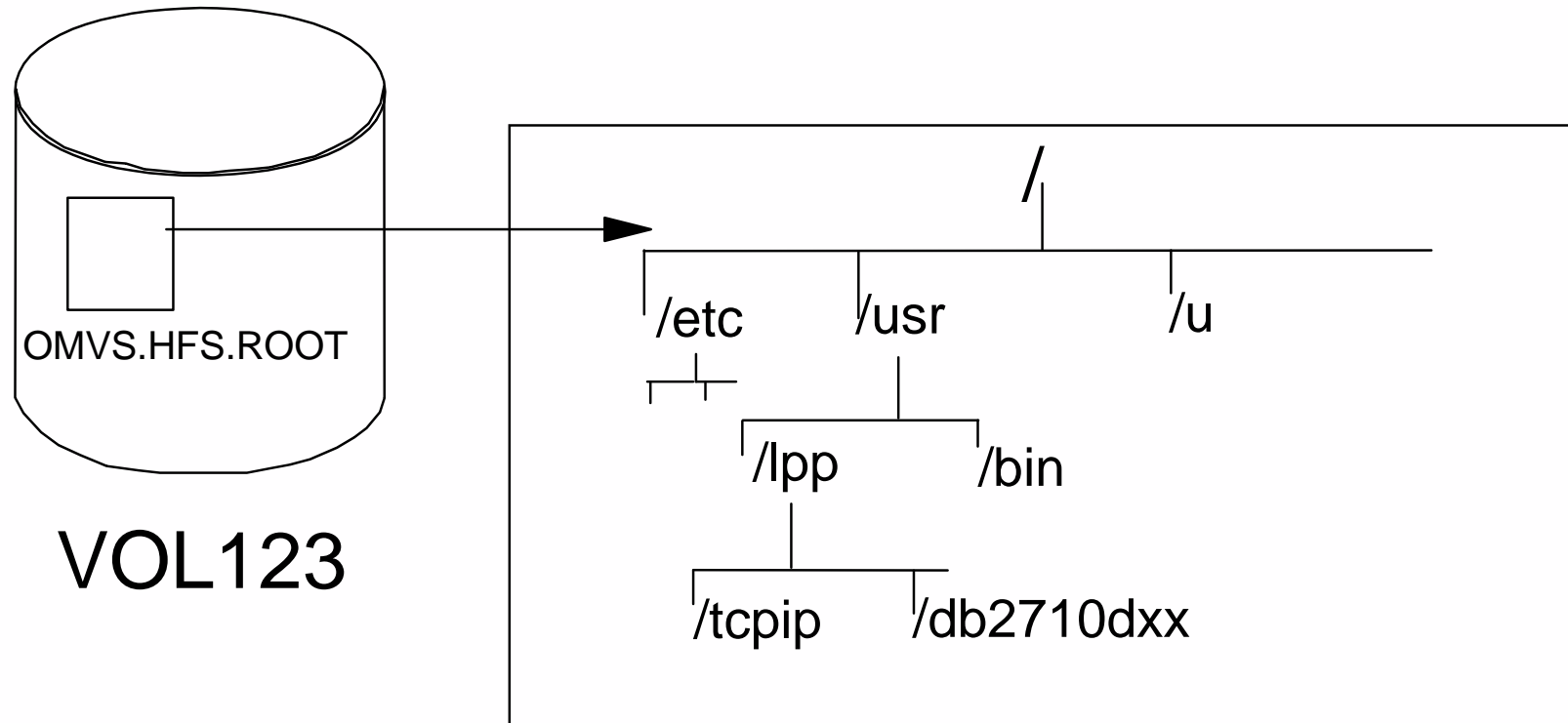
---

- TSO Shell (OMVS) & TSO Commands
  - ▷ Full Screen UNIX 'Session' Running Under TSO
  - ▷ Invoked with TSO Command OMVS
  - ▷ Provides function of a native UNIX Environment
- ISPF Shell (ISHELL)
  - ▷ Dialog based ISPF 'Style' Interface to USS
  - ▷ Invoked via TSO ISH Command
  - ▷ Mount HFS, Edit / Browse Copy HFS Members
- Other
  - ▷ Programming API's
  - ▷ dbx debugger

# USS - Typical Configuration



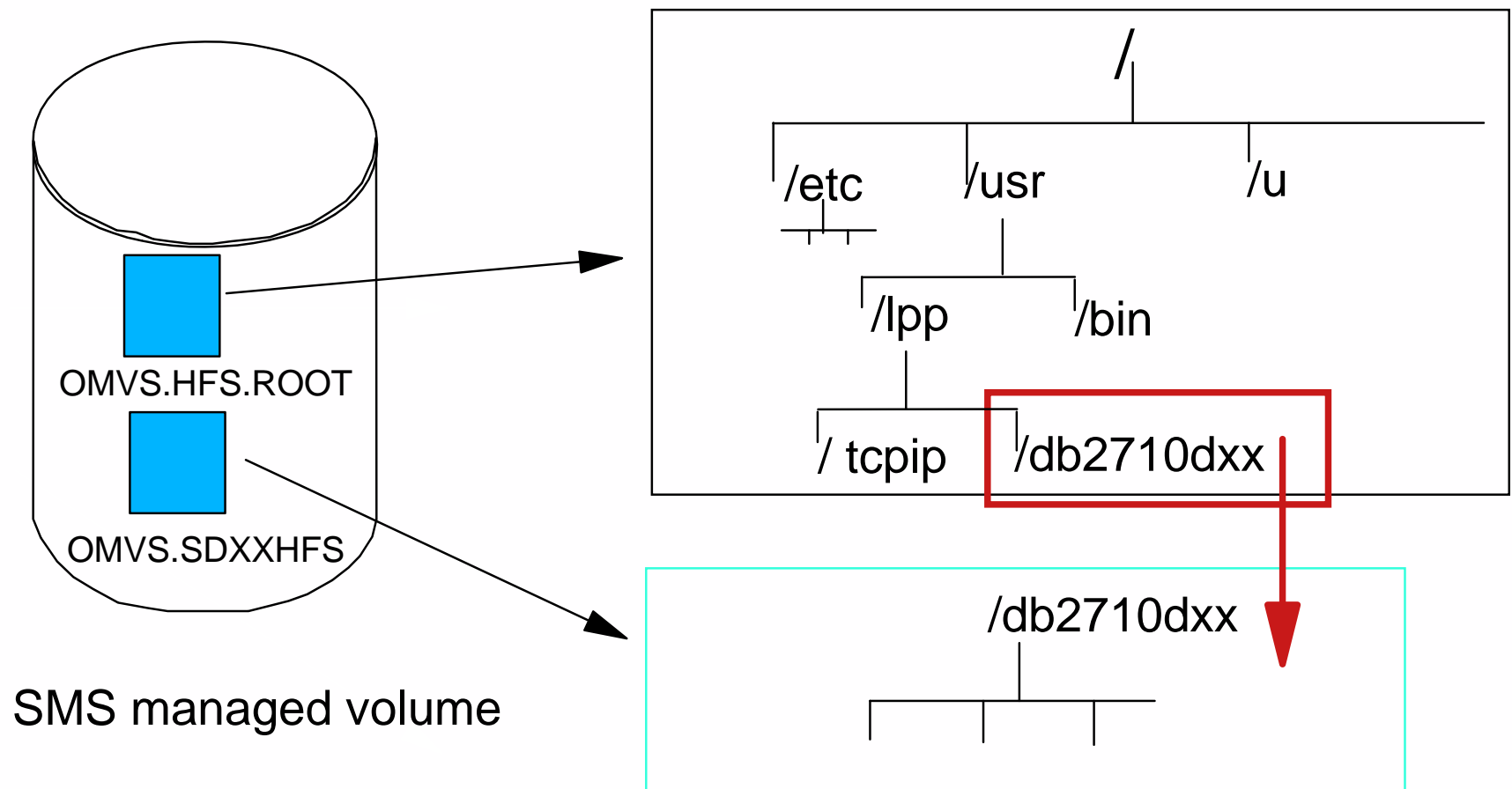
# Hierarchical File System - Root



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



# HFS - Mountpoints and XML /db2710dxx



SMS managed volume

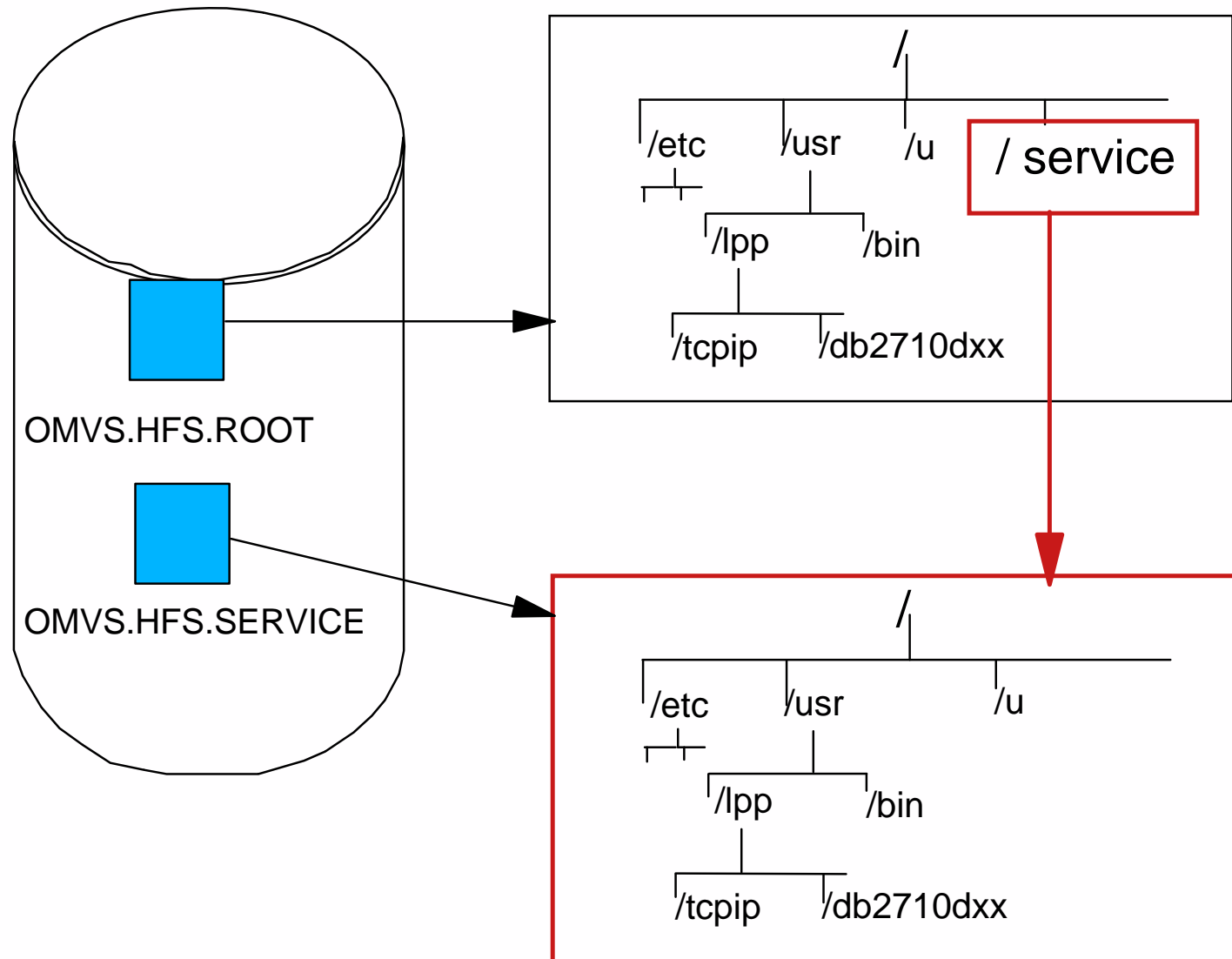
TSO MOUNT FILESYSTEM('OMVS.SDXXHFS')  
MOUNTPOINT('/usr/lpp/db2710dxx') TYPE(HFS) MODE(RDWR)

# OS390 UNIX Security

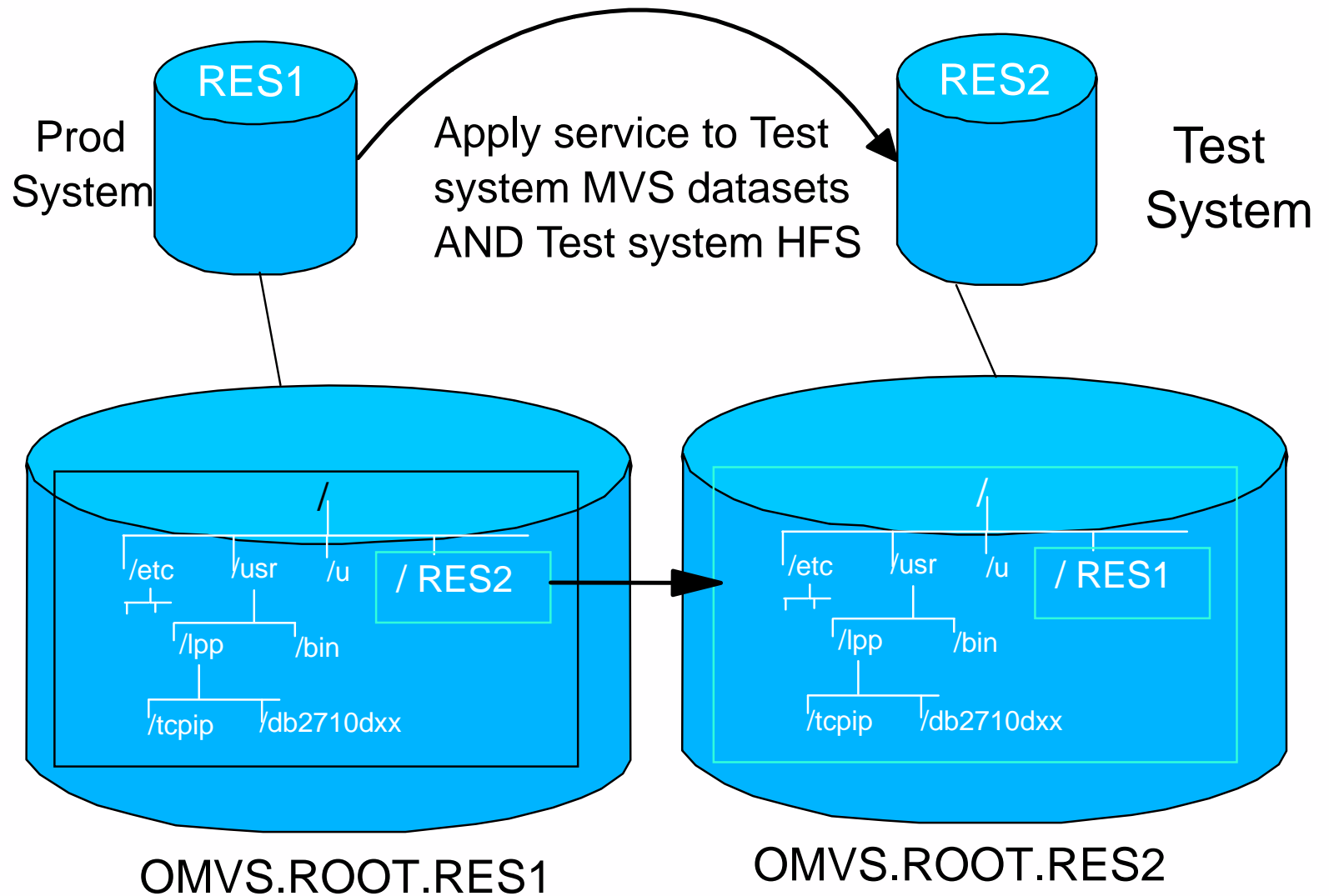
---

- OMVS segment.
  - ▷ Required to access UNIX services.
  - ▷ Consists of UID, GID, PROGRAM and HOME
- UID
  - ▷ UID or User ID is UNIX equivalent to a USER in RACF
    - ⇒ Value from 0 to 2147483647 !
    - ⇒ 0 is special - Superuser, has access to all UNIX resources
- GID
  - ▷ GID or Group ID is UNIX equivalent to a GROUP in RACF
    - ⇒ Value from 0 to 2147483647 !!
    - ⇒ No special values
- PROGRAM
  - ▷ Defines the shell program used for a given user
  - ▷ PROGRAM('/bin/sh')
- HOME
  - ▷ Defines the home directory for a given user
  - ▷ HOME('/u/armiges')

# OS/390 UNIX Maintenance



# OS/390 UNIX Maintenance - Full Scale Update



# Shared HFS Support

---

- It is now possible to have R/W access to a HFS from different systems at the same time, which enables:
  - ▷ HFS use by Web Server on multiple systems while retaining ability to make updates
  - ▷ System Programmers can maintain the HFS for any system without having to re-login to the related system to change each HFS
- XCF group **SYSBPX** provides communication
  - ▷ Sharing capability is similar to NFS - that is, ALL access (both write AND read) to the HFS must be done by the owning system
  - ▷ Owning system should be the most intensive user
  - ▷ Ownership can be moved - SETXCF command
- More information, including some performance numbers is available on the Web at: <http://www.s390.ibm.com/unix/release/shared.html>

# XML USS requirements

---

- Allocate a filesystem dataset e.g. OMVS.MVS20.DXX.V710.SDXXHFS
- Add the following statements to the active BPXPRMxx member in SYS1.PARMLIB:  

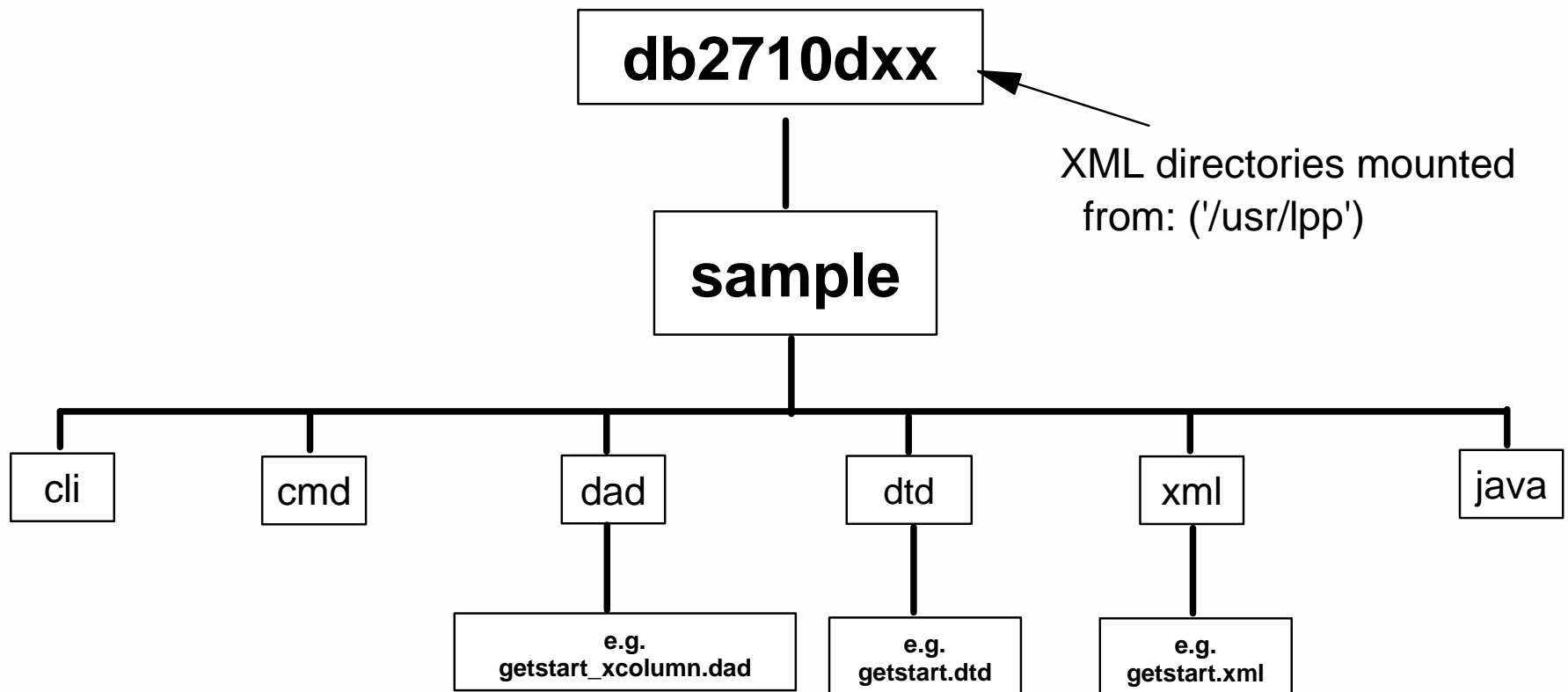
```
MOUNT FILESYSTEM('OMVS.MVS20.DXX.V710.SDXXHFS')  
MOUNTPOINT('/usr/lpp/db2710dxx')  
TYPE(HFS) MODE(RDWR)
```
- Ensure your Security Administrator provides you with the necessary authority to create further directories at the mountpoint created for XML Extender
- After the USS environment is initialized, you can perform administration tasks using either:
  - USS command line and odb2 command line
  - MVS batch and TSO batch

# Populating the XML HFS

---

- The DXXGPREP JCL performs the following tasks, including the population of the HFS
  - ▷ Frees / Binds DB2 Plans / Packages
  - ▷ Drops / Creates DB2 XML Procedures / LOB's
  - ▷ Grants authorizations
  - ▷ Copies XML scripts to USS
  - ▷ Enables XML server

# XML HFS Contents





# Useful USS References

---

- OS/390 UNIX System Services Command Reference - SC28-1892
  - ▷ This book describes USS commands
- OS/390 UNIX System Services Programming: Assembler Callable Services Reference - SC28-1899
  - ▷ This book describes the USS Assembler Callable Services
- OS/390 UNIX System Services Planning - SC28-1890
  - ▷ This book describes planning for USS
- OS/390 UNIX System Services User's Guide SC28-1891
  - ▷ This book provides tasks for using USS
- New Redbook on HFS on OS/390, including HFS Sharing:
  - ▷ HFS Usage Guide, SG24-5482-01

# Resource Recovery Services (RRS)

---

- RRS required for XML Extender stored procedures and UDF's that run in WLM managed address spaces:
  - ▷ Good information on setting up WLM, RRS and SPs in Redbook DB2 for OS/390 V6 Management Tools Package (SG24-5759)
- Provides services that implement two-phase commit protocol on OS/390 platform
- Protects resources by invoking these services and providing exit routines
- OS/390 MVS Programming: Resource Recovery (GC28-1739) is a good place to start for:
  - ▷ An overview of how to plan a resource manager
  - ▷ How to use RRS services and how to code the exit routines
  - ▷ Details about each resource recovery service
  - ▷ Details about each exit routine

# XML toolkit for z/OS and OS/390 - What is it?

---

- Under the covers feature providing:
  - ▷ A common data exchange format for both data and metadata
  - ▷ Capability for different applications and databases to exchange information
  - ▷ Publish data over the Internet while maintaining privacy of the sending system.
  
- 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, and benefit from the input of a large community of developers.
  - ▷ Tested and packaged for deployment on z/OS or OS/390 and include the enterprise servers' world class level of service.

# XML toolkit for z/OS and OS/390 -Where can I get it?

---

- How can I get the XML Toolkit for z/OS and OS/390:
  - ▷ Standard Product tape distribution
  - ▷ For latest product material and a download version, visit the XML Toolkit for z/OS and OS/390 Web site at:

<http://www.ibm.com/servers/eserver/zseries/software/xml/>

- ◊ SMP/E installable RELFILE format
  - ◊ Packaged using the SMP/E GIMZIP function
  - ◊ The SMP/E GIMUNZIP function is required to process the downloaded package
- Current PGMDIR support two levels:
  - ◊ Latest level V1R3 (FMID HXML130)
  - ◊ Previous level V1R2 (FMID HXML120)
  - ◊ Obtain the appropriate information based on install vehicle e.g. CBPDO

# Workload Management (WLM)

- Needs to be set-up if not already on your system
  - ▷ Good information on setting up WLM, RRS and SPs in Redbook DB2 for OS/390 V6 Management Tools Package (SG24-5759)
- Run WLM in goal mode so that new stored procedures address spaces are started automatically when required
- Create a WLM application environment for execution of UDF's and Stored Procedures. e.g. DXXENABLEDB.
  - ▷ Associated with a JCL procedure that starts the address space for their execution. e.g.

```
//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
//   DD DISP=SHR,DSN=SYS2.DB2.V710.SDXXLOAD
//   DD DISP=SHR,DSN=SYS1.SIXMMOD1 <===== N.B. XML Toolkit load library required.
//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)
```

# WLM Environments

---

- Multiple WLM environments can be configured for running XML UDF's.
- Specifying two WLM environments can improve performance:
  - ▷ All Stored Procedures will run in the first WLM environment
  - ▷ All UDF's will run in the second WLM environment
- Performance benefits of running in WLM "GOAL" mode
  - ▷ WLM will automatically start WLM established address spaces for UDF's 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 UDF's MUST be monitored to determine how many WLM address spaces are required
  - ▷ They must be manually started and stopped by an operator

# Security considerations

---

- Consider the implications the XML Extender has on security in the following areas:
- Accessing XML Columns in tables
  - ▷ Same security protection as traditional numeric and character data
  - ▷ User must have the required privileges to:
    - SELECT objects from
    - INSERT objects into
    - UPDATE objects in
    - DELETE objects
- Access to the contents of files including external entities or DTD's:
  - ▷ Partitioned Datasets (PDS)
  - ▷ File system compatible with OS/390 and z/OS UNIX Systems Services e.g. Hierarchical File System (HFS)

# Security considerations

---

- ▷ SECURITY option indicates how UDF's interact with an external security product e.g. RACF to control access to files:
  - SECURITY USER
    - XML UDF execution environment is assigned the Primary Authid of the process that called the function. i.e. ID used for non-SQL requests, not DB2 auths. such as package or plan owner
  - SECURITY DB2
    - XML UDF's access files using the authid associated with the WLM environment address spaces established for running the UDF's
- ▷ When UDF accesses a file, USS calls the External Security product to get the user ID (UID) and group ID (GID) associated with the UDF:
  - For SECURITY USER
    - UID and GID are those assigned to the auth ID for the process that calls the UDF
  - SECURITY DB2
    - UID and GID are those assigned to the WLM application environment address space for the UDF



# Security considerations

---

## □ EXECUTE authority

- ▷ When server is enabled the following privileges are granted to PUBLIC:
  - ⇒ Use privileges on the XML Extender's UDT's
  - ⇒ Use privileges on all UDF's and Stored Procedures
- ▷ Can be revoked and granted to specific authorisation ID's
  - ⇒ Does NOT affect the way the Extender operates
  - ⇒ Maintaining authority lists can be tedious

## □ Authority to administer the XML Extender

- ▷ Enable / Disable Server requires SYSADM authority to user ID DB2XML
- ▷ Enable / Disable Column, administrator must have table owner privileges on the table containing the column to be enabled and privileges for Bools and Table spaces
- ▷ Enable / Disable Collection, administrator must have table owner privileges on ALL tables in the collection and privileges for Bools and Table spaces

# odb2 command line tool

---

- Permits SQL from the OMVS Unix Shell.
- Will download from the IBM OS/390 USS Tools and Toys website:

<http://w3.s390.ibm.com/products/oe>

- ▷ Select Tools and Toys ==> Select OS/390 UNIX Tools
  - ▷ Ensure you download and print the README
- Mandatory requirement if running the sample applications from the OMVS command line
- Frequent users of the OMVS shell will find odb2 a very useful tool
  - ▷ Access DB2 data from the command line without exiting the shell

# DB2 XML Extender

---

- The DB2 XML extender on zSeries can be used from:
  - ▷ Local or remote Java applications using JDBC or SQLJ
  - ▷ Local or remote C or Cobol applications using ODBC or SQL
  - ▷ Batch, IMS, CICS, servlet applications
  - ▷ DB2 stored procedures in Java, C, Cobol, SQL Procedures (PSM), or Rexx
- The XML Extender code runs in WLM managed stored procedure address spaces and WLM
- Available with DB2 for OS/390 and z/OS Version 7
  - ▷ No charge feature - FMID JDB771X
  - ▷ Component id - 5740XYR06

# XML Extender implementation - DXXGPREP JCL

---

- Free / Bind DB2 Plans / Packages
  - ▷ Ensure correct code page (ENCODING) assigned to DXXADM Bind Package stmt.
  - ▷ BIND's using DB2XML owner id
  - ▷ Plan DXXADM for performing admin. through DXXADM. Uses collection DB2XML and includes the ODBC packages
  - ▷ Plans INSERT, RETRIEVE, SHRED, TESTS2X for Getting Started application
- DROPs / CREATEs DB2 XML stored procedure DB2XML.DXXENABLEDB
  - ▷ Used to Enable the Server
- Creates LOB tablespaces (Preferable to predefine)
  - ▷ CLOB data in the DTD\_REF table
- Grants Authorisations
  - ▷ To DB2XML
    - ⇒ DBADM rights on schema DB2XML and rights on the DB2 Catalog
    - ⇒ BINDADD and PACKADM to the XML collections DB2XML and DB2XML\_RUN
  - ▷ To PUBLIC, EXECUTE and BIND on collection DB2XML\_RUN

# DXXGPREP setup JCL

---

- Copies XML entities to the XML Extender /sample USS directory as the following sub-directories:
  - ▷ /cmd
  - ▷ /xml
  - ▷ /dtd
  - ▷ /dad
  - ▷ /cli
  - ▷ /java
- Enables XML server:
  - ▷ `ENABLE_SERVER -a ISC3 using XMLLOBTS,XMLLOBT2 wlm environment ISC3WLM`

# Set-up Hints and Tips

---

- We recommend you create a RACF group called DB2XML and connect users who will issue enable and disable commands to the group
- Ensure that APAR PQ47666 is applied so that tables are created in DB2XML database rather than the DSNDB04
- Predefine LOB tablespaces XMLLOBTS and XMLLOBT2

- OPUT copy command characteristics:

```
//COOKIDAD JOB CLASS=A,NOTIFY=COOKI,  
//      MSGCLASS=H,MSGLEVEL=(1,1),REGION=0K,TIME=NOLIMIT  
//GETOUT EXEC PGM=IKJEFT01  
//SYSPRINT DD SYSOUT=*  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *  
  OPUT 'COOKI.DB2.V710.ADXXDAD(DXXGCUDA)' -  
  '/usr/lpp/db2710dxx/dxx/sample/dad/getstart_xcolumn.dad'  
/*
```

# Set-up Process

---

- Run sample jobs either as batch jobs or commands from OMVS shell
  - ▷ May need to define a shell script to get round JCL limit of xx characters
  - ▷ Steps described in e-business redbook
  - ▷ Creates administration support tables DTD-REF and XML\_USAGE
  - ▷ Creates stored procedure DB2XML.DXXENABLEDB, UDFs and UDTs

# XML Administration options

---

- Tasks can be performed in the following ways
  - ▷ XML Extender Administration WIZARD, work station based tool
  - ▷ The DXXADM command issued from MVS Batch jobs.
  - ▷ The DXXADM command issued from the USS using the odb2 command line tool
    - ⇒ odb2 command line tool can be downloaded from the USS tools and Toys website:  
<http://www.s390.ibm.com/products/oe>
  - ▷ Issue SQL statements:
    - ⇒ Executed using the odb2 from the OMVS command line
    - ⇒ From batch jobs, based on samples found in the SDXXJCL data set.
  - ▷ Custom built applications using the XML Extender Administration Stored Procedures



# Overcoming the JCL EXEC parameter limit

- The JCL EXEC parameter length limit of 100 characters can be easily exceeded because USS HFS file names and paths tend to be very long

- ▷ For example Enabling a column using DXXGECOL JCL:

```
//STEP2 EXEC PGM=DXXADM,  
//          PARM='/enable_column -a V71A SALES_TAB ORDER /dxx/samples/dad/    X  
  
//          getstart_xcolumn.dad V SALES_ORDER_VIEW -R INVOICE_NUM'
```

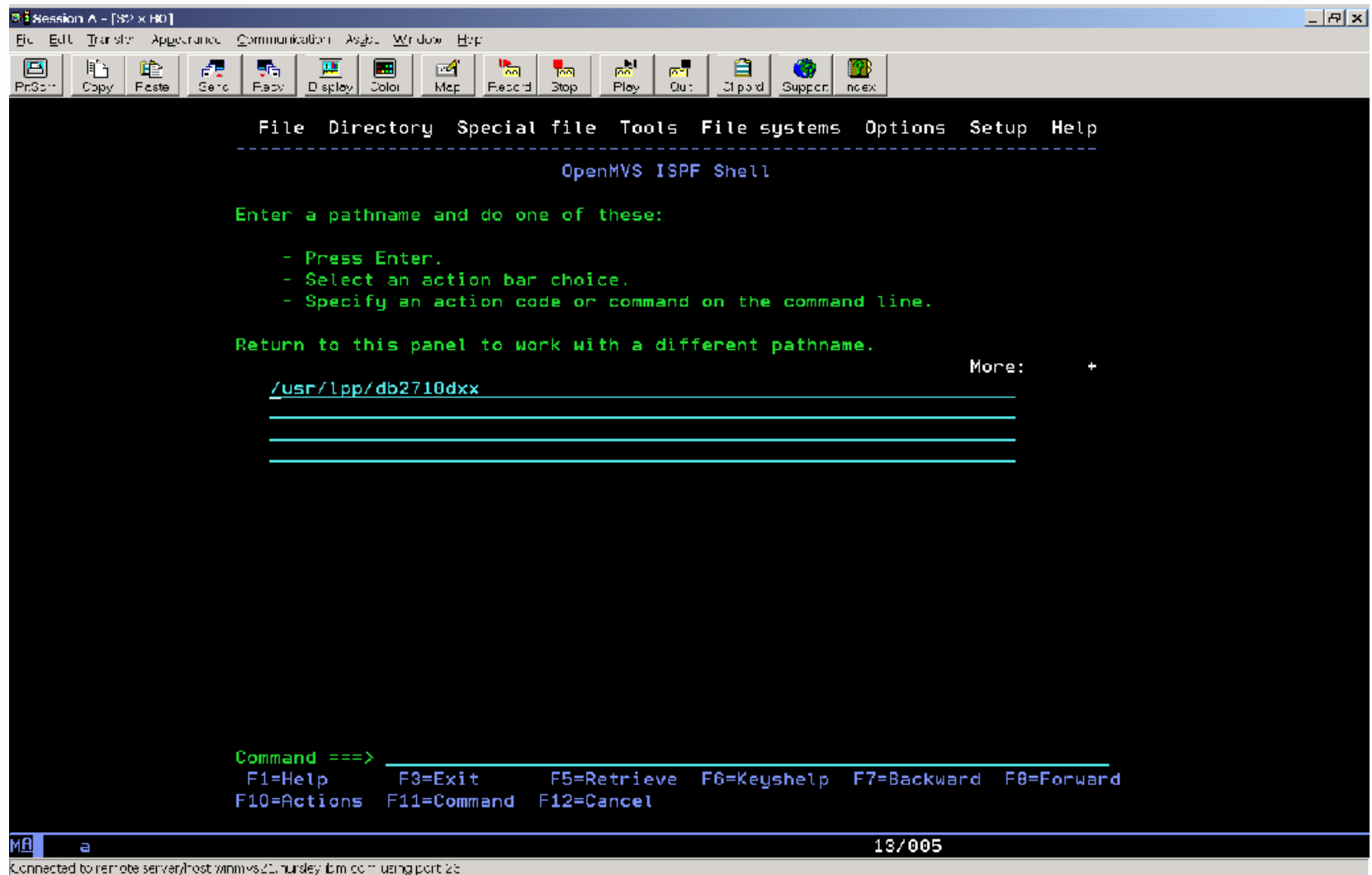
- Solution: code an executable script file:

- ▷ Create a sequential file of LRECL 255 bytes
- ▷ Edit it and code the entire dxxadm enable\_column command as a single line
- ▷ Use the OPUT command to copy its contents into the HFS as follows:

```
= OPUT 'COOKI.XML.SCR255' -  
  '/usr/lpp/db2710dxx/dxx/enable_col.scr'
```

- ▷ Invoke the USS and change directory to the location of the script
- ▷ Execute the enable\_col.scr file, to enable the ORDER column.

# Accessing the ISHELL from ISPF



# ISHELL view of XML Extender Home Directory

```
Session A [32 x 80]
File Edit Transfer Appearance Communicator Assist Window Help
PrintScreen Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Support Index

Directory List

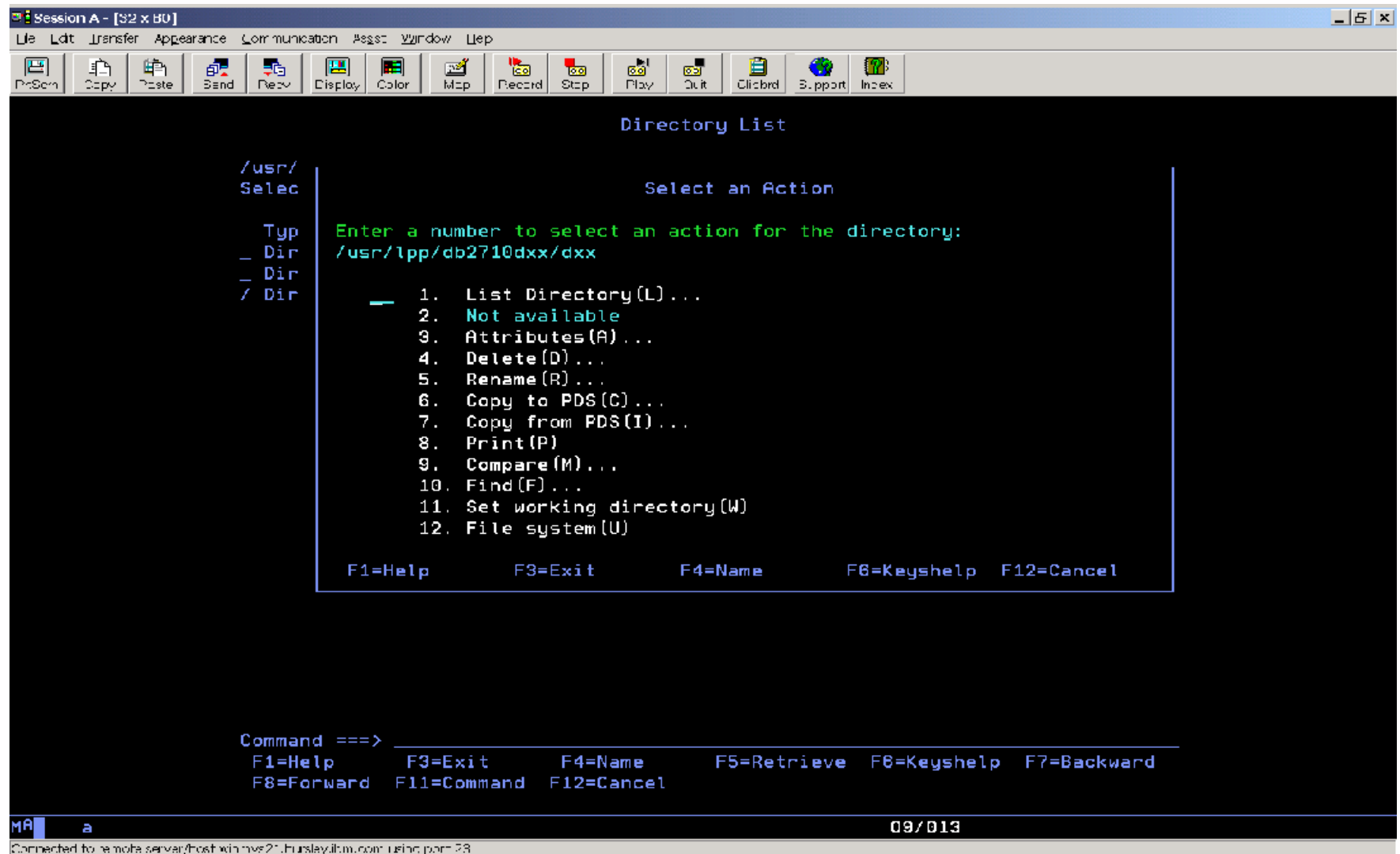
/usr/lpp/db2710dxx/
Select one or more files with / or action codes.

Type  Filename                                     Row 1 of 3
Dir   .
Dir   ..
Dir   dxx

Command ==>
F1=Help   F3=Exit   F4=Name   F5=Retrieve F6=Keyshelp F7=Backward
F8=Forward F11=Command F12=Cancel

30/015
Connected to remote server/host winvsa21.hurs.ibm.com using port 22
```

# Using the ISHELL "/" Line Command



The screenshot shows the ISHELL (IBM Systems Shell) interface. At the top is a menu bar with options: File, Edit, Transfer, Appearance, Communication, Assist, Window, Help. Below the menu bar is a toolbar with icons for various functions like Print Screen, Copy, Paste, Send, Receive, Display, Color, Map, Record, Stop, Play, Quit, Clipboard, Support, and Index. The main window displays a 'Directory List' for the path '/usr/'. On the left, a vertical list shows 'Dir' selected. The main area prompts the user to 'Select an Action' and lists 12 options: 1. List Directory (L)..., 2. Not available, 3. Attributes (A)..., 4. Delete (D)..., 5. Rename (R)..., 6. Copy to PDS (C)..., 7. Copy from PDS (I)..., 8. Print (P), 9. Compare (M)..., 10. Find (F)..., 11. Set working directory (W), 12. File system (U). At the bottom of the main area, function key shortcuts are listed: F1=Help, F3=Exit, F4=Name, F6=Keyshelp, F12=Cancel. Below the main area is a 'Command ==>' prompt with a list of function key shortcuts: F1=Help, F3=Exit, F4=Name, F5=Retrieve, F6=Keyshelp, F7=Backward, F8=Forward, F11=Command, F12=Cancel. The status bar at the bottom shows 'MA a' on the left, '09/013' in the center, and 'Connected to the node server/host with mvs21.hursley.ibm.com using port 23' on the right.

```
Session A - [32 x 80]
File Edit Transfer Appearance Communication Assist Window Help
Print Screen Copy Paste Send Receive Display Color Map Record Stop Play Quit Clipboard Support Index

Directory List

/usr/
Select an Action

Enter a number to select an action for the directory:
/usr/lpp/db2710dxx/dxx

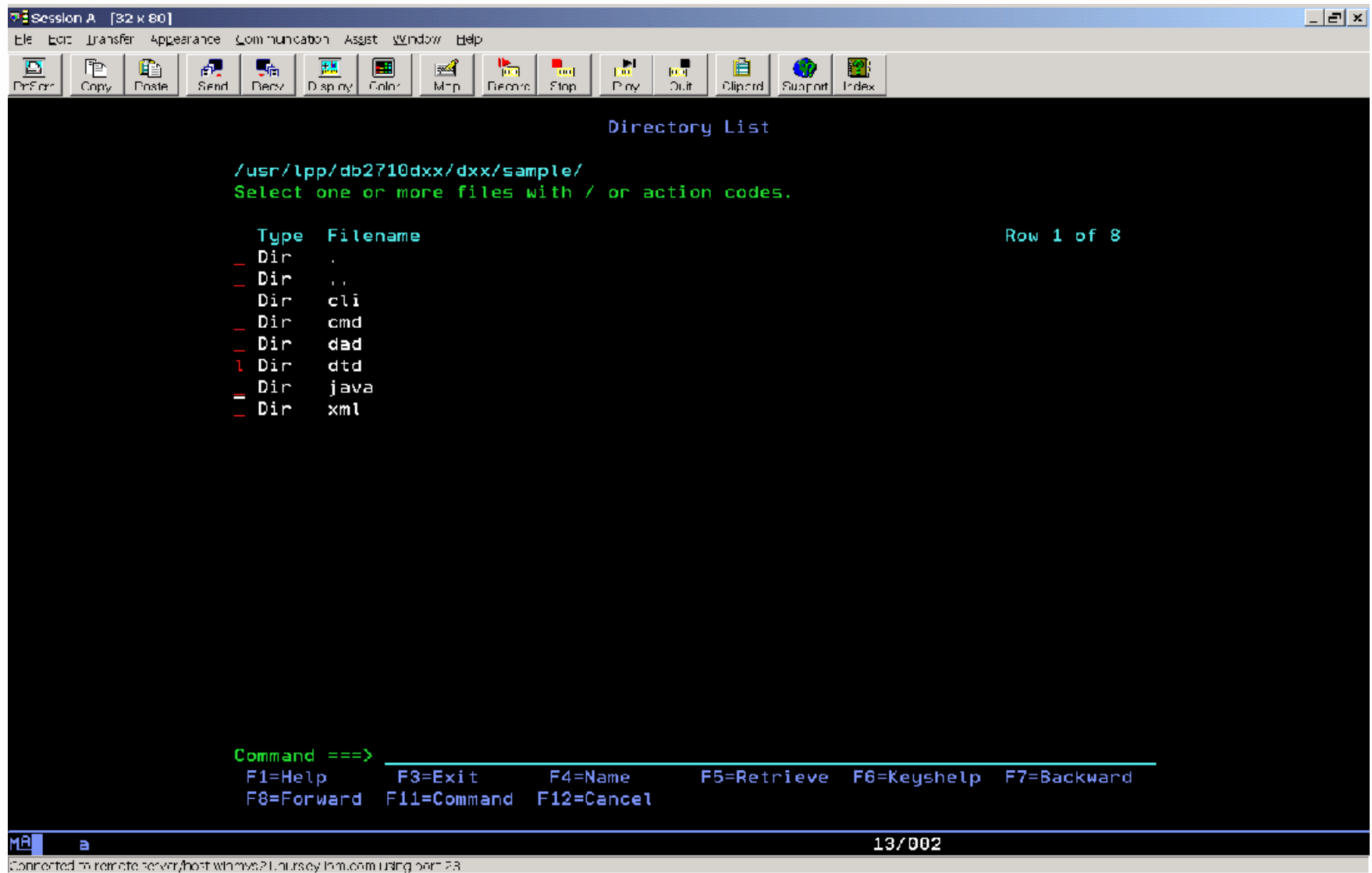
1. List Directory (L)...
2. Not available
3. Attributes (A)...
4. Delete (D)...
5. Rename (R)...
6. Copy to PDS (C)...
7. Copy from PDS (I)...
8. Print (P)
9. Compare (M)...
10. Find (F)...
11. Set working directory (W)
12. File system (U)

F1=Help F3=Exit F4=Name F6=Keyshelp F12=Cancel

Command ==>
F1=Help F3=Exit F4=Name F5=Retrieve F6=Keyshelp F7=Backward
F8=Forward F11=Command F12=Cancel

MA a 09/013
Connected to the node server/host with mvs21.hursley.ibm.com using port 23
```

# XML Extender Product Directories



The screenshot shows a terminal window titled "Session A [32 x 80]". The menu bar includes File, Edit, Transfer, Appearance, Communication, Assist, Window, and Help. The toolbar contains icons for various actions like Print, Copy, Paste, Send, Recv, Display, Color, Map, Record, Stop, Play, Quit, Clipboard, Support, and Index. The main display area shows a "Directory List" for the path "/usr/lpp/db2710dxx/dxx/sample/". It prompts the user to "Select one or more files with / or action codes." and displays a list of directories: ., .., cli, cmd, dad, dtd, java, and xml. The list is labeled "Row 1 of 8". At the bottom, there is a "Command ==>" prompt and a list of function key shortcuts: F1=Help, F3=Exit, F4=Name, F5=Retrieve, F6=Keyshelp, F7=Backward, F8=Forward, F11=Command, and F12=Cancel. The status bar at the bottom shows "MA a" on the left, "13/002" in the center, and "Connected to remote server/host winnys21.lursey.ibm.com using port 23" on the right.

```
Session A [32 x 80]
File Edit Transfer Appearance Communication Assist Window Help
Print Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Support Index

Directory List

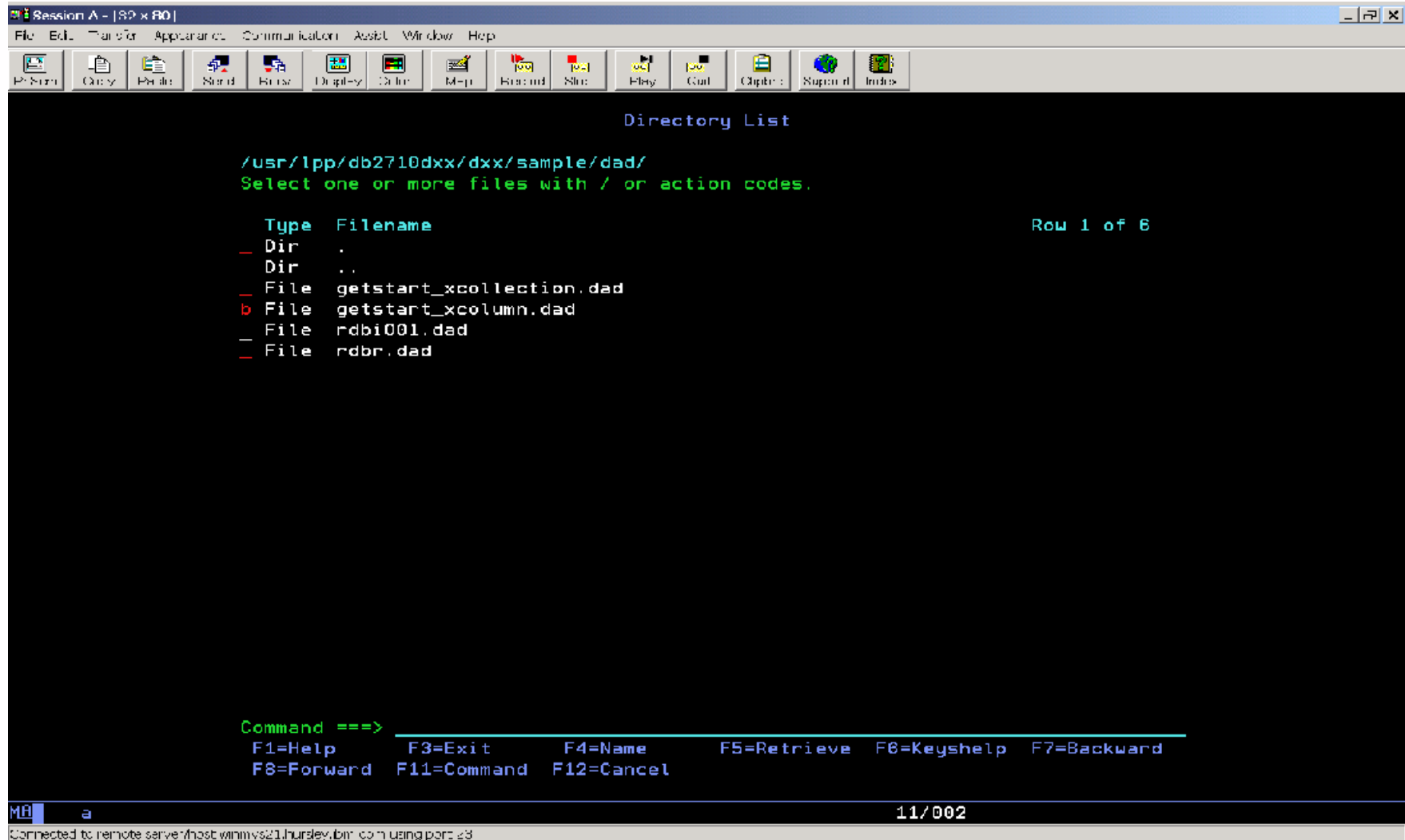
/usr/lpp/db2710dxx/dxx/sample/
Select one or more files with / or action codes.

Type  Filename                                     Row 1 of 8
- Dir  .
- Dir  ..
- Dir  cli
- Dir  cmd
- Dir  dad
- Dir  dtd
- Dir  java
- Dir  xml

Command ==>
F1=Help      F3=Exit      F4=Name      F5=Retrieve  F6=Keyshelp  F7=Backward
F8=Forward   F11=Command  F12=Cancel

MA a                                                13/002
Connected to remote server/host winnys21.lursey.ibm.com using port 23
```

# XML Extender DAD Directory



The screenshot shows a terminal window titled "Session A - [80 x 80]". The menu bar includes File, Edit, Transfer, Appearance, Communication, Assist, Window, and Help. The toolbar contains icons for various actions like Run, Copy, Paste, Stop, Run, Display, Color, Map, Record, Stop, Play, Quit, Capture, Support, and Index. The main display area shows a "Directory List" for the path "/usr/lpp/db2710dxx/dxx/sample/dad/". It prompts the user to "Select one or more files with / or action codes." and displays a table of files. The table has two columns: "Type" and "Filename". The files listed are: ". Dir", ".. Dir", "getstart\_xcollection.dad File", "getstart\_xcolumn.dad File", "rdbi001.dad File", and "rdbi.dad File". The status bar at the bottom shows "11/002" and a connection message: "Connected to remote server (host winmvs21.hursley.ibm.com using port 23)".

```
Session A - [80 x 80]
File Edit Transfer Appearance Communication Assist Window Help

[Icons: Run, Copy, Paste, Stop, Run, Display, Color, Map, Record, Stop, Play, Quit, Capture, Support, Index]

Directory List

/usr/lpp/db2710dxx/dxx/sample/dad/
Select one or more files with / or action codes.

Type      Filename
- Dir      .
- Dir      ..
- File     getstart_xcollection.dad
- File     getstart_xcolumn.dad
- File     rdbi001.dad
- File     rdbi.dad

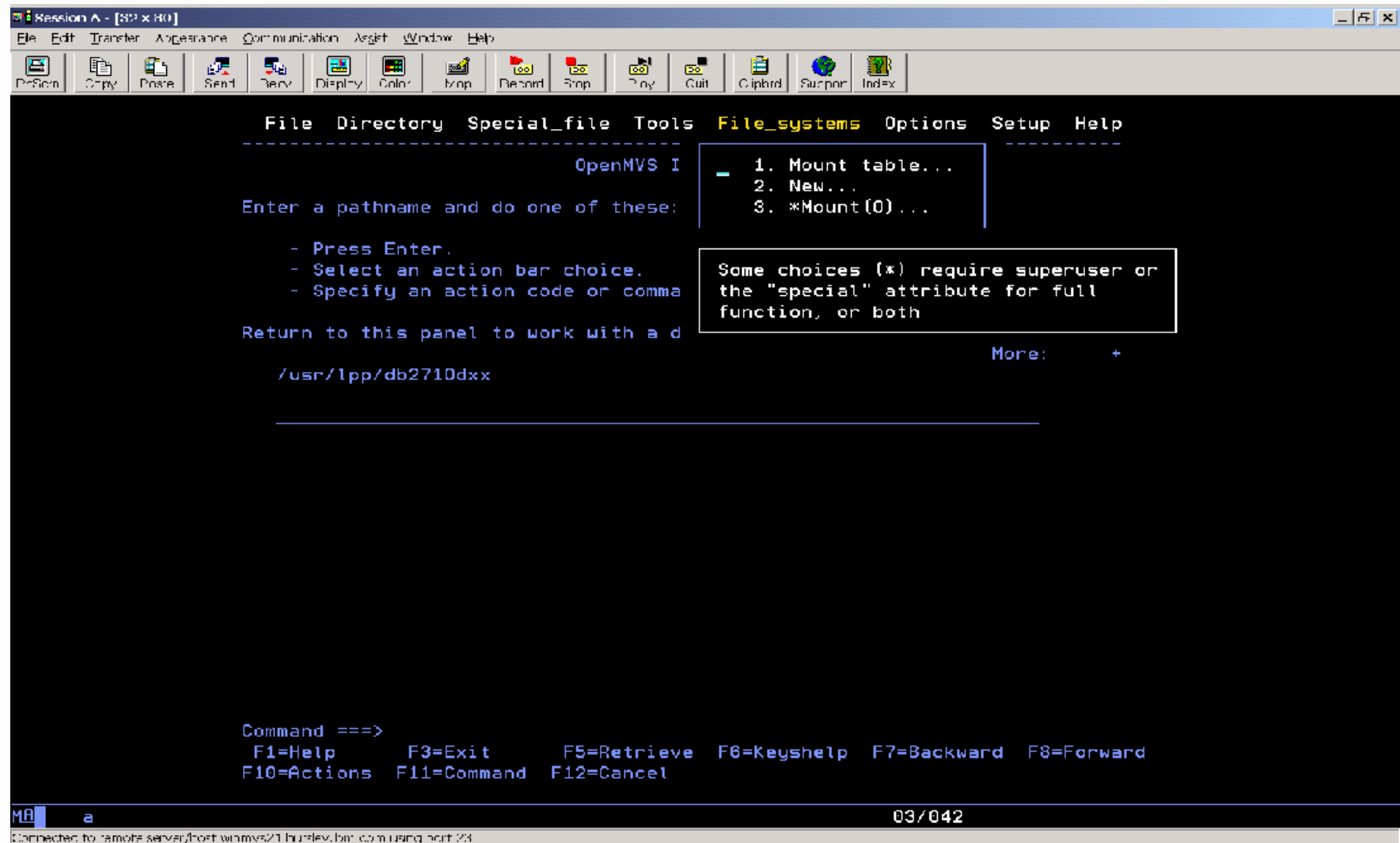
Row 1 of 6

Command ==>
F1=Help      F3=Exit      F4=Name      F5=Retrieve  F6=Keyshelp  F7=Backward
F8=Forward   F11=Command  F12=Cancel

11/002
Connected to remote server (host winmvs21.hursley.ibm.com using port 23)
```



# ISHELL - Displaying File Systems





# Mount Table highlighting XML HFS

```
Session A - [32 x 00]
File Edit Transfer Appearance Communication Assist Window Help

Work with Mounted File Systems

Select one or more file systems with / or action codes.
U=Unmount A=Attributes C=Change mode R=Reset unmount or quiesce
File system name Status Row 1 of 36
*AMD/u Available
OMVS.MVS20.CICSTS Available
OMVS.MVS20.CTG.V312 Available
OMVS.MVS20.DWC.V710.SIWHHFS.S011205 Available
OMVS.MVS20.DXX.V710.SDXXHFS Available
OMVS.MVS20.ETC Available
OMVS.MVS20.JAVA118.D2000038 Available
OMVS.MVS20.JAVA118.ROS.D2001108 Available
OMVS.MVS20.JAVA130.D2001078 Available
OMVS.MVS20.JDBC.V510.SDSNHFS.SF0301 Available
OMVS.MVS20.JDBC.V710.SDMBHFS.S011205 Available
OMVS.MVS20.JDBC.V710.SDSNHFS.S011205 Available
OMVS.MVS20.NOTES461.DATA.HFS Available
OMVS.MVS20.NOTES461.MAIL.HFS Available
OMVS.MVS20.NOTES504.PROD.HFS Available
OMVS.MVS20.OS39026.HPJ.D2000153 Available
OMVS.MVS20.OS39027.HOM.HFS.D1999302 Available
OMVS.MVS20.OS39028.ROOT.D2002093 Available
OMVS.MVS20.OS39028.WAS.V302.D2000325 Available
OMVS.MVS20.OS39029.WAS.V35.SEJSHFS2 Available
OMVS.MVS20.OS39029.WEBAS120 Available
OMVS.MVS20.PERL.V553 Available
OMVS.MVS20.SHIRAZ.DRIVER28 Available
OMVS.MVS20.TMP Available
Command ==>
F1=Help F3=Exit F5=Retrieve F6=Keyshelp F7=Backward F8=Forward
F11=Command F12=Cancel

a 06/002
Connected to remote server/fac: wlmvsv21.courtesy.ibm.com user ghol...25
```

# OMVS access to XML Extender sample DAD Directory

```
Session A - [82 x 80]
File Edit Transfer Appearance Communication Assist Window Help
[Icons: MScreen, Copy, Paste, Send, Recv, Display, Color, Map, Record, Stop, Play, Quit, Clipboard, Support, Index]

* to the contents of this computer system. Offenders are liable *
* to criminal prosecution. *
* IF YOU ARE NOT AN AUTHORISED USER DISCONNECT IMMEDIATELY. *
* *
* WINMVS20 - 29th October 1998. *
* *
* Welcome to OpenEdition Services & Utilities for OS390 rel 8. *
* *
* If you require additional help using this product, please *
* contact DS Systems Support. *
*-----*
COOKI:/u/cooki: >cd /usr/lpp/db2710dxx/dxx/sample
COOKI:/usr/lpp/db2710dxx/dxx/sample: >ls -l
total 96
drwxrwxr-x  2 SYSDSP  TTY          8192 Oct 12  2001 cli
drwxrwxr-x  2 SYSDSP  TTY          8192 Oct 12  2001 cmd
drwxrwxr-x  2 SYSDSP  TTY          8192 Nov 13  2001 dad
drwxrwxr-x  2 SYSDSP  TTY          8192 Oct 12  2001 dtd
drwxrwxr-x  2 SYSDSP  TTY          8192 Oct 12  2001 java
drwxrwxr-x  2 SYSDSP  TTY          8192 Nov 23  2001 xml
COOKI:/usr/lpp/db2710dxx/dxx/sample: >cd dad
COOKI:/usr/lpp/db2710dxx/dxx/sample/dad: >ls -l
total 32
-rw-----  1 COOKI  TTY          1894 Apr 22  10:38 getstart_xcollection.dad
-rw-----  1 COOKI  TTY           836 Feb 21  09:59 getstart_xcolumn.dad
-rw-----  1 COOKI  TTY          3633 Feb 14  10:19 rdbi001.dad
-rw-----  1 COOKI  TTY          3745 Feb 14  10:19 rdbn.dad
COOKI:/usr/lpp/db2710dxx/dxx/sample/dad: >
===> _

                                RUNNING
ESC=$  1=Help      2=SubCmd   3=HlpRetrn  4=Top      5=Bottom   6=TS0
        7=BackScr  8=Scroll   9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve

ME  a 29/007
Connected to remote server/host winmvs21.hursley.ibm.com using port 28
```

Permission  
Bits

Directories  
Files

# UNIX Security - 'Permission Bits'

```
drwxrwxr-x 2 SYSDSP TTY      8192 Oct 12 2001 cli
drwxrwxr-x 2 SYSDSP TTY      8192 Oct 12 2001 cmd
drwxrwxr-x 2 SYSDSP TTY      8192 Oct 12 2001 dad

-rwx --- --- 2 COOKI  TTY      8192 Oct 12 2001 getstart_xcolumn.dad
```

TYPE	USER	GROUP	OTHER
D	R W X	R W X	R - X
Binary	4 2 1	4 2 1	4 - 1
Dec	7	7	5

# Useful XML References

---

- XML Extender Administration & Programming - SC26-9949
  - ▷ This manual describes the XML Implementation process in detail
- DB2 for OS/390 and z/OS Powering the World's e-business Solutions - SG24-6257
  - ▷ Chapter 8 in this publication describes sample scenarios of using XML Columns and Collections
- Integrating XML with DB2 XML Extender and DB2 Text Extender SG24-6130
- DB2 XML Extender Hints and Tips (red paper)  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp0135.pdf>

Finally...

**Thanks!** We hope you have found these sessions useful

Don't forget to complete your feedback form