Chapter 14

Database Connectivity and Web Technologies
Database Connectivity

• Mechanisms by which application programs connect and communicate with data sources
  – Also known as database middleware

• Data repository:
  – Also known as a data source
  – Used to store data generated by an application program
Native SQL Connectivity

• Connection interface provided by database vendors
  – Unique to each vendor
• Example: Oracle RDBMS
  – Must install and configure Oracle’s SQL*Net interface in client computer
• Interfaces optimized for particular vendor’s DBMS
ODBC, DAO, and RDO

• Open Database Connectivity (ODBC)
  – Microsoft’s implementation
  – Widely supported database connectivity interface
  – Any Windows application can access relational data sources
  – Uses SQL via standard application programming interface (API)
ODBC, DAO, and RDO (cont’d.)

• Data Access Objects (DAO)
  – Object-oriented API
    • Accesses MS Excel, MS FoxPro, and dBase databases from Visual Basic programs
  – DAO interface can also be used to access other relational style data sources
Remote Data Objects (RDO)
- Higher-level object-oriented application interface used to access remote database servers
- Uses lower-level DAO and ODBC for direct access to databases
**FIGURE 14.2** Using ODBC, DAO, and RDO to access databases

Client Applications

MS Word  MS Access  MS Excel

RDO

Remote Data Objects

DAO

Data Access Objects

Jet Engine

Jet Engine supports MS Access databases and other SQL-aware data sources.

ODBC API

ODBC Driver Manager

ODBC Database Driver

Oracle Driver  MS SQL Driver  ODBC Driver

Oracle  MS SQL  Access

Database vendors provide ODBC database drivers so Windows applications can access their respective databases.

SOURCE: Course Technology/Cengage Learning
• Basic ODBC architecture has three main components:
  – High-level ODBC API through which application programs access ODBC functionality
  – Driver manager that is in charge of managing all database connections
  – ODBC driver that communicates directly to DBMS
1. From Excel, select Get External Data, From Other Sources and From Microsoft Query options to retrieve data from an Oracle RDBMS.
2. Select the Gradora ODBC data source.
3. Enter the authentication parameters. ODBC uses the connection parameters to connect to the data source.
4. Select the table and columns to use in the query.
5. Select filtering options to restrict the rows returned.
6. Select sorting options to order the rows.
7. Select Return Data to Microsoft Office Excel.
8. Excel uses the ODBC API to pass the SQL request down to the database. Oracle executes the request and generates a result set. Excel issues calls to the ODBC API to retrieve the result set and populate the spreadsheet.

SOURCE: Course Technology/Cengage Learning
OLE-DB

• Object Linking and Embedding for Database
• Database middleware that adds object-oriented functionality for access to data
• Functionality divided into two types of objects:
  – Consumers
  – Providers
Database Systems, 10th Edition

**Applications**

- Program directly to MySQL database API
- Direct access to SQL-based or other data sources via OLE DB API
- Program to ADO C-API to access any datasource type through OLE DB
- ADO.NET (2002)
  - Access other datasources manually

**Data Sources**

- SQL-based data sources
  - Relational databases
  - e.g. MySQL, MS-SQL, Oracle
- Non-SQL-based data sources
  - e.g. Directory Services, XML files, MS Exchange Server

**OLE DB (1996)**

- OLE DB uses ODBC to access SQL-based data sources

**ODBC (1992)**

- ODBC accesses any SQL-based data source for which there is an ODBC driver available
OLE-DB (cont’d.)

- ActiveX Data Objects (ADO) provides high-level application-oriented interface to interact with OLE-DB, DAO, and RDO
- ADO provides unified interface to access data from any programming language that uses the underlying OLE-DB objects
ADO.NET

• Data access component of Microsoft’s .NET application development framework

• Two new features for development of distributed applications:
  – DataSet is disconnected memory-resident representation of database
  – DataSet is internally stored in XML format
    • Data in DataSet made persistent as XML documents
• Specific objects manipulate data in data source
  – Connection
  – Command
  – DataReader
  – DataAdapter
  – DataSet
  – DataTable
Java Database Connectivity (JDBC)

• Java is an object-oriented programming language

• Advantages of JDBC:
  – Company can leverage existing technology and personnel training
  – Allows direct access to database server or access via database middleware
  – Provides a way to connect to databases through an JDBC driver
Database Internet Connectivity

- Web database connectivity allows new innovative services that:
  - Increase customer satisfaction through creation of Web-based support services
  - Allow anywhere, anytime data access using mobile smart devices via the Internet
  - Yield fast and effective information dissemination through universal access
Web-to-Database Middleware: Server-Side Extensions

• Web server is the main hub through which Internet services are accessed
• Dynamic Web pages are at the heart of current generation Web sites
• Server-side extension: a program that interacts directly with the Web server
  – Also known as Web-to-database middleware
Figure 14.8: Web-to-database middleware

1. HTTP page request from the client computer.
2. Web server receives the request.
3. Web server determines if the page contains script language and passes the script page to the Web-to-database middleware.
5. Database server passes the query results back to the Web-to-database middleware.
6. Web-to-database middleware passes the query results in HTML format back to the Web server.
7. Web server sends the HTML formatted page to the client.
8. The result of the database query is displayed in HTML format.

Source: Course Technology/Cengage Learning.
The Web Browser

- Software that lets users navigate the Web
- Located in client computer
- Interprets HTML code received from Web server
- Presents different page components in standard way
- Web is a stateless system: Web server does not know the status of any clients
Client-Side Extensions

• Add functionality to Web browser
• Three general types:
  – Plug-ins
  – Java and JavaScript
  – ActiveX and VBScript
Client-Side Extensions (cont’d.)

• Plug-in: an external application automatically invoked by the browser when needed
• Java and JavaScript: embedded in Web page
  – Downloaded with the Web page and activated by an event
• ActiveX and VBScript: embedded in Web page
  – Downloaded with page and activated by event
  – Oriented to Windows applications
Web Application Servers

- Middleware application that expands the functionality of Web servers
  - Links them to a wide range of services
- Some uses of Web application servers:
  - Connect to and query database from Web page
  - Create dynamic Web search pages
- Some features of Web application servers:
  - Security and user authentication
  - Access to multiple services
Web Database Development

• Process of interfacing databases with the Web browser
• Code examples
  – ColdFusion
  – PHP
ColdFusion code to query the VENDOR table

```html
<HTML>
<HEAD>
<TITLE>Rob & Coriolis - ColdFusion Examples</TITLE>
</HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<CFQUERY NAME="venlist" DATASOURCE="RobCor">
SELECT * FROM VENDOR ORDER BY VEN_CODE
</CFQUERY>
</BODY>
</HTML>

<CENTER><B>Simple Query using CFQUERY and CFOUTPUT</B></CENTER>
<CENTER>(Vertical Output)</CENTER>

<CENTER>
Your query returned #venlist.RecordCount# records
</CENTER>

<CFOUTPUT>

PRE
VEN_CODE: #VEN_CODE#
VEN_NAME: #VEN_NAME#
CONTACT_PERSON: #VEN_CONTACT_NAME#
ADDRESS: #VEN_ADDRESS#
CITY: #VEN_CITY#
STATE: #VEN_STATE#
ZIP: #VEN_ZIP#
PHONE: #VEN_PH#
FAX: #VEN_FAX#
E-MAIL: #VEN_EMAIL#
CUSTOMER_ID: #VEN_CUS_ID#
SUPPORT_ID: #VEN_SUPPORT_ID#
SUPPORT_PHONE: #VEN_SUPPORT_PH#
VENDOR_WEB_PAGE: #VEN_WEB_PAGE#
PRE</CFOUTPUT>
</PRE>
</BODY>
</HTML>
```

SOURCE: Course Technology/Cengage Learning
```php
<html>
<head>
<title>Rob &amp; Corone - PHP Example</title>
</head>
<body>
<?php
$dbc = odbc_connect("ExeCov", ".",".");
$sql = "SELECT * FROM VENDOR WHERE VENDOR_CODE = ";
$rs = odbc_exec($dbc, $sql);
while (odbc_fetch_row($rs))
{
    $VENDOR_CODE = odbc_result($rs, "VENDOR_CODE");
    $VENDOR_NAME = odbc_result($rs, "VENDOR_NAME");
    $VENDOR_CONTACT_NAME = odbc_result($rs, "VENDOR_CONTACT_NAME");
    $VENDOR_ADDRESS = odbc_result($rs, "VENDOR_ADDRESS");
    $VENDOR_CITY = odbc_result($rs, "VENDOR_CITY");
    $VENDOR_STATE = odbc_result($rs, "VENDOR_STATE");
    $VENDOR_ZIP = odbc_result($rs, "VENDOR_ZIP");
    $VENDOR_PH = odbc_result($rs, "VENDOR_PH");
    $VENDOR_FAX = odbc_result($rs, "VENDOR_FAX");
    $VENDOR_EMAIL = odbc_result($rs, "VENDOR_EMAIL");
    $VENDOR_CUS_ID = odbc_result($rs, "VENDOR_CUS_ID");
    $VENDOR_SUPPORT_ID = odbc_result($rs, "VENDOR_SUPPORT_ID");
    $VENDOR_SUPPORT_PH = odbc_result($rs, "VENDOR_SUPPORT_PH");
    $VENDOR_WEB_PAGE = odbc_result($rs, "VENDOR_WEB_PAGE");
    echo $VENDOR_CODE;  
    echo "VENDOR CODE: ",$VENDOR_CODE,"<br>";
    echo $VENDOR_NAME;  
    echo "VENDOR NAME: ",$VENDOR_NAME,"<br>";
    echo $VENDOR_CONTACT_NAME;  
    echo "VENDOR CONTACT NAME: ",$VENDOR_CONTACT_NAME,"<br>";
    echo $VENDOR_ADDRESS;  
    echo "VENDOR ADDRESS: ",$VENDOR_ADDRESS,"<br>";
    echo $VENDOR_CITY;  
    echo "VENDOR CITY: ",$VENDOR_CITY,"<br>";
    echo $VENDOR_STATE;  
    echo "VENDOR STATE: ",$VENDOR_STATE,"<br>";
    echo $VENDOR_ZIP;  
    echo "VENDOR ZIP: ",$VENDOR_ZIP,"<br>";
    echo $VENDOR_PH;  
    echo "VENDOR PH: ",$VENDOR_PH,"<br>";
    echo $VENDOR_FAX;  
    echo "VENDOR FAX: ",$VENDOR_FAX,"<br>";
    echo $VENDOR_EMAIL;  
    echo "VENDOR EMAIL: ",$VENDOR_EMAIL,"<br>";
    echo $VENDOR_CUS_ID;  
    echo "VENDOR CUS ID: ",$VENDOR_CUS_ID,"<br>";
    echo $VENDOR_SUPPORT_ID;  
    echo "VENDOR SUPPORT ID: ",$VENDOR_SUPPORT_ID,"<br>";
    echo $VENDOR_SUPPORT_PH;  
    echo "VENDOR SUPPORT PH: ",$VENDOR_SUPPORT_PH,"<br>";
    echo $VENDOR_WEB_PAGE;  
    echo "VENDOR WEB PAGE: ",$VENDOR_WEB_PAGE,"<br>";
    echo $VENDOR_CODE;  
}
odbc_close($dbc);
</body>
</html>
```

SOURCE: Course Technology/Cengage Learning
Extensible Markup Language (XML)

• Companies use Internet to create new systems that integrate their data
  – Increase efficiency and reduce costs
• Electronic commerce enables organizations to market to millions of users
• Most e-commerce transactions take place between businesses
• HTML Web pages display in the browser
  – Tags describe how something looks on the page
Extensible Markup Language (XML) (cont’d.)

- Extensible Markup Language (XML)
  - Metalanguage to represent and manipulate data elements
  - Facilitates exchange of structured documents over the Web
  - Allows definition of new tags
    - Case sensitive
    - Must be well-formed and properly nested
    - Comments indicated with < and >
    - XML and xml prefixes reserved for XML tags only
FIGURE 14.13 Contents of the productlist.xml document

<?xml version ="1.0"?>
<ProductList>
    <Product>
        <P_CODE>23109-HB</P_CODE>
        <P_DESCRIPT>Claw hammer</P_DESCRIPT>
        <P_INDATE>08/19/2009</P_INDATE>
        <P_QOH>23</P_QOH>
        <P_MIN>10</P_MIN>
        <P_PRICE>$5.95</P_PRICE>
    </Product>
    <Product>
        <P_CODE>23114-AA</P_CODE>
        <P_DESCRIPT>Sledge Hammer, 12 lb.</P_DESCRIPT>
        <P_INDATE>09/01/2009</P_INDATE>
        <P_QOH>8</P_QOH>
        <P_MIN>5</P_MIN>
        <P_PRICE>$14.40</P_PRICE>
    </Product>
</ProductList>

SOURCE: Course Technology/Cengage Learning
Document Type Definitions (DTD) and XML Schemas

• Document Type Definition (DTD)
  – File with .dtd extension that describes elements
  – Provides composition of database’s logical model
  – Defines the syntax rules or valid tags for each type of XML document

• Companies engaging in e-commerce transaction must develop and share DTDs

• DTD referenced from inside XML document
FIGURE 14.16 DTD and XML documents for order data

OrderData.dtd

OrderData.xml

"+" sign indicates one or more ORD_PRODS elements

Two ORD_PRODS elements in XML document

SOURCE: Course Technology/Cengage Learning
Document Type Definitions (DTD) and XML Schemas (cont’d.)

• XML schema
  – Advanced data definition language
  – Describes the structure of XML data documents

• Advantage of XML schema:
  – More closely maps to database terminology and features

• XML schema definition (XSD) file uses syntax similar to XML document
XML Presentation

- XML separates data structure from presentation and processing
- Extensible Style Language (XSL) displays XML data
  - Defines the rules by which XML data are formatted and displayed
  - Two parts:
    - Extensible Style Language Transformations (XSLT)
    - XSL style sheets
FIGURE 14.18 Framework for XML transformations

XML document → XSL transformations → XSL style sheets → HTML

- Extract
- Convert

New XML document → Apply formatting rules to XML elements → HTML

XSLT can be used to transform one XML document into another XML document.

The process can render different Web pages for different purposes, such as one page for a Web browser and another for a mobile device.

SOURCE: Course Technology/Cengage Learning
Displaying XML documents

```xml
<?xml version="1.0" ?>
<ProductList>
  <Product>
    <P_CODE>23109-HB</P_CODE>
    <P_DESCRIPT>Claw hammer</P_DESCRIPT>
    <P_INDATE>08/19/2009</P_INDATE>
    <P_QOH>23</P_QOH>
    <P_MIN>10</P_MIN>
    <P_PRICE>5.95</P_PRICE>
  </Product>
  <Product>
    <P_CODE>23114-AA</P_CODE>
    <P_DESCRIPT>Sledge Hammer, 12 lb.</P_DESCRIPT>
    <P_INDATE>09/01/2009</P_INDATE>
    <P_QOH>8</P_QOH>
    <P_MIN>5</P_MIN>
    <P_PRICE>14.40</P_PRICE>
  </Product>
</ProductList>
```

SOURCE: Course Technology/Cengage Learning
XML data binding

```xml
<HTML>
  <HEAD>
    <TITLE>BINDING THE PRODUCTLIST XML DATA TO HTML TABLE</TITLE>
  </HEAD>

  <BODY>
    <XML ID="PRODLIST" SRC="PRODUCTLIST.XML"/>
    <TABLE BORDER="1" DATASRC="#PRODLIST">
      <TR>
        <TD><SPAN DATAFLD="P_CODE"></SPAN></TD>
        <TD><SPAN DATAFLD="P_DESCRIP"></SPAN></TD>
        <TD><SPAN DATAFLD="P_QOH"></SPAN></TD>
        <TD><SPAN DATAFLD="P_PRICE"></SPAN></TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```

SOURCE: Course Technology/Cengage Learning
XML Applications

• B2B exchanges
• Legacy systems integration
• Web page development
• Database support
• Database meta-dictionaries
• XML databases
• XML services