

Building Distributed Web Applications Using ASP

Author: S. Ignat

IPA SA, Republicii Str. 109, Cluj-Napoca Romania
E-mail sorini@ipa.cjnet.ro

ABSTRACT

From the inception, ADO was designed to be a high-level interface to provide ease of access to data stored in a wide variety of database sources.

Active Server Pages proved to be an exciting, new technology that extended standard HTML by adding built-in objects, server-side scripts, access to databases, and ActiveX components.

Active Server Pages allows you to interact with any ODBC-compliant database, be it SQL Server, Access, Oracle, or Informix. Often times developers are connecting to a SQL Server database, but there are many who are connecting to Access databases as well.

Keywords: ADO, DAO, ASP, ODBC, SQL

1. INTRODUCTION

Generally are distinguished two different types of Web applications - Dynamic HTML (DHTML) applications and Microsoft Internet Information Server (IIS) applications. DHTML applications do their processing on the client computer; IIS applications do their processing on the server computer.

The introduction of the scripting languages, which could be embedded inside HTML code, opened new doors for dynamic site development. Active Server Pages (ASP), which managed to arrive just in time for the explosive growth of the World Wide Web, was a next logical step for Web-based application development.

ActiveX technology can be used to create components that you can use on HTML pages or as alternatives to HTML pages. These components can include controls (.ocx files), code components (.exe and .dll files) that run on the client, or ActiveX documents (.vbd files) that function much like an HTML page. All of these components help you create Internet solutions that meet your business needs.

The true power of ASP is the ease and rapidity with which developers can create and implement dynamic Web sites. Indeed, for today's modern Web commerce, a dynamic, database-driven, server-side application that interacts with the client is the norm.

2. DESIGNING DISTRIBUTED APPLICATIONS

With distributed applications, performance and scalability become key aspects of the design process. Distributed applications require that you consider the

application's components and infrastructure, including logical packaging, physical deployment modeling, remote server computer workload analysis, and evaluation of the available network bandwidth.

Ideally, to address performance requirements, each client request should be serviced on demand. On-demand delivery of requested application services to an ever-increasing number of client requests imposes certain scalability requirements. Application servers must be available to provide services whenever clients request them. When an application server is down, so is the business that it supports.

Servers can be unavailable for many reasons, including: a. server platform hardware problems; b. server platform software problems; c. networking problems; d. application problems.

In distributed applications, the code for the distributed components must be bulletproof. Component reliability must approach that of the operating system on which it runs. Obviously, distributed components must be thoroughly tested.

ASP With Databases. One of ASP's greatest assets is that it allows you to tap into a database. The default could be either an Access or a SQL database. You can connect to and open the database by using either a Data Source Name (DSN) or by making a DSN-less connection directly in your script.

3.OPTIMIZED DATABASE ACCESS

Database access often imposes the largest performance penalty on your application. This is especially a concern with distributed applications, where multiple clients simultaneously access common tables and rows.

While choosing the right data access technology (such as OLE DB with ADO) solves an important part of your high-performance requirement, the reality is that most of your application's database access speed will come from careful modeling of the data structures, query optimization, and careful handling of multiuser concurrency situations.

As companies begin to implement new Web-based information systems, data access technology must satisfy complex new scenarios. New developers building Web-based applications need access to data distributed throughout their organization. Valuable company information is, available from a wide variety of sources and data stores, including those in the following list: a)Customer Information Control System (CICS), b) Information Management System (IMS), c) Virtual Sequential Access Method (VSAM), d)AS/400 files, e)relational databases (such as Microsoft SQL Server and Oracle), f)personal spreadsheets (such as MS Excel), g)personal databases (such as MS Jet), h)word processing documents files, i)text files, j)e-mail.

4.UNIVERSAL DATA ACCESS USING OLE DB AND ADO

ADO (ActiveX Data Objects) -- A Microsoft technology that provides a unified way to access different data sources with a single data model. ADO allows you to access data in the same way whether it exists in a database, a spreadsheet, and even text files.

OLE DB - a set of ActiveX interfaces that provide applications with uniform access to data stored in diverse information sources. It supports proper amount of DBMS functionality appropriate to the data source, enabling it to share its data.

OLE DB components enables distributed applications to access diverse data sources (a.enumerators - search for available data sources; b.data source objects - contain the connection to a data source;c.session - provide a context transactions processing).

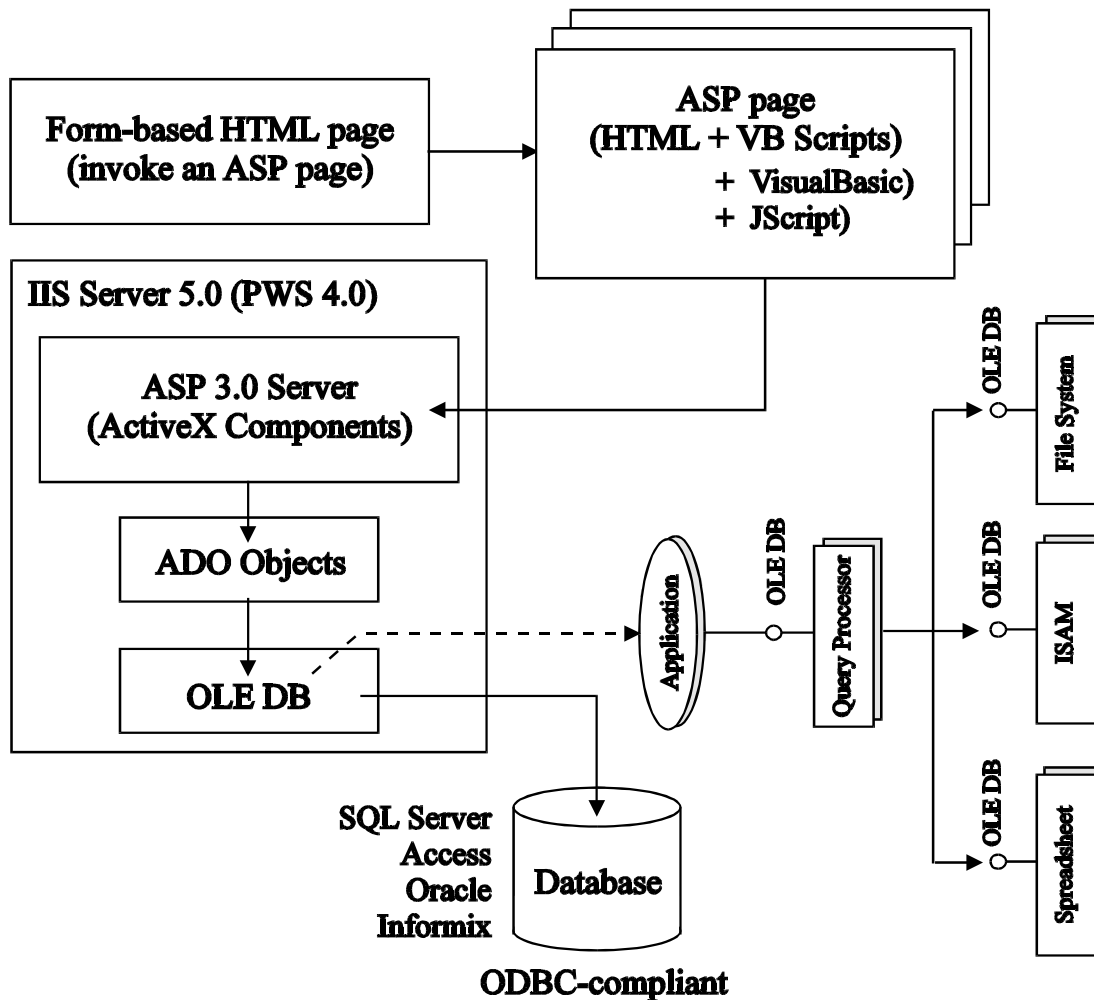


Fig1. ADO/ASP technology – an architectural overview

If a single data store for all types of data isn't the answer, what's the alternative? The solution to accessing different kinds of data throughout the enterprise is to use OLE DB as a data provider and ActiveX Data Objects (ADO) as the data access technology. Data access based on OLE DB and ADO is suitable for a wide range of application design requirements, from small single-workstation processes to large-scale Web applications.

OLE DB is an extensive set of Component Object Model (COM) interfaces that can provide uniform access, to data stored in diverse information sources. The interfaces, in turn, are enhanced to uniquely support the access functionality appropriate to each data source. The OLE DB interfaces allow an individual data store to easily expose its native functionality.

Because the OLE DB and ADO interfaces are based on COM, they support a rich set of integrated services (including transactions, security, and message queuing) to support the broadest range of application data access scenarios. This provides developers with a rich and consistent substrate of data data access design functionality.

It's worth nothing that OLE DB with ADO provides a significant development cost and performance advantage over Open Database Connectivity (ODBC) in two ways. First, ODBC drivers have to implement an SQL relational engine to expose no-relational data. Second, services, such as cursors and query processing must be implemented by every ODBC driver. This represents development cost for the ODBC driver as well as resource consumption in the form of multiple cursor engines and query processors.

ASP employs a scripting environment and VBScript is the default scripting language of choice. However, you can use other languages (such as JScript and Perl) as long as they have a scripting engine that is compatible with the ActiveX scripting standard.

ADO can be used with a variety of programming languages, including Visual Basic, VBScript, JScript, Visual C++, and Visual J++. Further, by using ASP, you can reference ADO components on HTML-generated Web pages.

5. CONCLUSIONS

Active Server Pages has ultimately proven to be of significant value to developers and fueled a revolution in the development of Web-based applications.

In general, the idea of universal data access is about an easily built, lower-cost application mode using a consistent OLE DB with ADO data access approach that will work with all of a company's data sources.

Applications use ADO to talk to differing data stores through existing OLE DB providers, although could be used OLE DB interfaces directly. As a general rule is not needed to develop custom OLE DB providers.

By using OLE DB data providers with ADO, you can: a. access most data stores with a single, consistent technology, b. benefit from the newest development tools, c. minimize training and support, d. reduce implementation and maintenance cost.

DCOM and Microsoft Transaction Server. You can implement Remote Data Service applications on an intranet without using HTTP. If the client- and middle-tier components are on computers within a LAN (local-area network), you can use DCOM to marshal the interfaces and method arguments across the network.

The future of ASP will continue to be very interesting. Later this year, ASP-plus, which is now referred to as ASP.NET, will be formally released by Microsoft. It allows the use of the Visual Basic, C#, and Visual C++ languages with ASP.

6. REFERENCES

- [1] Brian Jepson, O'Reilly. Perl Utilities Guide - Win32 Edition, 1st Edition, 1998.
- [2] A. Keyton Weissinger. ASP in a Nutshell, O'Reilly, 1st Edition, 1999.
- [3] Alex Homer et al, Active Server Pages 3.0, Wrox Press Ltd, 1999.
- [4] Tobias Martinsson. "Active Scripting with PerlScript", Microsoft Internet Developer August 1999, vol 4 no. 8.
- [5] Brian Jepson. "Someone else's database", WebTechniques March 2001, vol 6 issue 03.