Windows DNA - Relevance
Windows DNA - Relevance

A Technical Report

Technical Expertise Level: Intermediate
Requires knowledge of: Windows Operating Systems

INDEX

WHERE WOULD WINDOWS DNA BE RELEVANT TO US? ........................................ 3
WHAT IS IT AND WHAT IS IT NOT? ................................................................. 3
WINDOWS DNA BUILDING BLOCKS .......................................................... 4
BASIC FACTS ON WINDOWS DNA ............................................................. 5
MORE DETAILS ON DNA: WINDOWS DNA SERVICES ............................. 6
Where would Windows DNA be relevant to us?

Customers want these services:

- **Web enabling applications:**
  This is a very real possibility. We can market services for web enabling applications using the Microsoft Windows DNA framework.

- **Development of Large applications:**
  When customers ask for a complete end-to-end solution that involves a database and users in the internet/intranet – we come down to Windows DNA. We use the DNA framework to develop applications.

- **Component development as a part of large applications:**
  Customers want us to develop components for their Windows DNA applications: This might be a very typical development task a few years hence if Microsoft has their way with COM and Windows DNA.

Read on to find out what DNA is all about.

What is it and what is it not?

Windows DNA is not a solution: it is a methodology propounded by Microsoft for building web enabled applications.

Overall, the DNA methodology is an umbrella term that covers many existing technologies to help us design and implement robust, distributed applications. It visualizes this whole application as a series of tiers, with the client at the top and the data store at the bottom. The core of DNA is the use of business objects in a middle tier of the
application, and this is supported by two solutions: Microsoft Transaction Server (MTS), which is the component manager offering full transaction support; and Microsoft Message Queue Server (MSMQ), which provides the fault-tolerance required in a distributed application.

Windows DNA is like a map to help developers, technology planners, and IS managers avoid getting lost in the details of Microsoft's many operating systems, tools, technologies, and applications.

Windows DNA, at its core, is about integrating Web and client-server models of computing.

Explore this a little and get a glimpse of what the model comes down to.

- Want to incorporate area-wise tax figures into the application? Buy a custom tax component.
- Want to read data into a custom view? Buy an ADO component.
- Want to send custom email messages to customers after a transaction? Plug in an email component.

Windows DNA building blocks

Another way of phrasing the same question is “What are the technologies that make up the building blocks for Windows DNA?”

Here are the basic technologies:

1. COM and DCOM: Well, we cannot ever over-emphasize this. COM, COM and COM again.
2. ADO, OLE-DB
Here is approximately what it takes to round off the basic technology skills:

1. Win 32
2. Microsoft Message Queue server.
3. HTML, DHTML
4. Site Server
5. Java
6. Security (Kerberos)

We in CSWL have considerable expertise in most of the supporting technologies.

**Basic Facts on Windows DNA**

- Built around COM: Business objects are COM objects.
- Windows DNA applications can be built using existing tools such as VB, VC++, VJ++, Delphi, etc.
- Massively scaleable. Windows DNA architecture can support thousands of simultaneous users and can do so with the same application logic, tool set, and engineering effort as required for an application that supports a single user.
- Applications built for Windows DNA will support all platforms that support HTML

In DNA, business objects are implemented as software components. These components can be accessed by the client interface application or by another component, and can themselves call on other components, data stores, etc. Componentization of business rules brings many benefits, such as easier maintenance, encapsulation of the rules, protection of intellectual copyright, etc.
DNA is an approach to design that can speed up overall development time, while creating more reliable and fault tolerant applications that are easily distributable over a whole variety of networks. In this book, we're concentrating on the Internet, using TCP/IP as our protocol. However, the components can just as easily be used with compiled applications specially written to follow the DNA principles.

DNA centers around a set of services that are provided by Windows NT Server. These include both application and infrastructure services. The infrastructure services are provided by Windows NT Server itself, and consist of things like the Network service, the NT5 Active Directory service, the Remote File and Print service, the Security service and various other Component services that don’t easily fall into one of these categories.

What DNA means is that each of these services provides a common and easily accessible interface through which other components and scripts can access them. There is no 'natural' language, and developers use whichever language best suits their particular requirements. And, because the interfaces are published and open, independent (non-Microsoft) suppliers can create components and services of their own which can 'plug into' NT Server.

<table>
<thead>
<tr>
<th>Application Services</th>
<th>Example</th>
</tr>
</thead>
</table>

CSWL Inc, Pleasanton, California
<table>
<thead>
<tr>
<th>Database</th>
<th>Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java virtual machine</td>
<td>Microsoft JVM</td>
</tr>
<tr>
<td>Mail and collaboration service</td>
<td>Microsoft Exchange Server</td>
</tr>
<tr>
<td>Message queuing service</td>
<td>Microsoft Message Queue Server (MSMQ)</td>
</tr>
<tr>
<td>Scripting</td>
<td>Active Server Pages, Dynamic HTML, JScript, VBScript</td>
</tr>
<tr>
<td>Transaction service</td>
<td>Microsoft Transaction Server (MTS)</td>
</tr>
<tr>
<td>Universal data access</td>
<td>ADO, OLE-DB, ODBC, etc.</td>
</tr>
<tr>
<td>Web browser</td>
<td>Web browser Netscape Communicator, Internet Explorer.</td>
</tr>
<tr>
<td>Web server</td>
<td>Microsoft Internet Information Server (IIS)</td>
</tr>
</tbody>
</table>

**An important benefit: Scalability**

The most important feature of DNA is probably scalability – which has been one of the major factors blocking adoption of Windows NT in corporate environments.

Scalability is particularly important in a Web environment, where popular sites can attract thousands of visitors per day. It's a lot harder to plan capacity in this environment than on a local area network with a fixed number of users and a stable traffic pattern. To get round this, one of the new services supplied with Windows NT Server is the Microsoft Transaction Server (MTS). As we’ll see
in later chapters, this can bring benefits to DNA-compliant applications in a number of different, and not always obvious, ways.

Microsoft promises that, using Windows DNA, the very same design and architecture needs to be employed regardless of whether the application will have 10 or 10000 concurrent users.

**Trends that suggest we look seriously at Windows DNA**

Here are some things one can speculate on:

1. Component development is increasing in importance. Technological uncertainties are being plugged (to a large extent, in any case) by Microsoft solutions. All these technologies/solutions fit into one big map called Windows DNA. Developers will just have to write business logic code for business objects – which will be COM components.

2. Windows NT server is increasing in importance and performance. This might imply that large corporations would move their applications from other platforms to NT server. And application development on NT server would involve using many of the technologies used in Windows DNA.

3. Once the Y2k rush has spent itself – Web technologies and Microsoft technologies will be a prime focus for spending by large corporations. And Windows DNA and COM will be at the very heart of all that.

One can even proceed a step further and suggest that Windows DNA might be the map to application developers (for Microsoft technologies, at the least) that will finally move software development from the coding-intensive framework to a paradigm oriented towards design and component-reuse.
Copyright Notice:

2002 California Software Labs. All rights Reserved. The contents on the document are not to be reproduced or duplicated in any form or kind, either in part or full, without the written permission of California Software labs. Product and company names mentioned here in are the trademarks of their respective companies.