Oracle9*i* Application Server on Microsoft Windows

An Oracle White Paper
April 2003

ORACLE

EXECUTIVE OVERVIEW

The objective of this document is to provide organizations a view on how Oracle9*i* Application Server (Oracle9*i*AS) can interoperate with Microsoft Windows while leveraging open standards like J2EE. Oracle's goal is to allow organizations to keep their existing investments and continue to build on the latest technologies and ensure these technologies can interoperate in development and deployment.

Oracle offers Oracle9*i*AS – a 100% standards-compliant application server to simplify how internet applications are developed; to make internet applications fast, reliable, manageable, and secure; and to substantially lower the Total Cost of Ownership associated with developing, deploying, and operating internet applications.

Oracle9*i*AS is a competitive Java Application Server and is acknowledged as one of the leading Java/J2EE application servers by Java developers, IT managers and analysts for the three following reasons:

- 100% J2EE Standards Compliance
- Ease of installation, configuration and development
- Fast, scalable, secure, and highly available deployment platform

Oracle9*i*AS offers more than Java. It is a complete and integrated e-business platform. While the majority of application servers were all narrowly focused on Java, Oracle was the first to envision that application servers would evolve from Java servers to complete and integrated e-business platforms. To fulfill this vision, Oracle9*i*AS provides a complete platform that includes portal, wireless, integration, business intelligence, caching, security, directory, and systems management.

For enterprise deployment, Oracle9*i*AS provides maximum availability with Microsoft Windows by working with Microsoft Network Load Balancer and Cluster Service. Organizations can keep their existing Microsoft applications and go on to build new business components using technologies such as J2EE and Web Services, then deploy these mission critical applications with additional safeguards from Oracle on the Microsoft platform.

Oracle has always been committed to supporting Windows as an operating system. Since the late 1980s, the Oracle database has run on early versions of DOS and Windows. Windows is one of the base development platforms of Oracle's server products. As such Oracle9*i*AS is available on Microsoft Windows NT, 2000, XP and 2003. In addition Oracle9*i*AS Containers for J2EE (OC4J), part of Oracle9*i*AS, is available on 64 bit (IA64) Microsoft Windows.

PRODUCT OVERVIEW

It is important that we get a high-level overview of Oracle9*i* Application Server and Microsoft Windows before we delve into a detailed discussion on interoperability and comparison between technologies.

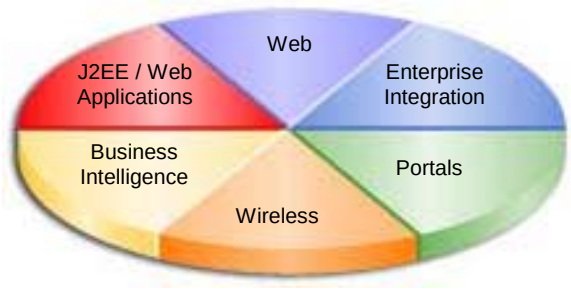
Oracle9*i* Application Server

Oracle9*i*AS is a single integrated product that provides facilities to develop applications based on internet standards and to deploy and manage them reliably. To understand its specific product facilities, let us look into its capabilities from the point of view of application development and deployment.

Application Development

Oracle9*i*AS provides a productive application server environment for developers to develop and deploy internet applications based on open standards such as Java2 Enterprise Edition (J2EE), XML, and emerging Web Services standards. With Oracle9*i*AS, you can simplify access to these applications by creating enterprise portals that provide a single point for users to access these applications from desktop browsers or wireless devices.

Through a comprehensive Enterprise Integration framework, modeling tools, pre-built adapters and Web Services, you can redefine your business processes and integrate your applications with legacy systems, databases, packaged applications, and business partners. With Oracle9*i*AS, you can deliver tailored 1:1 customer experiences via real-time personalization, assess and correlate Web site traffic patterns, and satisfy demands for up-to-the minute business intelligence about your applications. The figure below illustrates the capabilities in the product.



Specially, Oracle9*i*AS offers the following specific facilities:

- *J2EE Facilities* - Oracle9*i*AS (i) is compliant with latest J2EE open standards; (ii) is integrated with the best-of-breed HTML tools to design Web sites; (iii) provides facilities to create dynamic Web sites using JavaServer Pages and Servlets; and (iv) enables you to develop transactional internet applications using Enterprise JavaBeans.
- *Web Services* - Oracle9*i*AS enables you to (i) publish J2EE applications and existing legacy applications as Web services by generating WSDL interfaces; (ii) invoke Web services using the SOAP Protocol both synchronously and asynchronously; and (iii) publish and import Web services from a UDDI Registry. Oracle9*i*AS allows you to integrate Web services written in J2EE with those developed to Microsoft's proprietary .NET specification.
- *Portals* - As the number of Web sites and Web services proliferate within your organization, Oracle9*i*AS allows you to create enterprise portals that offer (i) a single point to access

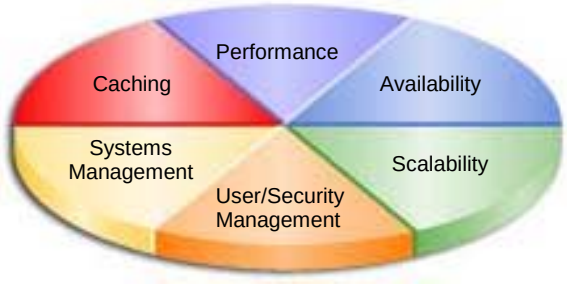
information personalized to individual user needs; (ii) a single point to aggregate information as "portlets"; and a (iii) single place to search for information across the enterprise.

- *Wireless* - Because users need to anytime, anywhere access to applications, Oracle9*i*AS allows you to (i) make any application accessible across any wireless network or a gateway; (ii) make these applications accessible from any wireless device; and (iii) exploit wireless facilities such as voice, location-based services, and wireless security to make applications easy to use.
- *Integration* - Oracle9*i*AS provides standards-based EAI, Web services and B2B including support for Business Activity Monitoring (BAM) and Business Process Management (BPM). Organization can integrate their enterprise Web sites, internet applications, and portals with (i) legacy systems; (ii) Oracle and non-Oracle databases; (iii) packaged applications; and (iv) business partners using a set of open standards such as J2EE, JMS, JCA, and XML messaging standards such as ebXML and RosettaNet.
- *Business Intelligence* - Finally, Oracle9*i*AS offers users comprehensive business intelligence about their applications. It (i) provides real-time "personalized" recommendations based on visitors' behavior; (ii) analyzes click stream logs and extracts intelligence about overall Web traffic on Web sites; and (iii) generate Web-based reports and decision analysis using powerful business intelligence tools.

Application Deployment and Management

With Oracle9*i*AS, you can deploy and manage your applications reliably. With Oracle9*i*AS, you can deploy any of these applications with excellent performance and scalability, reducing the amount of hardware you need to provide the same quality of service to users. You can further improve the performance, scalability, and availability of your systems through built in Web caching. Your applications will remain highly available across system failures, and when software maintenance operations are carried out. Finally, you can implement a centralized system management, security and directory framework to monitor, manage, and secure all of your distributed systems and users.

Specifically, Oracle recognizes that Oracle9*i*AS offers six specific facilities, illustrated below:



Lowest Total Cost of Ownership

Oracle9*i*AS is designed to lower your Total Cost of Ownership (TCO) in developing, deploying and operating applications by offering users a single integrated product. There are a number of features that unify the product:

- *Single Integrated Product* - Oracle9*i*AS is a single integrated product that is unique relative to other application servers in a number of ways. Users (i) install and configure all of its elements as a single product; (ii) secure it by defining users and their security privileges in a single place; (iii)

manage all the systems from a single management console; (iv) globalize applications in a single way; (v) use a common, consistent certified technology stack; (vi) develop applications to a set of open standards - J2EE, Web Services and others; (vii) have these applications integrate automatically into an enterprise portal; and (viii) make these applications and services transparently accessible from browsers and wireless devices. By unifying all these of elements into a single integrated product, Oracle9*i*AS lowers the cost associated with developing, deploying, and operating applications.

- *Lowest TCO* - Oracle9*i*AS provides lowest cost of ownership through (i) lowest software cost due to Oracle9*i*AS' competitive price and more capabilities "in-the-box"; (ii) lowest hardware cost due to the improved scalability and caching facilities that allow you to serve more users with less hardware; (iii) lowest integration cost due to a broader, better integrated product; (iv) lowest administration cost since users manage one product instead of a bundle of different products; (v) lowest support cost due to a single support contract for these different capabilities. This technical white paper highlights the inherent value in leveraging a single e-business platform so companies can minimize spending on IT infrastructure and focus on their core competencies to reduce their total cost of ownership and create a competitive advantage.

Oracle9*i*AS is available in three editions:

- *Oracle9iAS Java Edition* consists of the Oracle HTTP Server (based on Apache plus Mods provided by Oracle), Oracle9*i*AS Containers for J2EE, Web Services, Oracle9*i*AS Toplink, Oracle9*i* JDeveloper (100% Java IDE), and Management (Oracle Enterprise Manager).
- *Oracle9iAS Standard Edition* comprises Oracle9*i*AS Java Edition plus Oracle9*i*AS Portal and Content Management SDK.
- *Oracle9iAS Enterprise Edition* comprises Oracle9*i*AS Standard Edition plus Forms, Security, Directory, Integration, Caching and Business Intelligence.

Wireless and Personalization are additional options to Oracle9*i*AS Enterprise Edition.

MICROSOFT WINDOWS AND .NET ENTERPRISE SERVERS

Microsoft defines the Windows operating system as the "Application Server" but this definition is much narrower than what Oracle9*i*AS provides. Microsoft Windows "application server" does not include portal, enterprise integration, business intelligence, caching or wireless. These components are separate Microsoft products under the branding of Microsoft .NET Enterprise Servers, whereas these are all an integral part of Oracle9*i*AS, ONE product. Each Microsoft .NET Enterprise Servers product is priced separately and includes separate installation, configuration, and management tools. Only the combination of Microsoft Windows and .NET Enterprise Servers can be compared with Oracle9*i*AS.

Microsoft's latest operating system ("application server") is Windows Server 2003.

Microsoft Windows 2003 comes in four editions:

- *Web Edition* - Building and hosting ASP.NET applications and Web Services. This includes IIS, Microsoft's Web server, and the Microsoft .NET Framework. This edition supports a maximum of 2GB of RAM and 2-way SMP.

- *Standard Edition* - Includes all features in the Web Edition plus support for a maximum 4GB of RAM and 4-way SMP.
- *Enterprise Edition* - Includes all features in Standard Edition plus 32GB of RAM, 8-way SMP and 8-node clustering. Server clusters are only available in Enterprise and Datacenter Editions.
- *Datacenter Edition* - Includes all features in Enterprise Edition plus 64GB of RAM and 64-way SMP.

Microsoft .NET Enterprise Servers

In order for Microsoft Windows "Application Server" to compare with Oracle9*i* "Application Server", we must include the Microsoft .NET Enterprise Servers. The .NET Enterprise Servers are individual Microsoft products:

- Application Center
- BizTalk Server
- Commerce Server
- Content Management Server
- Exchange Server
- Host Integration Server
- Internet Security and Acceleration Server.
- Operations Manager
- Microsoft Project Server
- Mobile Information Server
- Share Point Portal Server

Some of these .NET Enterprise Servers are under different versions (year). For example, BizTalk Server is 2002, but Application Center is 2000 but is supported by Windows 2003 and Windows 2000. Furthermore, some of these products are not yet built on the .NET Framework or ".NET Connected". In Oracle9*i*AS, Oracle provides one single integrated application server product. Oracle9*i*AS is available on Windows NT, 2000, XP and 2003 and supports the Windows 64 bit (IA64) environment. With Oracle's commitment to the Microsoft Windows platform, Oracle9*i*AS provides a choice for customers with Microsoft Windows. Beyond Microsoft Windows support, Oracle9*i*AS provides options in support of Linux on the Intel Platform.

ORACLE9*i*AS AND MICROSOFT .NET INTEROPERABILITY

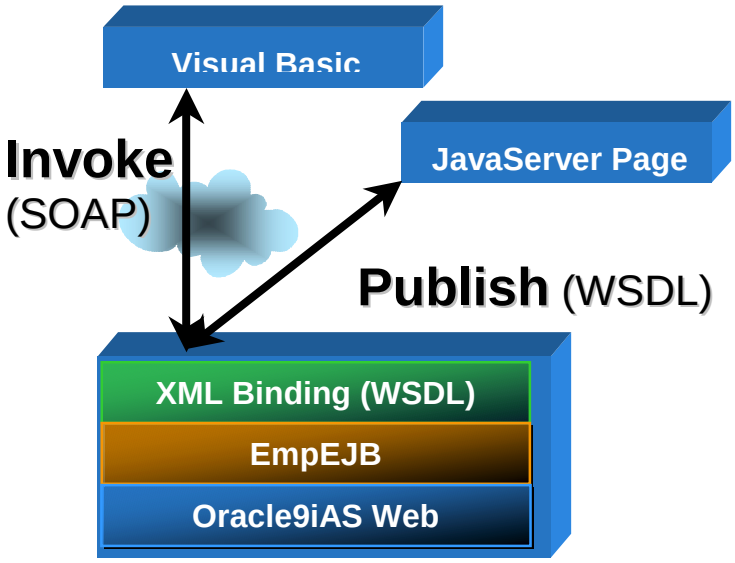
Many organizations will be faced with supporting both Microsoft .NET and J2EE environments simultaneously, either as a long-term co-existence strategy or a short-term transitional environment. Oracle9*i*AS is designed to integrate with the Microsoft .NET environment.

Development

Developers can use Oracle9*i*AS to build J2EE applications and then interoperate with existing and legacy Microsoft applications or recently developed Microsoft .NET applications. The fundamental interoperability technology surrounds the Web services open standards.

- *Microsoft SOAP Toolkit* – This allows interoperability with Microsoft COM applications, e.g. Microsoft Excel or Visual Basic.
- *XML Web Services* – This allows interoperability with Microsoft .NET Connected applications. If the application is written on the Microsoft .NET Framework, there is no need to use the SOAP Toolkit.

Let's examine the Oracle9*i*AS and Microsoft .NET interoperability in more detail. In this example, we have two applications built on Microsoft Visual Basic .NET and JavaServer Pages (JSP), respectively. JSP is part of the J2EE open standard. Both applications need to access the same business logic that is built on J2EE Enterprise JavaBeans (EJB).



Of course, JSP applications can interact with EJB, as this is part of the J2EE functionality. But how would the Visual Basic .NET application call the same EJB? The possibilities are:

- *Bridge and Wrappers* – One traditional answer is to use a bridge technology. Many vendors, including Oracle, offer bridge technology. Oracle9*i*AS includes a COM Bridge in the ASP Migration Kit. This is for customers who want to migrate from ASP to J2EE. The COM Bridge is provided as an interim step for migrated JSP applications to access existing COM objects. Microsoft provides bridges known as "COM Callable Wrappers" which expose .NET Framework objects to COM and "Runtime Callable Wrappers" which allow a .NET client, e.g. VB.NET, to call a COM object. Bridge and wrappers require marshalling between complex types, an expensive operation that can be costly to application performance.
- *J# (J-Sharp)* – Microsoft provides J#, which allows .NET applications like VB.NET to interact with Java. J# is based on an early version of the Java programming standard, JDK 1.1, and therefore not J2EE open standard compliant. Hence, J# cannot interact with EJBs without developers writing much of the custom code.
- *Web services* – Both Oracle and Microsoft embrace Web services as the open standards and key technology in interoperability. Web services provide a lightweight protocol in SOAP (Simple Object Access Protocol) and description in WSDL (Web Services Description Language) that is based on

XML. Oracle9*i*AS provides 100% Web services standards in the production release and continues to adapt to next generation Web services enhancements. Microsoft defines .NET as the XML Web Services platform.

Using our example, we examine how Web Services can simplify the interoperability picture and reduce development time, reuse existing components, and increase return on investment.

Oracle9*i*AS Containers for J2EE (OC4J) is the heart of the Oracle9*i* Application Server. OC4J provides a 100% J2EE compliant platform and allows developers to build and deploy compliant J2EE applications. Oracle9*i*AS supports both command line and GUI development environments. For GUI environments, Oracle offers Oracle9*i* JDeveloper, a pure Java IDE that includes the same OC4J included in Oracle9*i*AS. Developers can build, debug, and profile their applications using the same J2EE application server in development and deployment. Oracle9*i* JDeveloper is available on the Microsoft Windows platform.

When developers create J2EE components, OC4J automatically generates the Web services, e.g. WSDL and Java client stub. In our example, EmpEJB application serves as the business logic and reads the employee table in the Oracle9*i* Database. It retrieves the employee name that matches the department number entered by the application users either using VB.NET or JSP from browser like Internet Explorer.

Developers can select all or any one of the application interfaces to expose as Web services. In Oracle9*i* JDeveloper, a wizard is provided to make this selection simple. Once the selection is completed, the generations of the Web services are automatic, with no additional coding required. With WSDL, developers can publish it to UDDI (Universal Description, Discovery, Integration) repository or directly consume the WSDL using Microsoft Visual Studio .NET.

Because of Web services standards, VS.NET can parse WSDL generated by Oracle9*i*AS. Once VS.NET consumes the WSDL in the VB.NET application, a .NET stub is generated and developers can call the EmpEJB Web service from the VB.NET application.

In order to truly appreciate the power of Web services to simplifies interoperability, let's explore these examples:

- Visual Basic .NET calls J2EE (EJB)
- Java client calls ASP.NET with access to public Microsoft UDDI registry

- Microsoft SOAP Toolkit

Because this is an overview document, we will not show the complete code but rather the most important line(s) of code.

Visual Basic .NET calls J2EE (EJB)

The EmpEJB application has the getEmpNameList() EJB method that retrieves the EMP table (employee name) information from the database. A Java client application can call this EJB with the following Java code:

```
String[] nlist = empEJB.getEmpNameList("20");
```

This calls the getEmpNameList() EJB method with input of 20 which generates the following SQL statement

```
Select ename from emp where deptno = 20;
```

This returns the employee name(s) to the Java client application. To call this same EmpEJB with a Web service, developers can use the EmpEJB Java stub generated by OC4J. To call the Web service from the JSP application where application users can enter their value, the sample code is:

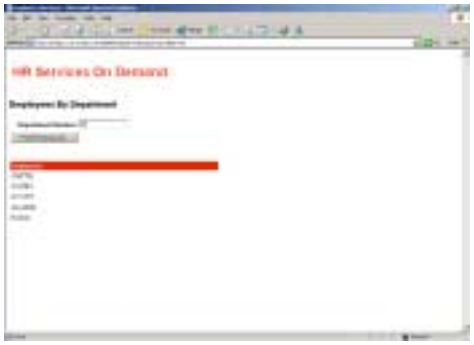
```
EmpEJBStub emp = new EmpEJBStub();
n = emp.getEmpNameList(deptno);
```

In review, OC4J generates the EmpEJB WSDL and Java stub. This allows the JSP application to call the EmpEJB web service using SOAP over HTTP. Because WSDL is part of the Web services standards, Visual Studio .NET can consume the EmpEJB WSDL. Once VS.NET registered the WSDL with getEmpNameList as the interface as a VS.NET web reference, VB.NET developers can call getEmpNameList.

Here is a sample VB.NET code:

```
Dim ws As New localhost.labEmpEJB()
ds = ws.getEmpNameList(TextBox1.Text)
```

In VB.NET application, the same EmpEJB method is available. You can see the end result from both applications calling the same back end business logic running on Oracle9*i*AS.



Java Client Calls ASP.NET with Access to Public Microsoft UDDI Registry

If you are building a .NET application and want to call it from J2EE, this can be accomplished using Web services similar to the previous example. With Oracle9*i* JDeveloper, developers can consume WSDL

generated by third party Web services tools or query Web services from a public UDDI registry including Microsoft UDDI. In the Oracle9*i* JDeveloper wizard, developers can search a UDDI registry by name or category. In this example, we will use the WeatherForecast Web Service available in the public Microsoft UDDI registry. Users enter a zip code (in the US) and the Web Service returns the weather forecast for that zip code.

Oracle9*i* JDeveloper can generate the WeatherForecast Web Service stub. This includes generating the client side stub, main Java method, and if necessary HTTP authentication code for proxy authentication. The Web Service Stub/Skeleton Wizard will generate three files. DailyForecast.java and WeatherForecast.java are used to serialize complex data types from the Web service and WeatherServicesStub.java is used to invoke the Web service itself.

The sample code in WeatherServicesStub.java to invoke the Web service is

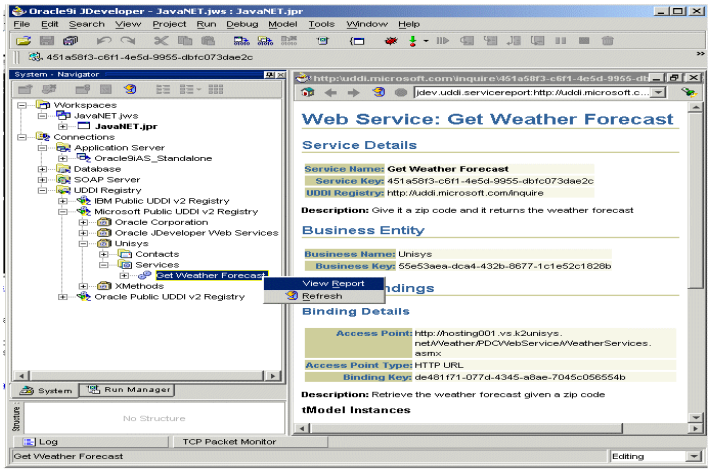
```
// Text output of forecast
System.out.println(stub.GetWeatherText("94065"));

// Object output of forecast
WeatherForecast wf = stub.GetWeather("94065");
System.out.println("Temperature is: " + wf.getCurrentTemp());
```

This example, “Automated Invocation of a .NET Web Service” is available in its entirety, including complete source code, on Oracle Technology Network (OTN). Developers can download this and other Web services examples and tutorials, complete release of Oracle9*i* JDeveloper and Oracle9*i*AS from OTN Web Services Center.

Microsoft SOAP Toolkit

In addition to .NET, developers can interoperate existing (legacy) Microsoft applications such as DNA, COM/COM+ without using expensive bridge or wrapper technology. With the Microsoft SOAP Toolkit, developers can call J2EE in similar manner as .NET. This is a huge savings of redevelopment and provides the ability to leverage open standards like J2EE.

Many customers and developers are using this interoperability to reduce their development time and therefore increase their return on investment. For example, a customer used Web services to transfer



information from Microsoft Content Management Server to Oracle9*i*AS J2EE, EJB application running on OC4J. Because SOAP supports HTTP and HTTPS, this customer can transfer this information in a SOAP packet between firewalls using a standard Web listening IP port. For security, this SOAP packet is encrypted with SSL.

Web Services Interoperability (WS-I)

So far we presented Oracle9*i*AS interoperability with Microsoft with different development methods. For example, with Web services, developers can use Visual Basic .NET applications and call J2EE functionalities. Also with Web Services, applications can receive and send messages between J2EE and MSMQ. Because Web services are open standards, Oracle and Microsoft applications can seamlessly interoperate. These have shown the capabilities of Web services but we are only at the basic building block for e-business applications. Web Services must also include important e-business functionalities such as the ability to build identity management with credential propagation and access list, and the ability to build business management processes and activities with orchestration and choreography.

To counter proprietary tendencies in Web services, the Web Services Interoperability Organization (WS-I) was created as a vendor-neutral body to certify interoperability between Web service implementations. Vendors like Oracle, IBM and Microsoft co-founded WS-I to ensure standards compliance in this area. WS-I is chartered to promote Web services interoperability across platforms, operating systems, and programming languages. The organization works across the industry and standards organizations to respond to customer needs by providing guidance, best practices, and resources for developing Web services solutions.

WS-I was formed specifically for the creation, promotion, and support of Generic Protocols for Interoperable exchange of messages between services. Generic Protocols are protocols that are independent of any specific action indicated by the message beyond actions necessary for the secure, reliable, or efficient delivery of messages; "Interoperable" means suitable for and capable of being implemented in a neutral manner on multiple operating systems and in multiple programming languages.

“A steadfast commitment to open standards is essential to unlocking the value of Web services. As a founding member of WS-I, Oracle will collaborate with other industry leaders to drive the development and adoption of Web services throughout the enterprise.” - Charles Rozwat, Executive Vice President, Server Technologies Division, Oracle Corporation

As a founding member of WS-I, Oracle is fully committed to the WS-I effort and ensures Oracle9*i*AS is fully compliant with WS-I. Oracle is leading the effort to create WS-I Profiles, which names groups of Web services specifications for interoperability purposes. These profiles enable the industry to define interoperability levels and how best to use combination of specifications together.

Deployment

Organizations require fast, scalable, easy to manage application servers to deploy internet applications and Web services. We already discussed Web services that interoperate with Microsoft environments. As an enterprise platform, Oracle9*i*AS provides complete interoperability with applications and products running on Microsoft platforms; reliably deploying internet applications such as Java/J2EE and non-Java applications such as Oracle9*i* Forms.

- *IIS Plug-in* – This is a reverse HTTP proxy that forwards incoming HTTP requests from IIS to an Oracle9*i*AS instance. Because Oracle9*i*AS provides this IIS plug-in, administrators can use the

Oracle9*i*AS Single Sign-On (SSO) Server to manage security and authentication. In a future release, Oracle9*i*AS will provide a native plug-in so administrators can use IIS as the Web server in Oracle9*i*AS environments.

- *Web Cache* – Oracle9*i*AS Web Cache increases the performance of Microsoft ASP applications while remaining transparent to the application Web server, IIS. IIS treats HTTP requests from Oracle9*i*AS Web Cache the same as any other HTTP request coming directly from the browser. In turn, IIS generates the response and sends it back to Oracle9*i*AS Web Cache as an HTTP message. Administrators can cluster Oracle9*i*AS Web Cache for high availability systems.
- *Connectivity* – Oracle9*i*AS provides data and application connectivity to Microsoft environments with standard J2EE in J2EE Database Connectivity (JDBC) and J2EE Connector Architecture (JCA). Incorporated in Oracle9*i*AS are adapters for connectivity, including connection pooling, from J2EE applications to SQL Server using JDBC.
- *Rapid Application Development* – Oracle9*i*AS Forms provides the ability to easily and rapidly build and deploy Web-based database applications in a declarative and productive manner. Web deployed Forms integrate with Microsoft Office such as Microsoft Word and Excel. For example, data can be queried by a Web deployed form and write the data to a tab delimited operating system file on the client. This file can be manipulated using Microsoft Excel.
- *Corporate Desktop* – Oracle9*i*AS Portal provides an extensible framework that integrates Web-based resources such as Web pages, applications, business intelligence reports, and syndicated content feeds, within standardized, reusable information components called portlets. Within a portlet, these resources are personalized and managed as a service of the Oracle9*i*AS Portal. The Portlet Development Kit (PDK) is provided as part of the Oracle9*i*AS Portal including capabilities to build portlets using Java. Many existing portlets are available in the Portal product and downloadable from the Oracle Technology Network (OTN). These portlets also provide interoperability with Microsoft.
    - *OmniPortlet and Web Clipping Portlet* –These capture and integrate data from multiple data sources including Web services, spreadsheets, XML and Web sites.
    - *Web Services Portlet* – This exposes Web services as portlets within Oracle9*i*AS Portal by enabling developers to easily create portlets from their own or existing Web services declaratively. Two popular styles of Web services are supported, controlling the look and feel of the portlet, as well as enabling automation of the Web Services Portlet.
    - *Exchange Portlet* – The Microsoft Exchange Portlet provides three portlets: the Inbox portlet, the Calendar portlet, and the Contacts portlet. Each of these portlets renders information associated with a user's Exchange mailbox. The Inbox portlet displays messages from the user's Exchange inbox. The Calendar portlet displays today's appointments and meetings. And the Contacts portlet displays the user's list of contacts.
- *WebDAV* – Using a WebDAV client, such as Windows Explorer, you can map Portal page groups to your desktop as Web Folders and seamlessly drag and drop content, files, and folders back and forth between Oracle9*i*AS Portal and your desktop.
- *Mobile and Wireless* – Oracle9*i*AS Wireless can wirelessly enable Microsoft applications including Exchange emails and seamlessly integrate content with Microsoft PocketPC devices. With Oracle9*i*

Lite, developers can build and deploy mobile applications on Windows CE and PPC and synchronize with Oracle Databases. Many adapters are included with Oracle9*i*AS Wireless to allow seamless integration with Microsoft environments.

- *Web Integration Adapter* – This retrieves and adapts Web content. The Web Integration Adapter works with Web Interface Definition Language (WIDL) files to map source content to Oracle9*i*AS Wireless XML. This allows developers to map Microsoft IIS sites content to WIDL and allow access from wireless devices.
- *SQL Adapter* – This allows services designers to create services based on SQL statements on Stored Procedures. Any database with a JDBC driver is supported including Microsoft SQL Server. The SQL Adapter uses a pool of database connections. The connection pool parameters can be specified as init arguments of the adapter.
- *HTTP Adapter* – This fetches the Oracle9*i*AS Wireless XML content from the external HTTP/HTTPS URLs. It acts as a proxy browser (which understands the Wireless XML) on behalf on the mobile device.

- *Directory* – Provides integration between Oracle Internet Directory, Microsoft Active Directory and Microsoft Windows NT Domain.
  - *Active Directory Connector* – In conjunction with the Oracle Directory Integration and Provisioning platform, this module synchronizes information between Microsoft Windows and Oracle Internet Directory.
  - *Oracle Directory Integration and Provisioning Platform* – This platform executes Microsoft Windows connector at specified intervals.
  - *Microsoft Windows Password Plug-ins* – This includes:
    - (1) Active Directory external password authentication Plug-in that supports Oracle9*i*AS components;
    - (2) Active Directory password modification Plug-in which enables Windows user password modification through the Oracle9*i*AS self-service administration interfaces;
    - (3) Oracle Internet Directory Password Synchronization Plug-in to generate password verifiers in Oracle Internet Directory where this plug-in module resides in Microsoft Windows, capturing the user password modification in Microsoft Windows and sending it to Oracle Internet Directory.
- *High Availability* – Integration with Microsoft Network Load Balancer (NLB) and Cluster Service (MSCS) and Oracle9*i*AS clustering provide maximum availability to critical systems. This offers seamless operations even at times of high demand. Systems then can be scaled in accordance with organizational growth. We will discuss high availability in more detail later in this paper.

Oracle9*i*AS provides portlets for fast development, IIS Plug-ins for easy to manage deployment, wireless for mobility, Web cache for scalability and adapters for interoperability with Microsoft environments. This allows organizations to leverage existing Microsoft investments without the heavy cost of conversion. At the same time, organizations can rapidly transition to an open standards environment in J2EE. This myriad of environments connects and operates seamlessly under Oracle9*i*AS, providing the most diverse user

environments with minimum administrative effort. Now that we have discussed how Oracle9*i*AS works with Microsoft, let's explore critical areas such as performance, availability, system management, security and directory of Oracle9*i*AS in more detail.

PERFORMANCE AND AVAILABILITY

Currently, internet applications are facing growing demands for complex e-business information from growing numbers of users. As internet applications are required to perform more computation under growing transactional loads, the performance of these applications declines sharply. Oracle9*i*AS includes key features to guarantee the performance and scalability of internet applications. Oracle's performance strategy is focused around the following objectives:

- *All Aspects of Performance* - To improve performance, Oracle focuses on all aspects of its Oracle9*i*AS runtime including its J2EE and Web Services code path and the network invocation costs (HTTP, RMI, RMI-over-IIOP, SOAP) to efficiently use in-process calls where appropriate for highly optimized local requests, to manipulate data, and for highly efficient transaction scheduling.
- *All Types of Hardware* - Since application servers are deployed on a variety of different kinds of hardware including 1-2 CPU commodity hardware and high-end SMP clusters, Oracle designed its application server to provide fast performance on both low-end hardware and high-end clusters.
- *Application Transparent* - To make applications run efficiently, Oracle ensures that applications require **no modification,** and reduces the number of parameters that administrators need to configure to make applications run efficiently. Oracle9*i*AS is designed in this fashion to ensure that static parameters, which are optimized for a specific user and transaction load, do not constantly need to be adjusted as loads on the system grows.
- *Caching to Further Improve Performance* - Finally, Oracle provides a number of different types of caching to improve application performance - (i) a Java Object Cache to cache frequently accessed Java objects and to cast SQL query results into Java objects to reduce type conversion overhead; and (ii) Web caching to cache static and dynamic Web content for performance including caching Microsoft ASP applications.

When evaluating application servers, two vital factors to consider are performance and price/performance. To measure J2EE performance, the Java community created ECperf benchmarks. Because Microsoft does not participate in the Java community, vendors like Oracle can only rely on reference implementations to measure application server performance on the Microsoft Windows platform.

ECperf

Until recently, the only reliable way to compare application server performance and cost-effectiveness was to conduct custom benchmark tests because no objective, accurate, and real-world benchmarks were available. ECperf™ is the first such benchmark broadly accepted as the means to estimate and compare application server performance and price. Microsoft does not participate in this open standard benchmark.

The ECperf benchmark has two metrics:

- **BBops/min** is the ECperf performance metric and denotes the average number of successful Benchmark Business operations per minute completed during the benchmark Measurement Interval.

- **$/BBops/min** is the ECperf price/performance metric and measures the price per business operation, calculated by dividing the total price of the test system in US$ by the BBops/min.

Oracle9*i*AS achieved the industry's best-recorded J2EE performance on the ECperf benchmark, beating the best results achieved by IBM and BEA significantly and also achieved the best result in the ECperf Price/Performance category. A number of performance enhancements such as code path optimizations, end-to-end application performance tuning, optimizations to the threading model, network and I/O model, event model, and various other elements to improve performance for both RPC and stream-based applications are responsible for making Oracle9*i*AS the fastest Java application server in the industry.

The other Oracle ECperf price/performance result, announced June 18, 2002, also shows Oracle9*i*AS to be the least expensive for a given performance. With such results on the ECperf benchmarks, Oracle9*i*AS has beaten both BEA WebLogic Server and IBM WebSphere Application Server.

Oracle9*i*AS - Best Overall Performance

Oracle's ECperf performance result shows Oracle9*i* Application Server to be the fastest application server to date. Oracle9*i*AS demonstrates significant high-end scalability and performance than BEA and IBM - Oracle9*i*AS' result of 51,007 BBops/min is 35% more than BEA WebLogic Server's best figure of 37,791 BBops/min and 56% more than IBM WebSphere Application Server's best of 32, 581 BBops/min.

Oracle9*i*AS - Best Price/Performance

Oracle also achieved the best-ever price/performance result, bettering BEA's and IBM's results. An interesting point in this benchmark is that although both Oracle's and BEA's price/performance figures appear as $7/BBops/min, Oracle actually achieved $6.07/BBops/min against $6.62/BBops/min for BEA. Both figures are reported as $7/BBops/min to conform to ECperf's rounding requirements.

In summary, Oracle's ECperf results show Oracle9*i*AS to significantly surpass BEA WebLogic Server and IBM WebSphere Application Server in performance; they also show that Oracle9*i*AS provides the best price/performance in the industry to date. With broad acceptance of ECperf as an objective and realistic measure of performance and price/performance of application servers, Oracle's ECperf results make Oracle9*i*AS the leader in both categories. As we mentioned, since Microsoft does not participate in the J2EE community and therefore the ECperf benchmark, there is no entry from Microsoft for comparison.

HIGH AVAILABILITY

Oracle9*i*AS High Availability with Microsoft Windows

Clusters enforce homogeneity between member instances so that a cluster of application server instances appears to function as a single virtual instance. With application servers, clustering provides two important benefits - (i) load balancing where the distribution of a workload is distributed in the most optimal fashion across a set of instances for efficient processing; and (ii) failover where requests can be handled by other instances in a cluster when a specific instance fails without interrupting the application.

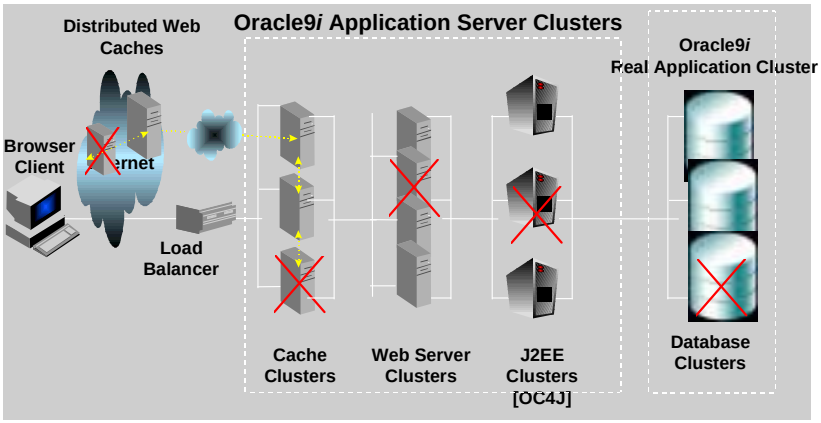
Oracle9*i*AS not only provides the industry's most productive development environment for internet applications but also provides a number of features that make it a mission-critical platform for highly

scalable, highly available internet applications. Oracle9*i* Application Server can support different middle-tier availability methods with Microsoft Windows:

- *Oracle9iAS Cluster* – Clustering with Oracle9*i*AS essentially means the use of a group of Oracle9*i*AS Servers that coordinate their actions in order to provide scalable, highly available services in a transparent manner. This runs on standard versions of Microsoft Windows including Microsoft 2000 Professional and Windows Server 2003 Web and Standard Editions.
- *Load Balancing* – Administrators can use Microsoft Network Load Balancer (NLB) instead of the hardware load balancer. NLB works with Microsoft Cluster Service and it does not require special hardware. This allows for not only load balancing but for failure detection and failover.
- *Microsoft Cluster Services* – The combination of Oracle9*i*AS clusters and Microsoft Cluster Service provides Windows customers with an easy-to-deploy disaster-tolerant high-availability solution that keeps business critical systems running with greater levels of availability and data protection.

Oracle9*i*AS Clustering

Application Server Clustering (not to be confused with Database Clustering) essentially means the use of a group of application servers that coordinate their actions in order to provide scalable, highly available services in a transparent manner.



Oracle9*i* Application Server is unique in providing end-to-end clustering. Examples of Oracle9*i*AS clustering on the middle-tier include:

- *Web Cache Clusters* - Oracle9*i* Application Server Web Cache can be deployed in a clustered environment, thereby increasing total cache capacity and application scalability and availability. This enables further cost savings as you can serve more content with the same hardware. To reduce the risk of denial-of-service attacks, and improving availability of dynamic/static content, Oracle9*i*AS Web Cache enables multiple cache instances to work together as a single logical cache.
- *Web Server Clusters* - Oracle9*i*AS Web Clusters enables HTTP processes to work in a cluster configuration to support automatic failover and efficient resource utilization.
- *J2EE and Web Services clusters* - Oracle9*i* Application Server enables the creation of J2EE "cluster islands" – a technology that is hardware and operating system independent – providing automatic

failover of both stateful and stateless J2EE applications and Web services. Administrators can easily scale hardware by upgrading to a new low-cost platform without rewriting their applications.

There is no hardware or special service required to use Oracle9*i*AS Cluster. Whereas other cluster technology requires Microsoft Windows 2000 Server or above, Oracle9*i*AS Cluster works with Microsoft Windows NT Workstation, Windows 2000 Professional and Windows 2003 Web and Standard Editions.

Load Balancing

Load distribution or load balancing means requests from clients can be distributed across multiple Oracle9*i*AS instances on a single CPU or on multiple CPUs. Oracle9*i*AS provides a number of advanced capabilities including:

- *Load Balancing at HTTP Server* - The Web server uses a simple but efficient mechanism to load balance between HTTP server processes within a single instance of the service. The master HTTP server process does not service client requests itself but spawns and monitors a group of child processes which take turns accepting HTTP requests from a shared socket by using a mutex. Once a child receives a request but before it begins servicing it, it releases the mutex, which can be acquired by another child. As a result, access to the socket is serialized but children may service requests in parallel. Additionally, Oracle HTTP servers can be run on multiple Microsoft nodes where client requests can be load balanced over the separate host instances using a variety of techniques including DNS round robin or a dedicated load balancer with Microsoft Network Load Balancer. Oracle HTTP Server load balancing does not require special hardware and is available on all editions of Microsoft Windows NT, 2000 and 2003.
- *Load Balancing at Oracle9iAS J2EE Container* - Servlet and EJB container instances load balance requests either across instances on a single node or across multiple nodes using a variety of load balancing algorithms. This includes support for RMI, RMI-over-IIOP, and SOAP requests.
- *Integration with Microsoft Network Load Balancer* - Oracle9*i*AS works with load balancing appliances - Cisco Local Director, BigIP, and Nortel - and Microsoft Network Load Balancer (NLB) for stateless and stateful load balancing. As NLB will be included with all editions of Microsoft Windows 2003, this provides administrators with a cost effective way to load balance requests across Oracle9*i*AS.
- *Connection Re-direction and Node Affinity* - Finally, for stateful applications, Oracle9*i*AS supports standard facilities such as cookies and dynamic URL-rewriting to bind and redirect clients to an existing session on a specific instance.

Microsoft Cluster Services

Oracle supports clustering not only at every layer in the middle-tier but also at the database level with Oracle Failsafe and Real Application Clusters, thereby providing the only true end-to-end clustering capability in the marketplace. With Oracle9*i*AS and Microsoft Cluster Services, administrators can provide cold or hot failover cluster.

- *Cold Failover* – The active node in the cluster enables a logical IP and maps it to the virtual name. Clients then access the application via the virtual name. When the application fails over to the second node, it enables the logical IP and maps the virtual name on the second node. Microsoft Cluster Services runs on top of this cluster setup and provides the monitoring facilities necessary to detect

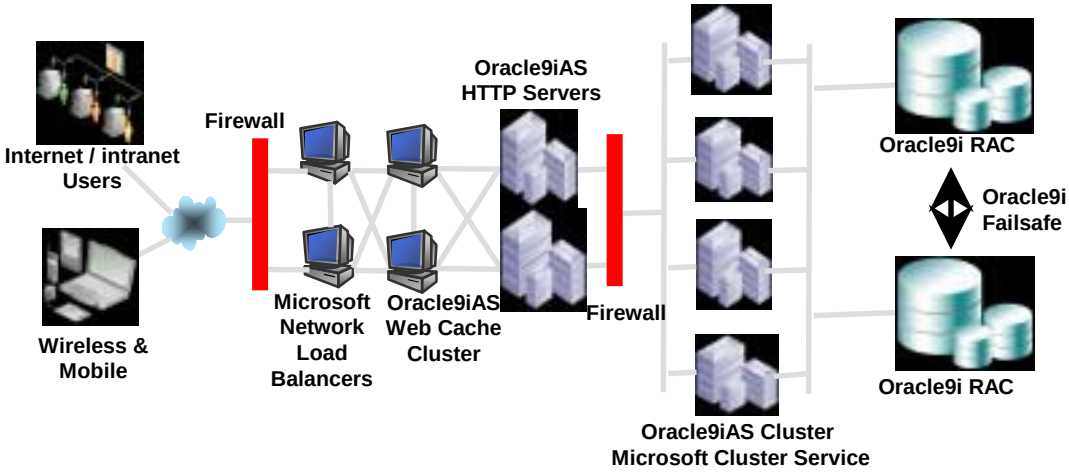
failures on the active node and take appropriate action to restart the failed services on the passive node.

- *Failsafe* – Oracle Failsafe works with Microsoft Cluster Server to ensure that if a failure occurs on one cluster node, then the Oracle database and Oracle9*i*AS running on that node will failover automatically and quickly to a surviving node.
- *Real Application Cluster (RAC)* – With combination of RAC and Microsoft Cluster Services, Oracle9*i*AS provides maximum availability by ensuring zero planned and unplanned downtime. Despite the failure of any node in the system, Oracle9*i*AS will continue to function and service client requests.

Maximum Availability Architecture on Microsoft Windows

Availability of an overall system or of an individual system component is defined as the percentage of time that it works normally or alternatively the mean-time-to-failure for the system. Maximum Availability Architecture (MAA) provides a simple, redundant and robust architecture that prevents different outages or recovers from an outage within a small mean time to recovery (MTTR). The goal is that most outages have no or minimal impact to availability while catastrophic outages can be repaired in short amount of time.

Oracle9*i*AS is designed to provide a highly available and highly reliable infrastructure to deploy applications. MAA is achieved on Microsoft Windows with the combination of Oracle9*i*AS Clusters, RAC, Failsafe, and Microsoft Network Load Balancer and Cluster Service. This diagram is an example of Maximum Availability Architecture in a Microsoft environment.



In this multi-tier cluster deployment example, there is no single point of failure. The previous diagram represents three tiers: De-Militarized Zone (DMZ), middle-tier, and data-tier. DMZ is the combination of Microsoft NLB, Oracle9*i*AS Web Cache and Oracle9*i*AS HTTP Server (OHS). Middle-tier is the Oracle9*i*AS Cluster and we will use Oracle Containers for J2EE (OC4J) in this example. The data-tier is the Oracle9*i* Databases.

DMZ

A single host in the NLB array listens to the virtual IP address. In the event of a failure, another member of the NLB array takes over. In addition to load balancing, this configuration provides failure detection and failover. NLB directs connections to any member of the Web Cache cluster. No changes are necessary to enable Web Cache with NLB.

In the Web Cache cluster, multiple instances of Web Cache operate as a single logical cache. By using NLB, it prevents Web Cache from being the single point of failure. Each cluster member can propagate object changes – including invalidations, destroys and replaces – through the cluster's messaging system to other cache in the cluster. This allows for the content to stay synchronized, without the overhead of centralized control. A clustered Web Cache can mask the failure of a member by either fetching the contents from another member or by going to the back-end server.

Oracle9*i*AS includes a lightweight process monitoring component in OPMN. OPMN uses a combination of ping, reverse ping and system specific process ID information to monitor a process. OPMN in all Oracle9*i*AS instances notify each other of the up/down status of Oracle9*i*AS components such as OC4J within their instance. For example any failures of an OHS can transparently failover by Web Cache. Using OPMN mechanism, Web Cache can ping OHS and make sure that the server is available. If Web Cache finds OHS to be unavailable, Web Cache removes that OHS instance from its routing table and then adds it back when it returns to service.

Middle-tier

OHS includes mod_oc4j which (1) identifies the requests it needs to act on, (2) determines which OC4J to route those requests to, and (3) communicates with that OC4J process. OPMN notifies local mod_oc4j of changes in the OC4J status on all machines within the cluster. This allows mod_oc4j to keep its routing table updated, without any intervention from administrator. Oracle9*i*AS Cluster replicates application state to all nodes in the cluster using Java Messaging Service (JMS) multicast topic. With OC4J Cluster, NLB and Web Cache could be configured without requiring session stickiness. NLB can route even stateful requests to any of the backend Oracle9*i*AS instances.

By using NLB with Oracle9*i*AS, administrators increase the availability of application servers used to handle incoming user requests. By adding Microsoft Cluster Service (MSCS) with Oracle9*i*AS, administrators have additional safeguards against system failure. Microsoft clustered servers are physically connected by cables and programmatically connected by MSCS. With Oracle9*i*AS, administrators have a choice of either Oracle Failsafe or Oracle Real Application Clusters (RAC) as the application repository and both work with MSCS to provide maximum availability.

Data-tier

Oracle9*i*AS with Oracle Failsafe provides a low-cost high availability solution on Microsoft Windows. A deployment example can be a single instance database that has two nodes managed by MSCS, one active and one passive. The Oracle software is installed on a private disk on each cluster node, whereas the data, control and log files are installed on cluster disks attached to the shared storage interconnected between the nodes. Effectively one of the nodes is in stand-by. This solution is the least expensive and provides fast failover.

Whereas Failsafe provides single instance failover, RAC uses two or more nodes, each running an Oracle instance that accesses a single database residing on MSCS. In a RAC environment, all active instances can

concurrently execute transactions against the shared database. RAC automatically coordinates each instance's access to the shared data to provide data consistency and data integrity. Oracle9*i*AS accesses a RAC database at the primary site in the database tier. If the primary site fails, then the server farm at the secondary site will be activated. Oracle9*i*AS directs all subsequent application requests to the secondary site.

This is only one MAA example with Oracle technologies on Microsoft Windows. Many industry experts believe that Microsoft Windows does not provide enterprise high availability and scalability. But with the combination of Oracle technologies including Oracle9*i*AS, Oracle9i RAC and Failsafe, Oracle can provide maximum availability on Microsoft Windows. There are other considerations that we have not discussed, including network infrastructure and operational best practices. For complete discussion on MAA, please download the MAA white paper from Oracle Technology Network (OTN).

SYSTEM MANAGEMENT

Oracle believes a well-integrated product, managed by a single tool embedded in the application server provides the lowest total cost of ownership. Oracle Enterprise Manager provides a comprehensive systems management platform for managing Oracle9*i*AS and Oracle9*i* Database. It combines a graphical console, Oracle Management Servers, Oracle Intelligent Agents, common services, and administrative tools. By integrating all components with Oracle Enterprise Manager, Oracle9*i*AS streamlines and significantly lowers management costs. There is no need to purchase separate products to manage and deploy applications.

With Oracle9iAS, administrators can use a Web-based Oracle Enterprise Manager for:

- *Comprehensive Monitoring* - Oracle Enterprise Manager provides a comprehensive set of facilities to monitor the status (up/down) of Oracle9*i*AS facilities; to measure their resource utilization; to diagnose faults; and to monitor the performance/throughput of Oracle9*i*AS facilities.
- *Systems Configuration and Management* - Based on information gleaned from monitoring the systems, Oracle Enterprise Manager provides administrators with facilities to centrally control the configuration (and re-configuration) and administration of one or more Oracle9*i*AS instances.
- *Application Management* - Finally, Oracle Enterprise Manager provides facilities to deploy, track, and administer J2EE Applications and Web services.

By making the management facilities accessible from a HTML browser, administrators can mange Oracle9*i*AS either locally or remotely (for instance when out –of –the office), including through firewalls. Furthermore administrators can manage from any Microsoft Windows environment with a browser, e.g. Internet Explorer. Neither additional software nor a special version of Microsoft Windows is necessary

SECURITY AND DIRECTORY

The Oracle9*i* Application Server provides a comprehensive, standard, and extensible set of security services for deploying business applications on the Web, including single sign-on (SSO), directory, and Java security services. These services allow users to integrate custom business logic and third party products with Oracle Web applications and tools in a single security framework.

Single Sign-On

Oracle9*i*AS Single Sign-On (SSO) addresses the problem of 'too many passwords.' Oracle9*i*AS Single Sign-On resolves this problem by enabling authentication to all appropriate applications in an enterprise by entering a user name and password only once and by providing centralized administration of user name and password combinations for all users in an enterprise.

Delegate Authentication to Oracle9*i*AS SSO

If administrators want to continue using Microsoft IIS as the Web server, administrators can delegate authentication to Oracle9*i*AS SSO using the Oracle9*i*AS IIS Single Sign-On plug-in. This plug-in (a single DLL) is included in Oracle9*i*AS. IIS administrators can install and configure this plug-in using IIS Management Console. With the Oracle SSO plug-in, users can authenticate only once to the Web server; thereafter, the user name and password are relayed invisibly to Oracle9*i*AS applications.

Oracle Internet Directory

Oracle Internet Directory (OID) is a critical component of Oracle9*i*AS management and security infrastructure. It provides facilities to centrally manage user accounts, their access control privileges, and groups. OID provides the following key directory features:

- Native LDAP v3 server supporting all LDAP2000-compliant RFCs, including LDAP v2 and v3 RFCs
- Supports the X.500 information, naming, and storage model
- Extensible directory schema for online modifications with no downtime
- LDAP developer APIs in Java, C, and PL/SQL to assist with application development

Microsoft Active Directory

If your enterprise stores and manages user security credentials in Microsoft Windows, then you can use Oracle Internet Directory Active Directory plug-ins to enable users to (1) authenticate themselves to Oracle components and (2) modify their passwords. Administrators do not need to store the credentials in Oracle Internet Directory and then worry about keeping them synchronized.

The Active Directory plug-ins use this process to authenticate a user to an Oracle component or change user password when the user security credentials are stored and managed in Microsoft Windows.

1. A user seeks to authenticate to an Oracle component, or to modify the user password.
2. Depending on the request, the Oracle Internet Directory launches either the Active Directory authentication plug-in or the Active Directory password modification plug-in and passes the user request to it.
3. The Active Directory plug-in passes the request to Microsoft Windows where the user authentication or modification occurs.
4. The Active Directory plug-in receives the results from Microsoft Windows.
5. The Active Directory plug-in sends the result to the Oracle Internet Directory.
6. The Oracle Internet Directory returns the result to the user.

The Oracle Internet Directory Active Directory plug-ins is integrated in the Oracle9*i*AS product.

CONCLUSION

Oracle9*i*AS provides compelling features for co-existence and interoperability with Microsoft Windows. This allows organizations to leverage existing Microsoft investments without the heavy cost of conversion. In summary, Oracle9*i*AS provides:

- Interoperability with XML Web services and SOAP Toolkit. There are many examples:
  - Visual Basic .NET calling J2EE EJB
  - Java Client calling ASP.NET
  - Legacy Microsoft applications with J2EE using Microsoft SOAP Toolkit
  - Interoperability with Microsoft UDDI
- Co-existence and integration with existing Microsoft applications using:
  - IIS Proxy and SSO Plug-ins
  - Web Cache
  - Connectivity
  - Oracle9*i*AS Forms
  - Corporate Desktop with Portal
  - WebDAV
  - Mobile and Wireless
  - Security and Directory

In addition, Oracle9*i*AS provides Maximum Availability Architecture with Microsoft Network Load Balancer and Cluster Services on Microsoft Windows. MAA embraces high availability so that any failure is handled transparently or with a thorough, automated recovery procedure that can achieve a low MTTR.

Oracle offers Oracle9*i* Application Server, a100% standards-compliant application server to simplify how internet applications are developed; to make internet applications fast, reliable, manageable, and secure; and to substantially lower the Total Cost of Ownership associated with developing, deploying and operating internet applications.

Oracle9i Application Server on Microsoft Windows
April 2003
Author: Mason and Christina Ng
Contributing Authors: Rakesh Dhoopar

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com