Building and Deploying an Enterprise-Quality Management Solution for Web Services

Technical Case Study

Published: May 2004

Line-of-business applications have often been designed and built in an isolated and monolithic manner making it difficult and time consuming to share information between applications. Microsoft recognized the need for a Web services management solution that would make it easier to develop, deploy and manage Web services for accessing all of Microsoft's internal line-of-business applications.

Situation

Line-of-business systems have often been designed and built in an isolated and monolithic manner, making it difficult, time consuming and costly to share business information in an organization. Microsoft recognized the need for an enterprise-wide Web services management solution that would make it easier to develop, deploy and manage Web services.

Solution

The Microsoft Information Technology group developed and deployed an internal Web services development and management solution to provide a secure, production quality, enterprise infrastructure for managing Web services across the line-of-business applications at Microsoft.

Benefits

- Seamless and easy integration of application data and services that reduced integration costs by as much as 4:1.
- Ten-fold reduction in deployment and operations management costs by deploying fewer server hardware and software resources.
- High availability and faster responses to service issues through proactive management of service level agreements.

Products & Technologies

- Microsoft Windows® Server 2003
- Microsoft SQL Server™ 2000
- Microsoft Visual Studio® .NET 2003
- Microsoft .NET Framework 1.1
- Web Services Enhancements (WSE) for Microsoft .NET

This technical case study is written for IT architects, developers and support professionals interested in the design, deployment and management of an enterprise-wide Web services solution. This case study will also be of interest to line-of-business application owners and analysts needing to understand the issues involved in adopting a services-oriented architecture (SOA) in their organization.

Background

Over a period of more than four years, the Microsoft Information Technology group (Microsoft IT) gained valuable experience using Web services to integrate line-of-business application services and data.

Prior to the development of the current version of its internal Web services development and management solution (referred to as the "Alchemy" project), Microsoft began developing Web services using the basic Web services standards (e.g. SOAP, WSDL, XSD and UDDI). The initial goal was to enable easy integration of business data from four core line-of-business applications at Microsoft:

- Siebel for customer relationship management,
- Clarify for product service request management,
- MS Sales for sales revenue reporting, and
- World-Wide Sales and Marketing Database for managing e-mail newsletter subscriptions, event invitations and diverse marketing campaigns.

Microsoft IT's early Web services supported read and update access to several business applications. Microsoft client and server-based applications used these Web services as the Microsoft IT standard for accessing line-of-business application services and data. For more information on Microsoft's early experiences deploying Web services including Microsoft's

first internal Web services application, “Account Explorer”, refer to the resources at the end of this case study.

Situation

The basic Web services standards (SOAP, WSDL, XSD and UDDI) provide the functionality necessary to build and connect distributed applications. However, by themselves, the basic standards are not sufficient for supporting a production quality, enterprise environment for the development, deployment and management of Web services. The second release of Microsoft’s internal Web services development and management solution addressed this situation in addition to the following issues:

- Lack of common development standards
- No common deployment and change management processes
- No common process for managing deployment and change
- Requirement for a Web services operations management infrastructure

Microsoft IT decided that an enterprise-class service-oriented architecture based on Web services would address these issues. The Web services development and management solution added the deployment, management and support features needed by line-of-business developers, operations staff and line-of-business application owners.

Lack of Common Development Standards

Microsoft IT’s early experience developing and deploying Web services was a common one: each line-of-business application support group created and deployed Web services for their own particular application. Microsoft IT found that when they relied on the basic Web services standards, there were too many options available for developers to perform user and process authentication, digital signing and encryption of service requests and responses, formatting of parameter data and handling error conditions. This resulted in incompatible security solutions, duplicated development effort, inconsistent management processes and unpredictable service levels. When only basic Web services standards were used without a common, integrated management solution, the outcome was incompatible Web service “silos” and the full benefit of Web services was not being realized.

No Common Deployment and Change Management Processes

The early deployments of Web services at Microsoft were made on an application-by-application basis. This created a Web services environment that was difficult to manage from both an operations and change management perspective.

It was not easy for a group of developers to take a Web service from development through testing to production. The development, testing and deployment processes placed a heavy burden on the server infrastructure and personnel who manually deployed and configured the Web services. Additional servers were required which resulted in increased hardware and operations management costs. Furthermore, manual installation and configuration of application software produced inconsistencies that were difficult to detect, correct and manage.

No Infrastructure for Operations Management

It is essential that an enterprise computing environment include an integrated approach for operations management and enforcement of service-level agreements (SLA); including the

need to support monitoring, logging, analyzing, and reporting on all Web service transactions (Web service requests and responses) and issuing alerts for out-of-bound SLA conditions.

Without an adequate infrastructure for Web service operations management, Microsoft IT found that it was impossible for Web service owners to provide predictable service levels.

Lack of Automation to Support Web Service Subscriptions and Service Level Agreements

Based on Microsoft IT's experience deploying Web services and having multiple development projects consume these services, they discovered a number of additional requirements. One of these included the need for a formal process by which Web service providers are notified of an application developer's intent to subscribe to a Web service. Subscribing to a Web service establishes the relationship between Web service consumer and Web service owner. The subscription process enables Web service owners to receive notification of a consumer's intent to use a particular Web service and for the appropriate SLA parameters to be established between the two entities. The subscription process requires an exchange of consumer and owner contact information so that two-way notification of pending changes, outages and out-of-range SLA alerts can occur.

Solution

To address the issues found in its original deployment of Web services, Microsoft IT developed and deployed an internal Web services development and management solution. This Web services development and management solution provided a secure, consistent, enterprise-wide infrastructure for developing, deploying and managing Web services across a diverse range of Microsoft's internal business applications.

The goals for the first release of Microsoft's internal Web services development and management solution were to:

- Create a comprehensive Web services management solution using Microsoft technologies.
- Provide a reusable, agile and global Web services framework that supported all Microsoft line-of-business applications.
- Make it easier for development teams to build and consume Web services in a line-of-business application environment.
- Reduce the tendency for developers to create stand-alone Web services that use different technology solutions for authentication, digital signing and encryption, formatting of parameter data and handling error conditions.
- Establish a single comprehensive repository for Web services that enabled the categorization, discovery and re-use of those Web services.
- Reduce development and deployment complexities as well as overhead costs when providing operational support for Web services.

Understanding the Service-oriented Architecture Model

In a service-oriented architecture (SOA), applications are structured as a fabric of interacting and cooperating services. Each service is designed to be discoverable and accessible in a standard manner. Microsoft IT found that software designers who use an SOA model are more able to solve business system development problems in a cost-effective, standards-based way. Developers were able to address system interoperability and data integration

requirements more quickly. Microsoft IT wanted to move beyond traditional object-oriented component-based designs to a model based on common business services that can be easily combined to create new services and end-user applications.

Microsoft observed that applications are expanding beyond the scope of a single system running on a single computer bounded by a single organization. Web services were developed and deployed as discrete units of application logic exposing message-based interfaces that can be accessed across a network. As a result, Microsoft IT expected that a service-oriented architecture built on industry standards and specifications would enable applications to interoperate more easily and new solutions to be created more quickly.

To successfully deploy and maintain a services-oriented architecture, Microsoft found it necessary to deploy an enterprise-quality Web services management solution.

Microsoft's Web Services Development and Management Solution Overview

Microsoft IT's internal Web services development and management solution simplifies the discovery, security, management, notification and analysis tasks required when using Web services in a production environment. The solution incorporates a distributed design philosophy that works in concert with a centralized management system. The solution design can be divided into two main subsystems: the "Alchemy Backend" and the "Alchemy Interface".

The "Alchemy Backend" provides services for data management, transaction history storage, service registration and execution verification. It also handles the interaction and storage of Web services configuration data including: system users, roles and operational metrics. An identical "Alchemy Interface" runs in-process in the consuming application and the Web service provider. The "Alchemy Interface" handles all incoming and outgoing SOAP request/response transactions.

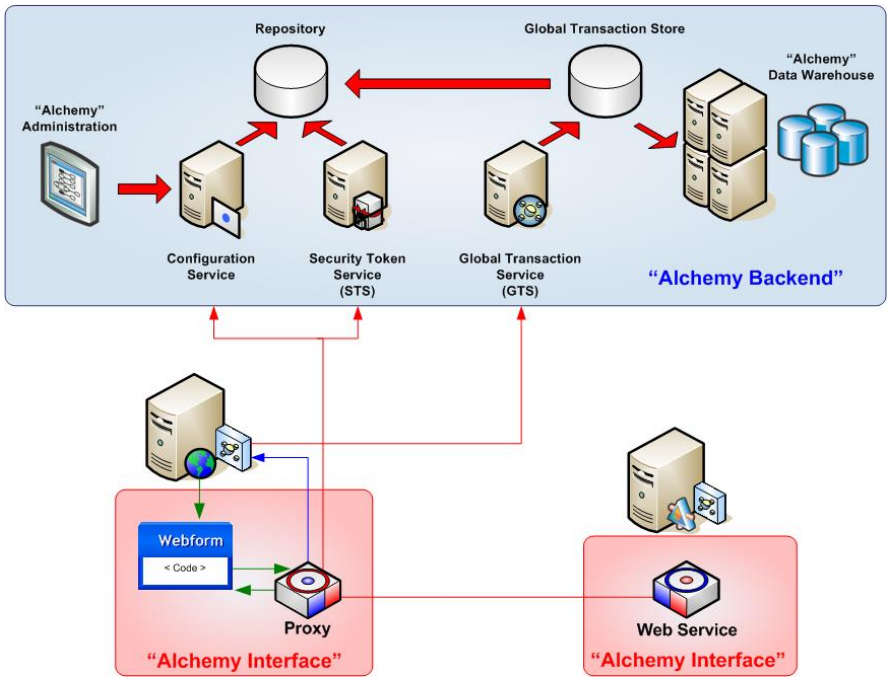

Figure 1. Microsoft IT Web Services Development and Management Solution

Microsoft IT's Web services development and management solution, detailed in Figure 1, is comprised of two subsystems:

- An instance of the "Alchemy Interface" which runs in-process in each Web services consumer and provider.
- The single "Alchemy Backend" which provides management services for all Web service calls exchanged between to "Alchemy Interface" endpoints.

By default, interaction with the "Alchemy Interface" is abstracted entirely from the host application. Microsoft IT's implementation of the "Alchemy Interface" was based on Microsoft Web Services Enhancements (WSE) for Microsoft .NET.

Web Services Enhancements (WSE) for Microsoft .NET

Web Services Enhancements for Microsoft .NET (WSE) is supported as an add-on to Microsoft Visual Studio .NET and the Microsoft .NET Framework. It provides developers with the latest capabilities for developing Web services using advanced Web services specifications. WSE provides customers who are early adopters of the advanced Web services technology with supported implementations of selected evolving Web services specifications.

WSE provides support for security features such as digital signatures and encryption as well as the ability to use binary message attachments. In addition, WSE provides a policy framework and message-based programming model along with support for multiple host application environments. These capabilities were provided using the WS-Security, WS-Policy, WS-SecurityPolicy, WS-Trust, WS-SecureConversation and WS-Addressing specifications. WSE simplifies Web service development by making it easier for developers and administrators to implement Web services based on these specifications. Web service requests and responses can be signed and encrypted using Kerberos tickets, X.509 certificates, username/password credentials, and other custom binary and XML-based security tokens.

Microsoft IT chose to implement its internal Web services development and management solution using WSE for a number of reasons:

- It is the most efficient way to implement standards-based Web services.
- WSE is compatible with Microsoft's long-term vision for distributed application development using a services-oriented architecture.
- Out-of-the-box, WSE will interoperate with other platforms that implement the advanced Web services specifications.

By adopting WSE, Microsoft's Web services development and management solution is well positioned to interoperate with present-day as well as future Web services standards-based environments.

For more information on Web Services Enhancements (WSE) for Microsoft .NET, refer to the resources at the end of this case study.

Message Processing

Microsoft IT's internal Web services development and management solution uses a pipeline of filters to process inbound and outbound SOAP messages. Some filters add headers to outgoing SOAP messages while other filters read and check the validity of headers on inbound messages. Filters can also be used to transform the body of a SOAP message.

The active message processing components in the “Alchemy Interface” are implemented as WSE input and output filters. The processing of a Web services message is depicted in Figure 2.

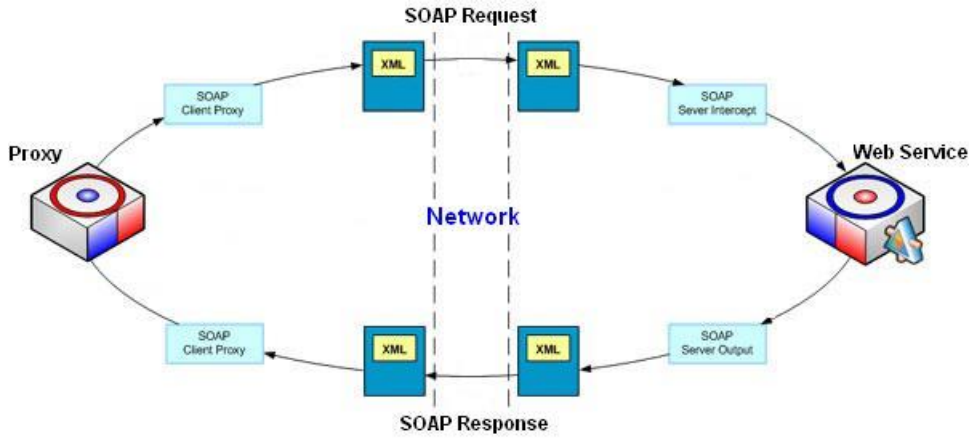

Figure 2. Message Processing

The “Alchemy Interface” handles all interactions with the “Alchemy Backend”. This includes communication with the “Alchemy Backend” configuration service, security token service (STS) and global transaction store (GTS). These services provide SLA information and tracking mechanisms, centralized authentication and metadata services, and global logging of transactions.

To optimize performance, each Web service consumer and provider log transactions to a local transaction store (LTS). The LTS batches transactions for periodic shipment to the GTS for final archiving, SLA monitoring, analysis, reporting and the processing of alerts. SQL Server 2000 Notification Services are used to trigger the creation of alerts.

In addition, the “Alchemy Interface” issues queries to the solution’s UDDI service to retrieve Web service entity and category information. The UDDI service uses an internally developed Microsoft taxonomy to organize, categorize and support searches for Web services. Each business service is also broken down into subcategories based on service priority, operating environment (development, test and production), geographic location of the Web service, required response time and other keyword and SLA related parameters.

Key Features and Capabilities

Microsoft IT’s internal Web services development and management solution provides the following capabilities:

- A standard development framework
- Access rights management
- Service Level Agreement (SLA) management
- Dynamic service selection
- Centralized management
- Monitoring and reporting systems
- Subscription automation and workflow capabilities

Standard Development Framework

Microsoft IT's internal Web services development and management solution provides solution developers and operations staff with a consistent framework for building, deploying, maintaining and managing Web services. Web service developers use a consistent framework and API for creating and maintaining Web services. The solution abstracts away most configuration settings as administrative parameters that can be configured centrally. These include settings for authentication, digital signing and encryption and messages. A common set of administrative tools are used to deploy Web services in development, test and production environments.

Microsoft IT's Web services development and management solution reduces the time and effort required to develop, test and deploy new Web services. Based on Microsoft's IT experience with several projects, a Web services implementation that would typically require four to six months of effort is reduced to four to six weeks of effort.

By using a Web services development and management solution, Microsoft avoided the creation of Web services "silos" that would otherwise require their own individual development, deployment and management infrastructures. Furthermore, the group reduced the server hardware and software resource requirements, as well as the amount of manual effort necessary to deploy and manage Web services at Microsoft.

Access Rights Management

Microsoft IT's Web services development and management solution was designed with a specific focus on security. In particular, it provides access rights management mechanisms to control call level access to particular resources. This is accomplished through policies configured on the central server and distributed to clients configured to access these resources. Checks are made by both the Web services consumer and provider to ensure that all principals involved in the transaction have the required access rights and adhere to the policy established for the specific service.

Multiple authentication types are validated locally and verified centrally. The "Alchemy Interface" provides support NTLM, Kerberos, Microsoft Passport, and X509 certificate based authentication. Data such as user roles, transport options, and SLA parameters are returned after the identities of the Web services consumer and provider have been successfully authenticated.

Service Level Agreement Management

Every Web services resource has an SLA associated with it. An SLA is comprised a specific values for service level parameters such as response time, time-out, execution environment (development, test and production), priority and keywords. These service level parameters enable a single application to service the needs of many consumers in varying ways without the need for additional software configuration or hardware.

Dynamic Service Selection

The "Alchemy Interface" determines the specific provider that a Web service request will be directed to. This is accomplished without requiring any host application involvement. In order to service the client's request most appropriately, SLAs are queried and matched against service provider capabilities. These selection options ensure that the most appropriate service will be used to respond to Web service requests from each Web services consumer.

Centralized Management

The Microsoft IT Web services development and management development team designed the configuration service and Web services filter to support centralized configuration of its Web services. This provides a single point of control for making changes to a Web service configuration. The use of a centralized configuration service results in improved accuracy, decreased operations staff, and increased uptime. Overall, this enabled Microsoft IT to reduce system support costs and the staff needed to manage the Web services environment.

Monitoring and Reporting

With Microsoft IT's internal Web services development and management solution, Web service transactions are automatically logged in the local transaction store and then periodically sent in batches to update a global transaction store. This enables the monitoring, analysis, reporting and alerting capabilities required for operations management and enforcement of service-level agreements (SLA).

Microsoft IT's Web services development and management solution provides Web service owners with automatic notifications enabling them with the ability to proactively manage out-of-range SLA conditions such as poor response times and "service down" situations.

Subscription Automation and Workflow

Microsoft IT included an automated subscription workflow process in the latest release of its internal Web services development and management solution so that developers can register their intent to use an existing Web service. The subscription process includes the exchange of contact information that informs a Web services owner about the consumers of their Web services and the applicable set of service level parameters. Both providers and consumers have the option of receiving an email alert or an Instant Message when a Web service was operating outside its SLA.

Microsoft IT found that reducing the complexity developers experienced discovering and subscribing to Web services (including the exchange of provider and consumer contact information and SLA parameters) was a key factor in encouraging broad adoption of enterprise-wide Web services.

Development and Deployment

The development team consists of one program manager (contributing 80% of his time), four testers (100%) and 3 software developers (two at 100% and one at 50%) working for four months to build, test and deploy each release of Microsoft IT's internal Web services development and management solution.

Microsoft IT's Web services development and management solution is used to support most of the company's Web services. This includes the Web services published by eight line-of-business applications which are, in turn, consumed by 15 other applications. Approximately 120,000 transactions are processed and logged per day (3.6 million per month). The average CPU and memory utilization across all of the eight "Alchemy Backend" servers is 7% and 18%, respectively.

Benefits

The Web services development and management solution provided both Microsoft IT and Microsoft business divisions with significant benefits based on the deployment of a consistent Web services development and management framework:

- Seamless and easy integration of services from existing line-of-business applications
- Cost savings from deploying fewer server hardware and software resources
- Accelerated adoption and re-use of Web services
- Proactive management of service level agreements for Web services

Seamless and Easy Integration

Organizations need to be more agile and responsive than ever before, and Microsoft is no exception. Companies require an application architecture that enables the rapid implementation of services that can be quickly recombined to create new customer management, sales, marketing and product development solutions.

With Microsoft IT's internal Web services development and management solution, a typical Web service can be built and deployed in four to six weeks compared to the four to six month period required to deliver equivalent capabilities using only basic Web services standards. In one specific project, Microsoft IT was able to reduce their custom development costs by almost 20%.

Reduced Server Deployment and Associated Costs

Microsoft IT's early deployments of Web services required a separate server environment to be deployed to provide basic Web service support for each line-of-business application.

The Web services development and management solution made it possible for Microsoft IT to deploy and manage several Web services on a single server as well as small, medium and large server farm configurations. Reducing the need for server hardware and software also reduced the overall number of staff and cost required to manage server operations. Ultimately, two full-time operations staff members were required for the entire enterprise where previously one person was required, on average, to support the Web services for each line-of-business application.

From a Web services management perspective, all Web services deployed at Microsoft are supported by a single network of eight "Alchemy Backend" servers. To provide the same level of redundancy and performance on an application-by-application basis would require 80 servers, a tenfold increase.

Fewer resources and better management practices increased uptime for business users while reducing costs.

Accelerated Adoption and Re-use of Web Services

Automated subscription processes for managing Web services reduced costs and encouraged fast adoption and re-use of common business services. At Microsoft, automation reduced the process time from one week to one day for subscribing to a Web service.

Proactive Management of Service Level Agreements

Microsoft IT's internal Web services development and management solution enabled proactive operations management and practical enforcement of service-level agreements by providing standard monitoring, logging, analysis, reporting and alerting functions. These functions provided Web service providers and consumers with the tools needed to monitor, measure and manage out-of-bounds service levels such as poor response times or "server down" situations. These capabilities are mandatory for the deployment of a production quality, enterprise-wide management solution for Web services.

Lessons Learned

Adopt a Services-Oriented Architecture

In a service-oriented architecture, applications are structured as interacting and cooperating services. Microsoft IT found that by adopting an SOA model, they were able to respond to business changes more quickly and with more predictable results.

A service-oriented architecture helps organizations steer clear of building monolithic applications that are resistant to change, difficult to integrate with other applications and frequently contain duplicate and incompatible functionality.

Microsoft IT also found that a production quality Web services management solution is a pre-requisite for the successful deployment and maintenance of a services-oriented architecture.

Value of a Governance Body

It is necessary to adhere to Web service standards and specifications when building a robust, yet flexible, service-oriented architecture. Microsoft IT is establishing an internal Web services governance body that includes representatives from the central IT group as well as key line-of-business application owners. The role of this body is liaison between business groups and the central IT organization. It also plans to take responsibility for IT related business strategy, technical strategy and architecture as it relates to the design, deployment and management of Web services.

Use Readily Available Tools and Frameworks

Readily available tools and frameworks enable rapid development of standards-based Web services. These include Microsoft Visual Studio® .NET, Microsoft .NET Framework, the Web Services Enhancements (WSE) for Microsoft .NET and Microsoft® Windows® Server 2003.

Based on their previous Web services development experience, Microsoft IT was able to deliver its Web services development and management solution in a relatively short four-month timeframe using the development technologies that are available to Microsoft's customers and partners.

Line-of-business Application Owners Should Own Their Web Services

Within the context of the strategy, architecture and guidelines established by the Web services governance body, let each line-of-business application owner be responsible for building Web services for their application. In a majority of situations, they are the most knowledgeable in terms of how specific line-of-business application functionality and data can be re-used through Web services. Deploy and manage all Web services using a common, production quality Web services management solution.

Future Directions

In the next release, Microsoft IT's internal Web services development and management solution will include support for the Dynamic Systems Initiative (DSI), the Microsoft Enterprise Instrumentation Framework (EIF), and Web Services for Management eXtensions (WMX).

Dynamic Systems Initiative

DSI brings application development and IT operations closer together to cut costs by automating the monitoring and management of business systems. DSI will provide

distributed computing management to Web services supported by Microsoft IT's internal Web services development and management solution.

Microsoft's internal Web services management solution will implement DSI with support for the first two server components of the DSI architecture: Microsoft's System Management Server (SMS) 2003 and Microsoft Operations Manager (MOM) 2004. DSI-aware Web services will understand Systems Definition Model (SDM) information provided by Visual Studio .NET 2004 applications when this new version of Visual Studio .NET is released.

Microsoft Enterprise Instrumentation Framework

EIF enables enterprise applications built on the .NET Framework to be instrumented for manageability in a production environment. This framework provides an extensible event schema and unified API which leverages existing event handling, logging and tracing mechanisms built into Windows, including WMI, the Windows Event Log, and Windows Event Tracing. An application instrumented with this framework can publish a broad spectrum of information such as errors, warnings, audits, diagnostic events, and business-specific events. In addition, the Enterprise Instrumentation Framework enables tracing by business-process or application service and can provide statistics such as average execution time for a given process or service.

Microsoft IT's internal Web services development and management solution currently supports user-defined instrumentation within a Web service. A SOAP transaction delivers instrumentation data to the Global Transaction Store (GTS). The next release of of Microsoft's internal Web services management solution will increase the amount of available instrumentation data by implementing support for the EIF schema and API.

Web Services for Management eXtensions

The next release of Microsoft IT's internal Web services development and management solution will also standardize its wire-level management protocols by adopting the Distributed Management Task Force (DMTF) WMX protocol specifications. WMX provides a framework for end-to-end server management. WMX is a general-purpose SOAP-based systems management protocol layered over common Web services specifications.

WMX is based on a small number of fixed operations designed to support common systems management tasks. The WMX protocol is compliant with modern XML, SOAP, and Web services specifications but does not introduce any new concepts with respect to these specifications. Rather, WMX combines existing building blocks to support management operations**.** The protocol and operating environment are simple enough to be implemented on small devices as well as by major operating systems.

WMX is a part of the WS-* family of Web services specifications. Microsoft IT's internal Web services development and management solution currently uses Web Services Enhancements for Microsoft .NET (WSE) implementations of the WS-Security, WS-Policy, WS-SecurityPolicy, WS-Trust, WS-SecureConversation and WS-Addressing Web services specifications. WMX extends WSE to support WS-Transfer, WS-Enumeration, WS-Eventing and WS-ShellExecute.

Conclusion

Use of Microsoft IT's internal Web services development and management solution continues to grow across Microsoft enabling a more agile, service-oriented architecture. More than 15

groups are building, deploying and consuming Web services based on the new Web services development and management solution. Business capabilities are increasing as development, integration and operations management costs continue to shrink.

Development of the next release of Microsoft IT's Web services development and management solution is underway using WSE. Key features of the next release of the solution include:

- Faster Web service development using a point-and-click development model
- Tight integration with Active Directory security groups (to replace native "Alchemy Backend" security groups)
- More robust management services including support for DSI, EIF and WMX

Microsoft IT's Web services development and management solution adheres to the Web services standards and specifications implemented in the Microsoft .NET Framework and the Web Services Enhancements for Microsoft .NET. In this way, Microsoft IT was able to build and deploy the Web services development and management solution with current technologies and, at the same time, be well positioned to interoperate future Web services standards-based environments.

For More Information

Microsoft Web Services Development and Management Solution White Papers

- Microsoft Corporation, "*Alchemy" Deployment*, http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=13739, June 2003.
- Microsoft Corporation, "*Account Explorer"*, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/msit/crm/actexpcs.asp, January 2002.

Web Services

- Microsoft's Web Services Development Center, http://msdn.microsoft.com/webservices.
- Microsoft Advanced Web Services Web site, http://msdn.microsoft.com/webservices/understanding/advancedwebservices/.
- Web Services Enhancements (WSE) for Microsoft .NET Web site, http://msdn.microsoft.com/webservices/building/wse/.
- Tim Ewald, *Inside the Web Services Enhancements Pipeline*, Microsoft Corporation, http://msdn.microsoft.com/library/en-us/dnwse/html/insidewsepipe.asp, December 2002.

Microsoft Dynamic Systems Initiative

- Microsoft Dynamic Services Initiative, http://www.microsoft.com/dsi.
- Microsoft Enterprise Instrumentation Framework, http://msdn.microsoft.com/vstudio/productinfo/enterprise/eif/.

Additional Information

For more information about Microsoft products or services, call the Microsoft Sales Information Center at (800) 426-9400. In Canada, call the Microsoft Canada information

Centre at (800) 563-9048. Outside the 50 United States and Canada, please contact your local Microsoft subsidiary. To access information via the World Wide Web, go to:

http://www.microsoft.com/

http://www.microsoft.com/technet/itshowcase

For any questions, comments, or suggestions on this document, or to obtain additional information about Microsoft IT Showcase, please send e-mail to:

showcase@microsoft.com

© 2004 Microsoft Corporation. All rights reserved.

This case study is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. Microsoft, Active Directory, Visual Studio and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.