Chapter No. 1
"Lotus Notes 8.5.3 and SOA"
In this package, you will find:
A Biography of the authors of the book
A preview chapter from the book, Chapter NO.1 "Lotus Notes 8.5.3 and SOA"
A synopsis of the book’s content
Information on where to buy this book

About the Authors

**Tim Speed** is an IBM Certified Systems Architect with IBM Software Services for Lotus. In that capacity, he is responsible for designing, implementing, and supporting various engagements with its clients. Mr. Speed lives in Texas and has been an IBM/Lotus employee for over 16 years in a variety of networking, technical, hardware, and software support and consulting positions. He has been working with Notes for over 20 years focusing on administration roles and infrastructure. He also has international experience with working on infrastructure engagements in Spain, Japan, Hong Kong, Singapore, Malaysia, the UK, and Indonesia.

For More Information:
Knowledge is based on many different facets—what you know, knowing where information can be found, and who you know. The information in this book is a combination of all these facets. Data sources have been referenced in this book; these include references to people, URLs, and other books. But much of the knowledge that is in this book comes from very smart people. Not all the people listed in this acknowledgment participated in the writing of this book, but have influenced and guided me in my life that has culminated in this work. First and foremost, I need to thank my wife for helping me with the book and providing some of the editing throughout the various chapters. Next, I want to thank Johnny and Katherine for tolerating me during the months that I worked on this book. Next, I want to thank my mother, Lillian Speed, for teaching me to "think big". Thanks to Ed Speed for the inspiration to keep publishing. Thanks to Packt, in particular Kerry George, for their hard work in getting this book published.

For More Information:
Thanks to all the co-authors – you ALL did a great job!

Special thanks to Lotus/IBM (and ISSL), Martin Crotty, Mark J. Guerinot, Larry Berthelsen, Chris Cotton, Steven Stansel, Mark Steinborn, and Bob McDonald for their assistance in getting this book published. Special thanks to Victor Ross for his review. Many thanks to Brent Peters for writing the foreword to this book. Special thanks to the following: Gail Pilgrim, Jason Erickson, Jeff Jablonowski, John Allessio, Boris Vishnevsky, Adam Hanna, Brad Schauf, Scott Souder, David Byrd, Paul Raymond, David Little, Craig Levine, Mark Harper, Jeff Pinkston, Jordi Riera, Dave Erickson, David Bell, Mark Leaser, Dr. Fred Dahm, Gary Wood, John Kistler, Luc Groleau, Michael Dennehy, Robert Thietje, Francois Nasser, Kim Artlip, Marlene Botter, Mike Dudding, Stephen Cooke, Don Nadel, Tom Agoston, Carl Baumann, the very brilliant Dr Seshagiri Rao, Alistair Rennie, Tim Cardwell, Andrea Waugh Metzger, Barry Rosen, Bennie Gibson, Beth Anne Collopy, Keith Attenborough, Bill Hume, Brent A. Peters, Ivan Dell’Era, Carlos Gonzale, Charles K. DeLone, Don Bunch, the great Chuck Stauber, David R. Hinkle, Doug Parham, Kelly Ryan, the very smart Frederic Dahm, Gary Ernst, Gary Desmarais, Hassan C Waheed, Ian Reid, Jay Cousineau, Jayasree Gautam, Anthony (Joey) Bernal, Cynthia Oehmig, Dr. John Lamb, and special thanks to another very smart dude – "John Norton"; also thanks to Steve Mark, Joseph Anderson, Joyce Cymerman, Katherine Holden, Kathleen Kulkoski, Kevin Lynch, Michael Dudding, Lauri Jones, Marc Galeazza, Marco M.Noel, Mark Leaser, Marlene Botter, Mary Ellen Zurko, Naemi Engler, John Munnel, Paul Raymond, Peter Burkhardt, Robert Thietje, Sherry Price, Stephen Hardison, Tony Cusato, Steve Matrullo, Steven J Amadril, Terry Fouchey, Victor Ross, and William Destache.

Barry Rosen is an IBM IT Certified Managing Consultant with IBM Software Services for Lotus. During the last six years, he has worked on several global messaging and migration projects as well as performed Domino upgrades, messaging assessments, and client deployments. Currently, Mr. Rosen has been heavily involved in LotusLive Notes cloud migrations and hybrid deployments. Prior to IBM Software Services for Lotus, he was a Software Engineer in Lotus Support for over five years. While in support, Mr. Rosen was on several teams specializing in mail routing, Lotus Notes Client, calendaring and scheduling, and server core. Mr. Rosen is an SME on clustering, Lotus Notes for the Macintosh, and rooms and resources. Mr. Rosen is also a published author with several articles and books written on Lotus products.

For More Information:
There are so many people that I want to thank for their help. First, I would like to thank Packt Publishing, for publishing this book. Thank you to my wonderful family. Thanks to IBM where I have been allowed to grow personally and professionally. My manager, Martin Crotty, and John Kistler for approving this book, and all of their support. My co-authors who have poured countless hours and energy into making this book a reality. Tim Speed, without whom this book would not exist. Tim, it seems like yesterday we were eating lunch in Rockefeller Plaza brainstorming on this book. Thanks for your patience and gentle nudging, you have helped me grow personally and professionally beyond my own expectations. My Mother, Father, brother, and grandparents. To all of my colleagues and friends: Mark Guerinot, Chris Cotton, David Bell, Andrea Waugh-Metzger, Sherry Price, Doug Parham, Bob Thietje, Larry Berthelson, John Norton, Marc Hendricks, Jim Price, Yasmin Al-Shibib, Marc Allan, Matt Stien, Michael Granit, Stephen Rafoul, Seth Berk, Rob Buchwald, Aaron Greenberg, and Scott Sapire. Stella you too.

Scott O’Keefe is an IBM Advisory Software Engineer and the project lead for the Domino Configuration Tuner. He joined IBM via Iris Associates in 1999 and has been a part of the Domino Administration team since 2001. In addition to DCT, Scott works on Domino monitoring, system administration template development and LotusLive Notes.

I'd like to thank Tim Speed and Barry Rosen for the opportunity to work on this book. I'd also like to thank all of my Domino Administration buddies past and present for being great colleagues, mentors, friends, and inspirations.

My portion of this book is dedicated to my wife Tricia, and my children Kasey, Matthew and Molly. The best part of my day – every day – is the part that I get to spend with you.

For More Information:
IBM Lotus Notes and Domino 8.5.3: 
Upgrader's Guide

If you're reading this book, you're probably already familiar with Lotus Notes and the Domino server. You know about the powerful productivity features offered by this product and you know how much your company relies on it to communicate, collaborate, and manage its collective store of corporate knowledge. This book is intended to show you the new features of Lotus Notes and Domino 8.5.3. These incredible products keep evolving and growing with each release. This exciting new release will help your end users with new features; but it will also help the administrator with new management features. This book has been written by Notes/Domino "insiders". Collectively, we possess decades of Notes/Domino experience; we've been with the product since Notes 1.0, and since then have worked directly with customers to help them with their Notes/Domino upgrade and deployment issues.

This practical tutorial walks through the new features of the Lotus Notes/Domino 8.5.3 suite and documents technical features in a descriptive way, with examples and useful screenshots. The book also discusses likely problems you might face while upgrading, and shows you how to get the most out of the exciting new features.

What This Book Covers

This book will help you understand the new features in Notes client user interface, Domino 8.5.3, and the concepts of a Service Oriented Architecture (SOA) and how Lotus Notes 8.5.3 fits into SOA.

Chapter 1, Lotus Notes 8.5.3 and SOA, gives a high-level understanding of SOA, what it is, its value, and its characteristics. You will also learn how Lotus Notes 8.5.3 includes many of the characteristics of SOA components, and how it can help you assemble applications that can play a role in an SOA.

Chapter 2, Overview of New Lotus Notes 8.5.3 Client Features, includes a deeper dive into the new Lotus Notes 8.5.3 components and some of the new LotusLive features that have been included in this release (actually, the LotusLive features were included in 8.5.2)

Chapter 3, Productivity Tools, provides an overview of three productivity tools—Lotus Symphony Documents, Lotus Symphony Presentations, and Lotus Symphony Spreadsheets. You will see how these tools are integrated with Notes 8.5.3, and how they can now be controlled by Domino policy documents.

For More Information: 
Chapter 4, *Lotus Domino 8.5.3 Server Features*, reviews the major new and enhanced feature areas in Domino Server 8.5.3. This includes end user and messaging enhancements, administrator enhancements, performance enhancements, directory and security enhancements, and better integration with other IBM technologies.

In Chapter 5, *Deployment Enhancements in Notes/Domino 8.5.3*, we examine important Notes/Domino 8.5.3 features that can make rolling out your new deployment significantly easier. We discuss client provisioning, including Eclipse-based client and server provisioning functionality. We also look at policy enhancements and the new database redirect feature.

Chapter 6, *Domino 8.5.3 Enhancements*, looks at Domino 8.5.3 core features such as Domino Attachment and Object Service, or DAOS, ID Vault, Notes Shared Login, and auto-populated groups. We also cover iNotes up to 8.5.3. The differences and similarities between web and full Lotus Notes clients are reviewed.

Chapter 7, *Upgrading to Lotus Notes and Domino 8.5.3*, is divided into two main sections. The first takes a look at the Notes/Domino upgrade process in general, discussing concepts and steps that should be considered whenever you upgrade to any major release of Notes/Domino. The second section covers upgrade issues that are specific to Notes/Domino 8.5.3.

In Chapter 8, *Coexistence between Notes/Domino Releases*, we examine coexistence issues involved with running Notes/Domino 8.5.3 in a mixed environment with one or more previous releases. Also provides a discussion of Domino 8.5.3 server coexistence, including features such as Domino Directory, ODS, Domino Web Access, DDM, and ID files. LotusLive Notes hybrid requirements are also discussed.

In Chapter 9, *New features in Notes/Domino 8.5.3 Development*, we review some of the major new features and enhancements that affect Notes/Domino 8.5.3 application development. These include enhancements related to composite applications, Domino Designer 8.5.3, formula language and LotusScript, Lotus Component Designer, Web 2.0, and Lotus Expeditor.
Chapter 10, *Integration with Other Lotus/IBM Products*, discusses add-on products for a typical Notes/Domino infrastructure. The specific products covered in this chapter are the most common that you might encounter, including Lotus Quickr, Lotus Sametime, and Lotus Connections. Also, this chapter includes a short introduction to some of the IBM learning tools.

*Chapter 11, Domino Configuration Tuner*, gives you a special treat. Several years ago IBM released a tool known as Domino Configuration Tuner (DCT). When this tool was first released, it only included a small set of "testing" rules and server checks. The DCT tool is now a full enterprise tool that will help you analyze your Domino server environment.
Lotus Notes 8.5.3 and SOA

Service-Oriented Architecture (SOA) provides for a set of resources that are linked together on demand. This demand access can be from other systems, and/or users, or even applications that link resources together via a set of standards. Lotus Notes 8.5.3 is built on IBM's release of the Eclipse Rich Client Platform (RCP). As Lotus Notes 8.5 was built based on a set of standard components, it was a part of an SOA. This provides a more extensible server-managed client.

In this chapter, we will introduce the concept of SOA and how Lotus Notes 8.5.3 fits into one such architecture. We will explain what an SOA is, its value, and its characteristics. We will also cover how Lotus Notes 8.5.3 has many characteristics of SOA components, and how it can help you assemble applications that can play a role in SOA.

In this chapter, we will cover the following topics:

- What is an SOA?
- Why SOAs now?
- The SOA lifecycle
- How Lotus Notes 8.5.3 works with SOAs

What is an SOA?

Although the concept of SOA is simple, the components that make up SOA can be complex. Additionally, the value and perception of SOA varies with the perspective and role of organizations and people considering SOA. In this chapter, we will start with the basic definition of SOA and then examine specific aspects of SOAs.

The definition of SOA can often be taken further by adding the word "business", as the inherent value comes from business orientation and enablement.

For More Information:

In computing, the term SOA expresses a software architectural concept that defines the use of services to support the requirements of software users. In an SOA environment, nodes in a network make resources available to other participants in the network as independent services which they access in a standardized way. Most definitions of SOA identify the use of web services (using SOAP and WSDL) in its implementation.

SOAs can be used to:

- Build distributed systems that deliver application functionality as services to either end-user applications or other users
- Design and implement distributed systems that allow a tight correlation between the business model and its IT implementation
- Manage services made available by different software packages for re-use and reconfiguration

These uses of SOA highlight the fact that they encompass a wide range of interests.

There are many ways to implement and view an SOA. The specific approach and value proposition depends on the needs of the business and the role of the organization or person considering the SOA. In this chapter, we will focus on how businesses can re-use existing Lotus Notes-based functions, and how to take existing Eclipse and WebSphere Portal services and incorporate them into new Lotus Notes functions.

**The characteristics of an SOA**

There are some commonly understood characteristics of an SOA. These include the following:

- Services are re-usable and called by many applications
- Service access is with communication protocols rather than direct calls
- Services are loosely coupled so that they are autonomous
- Interfaces are defined in a platform-independent manner
- Services are encapsulated so that the interface doesn't reveal how the service was implemented (this is called abstraction)
- Services share a formal contract
- Services are composable (able to be assembled into composite applications)
- Services are stateless
- Services are discoverable

For More Information:
The following screenshot shows how to add an assembled composite application into Lotus Notes Designer 8.5.3:

Later in this chapter, we will examine some of these characteristics and see how Lotus Notes 8.5.3 can interact with an SOA.

**Perspectives on SOAs**

From a business perspective, SOA is about identifying, surfacing, and integrating business services to meet business needs.

From an IT perspective, SOA is about responding quickly to changing business needs. IT organizations must determine what style, patterns, or principles provide architecture capable of responding in a timely fashion. These questions must be answered with the understanding that existing applications and systems have been built over time and are hardwired together.

**Why SOAs now?**

One key factor in the emergence and success of SOAs is the evolution of standards. Standardization has made SOAs more useful now than ever before. In the past, companies have made numerous attempts to develop a standard to support some versions of SOAs. Standards such as CORBA and DCOM have existed for a while, but have not been widely adopted to allow true interconnection of companies and people.

For More Information:
Thanks to the Internet and standards such as HTML and HTTP, companies and customers are linked together as never before. This linkage is the key to the interconnection and combination of services that distinguish an SOA. As the Internet has matured, web service standards have emerged; they now have a common set of standards across vendors and businesses. Major vendors have agreed on standardization of web services and have incorporated these standardized services into products, providing an unprecedented breadth of tools for supporting an SOA. Standards for interoperability that have been widely adopted include the following:

- Hypertext Transfer protocol (HTTP)
- Extensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description, Discovery, and Integration (UDDI)
- OASIS standards, such as Open Document Format (ODF)

ODF is an open XML-based document file format for office applications, which can be used for documents that include spreadsheets, text, and rich text, along with chart types.

This particular standard was developed by the Organization for the Advancement of Structured Information Standards (OASIS) consortium and based on the XML format originally created and implemented by the OpenOffice.org office suite.

Other factors play key roles in the adoption of SOAs. For example, mature software and software frameworks are now available across a breadth of vendors, including Eclipse and OSGi. SOA-related governance models and best practices are defined and proven. With the development of the Internet, implementation is now practical, and business/IT collaboration is receiving renewed focus.

The SOA lifecycle

There are four distinct phases to the lifecycle of an SOA. These are as follows:

- The Model phase
- The Assemble phase
- The Deploy phase
- The Manage phase

For More Information:
This lifecycle provides a framework within which an SOA can be built. However, businesses and IT organizations can choose the place within the lifecycle from where to begin the SOA implementation. (One of the key values of SOA is the ability to get quick benefits by assembling and deploying services without waiting for a full-blown SOA definition.)

The Model phase

The Model phase of the SOA lifecycle starts with discovering which program assets can be re-used in new applications. You can discover these hidden assets and determine which programs are good candidates for re-use in web applications with a number of tools already in the market.

As we stated earlier, the key value of an SOA is the surfacing of business services. So, to properly identify the business services and understand how they fit into the business, SOA modeling establishes a common understanding of the business processes, objectives, and outcomes between business and IT. The SOA model helps to make sure that any IT application meets the needs of the business and provides a baseline for business service performance.

At the end of the Model phase, you should have a clear inventory of assets showing where they can be used in the business processes that you have modeled.

The Assemble phase

The Assemble phase is where programs are wrapped as services and used to create composite applications, which bring together core assets that often span multiple platforms. If you use legacy host transactional environments, the tools simplify the development of new web user interfaces, traditional terminal interfaces, and backend business logic.

During the Assemble phase, you can create services out of existing assets such as Enterprise Resource Planning (ERP) and financial systems, legacy host applications, and other solutions that are currently running your business. If no functionality exists, you can create and test a service to deliver the functionality required for your business process. Once the required services are available, you can orchestrate them so as to implement your business process.

Lotus Notes 8.5.3 includes features to support the Assemble phase of SOA development. We will review those capabilities later in this chapter.

For More Information:

The Deploy phase

During the Deploy phase, you can configure and scale the runtime environment to meet the service levels required by your business process. You can optimize the services environment to reliably run mission-critical business processes while providing the flexibility to make updates dynamically in response to changing business requirements.

Once it is configured, you can deploy your business process into a robust, scalable, and secure services environment. This service-oriented approach can reduce the cost and complexity associated with maintaining numerous point-to-point integrations.

The Manage phase

The Manage phase involves managing the underlying service assets, and establishing and maintaining service availability and response times, along with managing and maintaining version control over the services that make up your business processes. The management phase ultimately enables you to make better business decisions sooner than previously possible.

You can monitor key performance indicators in real time to get the information required to prevent, isolate, diagnose, and fix problems, enabling you to provide feedback to the business process model to enable continuous improvement.

Once the SOA has been deployed, you'll need to continue to secure, manage, and monitor the composite applications and underlying resources, from both an IT and a business perspective, so as to get full value from the SOA. Information gathered during the Manage phase on key SOA indicators can provide real-time insight into business processes, enabling you to make better business decisions, and feeding information back into the SOA lifecycle for continuous process improvement.

How Lotus Notes 8.5.3 works with SOAs

Now that we've covered the basics of SOAs, it is time to examine how Lotus Notes 8.5.3 fits in. Lotus Notes can help an organization achieve target architecture requirements with SOA characteristics by:

- Supporting service re-use. Lotus Notes 8.5.3 does this by providing a composite application development capability, and by providing web service consumer and producer capability.
- Enabling further extension of Lotus Notes to work with SOAs through an open technologies framework.

For More Information:  
Chapter 1

Composite applications

Lotus Notes 8.5.3 has the ability to assemble composite applications. This ability is useful in the Assemble phase of the SOA lifecycle.

A composite application is a loosely coupled collection of user-facing components brought together for a specific business purpose. Composite applications provide the frontend of an SOA. The ability to create and edit composite applications lets you easily combine and re-use different services, providing a tremendous platform for service re-use—a key characteristic of an SOA. With Lotus Notes 8.5.3, server-managed, NSF-based composite applications can be created or edited. An NSF-based composite application can consist of NSF, Eclipse, and WebSphere Portal components.

Elimination of information and service "silos" is a key benefit of composite applications for end users. With composite applications, the services are loosely coupled and independent (not hardwired into the infrastructure), so they can be easily re-used or extended, as business needs change. This is an important element in enabling a business to respond flexibly to business changes and to alter application interactions as needs dictate. Available online or offline, composite applications can facilitate self-service activities. Using the Composite Application Editor within the Lotus Notes 8.5.3 software, end users and LOB managers can create composite applications. IT staff can use their current development skills to build and modify reusable components, helping to reduce IT and development costs. Organizations can re-use previously developed Eclipse technology-based components within the composite applications experienced by Lotus Notes and Domino 8 software, helping to increase return on investment in application development tools and skills.

The Composite Application Editor is an install-time option of the Lotus Notes 8.5.3 client. Composite applications can be built with minimal or no NSF design changes to re-use existing Lotus Notes applications as components of the composite application.

Development responsibilities for building composite applications can be distributed across several types of application development and administration team members. The process does not have to be restricted to a highly skilled component developer. The roles in composite application development typically include the following:

- A component developer who designs and creates NSF and Eclipse components
- An application assembler who defines and assembles the composite application, and who may be a business user
- An application administrator who deploys portlets onto the WebSphere portal and maintains NSF-based composite applications on the Domino server

For More Information:
Lotus Notes 8.5.3 and SOA

The following section provides an example of how Lotus Notes 8.5.3 enables composite application assembly. The documentation and files necessary to build this example can be found by visiting http://www-03.ibm.com/developerworks/blogs/page/CompApps?entry=more_sophisticated_tutorial_of_composite.

The first component of this example is Lotus Notes contacts.

The following screenshot shows the Contacts view of Lotus Notes. On its own, this component shows a list of contacts and has a preview pane showing the details for the currently selected contact. Certainly, this is a valuable service by itself. But our example shows how you can re-use this service, combine it with other services, and extend the value of this component:

The second component of this example is a Lotus Notes discussion application that is shown in the following screenshot. This component allows users to discuss topics (in this case, Lotus Notes Designer) in a user discussion forum setting. Again, this on its own is a useful component, but our example will combine this component with Lotus Notes contacts to create a new, more useful service by loosely coupling these components:

For More Information:

Lotus Notes 8.5.3 and SOA

In the first step in this example, the Lotus Notes 8.5.3 Designer uses **Web Services Description Language (WSDL)** to expose properties and actions needed to navigate within the Lotus Notes forum application. The following screenshot is from the Lotus Notes 8.5.3 Designer where the final step of the WSDL creation is being completed:

In addition to using WSDL, the Lotus Notes 8.5.3 Designer also allows you to create actions to be implemented during the assembly of a composite application. In this case, the action is called **SelectPerson**, as shown in the following screenshot:

---

For More Information:
This action will select the forum entry for the current selection in the Lotus Notes contacts component, as shown in the following screenshot:

For More Information:
With the appropriate WSDL action associated with the Lotus Notes Forum application, the prerequisites for assembling the composite application are in place. In the following screenshot, the Composite Application Editor is used to wire the Lotus Notes **Contacts** view and the Notes forum application:

![Composite Application Editor Screenshot](image)

The result is a composite application in which the Notes Forum entry is displayed based on the selected Lotus Notes contact. As you can see in the following screenshot, **Betty Zechman** of **ZetaBank** is the currently selected Lotus Notes contact and the Lotus Notes Forum has been advanced to show the threads for **Betty Zechman**.

For More Information:
Next, the Composite Application Editor is used to include an Eclipse tag cloud in the composite application. This component displays tag data about the current contact. This is accomplished by linking the Eclipse component to the Lotus Notes Forum application.
Lotus Notes 8.5.3 and SOA

In the following screenshot, the current Lotus Notes contact selection is Betty Zechman and the interest selected is Applications Development. The result is a view of the Notes forum positioned in a thread by Betty Zechman related to Applications Development:

![Screenshot](image)

This simple example shows the power of Lotus Notes 8 Composite Application Editor. Minimal Lotus Notes designer effort, combined with use of the Lotus Notes 8 client Composite Application Editor, can produce a new business function re-using existing heterogeneous services.

For More Information:

Lotus Notes (version 8 and 8.5.3) and web services

Web service producer and consumer capability is not new to Lotus Notes 8 and Notes 8.5.3. However, the ability to produce and consume web services is a key characteristic of SOA. This section provides an overview of how Lotus Notes supports web service production and consumption.

A web service provider makes available a WSDL document that defines the service interface. The WSDL document is in XML format. What happens behind the interface is up to the provider, but most providers map the interface to procedure calls in a supported programming language. Incoming requests from a consumer are passed to the underlying code, and results are passed back to the consumer.

Lotus Domino maps the WSDL interface to an agent-like web service design element that can be coded in LotusScript or Java. The web service must be on a Domino server with HTTP enabled. (We can test the web service through an HTTP session in the Notes client preview.) Access is through one of the following Domino URL commands:

- **OpenWebService**: This invokes the web service in response to a SOAP-encoded message, sent through an HTTP POST request.
  An HTTP GET request (for example, a browser query) returns the name of the service and its operations.
- **WSDL**: This returns the WSDL document in response to an HTTP GET request.

Several approaches can be used to create a web service design element in Domino Designer. One approach is to code the service entirely in LotusScript or Java. In this case, saving the design element generates a WSDL document that reflects the LotusScript or Java code. Alternatively, an existing WSDL document can be imported. In this case, the LotusScript or Java code reflects the operations in the imported WSDL document. The web service design element saves the WSDL document as well as the code. If the public interface has not changed, the WSDL document stays as it is. If anything in the coding that affects the public interface is changed, a new WSDL document is generated.

For More Information:
Lotus Notes 8.5.3 and SOA

In Domino Designer, the web service design element resides below **Agents** under the **Shared Code** element, as shown in the following screenshot:

![Web Service Design Element in Domino Designer](image)

The Web Services design window looks a lot like the Agent design window. Clicking on the **New Web Service** button creates a new web service. Double-clicking on the name of an existing web service opens it for editing.

An example of a web service to access Domino databases can be found by visiting the following URL:

http://www-10.lotus.com/ldd/pfwiki.nsf/dx/ibm-using-web-services-to-access-a-domino-discussion-database

**Lotus Notes 8.5.x and open technologies**

Lotus Notes 8.5.x supports both OASIS/ODF and Eclipse open formats.

**OASIS/ODF**

Lotus Notes 8.5.3 supports both OASIS/ODF and Eclipse open formats. Word processing, spreadsheet, and presentation applications are basic, standard tools that many business users need and use on a daily basis. The Lotus Notes 8 and 8.5.3 product suite includes a suite of office productivity tools that allows end users to create, edit, and collaborate on a wide variety of file types. The Lotus Notes 8.x product suite is packaged with IBM productivity tools that support OASIS. Open Document Format (ODF) is an international standard for saving and sharing editable documents such as word processing documents, spreadsheets, and presentations.

For More Information:

Eclipse

At the core of the Lotus Notes 8.x software (and the higher versions), is IBM's version of Eclipse Rich Client Platform (RCP) technology, which introduces a new, open, standards-based SOA that makes the Lotus Notes 8 software more extensible. In fact, a number of the new features of Lotus Notes 8 are a direct result of this extensibility (for instance, Sametime integration and RSS feed integration).

IBM has built a common client platform named Lotus Expeditor (previously called WebSphere Everyplace Deployment or WED) that packages the Eclipse Rich Client Platform with some extra services such as security, synchronization, data, deployment, and more, which can be used across the IBM software product set. The Lotus Notes 8 client is a consumer of this Lotus Expeditor common platform. This provides additional functionality while ensuring forward compatibility for existing Lotus Notes and Domino applications.

Lotus Notes 8 and 8.5.x software supports nearly all custom Lotus Notes applications built for prior versions and incorporates the open standards of the Eclipse application development framework, allowing the use of a componentized SOA. This provides help in making it easy to aggregate, access, and deploy functionality from a mix of software programs and systems. It enables developers to build applications more quickly and to re-use existing assets as business needs arise.

Summary

In this chapter, we introduced Service-Oriented Architectures (SOAs) and covered how Lotus Notes 8 and 8.5.3 support it. We then looked at several Lotus Notes features and capabilities that can help you implement SOA-based architectures within your own organization.

We then discussed that with a foundation in open technologies such as Eclipse and with the introduction of the Composite Application Editor, Lotus Notes can be a key part of SOA.

We also saw that currently Lotus Notes 8 only participates directly in the Assemble phase of an SOA lifecycle. However, the open framework on which Lotus Notes 8 is based provides a highly flexible platform, and we can expect to see a significant growth of Lotus Notes as a key factor in the growth and adoption of SOAs.

In the next chapter, we will examine the new features in the Lotus Notes client, from version 8 continuing through the latest version, Lotus Notes 8.5.3.

For More Information:
Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.