



SOA Governance and ContraSite

Ensuring SOA success with effective, automated control throughout the lifecycle

Franco Castaldini

Director, Product Marketing, Software AG

May, 2008

CONTENTS

FUNDAMENTALS OF SOA GOVERNANCE	4
<hr/>	
CHOOSING THE RIGHT SOA GOVERNANCE TOOL	6
<hr/>	
STRUCTURE: ORGANIZED SOA	7
<hr/>	
SCALE: ENTERPRISE-CLASS GOVERNANCE	8
<hr/>	
SPEED: FASTER TIME TO VALUE	10
<hr/>	
STAYING ON THE SUCCESS TRACK	11
<hr/>	



Technology should always be an enabler, not an inhibitor, of achieving business goals. The inability of inflexible, tightly coupled legacy systems to respond quickly and effectively to business needs is a key reason companies have invested in flexible, loosely coupled SOA.

But without proper control, the flexibility of SOA can devolve into chaos. By definition, SOA brings a dramatic increase in the number of interdependent moving parts in the systems environment. In turn, an increase in the number of parts is accompanied by an exponential growth in the number and complexity of interdependencies.

Uncontrolled SOA allows services to be developed, invoked, and orchestrated at any time into complicated construction. Rather than creating a platform for effective reuse and responding rapidly to business goals and changing market conditions, uncontrolled SOA leads to redundancy in development and lack of visibility into systems that impact key processes.

Unfortunately, under this scenario not only do companies end up failing to achieve the return on investment they anticipate, they may even spend more time and money over the long haul than they would have under traditional development. IT may see more redundancy of services and greater infrastructure complexity as a result of poor SOA governance and through the uncontrolled proliferation of services that are that are difficult to locate or inadequately constructed or understood.

Industry analysts maintain that the key to successful SOA projects is to understand and control your SOA artifacts, and the key to understanding these artifacts is SOA governance. Forward-looking IT organizations have recognized the need for SOA governance: a strategy where each service is an asset that has to be properly designed to be useful within a larger portfolio of business services – versioned, secured, managed, and monitored to ensure that it performs with the expected quality of service (QoS).

SOA governance begins with creating standards for designs and processes that must then be applied to assets as they are created, used, and changed. However, to enforce these standards, time- and resource-strapped architecture organizations need powerful, automated SOA governance solutions. These solutions must also enable companies to efficiently and effectively apply and enforce governance throughout the entire SOA lifecycle—from design time, to run-time, to change time.

By making governance an up-front part of SOA implementation strategy, you will greatly enhance your chances of success and achieving a lasting return on your SOA investments. In this whitepaper, you will learn how effective SOA governance enables SOA success, and how to evaluate technologies that support your SOA governance strategy.

FUNDAMENTALS OF SOA GOVERNANCE

The demand for SOA governance comes not just from within IT. Across the enterprise, there has been a drive toward governance that provides visibility and accountability. Corporate governance therefore impacts overall IT governance, which in turn extends to SOA governance. In this context, SOA governance involves establishing the right ownership, the right roles, and the right level of visibility that provides trust in assets that are developed and consumed.

But specific to the technology of SOA, success is highly correlated with an organization's ability to manage complexity and to develop the necessary maturity and infrastructure—in the form of control and enforcement mechanisms—to maintain order over the SOA environment. According to Forrester Research, the number one benefit sought by adopters of SOA is flexibility.¹ However, without effective SOA governance, organizations will experience just the opposite, including:

- A fragile and brittle SOA implementation
- Services that cannot easily be reused because they are unknown to developers or because they were not designed with reuse in mind
- Lack of trust and confidence in services as enterprise assets, which results in a “I will build it myself” mentality (further compounding the lack of reuse with redundancy and unnecessary duplication of functionality)
- Security breaches that cannot easily be traced
- Unpredictable performance
- And, ultimately, disillusionment with SOA or complete SOA failure.

¹ Forrester Research, “SOA Adoption: Many Firms Got Started in 2007,” March 2008. The proportion of firms naming flexibility as a driver for SOA remains between 78% and 90%.

Governance Throughout the Lifecycle

SOA is also unique in that it involves the creation of discrete, well-defined services that exist not only as building blocks of larger systems and applications, but also as independent entities. SOA exposes standalone application functionality at a fine-grained level of granularity. It is no longer prudent to consider SOA governance a run-time or design-time activity; instead, SOA governance happens all the time.²

Service-level lifecycle governance includes:

Design-Time Governance

Design-time governance is an IT development function that involves the application of rules for governing the definition and creation of Web services. Policies might include ensuring that services are technically correct and valid and that they conform to relevant organizational and industry standards. If an organization has an SOA governance infrastructure in place—in the form of software that facilitates the implementation of SOA governance practices—these checks can be invoked automatically when developers check services into a registry.

Run-Time Governance

Governance at run-time revolves around the definition and enforcement of policies for controlling the deployment, utilization, and operation of deployed services. These run-time policies typically relate to non-functional requirements such as trust enablement, QoS management, and compliance validation. Run-time governance also involves service-level agreement (SLA) monitoring and reporting. By tracking the actual

² Gartner, “Key Issues for SOA Governance Technologies, 2008,” February 2008.

performance of a service and comparing it to the requirements specified in the SLA, the system can identify non-compliant services and prompt action.

Change-Time Governance

Change-time governance—the act of managing services through the cycle of change—is arguably more important in the long term than design-time governance. At some point, services deployed in the run-time environment will have to be changed to adapt to new business requirements. Traditional software development is only changed through recoding and deploying new code, but because of the rapidity of change enabled by SOA, appropriate controls are even more important to ensure managed outcomes.

Essential Technological Components of SOA Governance

While SOA governance is not a shrink-wrapped capability that can be implemented off the shelf without also addressing organizational and procedural issues, companies must incorporate SOA governance technology to effectively enforce and automate policies. Forrester observes that an enterprise-scale SOA is unwieldy without a level of process automation.³

At a basic level, an SOA governance system should facilitate service-level governance across the lifecycle from design-time to run-time to change-time. It should allow policies to be defined and created and provide mechanisms for these policies to be enforced at each phase of the service lifecycle. The main components of this system include:

- **Registry.** A registry is a catalog or index of services within an SOA. A registry is not designed to store the services themselves; rather, it indicates their location by reference. An SOA registry stores service descriptions and related information, allows services to be categorized and organized, enables users to publish new services into the registry and to browse and search for existing services, maintains service history, allowing users to see when a service was published or changed, and interoperates with other components of the SOA infrastructure.
- **Repository.** Because the registry itself does not provide sufficient context for SOA governance, such as policies that are enforced on services, a repository is needed for storing governance-related artifacts and supporting the full complexity of managing service metadata throughout the service lifecycle. A governance repository should provide an information model or taxonomy,

audit capabilities for tracking the trail of changes and authorizations applied to assets within the repository context, identity management capabilities and role-based access controls, and a notification system and content validation capabilities to provide additional assurances that policies are well-formed, consistent, and properly applied.

- **Integrated Registry/Repository.** When the registry and repository are unified with a single, normalized and standards-based information model and underlying datastore, the integrity of both governance related metadata and the information model can be assured. The unified approach also eases the enforcement of policies that apply across the boundary between the registry and the repository. Standard registry capabilities can be offered by an integrated registry/repository with a UDDI interface to the policy repository that allows access to the relevant data.

Policy Enforcement Points

Policies maintained in the registry/repository need to be enforced. Policy enforcement points change depending on the lifecycle stage.

During design-time, the registry/repository itself is the point of enforcement. Policies might include ensuring that services are technically correct and valid, and that they conform to relevant organizational and industry standards. Examples of this type of validation might include checking that a service is compliant with the Web Services Interoperability (WS-I) profiles by automatically verifying service schemas, validating namespaces, and other such controls. These checks can be invoked automatically when developers check services into a registry, and approval and notification workflows can be triggered to ensure that services pass through pre-defined review and approval steps.

During run-time, policies are generally enforced by the underlying message transport system that connects service providers with consumers. Run-time policy enforcement relies on an SOA infrastructure that is able to exercise policy enforcement in a way that is transparent to, and independent of, the service providers and consumers. This is achieved through an agent or intermediary that resides between provider and consumer and a registry/repository that addresses both the needs of run-time service discovery as well as policy enforcement. In an SOA, the run-time system is typically a message transport or mediation layer such as an Enterprise Service Bus (ESB). The message transport brokers transactions between service provider and service consumer and frequently offers additional functions such as data transformation, message queuing, reliable messaging, and other operational capabilities.

Change-time enforcement relies on IT change management practices and procedures more so than on enforced control points. The IT management system—the set of systems-management software tools and services that the IT organization uses to manage, monitor, and control its business applications and software infrastructure—and the registry/repository combine to play an important role in change-time governance.

Because SOA involves a dynamic network of interdependent services that are in an ongoing state of adaptation and optimization, it is critical that services, transactions, and SOA events of interest are monitored by the IT management system that, in turn, feeds information back into the registry/repository to facilitate the orderly evolution of the SOA environment.

³ Forrester Research, Inc., "The Forrester Wave: SOA Service Lifecycle Management, Q1 2008," January 2008.

Rules-Based Policy Management

Incorporating rules engine technology within the registry/repository enables a significant degree of flexibility and automation, while reducing the reliance on humans to perform mechanical governance tasks and the associated risk of error. Rules are typically associated with events, while the rules engine handles the firing and chaining of rules. A rules engine also provides the basis for creating complex policies based on reusable templates.

In addition to automating governance tasks, the rules engine can also help to deal with policy federation, or the ability to allow multiple policy authors and authorities. This is an important use case for enterprise SOA adoption where governance policies might not be authored and controlled by a single department or organization. A more robust model—which is the basis for policy federation—is to enable both centralized as well as distributed policy creation. Policy federation requires the establishment of guidelines and rules for reconciling policies that come into conflict, and the rules engine assists in the execution of these rules.

Automated Service Lifecycle Management

Finally, the SOA governance platform must provide a human interface to the registry/repository that also incorporates the governance lifecycle processes and workflows, including:

- Publishing a service by an authorized provider
- Discovering a service by a potential consumer
- Requesting use of the service by the consumer

- Agreeing on the terms of delivery of the service
- Authorizing the consumer
- Provisioning the service
- Monitoring the service delivery

CHOOSING THE RIGHT SOA GOVERNANCE TOOL

You need a tool in order to govern your SOA, but the wrong tool can complicate, rather than facilitate, effective and efficient governance and inhibit SOA success.

CentraSite is an enterprise SOA Governance and Lifecycle Management platform. It is designed for architects and IT managers leading SOA initiatives who need visibility and control over SOA development and deployment and for other users throughout the enterprise who are involved in the provision or consumption of SOA assets.

Forrester Research states, “CentraSite provides solid functionality in most areas, with very strong core capabilities and basic out-of-the-box reporting.”⁴ CentraSite’s advantages fall into four key categories.

1. Integrated Registry/Repository

Disjointed governance tools provide only piecemeal solutions and lock companies in to proprietary architectures. Unfortunately, the evolution of the SOA governance marketplace has done just that by traditionally separating the registry and repository. Additionally, as an SOA scales beyond just one project with a small set of services and assets, it becomes challenging to keep the registry and repository synchronized if they are not unified.

⁴ Forrester Research, Inc., “The Forrester Wave: SOA Service Lifecycle Management, Q1 2008,” January 2008.

CentraSite’s unified registry/repository provides companies a single system of record for information regarding both SOA assets and SOA governance artifacts. Its simplified architecture increases speed of implementation and adoption and is a more intuitive platform than segregated models.

2. Active Policy™ Management

If companies do not have an effective way to automate governance policies, the approval and monitoring bottlenecks that result will quickly become a point of frustration for both providers and consumers of services. CentraSite’s Active Policy management is unique in the marketplace because it allows companies to create policies that can automatically engage—creating efficiency, delivering faster time to value, and enabling scalability.

3. Full Lifecycle Management

In the marketplace, most governance solutions are still focused on design-time or run-time. In contrast, CentraSite automates design-time, run-time, and change-time governance in a highly integrated environment, providing one system of record for your end-to-end SOA lifecycle.

4. Extensibility

Finally, CentraSite ships with pre-loaded functionality including policies, best practices, metadata models, and user roles. However, the solution is also easily customizable to accommodate unique policies and practices and existing IT governance constructs.

CentraSite brings **Structure, Scale** and **Speed** to SOA adoption by guiding reuse, automating SOA processes, and simplifying complexities and interdependencies.

STRUCTURE: ORGANIZED SOA

A Single System of Record

By incorporating a unified registry/repository, CentraSite gives companies a single system of record for SOA assets. CentraSite provides facilities to store and manage assets, asset metadata, supporting resources, documentation, relationships, policies, contracts and run-time information.

Asset Types

Within CentraSite, both Web-Service and non-Web-Service assets are supported. Also within this framework, business processes are defined as SOA assets themselves. Users can publish business processes into CentraSite and share them with other stakeholders. Users can find existing processes and their interdependencies with other SOA assets. CentraSite provides full policy-based governance capabilities around approving, changing, and deprecating business processes.

CentraSite also supports an “interface-first” approach, where potential services can be proposed in the SOA by service consumers and architects. CentraSite provides governance and workflow capabilities to take such requests through the appropriate review and implementation process. Since lifecycle management is core to CentraSite functionality, users can note and track the evolution of a service from proposal to deployment.

Version Control

CentraSite manages multiple versions of a service or any SOA asset. New versions can be easily created from existing versions, including transferring all metadata and supporting information to the new version. Users have the option to change the metadata for the new version as needed.

Additionally, CentraSite eliminates the need to create new versions of a service

simply to support different service levels by featuring advanced service level management functionality, called Service Delivery Contracts™. Multiple contracts can be created for the same version of the service, providing different service levels and QoS.

Repository Architecture

CentraSite is built using JAXR repository technology and the Tamino XML database. Therefore, CentraSite is extremely flexible and extensible, having the ability to capture and process a rich variety of metadata.

Reporting and Analytics

CentraSite provides an intuitive, graphical impact analysis specific to each SOA asset. Relationships are also graphically presented so that users can easily identify which services are using which artifacts and vice versa. The analysis clearly shows relationships among assets, versions, business processes, contracts, policies and SOA mediation environments like ESBs, appliances and agents.

CentraSite analytics provide an interactive impact analysis during change management. Users can drill down into artifact details, such as owner and creation date, and define filters based on asset types, association types, and taxonomies that can be applied to impact analysis. Notification requests can be set on assets used or tracked.

Also important, CentraSite tracks both unidirectional and bidirectional relationships and dependencies at a fine-grained level. Relationships are tracked among all type of services and assets, policies, contracts and target deployment environments. Examples of relationships include Enforces, Hosted On, Invokes, References, Similar To, and Version. Users can also create custom relationship types as needed, and CentraSite can automatically infer relation-

ships based on available asset metadata. Distinct sets of relationships can be captured, managed and presented for a variety of contexts from design to development to deployment.

CentraSite also allows users to create custom reports using X-Query, making it possible to define custom metrics and integrate them into report templates. CentraSite ships with a set of Eclipse plugins that assist in defining such queries and integrating them into reports.

Service Validation

CentraSite provides advanced, policy-based service interface validation. Validation can be performed automatically, at any point in the lifecycle from design to deployment. Services can be automatically prevented from entering the SOA or progressing through the SOA lifecycle if they fail compliance checks.

CentraSite provides dozens of built-in validation policies that cover everything from WSDL validation to a full range of WS-I compliance checks. Further, users routinely add their own custom validation policies, including manual reviews and approvals if needed. Policy-based validation is also available for non-WS services, since those services are also SOA assets in CentraSite.

Service Metadata

CentraSite’s out-of-the-box meta model includes dozens of predefined asset types, profiles and attributes. Additionally, the CentraSite meta model is flexible and completely open to definition of custom attributes, attribute profiles and asset types using an AJAX-based UI. For any custom types, users can also provide custom metadata extractors and custom policies. Since custom asset types are first-class SOA assets, users can easily track relationships and dependencies with other interface types.

SCALE: ENTERPRISE-CLASS GOVERNANCE

CentraSite is an enterprise-class governance solution, enabled both by architecture and by design.

The unified registry/repository not only ensures consistent lifecycle management, but also provides a single platform for the enterprise that automatically governs each step in relationships between customers, suppliers and internal stakeholders. This provides an assurance of services' quality and enables collaboration and trust around the creation and consumption of SOA assets.

By being designed for heterogeneous system environments, CentraSite is also able to merge fragmented silos into a single governance context

Multi-Role Support

Throughout the service lifecycle, there are different stakeholders who participate in service production, management, and consumption. Within CentraSite, suppliers and consumers have their own customized view into the system of record, enabling communication between internal stakeholders, providing visibility into the enterprise SOA, and increasing confidence around the consumption of individual services.

CentraSite ships with more than a dozen pre-defined roles, including Consumer, Policy Administrator, Operator, Deployer, Asset Administrator, and others. Companies can create additional roles as needed. Metadata for assets in the catalog is presented via profiles that allow sets of metadata to be grouped, such as Technical Details, Resources, Deployment, Support, and others. Access control is available not only at the individual asset level, but also at the profile

level within assets. Therefore, companies can establish roles and access controls so that different stakeholders have different view of the catalog and its assets.

CentraSite provides an easy-to-use, shopping-cart-like functionality for consuming services. When consumers find services of interest, they simply add them to a request for access. The request for access is routed to the services' provider organizations. Provider administrators can then take the appropriate action, such as negotiating terms and SLAs, ultimately provisioning the requestor as a consumer of the services.

Service Delivery Contracts, which actively control the actual delivery of services, are a unique and powerful capability in that they not only allow definition of different ways in which the same service version can behave, but also that they are enforceable at run-time.

End-to-End Lifecycle Management

CentraSite provides end-to-end SOA lifecycle governance and supports managing multiple versions of any SOA asset. New versions can be easily created from existing versions, and multiple versions of the same service can be available in any lifecycle state at the same time. CentraSite provides easy means to identify versions and to navigate between different versions. Impact analysis pinpoints dependencies down to the service version.

Service Interface Design

CentraSite features built-in support for the WSDL interface standard and WSDL viewing and editing. Interfaces can be published into the SOA catalog even if an implementation does not exist for the interface through CentraSite's interface-first design. Requirements documents

and additional material can also be published to aid collaboration around interface design. Governance and workflow capabilities are provided to take such requests through the appropriate review and implementation process.

CentraSite provides built-in support for XML schema-based interface definitions. It can also be extended to support any asset type, such as BPEL processes, CORBA services, .Net assets, JMS artifacts, BIRT Reports, Service Component Architecture archives, and others.

In addition to describing the interfaces to technical services, CentraSite provides an extended set of templates for creating description profiles for business services. This ensures that business services are constructed conform to industry best practices and are well described, classified and documented.

Customization

CentraSite includes built-in taxonomies, can import external taxonomies, and enables users to create their own taxonomies using an AJAX-based UI. Many of the built-in taxonomies like 'Organization' and 'Asset Type' are automatically maintained by the system. Privileges can be managed over taxonomies to control which stakeholders see specific taxonomies when browsing.

Within CentraSite, policies can be defined to automate governance processes regarding users, groups, organizations, consumers, taxonomies, and other criteria. Policies can be defined so that they are only applicable to a specific organization or so that they are applicable to all organizations in the SOA. Policies are further scoped by entity type and governance event, such as creation, change, certification, or promotion. A policy can

be defined to be narrow in scope (for a specific entity type and specific event) or broad in scope (for multiple entity types and multiple events), as needed.

The asset type framework is flexible and completely open to definition of custom interface types. Companies can create interface types for non-WS services, legacy service technologies, homegrown APIs, controls, portlets, and other assets.

Service Deployment and Promotion

CentraSite provides the ability to promote services and assets from development through production. If the executable asset is managed in the CentraSite repository, the entire asset is also promoted. Promotion is a distinct event in the governance framework to which policies can be attached, and it is possible to trigger custom deployment services by attaching them to events in the registry.

For instance, it is possible to trigger a deployment service every time a specific asset type instance changes into a specific state of the asset's lifecycle. By doing this, companies can easily leverage their own deployment infrastructure. Services can also be virtualized using Service Delivery Contracts to provide mediation capabilities such as routing and credential mapping.

Scalability

The CentraSite repository solution employs J2EE architecture. Therefore, it can be scaled using capabilities of J2EE containers. Data and application tiers can be distributed on different machines.

CentraSite's run-time enforcement solution is built using a stateless Java architecture. As a result, it can be scaled horizontally by simply adding additional instances behind a load-balancer.

Heterogeneous Federation

CentraSite is designed for heterogeneous system environments. Federation is supported through use of the UDDI v3 standard, providing federation capabilities between several installations of CentraSite. Any specific installation of CentraSite can, at the same time, act as master and slave for federated processes

Federation across UDDI compliant 3rd party registries is supported. Based on CentraSite homogeneous federation abilities, extended federation with other third-party repositories and ESBs can be added.

SPEED: FASTER TIME TO VALUE

CentraSite's event-driven Active Policy technology ensures automatic and consistent policy enforcement. By delivering service lifecycle management "out of the box," CentraSite also speeds time to deployment. Pre-loaded policies and best practices enable companies to avoid the risks associated with starting from scratch, providing a proven path to success through a structured approach.

Active Policy Technology

Attempting to apply governance throughout the SOA lifecycle without effective automation will quickly result in bottlenecks. Architects are forced to manually review events, assess policy violations, and determine remedial action needed. Forrester stresses the need by enterprise architects for an automated process that coordinates the service lifecycle, ensures that each step is subject to review and approval, and provides robust electronic notification of important events.⁵

CentraSite's Active Policy technology not only speeds SOA processes through automation but, equally important, ensures consistent outcomes and provides quality control throughout the SOA lifecycle. Controls can be established quickly that govern service registration, validation, verification, approval, and certification. The workflow engine seamlessly connects to the policy engine to fully automate how approvals are invoked which, in turn, determine how further policies are executed.

Event notification is supported through standards such as SMTP (email) and SNMP. CentraSite supports user notification through triggering of any notification mechanism that is accessible as a web service. Notification can also be initiated manually if needed. CentraSite also integrates with third-party SOA management solutions, allowing services to be discovered and published into the registry/repository.

Pre-Loaded Policies and Best Practices

Pre-loaded policies cover processes such as:

- Metadata validation
- WS-I Compliance Check
- WSDL Validation
- Asset Certification
- Approval Workflow
- Setting Attribute Values
- Entity Promotion
- WS-Security
- Monitoring & Alerts
- Routing
- SLAs
- Depreciation
- Failover
- Logging

User Experience

CentraSite is designed from the ground-up to be deployed in a multi-stakeholder environment. As a result, access control is role-driven and very granular. All actions within the system are controlled through permissions and privileges, which in turn are managed via roles.

CentraSite's intuitive user interface accelerates enterprise SOA adoption. The interface is customized for a range of users, from architects and power users to consumers of SOA assets within business units.

Access control can also be automated through policies. Roles and access controls are enforced, regardless of how functionality is accessed, whether through our browser UI or through APIs from external tools. CentraSite integrates with over a dozen major IDEs using open standards like UDDI, and plug-ins are provided for popular IDEs like Visual Studio. These plug-ins provide advanced functionality and a more seamless user experience.

Architecture

CentraSite's J2EE solution architecture itself enables rapid deployment. In fact, the CentraSite solution itself can be installed and configured in a matter of hours. Typically, complete implementation ranges from a few weeks to a few months depending on the complexity of the enterprise, with the majority of the additional time spent on personnel and organizational issues related to SOA governance.

CentraSite features a built-in workflow engine that supports SOA processes such as consumer provisioning, approval of service interfaces, and change management, and integrates with business process management systems (BPMS). CentraSite offers BPEL 2.0 import capabilities, fully supporting decomposition of BPEL artifacts, either published or imported.

⁵ Forrester Research, "The Forrester Wave: SOA Service Lifecycle Management, Q1 2008," January 2008.

STAYING ON THE SUCCESS TRACK

Keeping SOA from turning into chaos requires keeping it under control through effective governance, grounded in a solution that can automate end-to-end SOA lifecycle governance. Yet in order for a solution to be embraced, it must also be quick to deploy, flexible, and easy to use. An enterprise class solution for SOA governance and lifecycle management must also be designed to be deployed in a multi-stakeholder, multi-organizational environment.

Forrester Research evaluated leading SOA service lifecycle management vendors against 96 criteria and found that Software AG is an established leader, thanks to SOA service lifecycle management capabilities and breadth of process focus.⁶ Software AG's CentraSite is an automated end-to-end SOA lifecycle governance platform that is an enabler and not a barrier to SOA adoption. It provides a unified registry/repository that is a single system of record for all SOA assets, automates governance policies for all stakeholders affected by SOA, and helps businesses keep SOA on the track to success.

⁶ Forrester Research, Inc., "The Forrester Wave: SOA Service Lifecycle Management, Q1 2008," January 2008.

TO FIND THE SOFTWARE AG OFFICE NEAREST YOU,
PLEASE VISIT WWW.SOFTWAREAG.COM

Take the next step to get there – faster.

ABOUT SOFTWARE AG

Software AG is the world's largest independent provider of Business Infrastructure Software. Our 4,000 global customers achieve business results faster by modernizing, automating and improving their IT systems and processes to rapidly build measurable business value and meet changing business demands. Using our solutions, organizations are able to liberate and govern their data, systems, applications, processes and services -- achieving new levels of business automation and transparency.

Our industry-leading product portfolio includes best-in-class solutions for managing data, developing and modernizing applications, enabling service oriented architecture, and improving business processes. By combining this proven technology with industry expertise and best practices, our customers improve and differentiate their businesses – faster.

Software AG – Get There Faster

Copyright © 2008 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their suppliers. All rights reserved.

The name Software AG™, webMethods™, Adabas™, Natural™, ApplinX™, EntireX™ and/or all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.