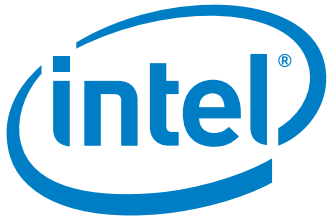


White Paper

Intel Information Technology

Computer Manufacturing

Service-based Architecture

Implementing a Service-based Architecture at Intel

To improve the responsiveness of enterprise solutions while reducing development time and development costs, Intel IT has been implementing a service-based architecture over the last few years. This strategy has proved to be very effective in delivering quick responses to changing business requirements and efficient use of scarce IT development resources. Our experience with building a service-based architecture shows that an incremental approach to developing reusable modular building blocks provides better results and return on investment than a more ambitious approach. We have also learned that governance and a centralized enterprise services repository and registry for service discovery and reuse are critical components of a successful services implementation.

Thiru Thangarathinam, Eric Hamer, Anil Varhadkar, Divya Bajaj Kumar, Abby H. Brown, and Tom Rynkowski, Intel Corporation

August 2009

IT@Intel

Executive Summary

To improve the responsiveness of enterprise solutions while reducing development time and development costs, Intel IT has been implementing a service-based architecture over the last few years.

Key benefits of implementing a service-based architecture include increased IT agility, improved time-to-value, better IT utilization and ROI, and simplification of our enterprise architecture and computing model.

Our service-based architecture strategy has proved to be very effective in delivering quick responses to changing business requirements and efficient use of scarce IT development resources. Our experience with building a service-based architecture shows that an incremental approach to developing reusable modular building blocks provides better results and return on investment, compared to a more ambitious approach.

Our continued success rests on a strong foundation of:

- **Governance.** We work with key stakeholders and in-flight projects to identify highly reusable enterprise services that address critical business needs. We also carefully analyze the solution design and look for reusable service candidates.
- **Infrastructure for easy access to services.** We standardized an enterprise services repository (ESR) that enables Intel IT developers and business group project managers across Intel to discover and repurpose existing services.
- **Education.** We have produced prescriptive guidance including how-to documents, video-recorded demos, and workshops that help IT developers learn how to use the ESR and develop services that align with our IT frameworks.

Key benefits of implementing a service-based architecture includes increased IT agility, improved time-to-value, better IT utilization and ROI, and simplification of our enterprise architecture and computing model.

Contents

Business Challenge

Intel IT, like many IT organizations, continues to improve its computing environment with the latest software and tools. Although critical, these improvements cause information consumers to retool their information consumption methods according to new data sources. These changes also imply testing overhead and the uncertainty associated with new releases.

In addition, Intel IT manages a heterogeneous computing landscape that successfully meets vertical business needs—but this makes integration and maintenance across the business and technical domains an ongoing challenge. Maintenance issues are further amplified by point-to-point interfaces that are fragile in nature.

These conditions result in a slow and inflexible IT response to demanding business needs. To counter this situation, we began to look at a service-based architecture, which proposed a solution to the challenges we faced. It offered interoperability between different off-the-shelf providers of services, shorter development cycles, and higher rates of reuse.

Our research and previous attempts to implement a service-based architecture indicated that it is important to look at service-based architecture as a methodology rather than as a technology.

To make a service-based architecture work for us, we needed to answer some tough questions:

- How could we determine which services were important to running the business?
- How do we identify the right set of services for enterprise-wide reuse?
- How could we train our developers to access and reuse existing services?
- How do we promote the right services to the enterprise level?

Solution

We answered these questions by adopting the combination of top-down and bottom-up approaches.

Top-down approach. We established a service-based architecture ecosystem that includes:

- **Governance.** Prescriptive guidance identifies which services should be developed, who should be involved, and how services are deployed.
- **Easy access to services.** An enterprise service repository (ESR) and registry makes existing services available for discovery and reuse, thereby increasing each service's return on investment (ROI).
- **Education.** Training for service consumers and contributors helps them learn how to use the ESR and develop services that align with our IT frameworks.
- **Landing zone.** Infrastructure hosts services with appropriate service-level agreements (SLAs).

Bottom-up approach. We created an attainable enterprise solution by following a methodical and consistent approach to building services that:

- Meets tactical project needs.
- Incrementally proves the value of a service-based architecture through real proof points.

Success through Governance

We have learned that about 70 percent of the work in successfully adopting a service-based architecture lies in defining and implementing the governance model.

First, it is important to realize that not everything needs to be an enterprise-level service. We focus on two ways to identify and manage services:

- A top-down approach by enterprise architects.
- A more tactical, bottom-up approach where services are identified through in-flight projects but are developed and supported as enterprise-class services.

We don't build infrastructure services before there is a clear business need to do so. We use several strategies to help identify these business needs, summarized in Table 1.

Goal Model: An Approach to Partnering with the Business to Develop a Service-based Architecture

The Goal Model captures the business strategy and supports a top-down approach by managing goals and supporting technical roadmap decisions. It directs our service-based architecture to deliver the key services required for business program success by establishing a framework for a services-oriented dialogue between the internal business stakeholder and the enterprise architects.

Within the Goal Model, we break down business goals to three levels and balance these goals against key performance indicators (KPIs) and metrics. Finally, we match the goals to supporting business processes.

KPIs and metrics serve as the blueprint for developing dashboards that are used by the business to guide the business intelligence lifecycle. Business processes associated with the goals connect project releases and teams to the business program strategy.

Table 1. Strategies for Building an Effective Service-based Architecture

Strategy	Rationale
Use a Goal Model.	Helps us manage goals and related business decisions.
Work with interested third parties and industry experts.	Helps us find services that will be used immediately. One major aspect of service-based architecture is that it enables key business capabilities and functions. Identifying long-term business stakeholders to "own" a service helps ensure that the service will be supported long after the first project that developed it is closed.
Reuse existing infrastructure, frameworks, and tools.	New services are compatible with previously developed services.
Look in the service registry.	Determine if there are services that could be reused by the project.
Look at prior implementations and evaluate what worked and what didn't.	Apply what we learn from this analysis to future implementations.

The Goal Model steers the relationship between the business stakeholders and enterprise architect by enabling decisions that keep pace with the changing business environment and shape a solid, reliable program roadmap.

Reuse through an Enterprise Services Repository

To maximize ROI on any service, it needs to be discoverable and easy to use. To address these two criteria, we have standardized on an ESR and registry as a central location to publish and register our Web services. For each service, the ESR publishes a standard service contract that provides details about the service, making it easy to use.

As shown in Figure 1, the ESR enables service consumers to connect with service providers, exposing services for enterprise-wide reuse.

Our ESR is made up of two components:

- **Enterprise service repository.** The design-time entity that enables end-to-end service life cycle management and governance.
- **Service registry.** The run-time entity that is based on the Universal Description, Discovery and Integration (UDDI) 3.0 standard.

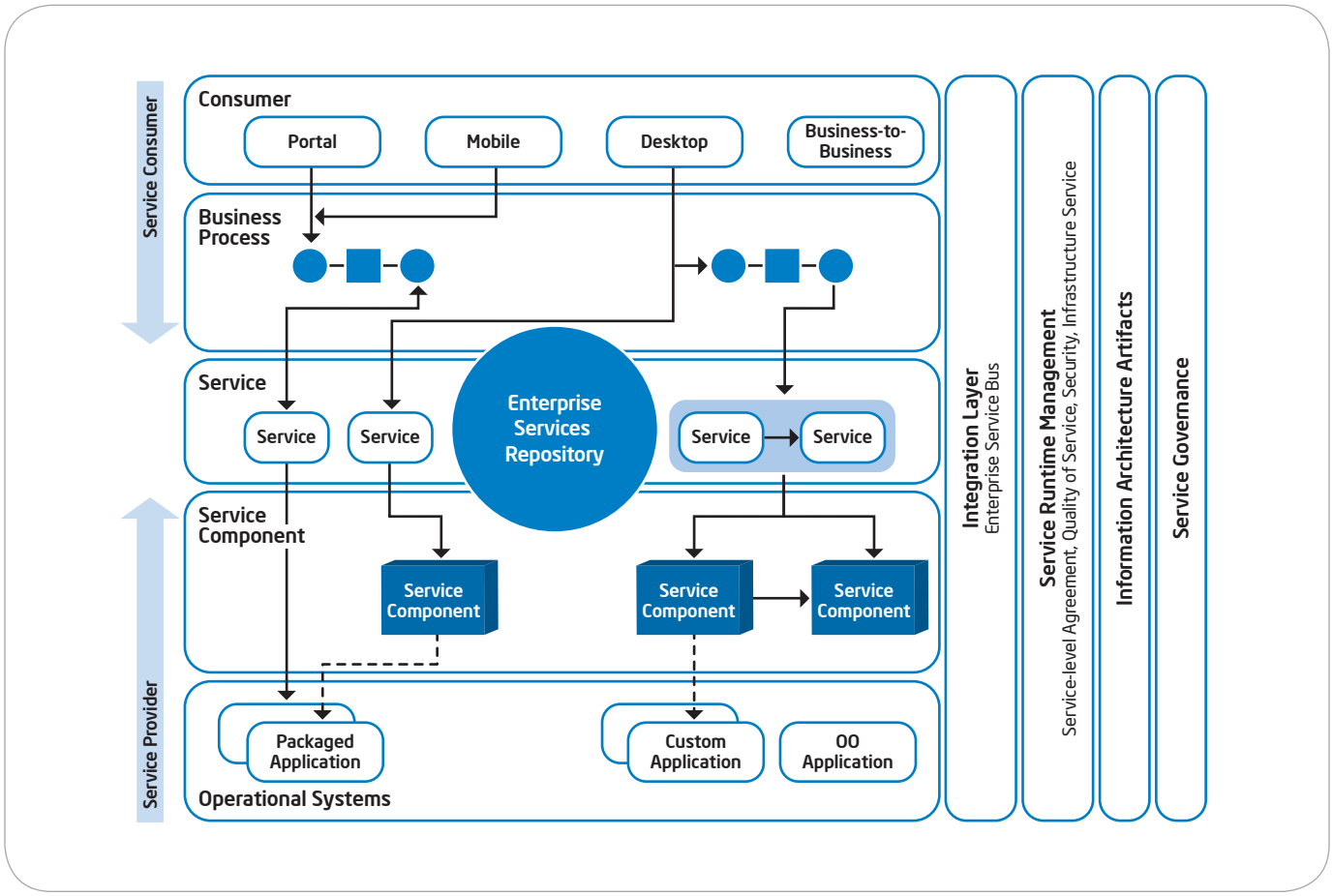


Figure 1. By allowing users to discover available services, more services are reused and repurposed, increasing the return on investment (ROI) of each service.

In the service registry, services are classified according to a hierarchy that is based on existing IT frameworks. These IT frameworks identify the capabilities required to run business-specific functions, horizontal business agnostic functions, and infrastructure.

Adoption through Education

To further encourage the reuse of developed services, we provide communications and training to educate Intel IT developers on how to create, publish, discover, and consume reusable services. We post how-to documents, workshops, and video recordings of hands-on demos online to help our developer community understand what services are available and how to use the ESR.

Organization Using a Common Landing Zone

It is important to host enterprise services on an infrastructure that:

- Is well supported.
- Meets required SLAs.
- Can easily scale with service consumption.

Meeting this challenge was one of our key hurdles, because typically development teams can develop enterprise-class services but don't have an appropriate infrastructure on which to host them. We removed this hurdle by establishing a common landing zone and implementing required support from the very beginning.

Examples of Service-based Architecture

The examples of services described in this section illustrate the types of services we have identified as viable service-based architecture candidates and how they have met our criteria for reuse as an ROI metric.

Employee Data Service

By implementing a service for Intel's employee information, we streamlined a complex data structure into a central database and greatly simplified data updates. To gain flexibility, we invested in creating components that can be easily assembled into various solutions.

Previously, applications that used employee information existed across a heterogeneous landscape of databases, OSs, and programming platforms stored on servers around the world; each database needed its own technology-specific structured query language (SQL) connection and/or a full data copy. Changing the employee data source affected hundreds of applications, creating a huge ripple effect that amplified costs and time to market (TTM).

Our Employee Data Service provides a service interface to the central database using Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP). This interface provides the downstream applications with a consistent way to programmatically retrieve up-to-date employee information. This reduces the development time required to build new applications, and data updates are quick and seamless.

We have seen good ROI from our employee data service:

- In production for seven years.
- About 250 applications receive employee data through this service.

Table 2. Cleansed Address Example

Submitted	Returned (Cleansed)
Intel Corporation	Intel Corp
prairry city road, 1900	1900 Prairie City Rd
folsum, ca. 95670	Folsom CA 95630-9599

Address Cleansing Service

Business groups across Intel, in particular the customer team, needed to "cleanse" addresses when new customer records were created or updates made to existing records. Cleansing addresses in our customer databases helps ensure that many different business groups consume consistent, correct data.

The address cleansing service allows multiple sources to standardize and validate a supplied address. This is important because several downstream solutions rely on this data as part of their business processes.

- We have been using the address cleansing service to clean up customer data since 2006.
- Since then, we've cleansed about 45 percent of our customer data.

For the approximately 50 percent of our customer base that uses the address cleansing service, we've seen an average 79 percent improvement in our customer address data quality.

Table 2 shows the cleansed output produced by the address cleansing service when provided with an incorrectly formatted address.

File Transfer System

Intel needed a more flexible, secure, and reliable service that could be reused by any Intel business group to exchange files externally. Intel's file transfer system (FTS) was developed within a service-based architecture and built using industry standards to meet this critical business need.

Since it was originally developed, our FTS has also enabled us to consolidate other file transfer solutions within Intel and has provided additional critical file transfer capabilities to Intel's business groups. Our FTS's service-based architecture is a key factor that has enabled this type of agility, with very low cost to Intel.

- Over 400 unique file transfer interfaces now exist in Intel's FTS to exchange files externally. Without a reusable service such as the Intel FTS, other file transfer solutions would have to be developed and maintained.
- Approximately 80,000 files are transferred, and over 80 percent of these files are exchanged without human interaction—further increasing the ROI for the FTS.

Enterprise Resource Planning Middleware

We have used the middleware component of our enterprise resource planning (ERP) system to develop services that access the capabilities in the ERP backend.

Previously, enterprise applications connected directly to the ERP system; this tight coupling led to systems that were complex and difficult to upgrade. Figure 2 illustrates our overall integration architecture and shows how a domain bus is used to encapsulate the services inside the domain and expose them to the enterprise.

During the past two years, all interfaces into the ERP backend have been required to go through the ERP middleware component (ERP domain bus) and have been required to use Web services or Java* Message Service (JMS) protocols. This service enablement was an early step toward service-based architecture and has simplified integration in our heterogeneous environment using industry-standard protocols.

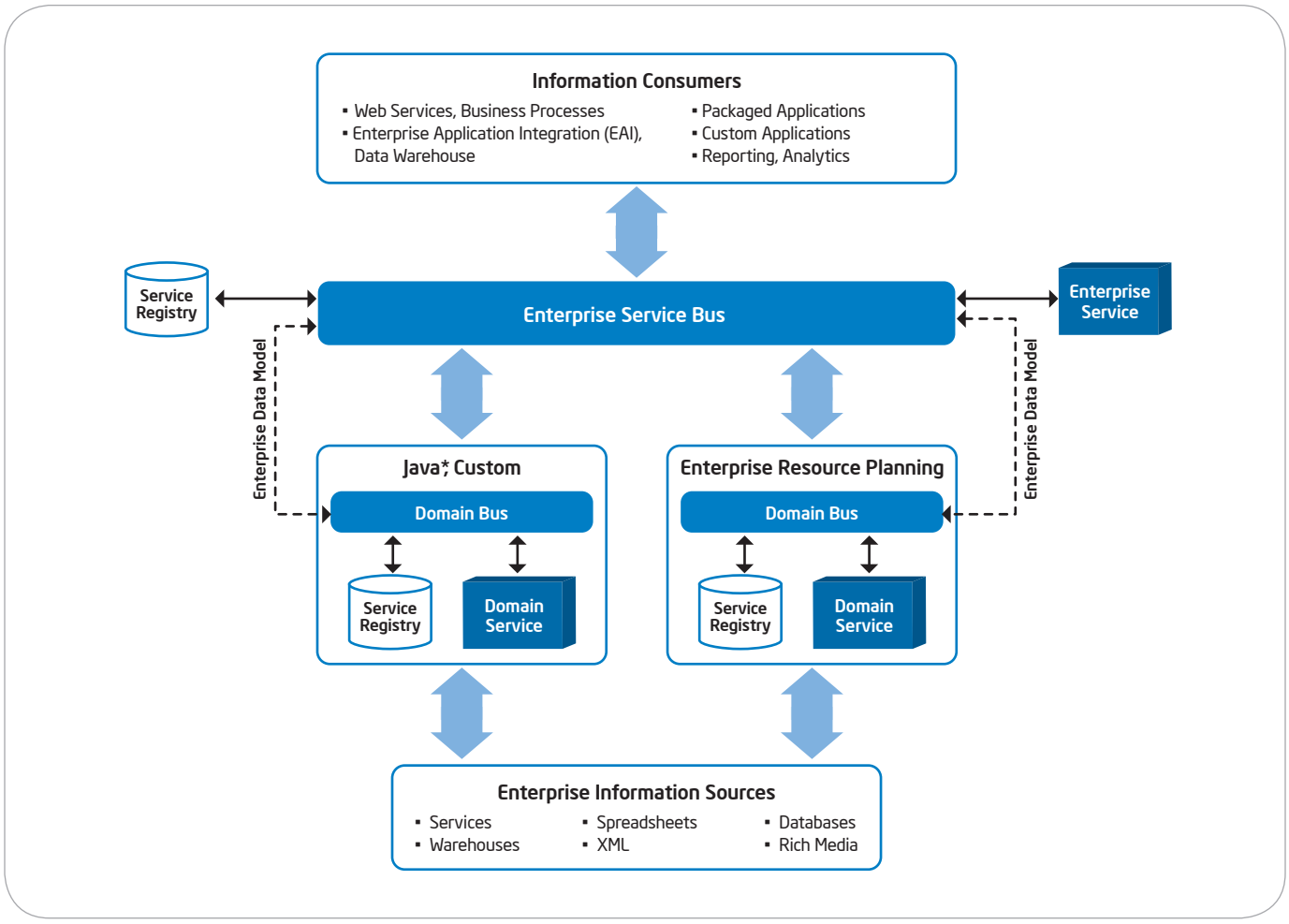


Figure 2. Integration across domains encapsulates the services inside the domain and exposes them to the enterprise.

By firmly decoupling the consumers from the ERP backend, we have greatly increased the ability to test and upgrade the ERP systems. By disallowing direct connects to ERP systems, we have created lightweight passthroughs into the ERP domain and have shifted the burden of backend compatibility from various clients and adapters onto the ERP middleware. Increased agility and decreased costs are benefits of providing service access through the ERP domain.

Next Steps

We are investigating implementing an enterprise service bus to further our service-based architecture efforts. These efforts are also a stepping stone to several important emerging areas of IT: business process management (BPM) and cloud computing.

Enterprise Service Bus

We have evaluated the applicability of an enterprise service bus architecture to connect multiple technology domains. Our evaluation determined that such a bus could deliver the common services and data layer that is needed for an enterprise-scale service-based architecture, including the following:

- Performance management
- Core services model
- Data management
- Service management
- Transaction management
- Security

Business Process Management

Industry experts predict that BPM will be among the fastest-growing software markets through 2011. The BPM approach to building IT capabilities views business processes as being composed of multiple steps, with services providing the capabilities for specific steps. With well-implemented BPM that is built on a strong service-based architecture foundation, Intel IT can save money, save time, and add business value.

With its focus on governance and re-use, our experience with developing a service-based architecture has taught us much that we can apply to the development of BPM capabilities.

Cloud Computing

Service-based architecture and cloud computing have something in common: They are both flexible architectures oriented toward providing a just-in-time response to business needs.

Cloud computing provides an alternative delivery mechanism of services to the enterprise, as shown in Figure 3.

Cloud computing is not the right delivery mechanism for every service in every situation. However, for certain categories of services, such as those with lower security requirements or where the Web is the desired platform, cloud computing offers cost-effective, efficient access to services.

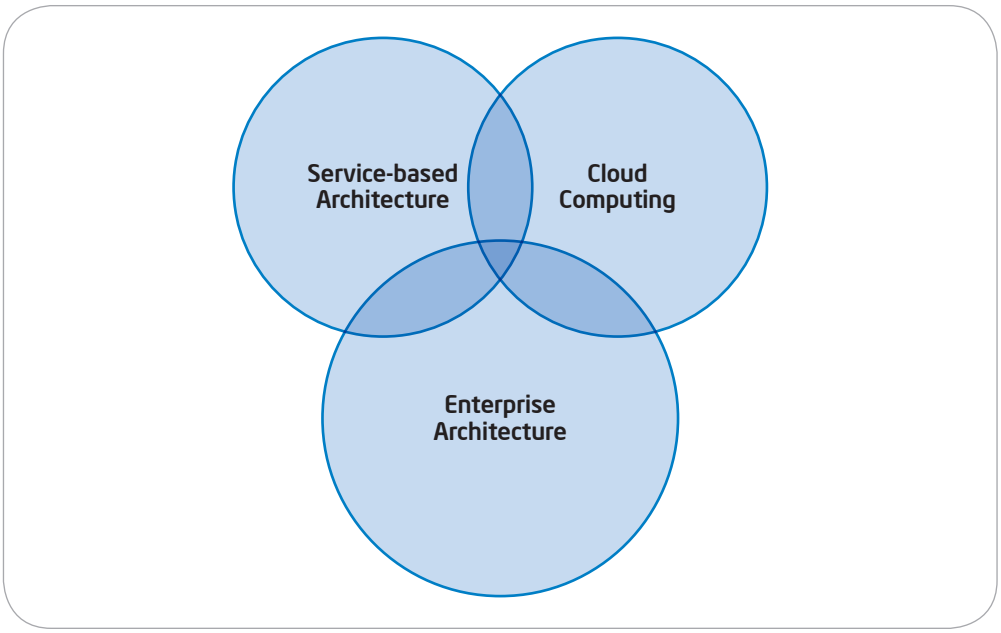


Figure 3. Service-based architecture and cloud computing complement each other, making delivery of services to the enterprise more flexible.

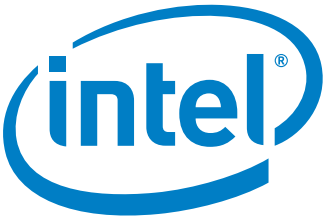
Conclusion

Our strategy to adopt a service-based architecture has helped us deliver agile enterprise solutions while reducing our development, maintenance, and integration time. Key learnings from our service-based architecture implementation include:

- Governance, including publishing prescriptive guidance and policies, is very important.
- We need an infrastructure for easy access to services.
- Education is necessary to help ensure services are used and reused.
- A common landing zone infrastructure is required, to host services with appropriate SLAs.

With an incremental approach to service-based architecture that focuses on meeting actual business needs and optimized ROI, we are starting to reap benefits that include increased agility, improved TTM, better IT utilization and ROI, and simplified enterprise architecture and computing model.

For additional content on Intel IT's best practices on this topic, go to www.intel.com/it



Authors

Thiru Thangarathinam is an enterprise architect with Intel IT.

Eric Hamer is an enterprise architect with Intel IT.

Anil Varhadkar is an enterprise architect with Intel IT.

Divya Bajaj Kumar is a program manager with Intel IT.

Abby H. Brown is an enterprise architect with Intel IT.

Tom Rynkowski is a product manager with Intel IT.

Acronyms

BPM	business process management
ERP	enterprise resource planning
ESB	enterprise service bus
ESR	enterprise services repository
HTTP	Hypertext Transfer Protocol
JMS	Java* message service
KPI	key performance indicator
ROI	return on investment
SLA	service-level agreement
SOAP	Simple Object Access Protocol
SQL	structured query language
TCO	total cost of ownership
TTM	time to market
UDDI	Universal Description, Discovery and Integration

Printed in USA
0809/IPKA/KC/PDF

Please Recycle
322028-001US