

# > ARTIX TECHNICAL BRIEF

Advanced SOA Infrastructure Suite

---

## TABLE OF CONTENTS

Executive Summary .....	1
The Need for SOA .....	2
The Integration Dilemma .....	2
Why Artix ESB .....	2
The Role of ESB .....	3
What is Artix ESB .....	4
Benefits of Artix ESB .....	5
Distributed and Standards-Based .....	5
Complex SOA Enablement .....	7
Incremental Journey to SOA .....	8
How Artix ESB Is Used .....	8
System Migration .....	8
Client Renovation .....	9
Process Automation .....	9
Point-to-Point Integration .....	10
Protocol Adapter .....	10
The Parts of Artix ESB .....	11
Design-Time Tools .....	11
Runtime System .....	13
Enterprise Qualities of Service .....	18
Progress Artix: Advanced SOA Infrastructure Suite .....	20
A Different Approach to Integration .....	20

---

## EXECUTIVE SUMMARY

The Progress® Artix® ESB is an enterprise service bus (ESB) and a major component of the Artix advanced SOA infrastructure suite. The suite is designed for complex and heterogeneous IT environments running mission-critical applications, and uniquely supports a fully distributed architecture for flexibility and agility.

The Artix ESB service-enables existing servers directly at the endpoints without the need for a separate integration server, preserving enterprise qualities of service. Artix ESB uses any existing message passing infrastructure or protocol without adapters or translation to a canonical message format. All Artix ESB features are dynamically configurable at runtime. Artix ESB provides the maximum degree of flexibility for implementing a scalable and high-performance, fully distributed service-oriented architecture (SOA). Artix ESB can complement an existing centralized integration solution or it can replace centralized solutions altogether.

Artix ESB integrates your existing infrastructure with Artix endpoints. Artix endpoints provide integration capabilities alongside existing servers and enterprise functionality. An Artix endpoint can be deployed in the following ways:

- > In-process with existing applications as a runtime library
- > On the same machine with existing applications, running in a separate process
- > Standalone intermediary

Artix endpoints can be deployed one-at-a-time which allows customers to start small and scale up slowly or quickly as needs demand.

Artix endpoint interfaces and features are specified using the standard Web services description language (WSDL) that can be easily composed in the Eclipse-based Artix Designer user interface. Code stubs can be automatically generated from WSDL simplifying service-enablement of existing functionality.

---

## THE NEED FOR SOA

Many large companies are adopting service-oriented architecture (SOA) and standards-based technologies to cut costs, facilitate system reuse and increase business agility. SOA allows loosely coupled systems from different vendors to communicate using an enterprise service bus (ESB), creating a back plane of services across the enterprise.

### *THE INTEGRATION DILEMMA*

In recent years, IT managers have spent millions of dollars on integration solutions that have not provided the return they expected. In these environments, where IT assets include solutions from many vendors, ROI needs to be increased and operating costs reduced. However, as competitive pressures, regulatory mandates, and customer demand grow, businesses must add new features and functions more quickly than ever before. It is more important than ever for Global 2000 firms with the most complex integration challenges to find new ways to leverage what they already have, using existing functionality to build new features for users. A traditional approach to integration requires complete refactoring, and building yet a new all-encompassing vision for the enterprise. But given the financial realities, that is simply not an option.

### *WHY ARTIX ESB?*

Artix ESB gives your IT organization a unique set of options for integration because Artix ESB approaches integration in a new way. Artix ESB focuses on the nontrivial task of service-enabling mission-critical infrastructure leveraging existing IT assets. By service-enabling existing systems directly, using smart endpoints that can run on the same machine as existing servers, Artix ESB increases the value of existing assets without a new centralized system. Artix ESB allows systems to connect in new ways, using any messaging system, without disrupting existing functionality and without requiring a large upfront investment. With Artix ESB IT managers can create the agile infrastructure required to offer new services while leveraging existing investments.

---

## THE ROLE OF AN ESB

Gartner defines an ESB as Web-services-capable middleware infrastructure that supports intelligent program-to-program communication and mediates the relationships among loosely coupled and uncoupled business components. In order to do this in complex enterprise environments where leveraging existing systems is a priority, it is essential that the ESB solution be

- > **Technology-neutral**—to work across all messaging systems and platforms
- > **Easy to incrementally adopt**—to allow step-by-step deployment and minimal disruption to existing functionality
- > **Dynamic and adaptable**—to react to rapidly changing business requirements

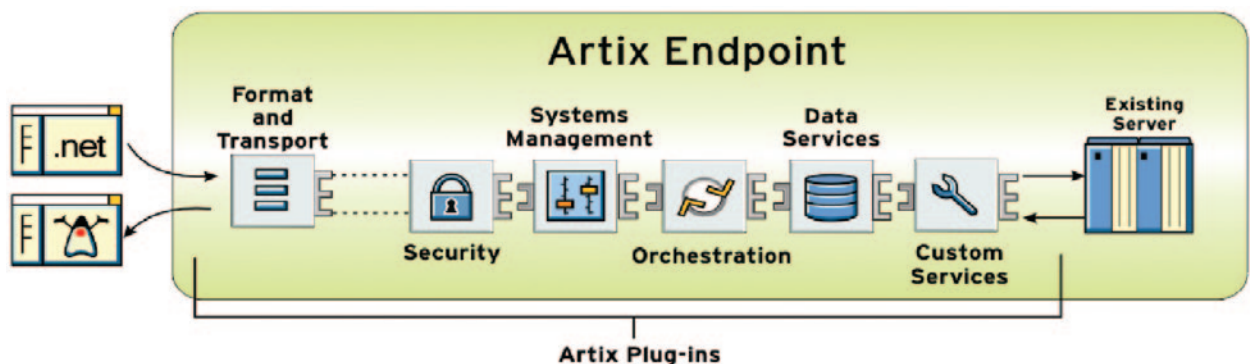
An ESB enables the passing of messages between enterprise systems in a SOA. An ESB service-enables systems to participate in the SOA. Artix ESB does this without a centralized intermediary server, leveraging existing messaging infrastructure, (like WebSphere MQ or TIBCO), and it provides enterprise qualities of service, like security, management, high availability, and transactions.

Artix ESB defines Web services interfaces using Web services description language (WSDL). Using WSDL, users can expose aggregate functionality of underlying systems that can be useful throughout a large organization in different applications. Artix endpoints configured using WSDL provide an abstraction layer between systems, and allow services to be independently implemented and maintained using the technologies and languages most familiar and appropriate for a particular business function. Tying these services together is the role of an ESB. To be effective in this SOA-enablement and integration role an ESB must support the widest array of disparate technologies: operating systems, device types, languages, and network messaging protocols. It must be extensible to add capabilities as needs change and as new technologies need to be supported. Artix ESB

provides the most flexible toolset available to service-enable a broad array of existing enterprise infrastructure with the enterprise qualities of service mission-critical operations demand.

## WHAT IS ARTIX ESB?

Artix ESB is a platform-independent, standards-compliant infrastructure product for building Java, C/C++ and mainframe Web services. For the enterprise, Artix ESB is an enabler of SOA, which makes it a cost-effective solution for integrating enterprise business systems and building integration-ready applications. Artix ESB consists of design-time tools for defining and implementing Web services and a runtime system of smart endpoints.

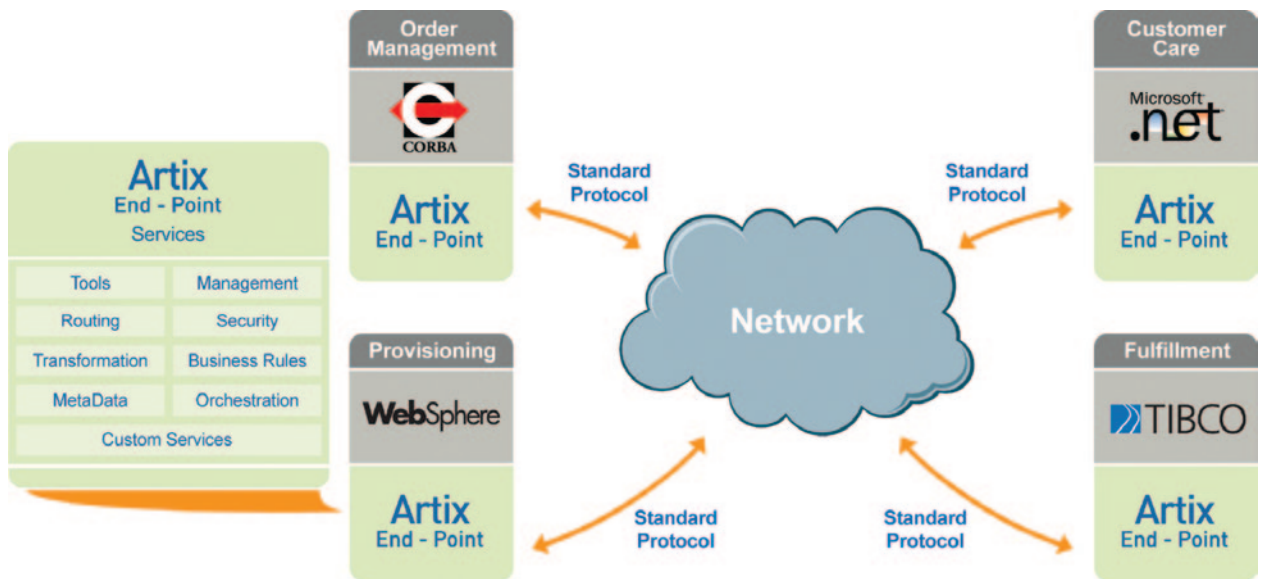


Artix endpoints provide security, orchestration, high availability, and other features through plug-ins. Plug-ins can be added at each endpoint separately providing just the functionality you need without unnecessary overhead. This allows you to expose your servers as Web services with little or no additional hardware requirements.

Artix ESB consists of design-time, runtime, and configurable enterprise qualities of service, each of which is described in more detail below (see The Parts of Artix ESB).

## BENEFITS OF ARTIX ESB

Artix ESB is an extensible ESB designed to meet the integration needs of large, heterogeneous enterprise environments in a cost effective way. With Artix ESB you can expose capabilities of your existing systems as Web services, using any messaging protocol, not just SOAP over HTTP, without the need for a centralized integration server. Artix ESB is designed to allow you to leave and leverage existing systems, so you don't need to rip out and replace what you have to get the benefits of a SOA.



### *DISTRIBUTED AND STANDARDS-BASED*

Artix ESB is a fully distributed, standards-based ESB. Unlike other ESBs, Artix ESB provides all features in a distributed way, at the endpoints of your network. Artix endpoints can be collocated with your enterprise systems, often on the same physical machine. Because it is distributed and standards-based, Artix ESB enables easy incremental SOA adoption, it is dynamic and adaptable, and it is technology-neutral.

*Distributed and standards-based approach with qualities of service embedded in the endpoint*

---

### ***Technology-neutral***

The standards-based, technology-neutral approach of Artix ESB enables your enterprise systems to interoperate using any messaging system, so existing messaging buses can be left in place, or partially replaced depending on what makes sense for a particular environment. Artix ESB reduces dependencies on any one technology and supports a diverse network of technologies from many vendors. Artix endpoints support the latest Web services standards, and are configured using WSDL their interfaces are easy to change.

### ***Enabling Incremental SOA Adoption***

Enabling incremental SOA adoption means customers can deploy SOA one function at a time with no disruption to existing functionality. Artix endpoints do not depend on one another so they can be added to existing systems one-at-a-time. For example you may use Artix ESB to replace hard-to-configure adapters for a centralized enterprise application integration server. This allows you to add additional capabilities to that endpoint, like the ability to talk directly to other enterprise servers without the interaction of the central server, or to respond to SOAP over HTTP requests directly.

### ***Dynamic and Adaptable***

Artix ESB features are determined at runtime and loaded as plug-ins as needed. This makes Artix ESB extremely lightweight and right-sized for your particular integration needs. Unlike heavyweight centralized servers, Artix ESB can provide just the features you need using minimal resources. As your needs grow, the Artix plug-in architecture allows you to add functionality that meets or exceeds the capabilities of centralized solutions, always allowing you to leverage what you already have. Plug-ins can be reconfigured without code changes and changes to one endpoint do not require changes throughout an enterprise.

---

## COMPLEX SOA ENABLEMENT

Artix ESB provides a SOA-enablement solution that focuses on the most complex IT environments where SOA-enabling existing systems is nontrivial. In these environments it is essential that an ESB be extensible, provide broad platform support, and provide enterprise qualities of service like security, transactions, and high availability. Following is a brief description of each of these three key attributes that make Artix ESB the right choice for complex enterprise integration:

- > **Extensibility**—For unique application specifications that are not met by its out-of-the-box platform support, Artix ESB is easily extended through its patented plug-in architecture. The extensibility of Artix ESB makes it possible to meet and future-proof even the most extreme requirements quickly and efficiently.
- > **Broad Platform Support**—From mainframes to mobile devices, Artix ESB runs natively inside a wide range of operating and application platforms and bridges the protocols and data formats of diverse, legacy systems.
- > **Enterprise Quality of Service (QoS)**—Security, manageability, high availability, performance, and reliability are the most important concerns for these customers, and the maturity and proven track record of Progress ensure that even the most taxing integration projects are durable and resilient. The high performance of Artix meets the throughput needs of demanding enterprise system. Including even mainframe applications Progress leverages its wealth of technology and expertise to deliver an ESB that is extensible to meet the most demanding IT requirements today, and into the future.

The unique features and capabilities of Artix ESB are transforming SOA adoption and use. Customers can now implement SOA incrementally, in a distributed environment while maintaining the utmost flexibility and adaptability.

---

## INCREMENTAL JOURNEY TO SOA

The process of adopting a SOA in a large enterprise is a journey. Artix ESB makes that journey addressable in small, manageable chunks. The incremental adoption that Artix ESB enables makes it possible to tackle each step of SOA adoption one at a time. SOA adoption follows several steps. These steps are as follows:

- > **Evaluate**—Evaluate existing capabilities and infrastructure components. Determine what systems need to be leveraged, and where Artix ESB can best be deployed to take the best advantage of existing functionality
- > **Service-enable**—connecting existing functionality, exposing it as Web services. Artix ESB specializes in service-enabling existing systems. In particular, Artix ESB service-enables the broadest array of platforms available, including mobile devices and mainframes.
- > **Configure qualities of service**—leveraging existing security and management solutions where appropriate (See technical notes “Artix and Security” and “Artix and Management” for details.)
- > **Orchestrate**—Build multistep processes using Artix® Orchestration. (See “Artix and Orchestration” technical note.)

## HOW ARTIX ESB IS USED

Artix ESB can be used to address a variety of enterprise integration challenges. Most uses fall into one of 5 categories: system migration, client renovation, process automation, point-to-point integration, and protocol adapter.

### *SYSTEM MIGRATION*

Artix ESB can be used to isolate an existing integration solution for the purpose of gradually migrating to a new solution. Replacing all outdated systems at once is disruptive and often impractical. Artix ESB allows new functionality to interoperate with existing systems as customers incrementally switch messaging systems. Artix ESB is particularly well-suited for this

---

purpose because it is technology-neutral and works with any messaging system, providing any-to-any access for enterprise systems.

For example a customer may have an integrated network based on Tuxedo messaging but now they want to migrate to WebSphere MQ, which has recently been chosen as the new enterprise-wide standard. Artix ESB allows some systems to be migrated to WebSphere MQ while still talking to other systems that use Tuxedo. As each piece of functionality is tested to work properly with both WebSphere MQ and Tuxedo, Tuxedo can gradually be replaced.

### ***CLIENT RENOVATION***

Many client-server installations have highly functional and reliable back-end systems, but are suffering from antiquated presentation layers. Artix ESB can help customers replace green-screen terminals and other outdated user interfaces with Java, .NET, or Web-based front ends without changing legacy servers.

An Artix ESB customer rebuilt their customer service desktops using .NET but wanted to connect those client applications to existing mainframe services, J2EE services, and CORBA services. Artix ESB provided an abstraction layer that enabled clients to talk to existing and legacy systems on the back end in the old protocols. Mainframe systems were left unchanged, but users no longer needed to interact using green-screen terminal emulators.

### ***PROCESS AUTOMATION***

Artix ESB can enable existing infrastructure to meet changing business requirements by automating processes that include legacy and new systems. Artix ESB provides a common developer interface for all systems, which can allow companies to react faster to changing business conditions.

An example of how process automation is essential is in the context of a large telecommunications firm built through mergers and acquisitions of wire-line telephone, mobile telephone, cable TV, and internet providers. In order to provide combined services with unified billing, providers like this need a way to automate billing across the billing systems of all the constituent merged subsidiary companies.

---

One company may have billing systems implemented using TIBCO, another may use an IBM z-series mainframe. Each silo of functionality may also have its own IT department with skills to match their internal integration solution. Indeed there may be functional reasons, or important legacy reasons, why a particular solution makes sense for a particular business unit that don't necessarily apply to other business units. Choosing one of the systems in use as the new standard for all may not be practical, especially in an environment where time to market is an essential component of success. And the cost of ripping these systems out and replacing them can be astronomical, which makes this approach impractical.

Artix ESB allows disparate systems like these to be integrated without completely reengineering. It allows integration goals across a large enterprise to be addressed incrementally, one system at a time. As small, compartmentalized integration projects are completed ROI can actually be calculated, as opposed to the reality of ongoing "boil the ocean" all-or-nothing integration projects.

### ***POINT-TO-POINT INTEGRATION***

Artix ESB fills a tactical need to connect a system with other parts of the enterprise. Simple integration between two incompatible systems is an important need not easily addressed with heavyweight integration or application servers.

An example of this is when a customer has an order processing system built using TIBCO Rendezvous messaging and now has a need to access parts of that system from a WebSphere MQ-based system. Artix ESB can allow that access from one system with just the functionality required.

### ***PROTOCOL ADAPTER***

Artix ESB provides multi-protocol connectivity in one package that can be configured using WSDL. Artix ESB is more flexible than protocol adapters provided with enterprise application integration servers because it allows endpoints to communicate using any protocol, or multiple protocols simultaneously.

---

Customers have chosen Artix ESB to provide the widest possible set of connectivity options even in new applications. One provider of medical document management and viewing software chose Artix ESB to connect that system to the greatest number of hospital IT infrastructures.

Artix ESB works well in each of these use cases because it is technology-neutral, can be deployed incrementally, and is dynamic and adaptable as needs change.

## THE PARTS OF ARTIX ESB

Artix ESB is a set of tools to build, configure, and deploy, secure, and manager a set of smart endpoints throughout your network to make existing systems available in a SOA context. The interface of each endpoint is entirely defined using WSDL interface contracts. For each endpoint WSDL specifies the name and type signature of each function, the message serialization formats (i.e. SOAP or TibMsg), and the network protocols to use (i.e. HTTP or WebSphere MQ). Artix ESB allows architects to specify endpoint functionality in WSDL first, providing a clearly defined interface abstraction for each endpoint in the network. Using Artix Designer and the included code generators, developers can create WSDL interface contracts as well as stub implementations of WSDL interfaces in C++, Java, or COBOL.

To explain further how Artix ESB works let's first examine what comes with Artix ESB and how it is organized. Artix ESB consists off design-time tools, a runtime system, and a set of enterprise qualities of service.

### *DESIGN-TIME TOOLS*

Artix ESB design-time tools are provided to make creating, testing, and deploying services easy. Design-time tools consist of the following:

- > **Artix Designer**—a tool based on the familiar Eclipse development environment that is the primary interface for defining and implementing service interfaces
- > **Code generators**—(typically invoked from Artix Designer) help you build C++ and Java implementations of services from Web services description language (WSDL)

- > **Management Console**—a tool to view the status of Artix Container hosted services. This tool can be useful for testing services and functionality hosted in the Artix Container (see below) or for monitoring the status of deployed services

### ***Artix Designer***

The Artix Designer is an integrated development environment (IDE) based on the open source Eclipse platform. IONA, as a leader of the Eclipse SOA Tools Project, is committed to extending this popular industry-standard toolset to provide the most useful vendor-neutral SOA design and implementation environment. Within Artix Designer you can generate WSDL contracts for your services and invoke the code generators that build Java or C++ code that implement those WSDL interfaces. Also you can graphically connect databases to your SOA and compose orchestration flows using business process execution language (BPEL). Artix Designer is the integrated front-end for Artix ESB and provides a single toolset to create and maintain all endpoints, even those running on IBM's mainframe operating system z/OS.

### ***Code Generators***

Artix Designer generates code for a variety of languages from WSDL interface definitions. After defining an endpoint interface you can auto-generate client and server stub code for C++ and Java. Artix ESB also includes the following code generation features:

- > **WSDL-to-ACL security support**—generate XML based Access Control List configuration to be used in support of role based authorization
- > **WSDL-to-CORBA IDL generation**—auto-generate CORBA IDL directly from WSDL file
- > **COBOL-to-CORBA IDL generation**—auto-generate CORBA IDL directly from COBOL Copybook
- > **WSDL-to-SOAP for immediate Web service access**—automatically generate Web service endpoint definitions (SOAP over HTTP) for all services defined in the WSDL

- > **Java-to-WSDL support for stateless session Beans**—create WSDL directly from Java for fast creation of Web service interfaces
- > **IDL-to-WSDL support for enabling CORBA services**—create WSDL directly from CORBA IDL for fast creation of Web service interfaces for CORBA servers
- > **WSDL validation tools**—test the validity of WSDL documents prior to deployment to aid interoperability and error free deployment
- > **WSDL service tools**—generate services from the command line for mass service creation and scripting
- > **WSDL to PL/I code generation**—create PL/I copybooks from WSDL for use in mainframe Web service applications
- > **WSDL to COBOL code generation**—create COBOL copybooks from WSDL for use in mainframe Web service applications
- > **Copybook-to-WSDL for enabling mainframe applications**—create WSDL directly from COBOL copybooks for fast creation of Web service interfaces for COBOL servers

### ***Artix Management Console***

A new Eclipse-based tool allows you to manage your Artix Web services. Accessible either from within Artix Designer or as a stand-alone application, this simple console takes full advantage of the Artix container functionality.

### ***RUNTIME SYSTEM***

The Artix Runtime engine is based on a high-performance, patented microkernel architecture originally from the successful CORBA integration tool, Progress® Orbix®. Artix provides an ESB on top of this time-tested engine design.

### ***Deployment Options***

Artix ESB provides a wide array of deployment options for the greatest flexibility in incremental SOA deployment. An Artix endpoint can be deployed in the following ways:

- > **In-process with existing applications**—Artix ESB can be deployed as a runtime library to directly provide integration functionality and enterprise qualities of service within your client or server applications.
- > **On the same machine with existing applications**—The Artix Container can provide intermediary functions like protocol transformation, routing, location directory services, security, high availability, on the same machine with existing servers, leaving legacy applications totally unchanged.
- > **Standalone intermediary**—Artix ESB can serve as a service intermediary between islands of integration within a large organization. Being able to deploy Artix ESB as a stand-alone server is sometimes the best option to provide integration between two or more sets of systems. The Artix Container can also host simple services implemented as Java or C++ classes, providing easy scalability and security.

The runtime system that enables these deployment options consists of the following elements:

- > Artix Runtime Library
- > Artix Container
- > Runtime Plug-ins

### ***Artix Runtime Library***

Artix ESB can be deployed as a library module which can be embedded in your applications on the service producer or consumer side. The runtime is available in several forms including a J2EE Connection Architecture (JCA) connector, a .NET assembly, a Java class library, and a C++ linked library.

### ***Artix Container***

The Artix Container allows Artix ESB to operate as a standalone process. The Artix Container can host Artix services, like the router or locator services, or custom-made services including plain-old Java objects (POJOs).

---

The Artix container is a server that allows you to run service plug-ins, either those that come with Artix ESB or custom-made ones, as a stand-alone process. The Artix container provides, threading, resource management, and network management services to simplify the deployment of stand-alone service intermediaries and shared services.

For example, you may want to store location information about your various Web services in a centralized repository. This can be done using the Artix locator service hosted in the Artix container. You can also host application logic developed in the form of a Java or C++ class directly to your SOA using the Artix container. The Artix container can provide all the Artix plug-in features to your application logic.

The container frees you from having to worry about message passing, serialization, or low-level details of implementing your functionality as a feature-rich server.

### ***Runtime Plug-ins***

Plug-ins provide a variety of functions to the Artix runtime. These can be loaded as needed by the Artix runtime, which keeps endpoints lightweight and extensible. Since only the features needed are loaded runtime overhead requirements are minimized. Plug-ins that ship with Artix ESB include transport plug-ins to support popular messaging systems like WebSphere MQ, Tuxedo, TIBCO Rendezvous, and JMS.

Plug-ins are loaded as needed by the patented ART runtime (see “Artix and ART” technical note) and are selected based on the content of metadata. This metadata is provided in two principal forms: endpoint configuration files and WSDL interface contracts. Depending on what kinds of features or protocols are needed to expose a particular interface in a WSDL file, an endpoint will load various plug-ins at startup. Plug-ins fall into 3 categories: transports, message formats, and services. These plug-ins form the base set of features provided in an Artix endpoint, but they can be easily extended using a well-documented API. (See Artix ESB programming manuals for details.)

---

## Transports

Transport plug-ins define how messages are packaged into on-the-wire protocols, like HTTP or WebSphere MQ. Artix ESB comes with the following transport plug-ins:

- > HTTP
- > HTTPS—secure HTTP over SSL
- > IIOP—CORBA Internet Inter-Orb Protocol
- > GIOP—CORBA General Inter-Orb Protocol
- > BEA Tuxedo
- > IBM WebSphere MQ
- > TIBCO Rendezvous
- > Java Message Service (JMS)
- > FTP

## Payload Formats

Payload format plug-ins define how messages are serialized into a standard message format. Artix ESB supports these payload formats directly:

- > SOAP
- > G2++ Record Format
- > FML
- > Tagged variable-length records
- > TibMsg
- > CORBA messages

## Services

Service plug-ins provide features within the context of a user-implemented endpoint or in a stand-alone Artix container instance. Service endpoints that can be used stand-alone offer cross-SOA management features, like location and session management or service routing.

---

### Stand-alone service plug-ins:

- > Session Manager—Enables service sharing via short-term lease
- > Service Locator—A centralized node to post and look up service locations
- > UDDI Proxy—Dynamically locates Web services using a UDDI service
- > Routing—a service intermediary to conditionally route service requests
- > Coordination Service—WS-Coordination service for 2-phase-commit transactions
- > XSLT processing

### Endpoint-embeddable service plug-ins

- > Locator endpoint service—allows usage of a service locator
- > OTS—enables the CORBA OTS transaction system
- > Session endpoint manager—loaded by endpoints use session management
- > WSDL Publish—Enables Artix endpoints to publish and use object references.
- > Collocation Service—Allows services on the same box to talk without messaging
- > XML Logging

### Messaging Options

Artix ESB works with any messaging system, and unlike many ESBs, Artix ESB is not built on top of a message passing infrastructure. Instead, message passing systems are directly supported via plug-ins to Artix ESB. Artix ESB has no native messaging protocol or format so it can support any system you may already have without translating and without translation adapters.

---

## Bundled JMS Implementation

Artix ESB comes with a message passing system in the form of a Java Messaging System (JMS) implementation to help you get started easily, but it is fundamentally not based on any one messaging framework.

This new JMS implementation is provided in addition to the JMS transport support Artix ESB has had previously. Artix ESB supports all JMS implementations, but for customers who do not already have a messaging system that suits their needs including JMS can help get started with Artix ESB or provide a way to migrate away from another undesirable messaging solution.

## WS-ReliableMessaging Support

Artix ESB supports reliable SOAP messaging over unreliable HTTP via a runtime transport plug-in. This provides mission-critical, reliability to standard SOAP messaging, so customers do not have to extend investments in expensive proprietary reliable messaging solutions, like WebSphere MQ.

## *ENTERPRISE QUALITIES OF SERVICE*

Artix ESB provides proven enterprise services for security, systems management, performance and reliability. As our customers have learned, these sophisticated enterprise services are a mandatory feature of integration systems that involve mission-critical applications and technologies. Artix ESB provides enterprise qualities of service for mission-critical deployments that other ESBs cannot match. Artix ESB users build secure, fault-tolerant systems that are easier to manage and deploy, even when these systems involve a complex, heterogeneous mix of technologies and applications.

### **Security**

Artix ESB provides unmatched capabilities for building secure systems and enabling enterprise security across the ESB:

- > Security plug-ins for existing security mechanisms, including: packaged solutions and platform standards like Microsoft® Active Directory, LDAP and other industry standards
- > Support for RBAC, authentication and authorization functions

- 
- > Support for single sign-on across diverse ESB endpoints
  - > Support for industry standards, including WS-Security, Kerberos Security Tokens and X509 certificates
  - > Full integration with existing mainframe security software using SAF, e.g. RACF, CA-ACF/2 and CA-Top Secret
  - > TLS plug-in enables TLS applications to use a PKI system for authenticating each side of a TLS connection

See “Artix and Security” technical note for more information.

### ***Management***

Artix ESB makes every endpoint in every system manageable by in-place enterprise systems management infrastructure:

- > Management plug-ins for such enterprise systems management vendors as IBM’s Tivoli® platform, CA-WSDM, and BMC® Patrol
- > Support for Web services management standards such as WSDM
- > Support for technical management standards, including SNMP, and extensible operational logging support
- > Logging of key operational data, which includes request per interval and minimum, maximum and average response times
- > JMX instrumentation for managing services deployed in an Artix container available through any JMX client

See “Artix and Management” technical note for more information.

### ***Transaction Support***

Artix ESB can extend transactional protocols and semantics across the ESB, with support for Web services transactions specifications including WS-AtomicTransactions, two-phase commit, and one-phase commit XA transactions.

### ***Enterprise Capabilities***

Artix ESB supports demanding enterprise requirements used in mission-critical operations including the following:

- > **Routing**—support for port (transport), port property, operation and application based routing of Web service messages
- > **Locator**—supports dynamic Web services registration that can be configured to support high availability, load-balancing, failover and more.

## PROGRESS ARTIX: ADVANCED SOA INFRASTRUCTURE SUITE

Artix ESB is one component of Progress Artix, a comprehensive suite of products to streamline, modernize and lower the operating costs of complex and heterogeneous IT environments. The Artix suite includes

- > **Artix ESB**—connects any service consumer with any service providers using any middleware by deploying, managing and securing a SOA without requiring a centralized hub
- > **Artix Registry/Repository**—a phonebook-style listing of all available services with automatic provisioning and monitoring of services, to maximize reuse and ensure continued adherence to enterprise policies
- > **Artix Orchestration**—facilitates the composition of fine-grained functionality into reusable services using BPEL to create business-level services
- > **Artix Data Services**—a metadata management, data modeling, transformation and integration toolkit to abstract data services from the underlying transport and integration infrastructure

## A DIFFERENT APPROACH TO INTEGRATION

Artix ESB provides a fully distributed and standards-based approach to integration that gives architects a new set of tools to integrate disparate systems with minimal disruption to existing systems and functionality. The distributed architecture of Artix ESB is critical to the SOA approach, making it possible to provide a back-plane of services that work across the broadest

array of platforms and systems, regardless of messaging system. And WSDL provides a standard way to represent endpoint functionality throughout a complex enterprise, and an easy way to reconfigure endpoints to support new protocols and messaging transports.



---

## PROGRESS SOFTWARE

Progress Software Corporation (NASDAQ: PRGS) is a global software company that enables enterprises to be operationally responsive to changing conditions and customer interactions as they occur. Our goal is to enable our customers to capitalize on new opportunities, drive greater efficiencies, and reduce risk. Progress offers a comprehensive portfolio of best-in-class infrastructure software spanning event-driven visibility and real-time response, open integration, data access and integration, and application development and management—all supporting on-premises and SaaS/cloud deployments. Progress maximizes the benefits of operational responsiveness while minimizing IT complexity and total cost of ownership.

## WORLDWIDE HEADQUARTERS

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA  
Tel: +1 781 280-4000 Fax: +1 781 280-4095 On the Web at: [www.progress.com](http://www.progress.com)

Find us on  [facebook.com/progresssw](https://facebook.com/progresssw)  [twitter.com/progresssw](https://twitter.com/progresssw)  [youtube.com/progresssw](https://youtube.com/progresssw)

For regional international office locations and contact information, please refer to the Web page below:  
[www.progress.com/worldwide](http://www.progress.com/worldwide)

Progress, Artix, Artix Orchestration, Orbix, and Business Making Progress are trademarks or registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and other countries. Any other marks contained herein may be trademarks of their respective owners. Specifications subject to change without notice.

© 2009, 2010-2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Rev. 07/11 | 6525-128137