WHITE PAPER

CORBA AND A SERVICE-ORIENTED ARCHITECTURE (SOA)

A CORBA Integration Planning Guide

EXECUTIVE SUMMARY

Many enterprises rely on CORBA for its high level of service availability, transactional integrity and security for their mission-critical applications. They are also implementing service-oriented architectures (SOAs) to facilitate system reuse and increase business agility.

Reconciling the two is challenging. For most organizations, rip-and-replacing deployed infrastructure is not an option and custom integration for legacy systems is too expensive. To implement demanding line-of-business objectives, architects need a solution that easily extends existing CORBA infrastructure components to work with other enterprise technologies.

Artix delivers on these goals. Artix is an extensible Enterprise Service Bus (ESB) that dramatically reduces operating costs for organizations with complex and heterogeneous IT systems by deploying, managing and securing a SOA without requiring a centralized hub. Artix uses distributed computing technology to leverage and modernize existing middleware investments to help the Global 2000 deliver products and services to their customers faster and more efficiently. Artix is

- **For incremental SOA adoption** – Artix creates a network of smart, standards-based endpoints using existing infrastructure so enterprises can begin with low-risk, high-value SOA projects and gradually add services as needed
- **Dynamic and adaptable** – Artix endpoints are independently configurable so services can be extended, modified and hot deployed without disrupting the rest of the enterprise
- **Technology-neutral** – Artix is a multi-platform and multi-protocol solution that connects diverse and lightweight endpoints without an expensive and cumbersome centralized server, and without promoting vendor lock-in

Progress has a proven track record of delivering mission-critical infrastructure, and has built many of the earliest and largest SOAs for Global 2000 customers, including Credit Suisse, BellSouth, Raymond James & Associates, Marconi, and Deutsche Post (DHL).

This planning guide is designed to help developers, architects, and technical staff members understand how to extend existing CORBA applications using the integration capabilities of the Artix Enterprise Service Bus. The material in this paper is intended to address both Progress Orbix and non-Progress CORBA applications including Borland's VisiBroker and many other implementations.

Copyright © 2008. Progress Software Corporation. All rights reserved.

WHY EXTEND CORBA SYSTEMS?

Large organizations have a huge investment in IT infrastructure and the middleware that holds it together, but economic pressures are forcing them to look for ways to take costs out of IT operations and untangle its complexity. One approach is to replace legacy environments with other newer alternatives such as J2EE or .NET, but a rip-and-replace approach is risky and difficult to execute.

CORBA is a proven solution for distributed computing and integration problems, and its high performance, dependability, scalability, and great flexibility have overcome challenges that were previously insolvable. Many CORBA-based systems have been running successfully in production environments for years and their owners are reluctant to "fix what ain't broke."

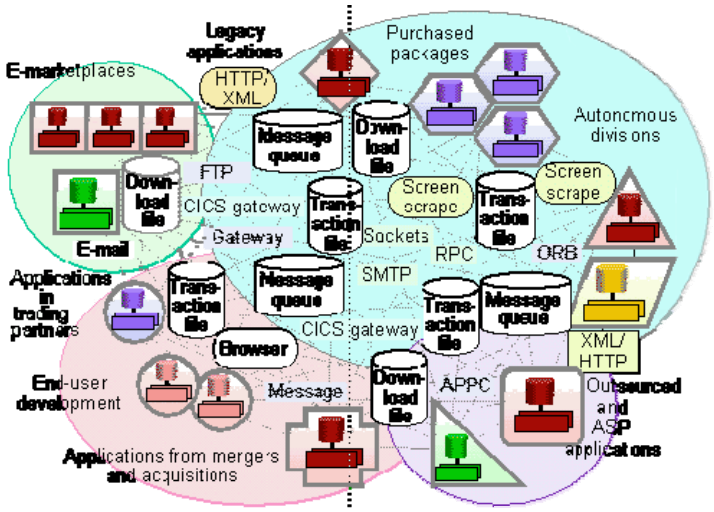

Figure 1: Today's large enterprise uses several middleware technologies

To further complicate the integration landscape, many other middleware products also continue to be successful within the enterprise. As shown in Figure 1, the typical enterprise infrastructure is a hodgepodge of different technologies, each operating within its own silo.

IT organizations are faced with daunting integration challenges. Point-to-point connections are one option, but the effort has to be repeated every time the system is reused. Replacing applications with new technology is impractical. A safer and more efficient approach is to establish a common IT architecture that bridges all systems – past, present, and future.

A service-oriented architecture (SOA) is the architecture of choice for rapid integration across diverse entities. By service-enabling existing components, all systems can communicate efficiently and transparently creating an agile

Copyright © 2008. Progress Software Corporation. All rights reserved.

environment that encourages system reuse. By extending and service-enabling CORBA components, CORBA servers can access other types of middleware and CORBA clients can access services developed with other technologies including Tuxedo, Tibco, MQ applications, and other assets already in place today.

TECHNICAL CHALLENGES

Although Web Services and CORBA are similar, they are by no means the same. Developers trying to service-enable CORBA applications face a number of technical challenges including

- **Differences in Application Protocols** - Getting two applications to work together requires the mapping of the concepts, functions, or data native to each application, and identifying the abstractions under which each application can perform correctly and efficiently. In CORBA these abstractions are expressed in IDL and in Web services these abstractions are expressed in WSDL[1].
- **Differences in Granularity**– For many CORBA applications, fine-grained object interfaces are the norm but SOA encourages modeling services as course grained business objects. While modeling differences is subtle, being more a matter of usage style, developers need to map fine-grained interfaces to much coarser grained services.
- **Supporting Object References –** CORBA object references are handles that identify an object instance within the ORB, and are needed to invoke methods on objects. Object references are not global identifiers that are valid across all machines in a distributed network. Web services references are based on URIs Integrating CORBA with other technologies requires mapping object references to the equivalent in other systems.
- **Supporting Factory Patterns** - A factory is a CORBA object that returns another CORBA object reference via one of its CORBA operations. There are many different types of factories as many CORBA services are defined as factories.
- **Mediating Stateful versus Stateless Systems** – CORBA applications sometimes have objects that maintain an internal set of values. In systems that are inherently stateless, as are Web services and many middleware systems, it is hard to duplicate or emulate this behavior.
- **Supporting Client Callbacks** – A CORBA client can pass an object reference to the server, enabling the server to make a method call back to a client-side object. This callback capability is useful if the client needs to be notified of some event on the server. It is also useful if the server needs customized client functionality to fulfill its implementation.

[1] Progress has been on the forefront of establishing standardized bindings within OMG WSDL/SOAP to IDL and IDL to WSDL/SOAP, and is developing a new version of the WSDL bindings for CORBA and a new WSDL to C++ mapping standard. Refer to http://www.omg.org/technology/documents/corba_spec_catalog.htm

Copyright © 2008. Progress Software Corporation. All rights reserved.

- **Supporting Enterprise Qualities of Service (QoS)** – Mission critical applications must have support for enterprise services. Mapping CORBA services such security, transaction support and high availability to their equivalent capability in other middleware systems is required for successful integration.

CORBA AND ARTIX

Artix is specifically designed for heterogeneous environments and is helping organizations leverage their existing CORBA and other IT investments as they move toward an SOA. The Global 2000 companies who use Artix today value it for its unique combination of

- **Extensibility** – Artix extends CORBA applications to interoperate with other enterprise systems and to support multiple high-performance protocols without requiring changes to existing systems or infrastructure components
- **Broad Platform Support** - Artix successfully leverages Progress' experience to deliver an ESB with the broadest support for enterprise technology. From mainframes to mobile devices, Artix runs natively inside a wide range of operating and application platforms and bridges the protocols and data formats of diverse, legacy systems
- **Enterprise Qualities of Service (QoS)** – Artix seamlessly integrates enterprise security models - including LDAP, Active Directory, RACF and third-party solutions - without hand coding. Artix enhances existing CORBA installations with value-added QoS services including enterprise management systems (EMS) support, SNMP support, operational logging, load balancing and session management for mainframe-class reliability, availability, and scalability

Artix is able to support numerous existing systems and multiple generations of architecture and technology without sacrificing performance or quality of service because of its extensible plug-in architecture. Rather than force applications to conform to predetermined transport and payload standards, Artix service-enables existing systems by extending the endpoint with targeted plug-ins, where an endpoint is defined as a service provider or a service consumer. Plug-ins add support for any protocol, transport, data model, security standard, development platform, etc., with a minimum footprint and little deployment overhead, while protecting the systems sovereignty as shown below.

Copyright © 2008. Progress Software Corporation. All rights reserved.

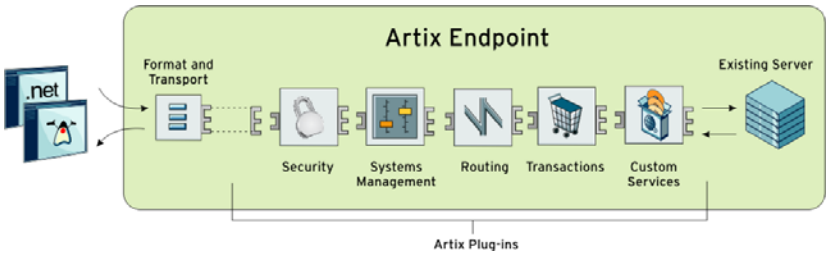

Figure 2: Artix is built on extensible plug-in technology

Service-Enabling CORBA Applications

Any existing CORBA application can be extended to support a Web services interface, where it can interoperate with other systems within an SOA.

Extending CORBA applications in this way has the following benefits:

> **Increasing reuse** – enables users to leverage the value of existing working systems rather than reinvent the wheel to get access to the same data

> **Broader accessibility** – CORBA Services can be accessed by applications in other business that may be using different technology

> **Increased agility** – by ascribing to SOA principles organizations are better able to respond to changes in the business environment

The Role of WSDL

The Web Services Definition Language (WSDL) plays a central role in Artix. WSDL defines the interface of any endpoint independent of the message format or transport used, providing Artix with a standard way of bridging middleware domains. WSDL defines the interfaces (or port types) and operations for a Web service. In this respect, the WSDL contract is analogous to an IDL interface in CORBA.

The WSDL contracts contain more than just interface definitions. As shown in Figure 3, it defines the binding, the wire-level data format, which together is called the physical contract. It also specifies a logical contract, which is the public interface that describes the interfaces at the application level. This logical contract is independent of transports, data formats, and programming languages.

Copyright © 2008. Progress Software Corporation. All rights reserved.

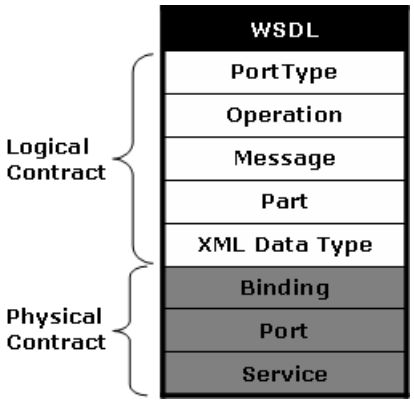

Figure 3: Service contracts specify physical and logical contracts

Using WSDL as an interface definition language, Artix provides the development and runtime tools to implement SOA across diverse middleware – including CORBA. The Artix Designer manages the mapping between WSDL and CORBA IDL. Figure 4 compares the elements of WSDL and CORBA IDL.

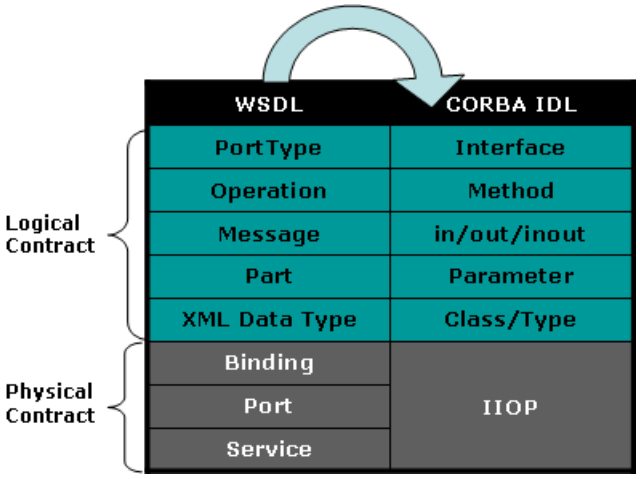

Figure 4: Mapping WSDL service definition to CORBA IDL

Alternate Transports

Protocol independence is a key feature of the Artix architecture. With the help of the plug-in development APIs, it is possible to extend Artix to support any custom protocol. Furthermore, Artix supports important combinations of payload and transport for CORBA customers including SOAP over IIOP which allow users to overcome inherent limitations in HTTP such as the ability to interleave requests/responses (i.e., request/reply correlation) while still being able to use high level tools to process soap requests.

Mapping Data Types

Artix provides extensive support for IDL to WSDL data mappings found in many implementations of CORBA (including Orbix Gen 3 and non-Progress CORBA versions). Data types must be mapped from IDL to WSDL. For primitive types, the mapping is straightforward. However, complex IDL types such as structures,

Copyright © 2008. Progress Software Corporation. All rights reserved.

arrays, unions, typedefs, arrays, sequences, and exceptions require more detailed descriptions to preclude ambiguity. Artix also supports mapping recursive types as well as abstract data types. Appendix A shows the way Artix handles the mapping of basic data types among WSDL, C++, Java and IDL.

Object References

CORBA object references identify an object instance within an ORB. Artix references provide a means of passing a reference to a service between two operations. Since Artix services are Web services, their references are very different than references used in CORBA. Artix provides a special mechanism for passing Artix references to CORBA applications over the Artix CORBA transport.

Handling Exceptions

CORBA applications can cause an ORB to raise a number of system exceptions when errors occur in the processing of requests from a CORBA client application. To facilitate interoperability between CORBA applications and Artix applications, Artix automatically maps between CORBA system exceptions and Artix faults.

Artix Router

Directing messages among endpoints and bridging the differences between protocols is the job of the Artix Router. Figure 5 shows an example of a SOAP/HTTP-to-CORBA router. The router is a plug-in that implements a general-purpose protocol bridge. Unlike proprietary integration products, no configuration or coding is required in the router and everything is encapsulated in the WSDL contract. In this example, it translates incoming SOAP/HTTP request messages into outgoing IIOP request messages. On the reply cycle, the router translates incoming IIOP reply messages into outgoing SOAP/HTTP reply messages.

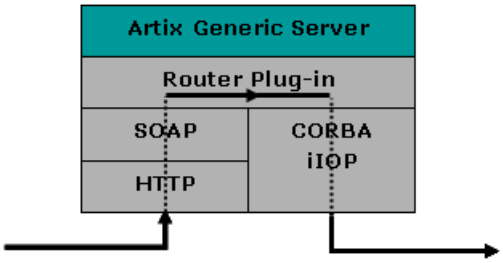

Figure 5: SOAP/HTTP-to-CORBA Router

Artix routers can be configured as either standalone processes, or for Orbix 6.x users, the router can be embedded in the Orbix executable. A standalone router is a noninvasive approach giving any Web services client to access to any CORBA server without requiring changes to either the client or the server.

Integrating Callbacks and Factory Patterns

Callbacks are one of many CORBA programming patterns supported by Artix. Since the server initially does not know about the client's callback object, the

Copyright © 2008. Progress Software Corporation. All rights reserved.

client must send details of this to the server. Once the server receives this, it is able to call back on the client's callback object at any later time. In Artix, the details about a service are encapsulated in an Artix reference. Figure 6 shows how existing CORBA clients can continue to communicate with a Web Services server by interposing an Artix router to translate between the IIOP and SOAP/HTTP protocols.

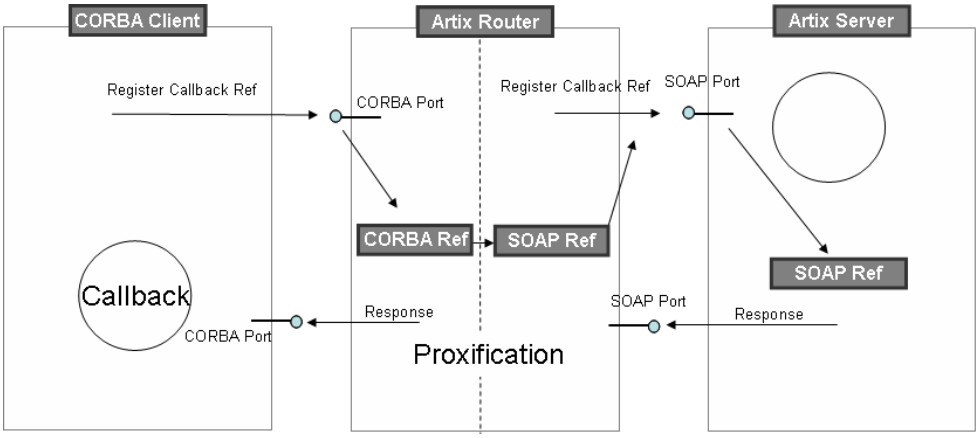

Figure 6: Artix router bridges services that pass references for callbacks

Dynamic proxy services are used when the Artix router bridges services that have patterns such as callback, factory, or any interaction that passes references to other services. When the router encounters a reference in a message, it proxifies the reference into one that a receiving application can use. Proxification means that the router automatically creates a new route for the references that pass through it. For example, an IOR from a CORBA server cannot be used by a SOAP client, so a new route is dynamically created for the SOAP client.

If we take this scenario one step further, we see how CORBA applications can be accessed using any of the application technologies supported by Artix. Figure 7 shows an existing CORBA application, which performs a search for customer information and returns the information to the requestor. By exposing an existing CORBA application as a service using Artix, applications written in a broad range of application technologies can access the data directly from the CORBA server. All of this is accomplished without making any changes to the existing application.

Copyright © 2008. Progress Software Corporation. All rights reserved.

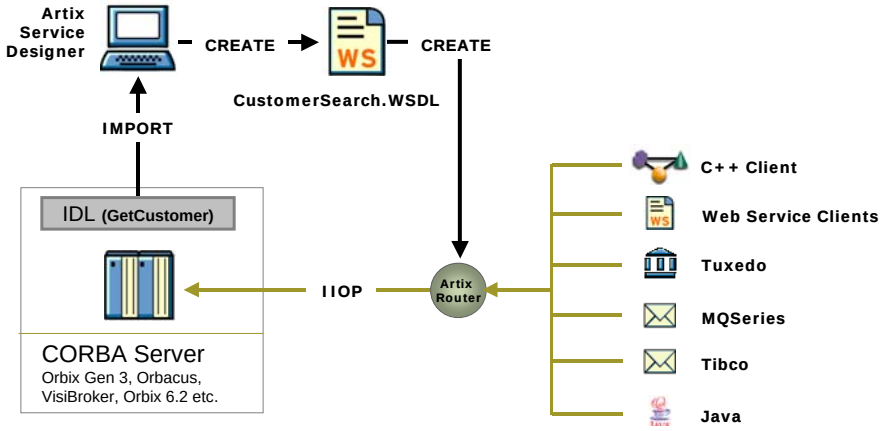

Figure 7: Existing applications accessible to other applications

Figure 7 shows the steps required to service-enable a CORBA application using a SOAP/HTTP Client as an example. The steps include

1. **Creating the logical contract:** the Artix Service Designer tool imports IDL to automatically generate the logical data model and interface (called Port Type) in WSDL
2. **Creating the Physical Contract:** the Artix Service Designer is used to add the physical elements, including a binding for each of the required client application data payloads (e.g. CORBA, SOAP), and a port definition that describes specific connection details o f the endpoint (an endpoint for the CORBA system might include an IOR or a reference to the Naming Service while an HTTP endpoint will be a URI, for example)

Accessing Web Services from CORBA Applications

Another capability gained from extending CORBA applications is the ability to access services developed on non-CORBA systems from CORBA applications. It is relatively straightforward to expose a Web service through Artix by adding the configuration of the relevant CORBA bindings to the WSDL contract.

The ability to access other technologies from CORBA has the following benefits:

> **Extending the life of CORBA applications:** Because applications now have better access to data and functions built on non-CORBA technology they can be more easily extended and therefore are more cost effective to keep in service.

> **Broader range of technology options:** Artix can accommodate several combinations of technology providing the best technical fit for an application. For example, where HTTP transport has certain limitations, delivering data as SOAP over IIOP may provide better performance and scalability than SOAP over HTTP. With Artix this is an easy configuration change.

Copyright © 2008. Progress Software Corporation. All rights reserved.

> **More timely:** CORBA applications do not have to rely on intermediate systems to participate in a business process. Since the CORBA applications can now invoke directly on the applications that participate in a business function there is less latency between the systems.

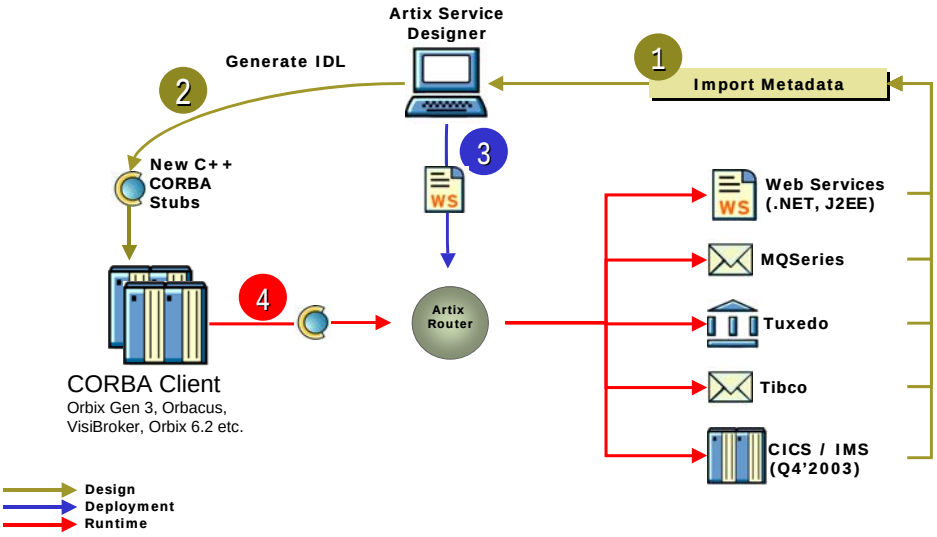

Figure 8: Accessing other middleware from CORBA

Figure 8 illustrates the 4 steps required to access non-CORBA systems form CORBA applications:

1. **Create WSDL contract**: The Artix Service Designer imports metadata from existing systems and generates a Web services contract for each system. A CORBA binding and port are added to these contracts to support a CORBA client application. A routing definition is also added that describes what the Artix Router has to do.

2. **Generate IDL from WSDL**: Artix generates IDL from the logical service contract. The generated IDL can then be used in the CORBA environment to create the appropriate CORBA client proxy for this service.

3. **Deploy Artix Router**: The Artix Router reads the WSDL contract and determines at runtime what routings need to be performed. Note that the Artix Router may be co-located with service endpoint or reside on the CORBA client side.

4. **Run the client:** The CORBA application accesses the CORBA endpoint defined in the contract and invokes the Artix Router that transparently routes the call to the backend application using the appropriate transport and payload format. Note there is no dependency on a specific backend technology introduced to the CORBA application.

Copyright © 2008. Progress Software Corporation. All rights reserved.

Integrating Enterprise Qualities of Service (QoS) and CORBA

While Artix provides extensive support for basic interoperability between CORBA and other supported middleware environments, it also offers support for integrating many of CORBA's enterprise services. As shown in Figure 9, enterprise service plug-ins allow Artix to provide mediation for qualities of service among several middleware domains.

Services	Other Middleware Requirements	Services Mediation	CORBA Systems Requirements
Security Requirements	Transport layer, Authorization, Authentication, Single sign-on	← →	Transport layer, Authorization, Authentication, Single sign-on
Systems Management	Correlated faults, automated recovery, diagnostics, response time tracking		Correlated faults, automated recovery, diagnostics, response time tracking
Session Management	State-full and Stateless		State-full and Stateless
Transaction Management	Guaranteed message delivery Coordinated activities		Guaranteed message delivery Coordinated activities
Resiliency	Load balancing, failover, disaster recovery		Load balancing, failover, disaster recovery

Figure 9: Mediating services

Artix provides a means for taking advantage of existing CORBA services while transmitting messages using a payload format other than CORBA. For example, Artix allows you to send fixed format messages to an endpoint whose address is published in a CORBA naming service. Table 1 provides an overview of critical enterprise services supported by Artix and their benefit to CORBA users.

CORBA Service	Description	Artix support	Benefit
Name Service	A name service is a directory that provides a mapping between names and object references. When an object is created, it is assigned an object reference which can be bound with a CosNaming::Name name into the namespace managed by the naming service.	Use of the IIOP tunnel or IIOP transports allows Artix to send messages to an endpoint whose address is published in a CORBA name service.	Artix can integrate CORBA applications that use CORBA naming service with non-CORBA applications that use other naming schemes.
Object Transaction Service	The OTS, a transaction service, provides a mechanism for distributed	Artix provides declarative control over transactional queuing and	Artix allows you to adopt a unified approach to

Copyright © 2008. Progress Software Corporation. All rights reserved.

(OTS)	CORBA Objects to participate in a distributed transaction through a two-phase commit protocol.	backout/exception processing across multiple transactional systems including CORBA's OTS. Users are free to use any supported transaction manager to control a distributed transaction although OTS is included with Artix and is transaction manager of choice for CORBA and J2EE applications.	transactions management across middleware systems. For example, with Artix you can have a CORBA endpoint, an MQ Endpoint and a Tuxedo endpoint all participate in the same transaction.
Security	CORBA applications have three levels of security: a transport layer called IIOP/TLS which provides secure communication between client and server; Common Secure Interoperability (CSI) which provides a mechanism for propagating username/password credentials; and the GSP which is concerned with higher-level security features such as authentication and authorization.	The Artix security framework is a comprehensive security framework that supports authentication and authorization using data stored in a central security service (the Artix security service). Refer to the technical note *Artix and Security* for more information.	Artix allows you to adopt a unified approach to security. A CORBA application can interoperate with applications that use a variety of security schemes for transport, authorization and authentication.
Systems management	CORBA applications need to be managed by Enterprise Systems Management products.	Artix end points are designed to integrate with Enterprise Management Systems such as IBM Tivoli, HP OpenView, CA Unicenter, or BMC Patrol.	Artix allows you to adopt a unified approach to systems management that includes CORBA and non-CORBA systems.
Session Management	CORBA applications may require lasting connections and stateful interactions.	Artix Session Manager supports stateful service providers by allowing a service consumer to establish a session that guarantees exclusive access to a single stateful service provider over a series of interactions. It is responsible for connection	The Artix session manager provides users with a way to integrate endpoints that require lasting connections.

Copyright © 2008. Progress Software Corporation. All rights reserved.

		management, including session leasing, session renewal, and session recovery. Refer to the technical note *Artix and Management* for more information.	

Table 1: CORBA and Artix Services

CONCLUSION

Historically, Progress helped its customers build complex, enterprise class CORBA systems even before the CORBA specification was fully capable of supporting enterprise requirements. The success of CORBA along with the success of many other technologies has introduced a new problem. For service-oriented architecture to succeed, CORBA systems will need to interoperate with other technologies. With Artix, Progress is alone in providing a single integration platform that employs Web services to deliver interoperability among CORBA, J2EE, .NET, mainframes and many other systems.

Artix customers are deploying Artix in projects that conform to at least one of three usage scenarios:

- **Universal Access** - By eliminating certain technology dependencies (e.g., CORBA client code), you can leverage CORBA services for multiple channels (Web, etc.) and reduce your cost structure by leveraging run-time support included at no charge in today's platform technologies.
- **System Migration** - Replacing existing systems using a big bang, rip and replace model is too disruptive and impractical. By masking physical deployment technology behind a well-understood, standard contract language (WSDL), organizations can encapsulate depreciated software assets for either permanent encapsulation or for incremental testing to facilitate a phased migration.
- **Process Automation** – Automation of business processes enables organizations to react much more quickly to changing business requirements or market conditions. By exposing CORBA applications as services accessible to a broad range of middleware, Artix allows CORBA systems to interoperate with a broad range of business process management tools without forcing development teams to learn other new technologies. Artix does much more than Web service-enable CORBA endpoints. It helps users overcome difficult technical hurdles, which are mandatory for proper interoperability.

The benefits of extending CORBA application using Artix include:

Copyright © 2008. Progress Software Corporation. All rights reserved.

- **Increasing operational efficiency** – Corporations look to automate common business processes without having to make a large-scale investment.
- **New business opportunities** – Competitive pressures force organizations to constantly provide new and better service to customers without disrupting existing services.
- **Reducing IT expenses** – With over 75% of IT budgets being consumed by fixed-cost expenses, any infrastructure decisions that reduce these costs liberate funds for new initiatives.

Progress is determined to help its customers extend and preserve their CORBA investments through new technologies such as Artix. And with its decade-long support of the evolving CORBA platform, Progress is uniquely qualified to help customers leverage new technologies to continue to build scalable and reliable CORBA systems.

APPENDIX A: DATA TYPE MAPPINGS

IDL Type	XML Schema Type	CORBA Binding Type	Artix C++ Type	Artix Java Type
Any	xsd:anyType	corba:any	IT_Bus::AnyHolder	com.iona.webservices.reflect.types.AnyType
boolean	xsd:boolean	corba:boolean	IT_Bus::Boolean	boolean
char	xsd:byte	corba:char	IT_Bus::Char	byte
wchar	xsd:string	corba:wchar		java.lang.String
double	xsd:double	corba:double	IT_Bus::Double	double
float	xsd:float	corba:float	IT_Bus::Float	float
octet	xsd:unsignedByte	corba:octet	IT_Bus::Octet	short
long	xsd:int	corba:long	IT_Bus::Long	int
long long	xsd:long	corba:longlong	IT_Bus::LongLong	long
short	xsd:short	corba:short	IT_Bus::Short	short
string	xsd:string	corba:string	IT_Bus::String	java.lang.String
wstring	xsd:string	corba:wstring		java.lang.String
unsigned short	xsd:unsignedShort	corba:ushort	IT_Bus::UShort	int
unsigned long	xsd:unsignedInt	corba:ulong	IT_Bus::ULong	long
unsigned long long	xsd:unsignedLong	corba:ulonglong	IT_Bus::ULongLong	java.math.BigInteger
TimeBase::UtcT	xsd:dateTime	corba:dateTime	IT_Bus::DateTime	java.util.Calendar

Copyright © 2008. Progress Software Corporation. All rights reserved.

Worldwide Headquarters
Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA
Tel: +1 781 280-4000 Fax: +1 781 280-4095
www.progress.com

For regional international office locations and contact information, please refer to www.progress.com/worldwide

© Copyright 2008 Progress Software Corporation.
All rights reserved. Progress, Actional, Apama, DataXtend, Shadow, Sonic ESB, and SonicMQ 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 trademarks contained herein are the property of their respective owners. Specifications subject to change without notice.

© Copyright 2008 IONA Technologies PLC, a wholly-owned subsidiary of Progress Software Corporation.
All rights reserved. IONA, IONA Technologies, the IONA logo, Orbix®, High Performance Integration, Artix® , FUSE™ and Making Software Work Together® are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that may appear herein are the property of their respective owners.