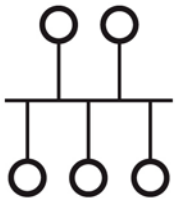

Orbix® 6.1

Technical Overview

IONA Technologies PLC
December 2003

Table of Contents

1 Introduction

CORBA makes software work together regardless of where it is located, what it is running on, or what language it's written in. Today, CORBA not only is mainstream technology, but it is mainstream technology in traditionally conservative IT environments, including banking, insurance, utilities, manufacturing, and government. IONA Technologies built its business around CORBA; Orbix has been in existence almost as long as CORBA.

Orbix is by far the market leading implementation of the CORBA standard. It facilitates interoperation of applications across network, language, CPU, and operating system boundaries. Beyond the implementation of the standard, it also provides enterprise-class features that reside at the core of thousands of distributed systems around the world. Orbix routinely handles integration and scalability problems in the most complex and largest systems.

Between the extremes of everyday commercial data processing and hard real-time control applications, Orbix operates across the entire spectrum of computing tasks, from billing systems and multi-media news delivery to airport runway illumination and aircraft radio control. Most of the world's telephone systems, as well as the truly mission-critical systems operated by the world's biggest banks, are built on Orbix.

Orbix has always been attractive to IT organizations in industries that need large, resilient systems to handle enormous peak volumes of data and service requests, while guaranteeing a high level of availability. Financial services companies constitute the most transaction-intensive industry in the world. For this reason, nearly 70% of the financial organizations listed in the Fortune Magazine Global 100 rely on Orbix. Orbix has also made similarly impressive inroads in aerospace, government, high-tech manufacturing, and several other industries. More developers today have used Orbix than any other ORB. This creates a larger pool of experience on which development organizations can draw, which lowers recruiting and training costs.

IONA's dominance of the CORBA market positions us to provide the best and broadest support for CORBA implementations. IONA now offers five-year support up front—so that your long-term deployment plans have the backing of IONA's support services. IONA now guarantees binary compatibility for future versions of Orbix 6, allowing an easy upgrade.

IONA supports CORBA on more hardware and operating system platforms than any other CORBA vendor. IONA's continuing support for the latest operating systems and compilers allows you to take advantage of the latest performance improvements in hardware. This technical white paper is a testament to the breadth and depth of features that are available in Orbix 6.1.

IONA's core competence is the production and support of Service Oriented Architecture (SOA)-based middleware that is ideal for extending CORBA to other technologies and standards. IONA is now applying the same expertise learned from 4500 deployments to other middleware technologies that have emerged since CORBA.

IONA now provides:

- Native .NET to CORBA Connector—bringing enterprise CORBA scalability to .NET
- Tools to easily expose CORBA objects as Web Services
- Orbix Connect—J2EE to CORBA connector
- Orbix Mainframe—turn CICS and IMS transactions into CORBA services

2 ORB Technical Overview

Orbix allows systems to communicate directly across a network, regardless of the programming languages used to create them, and regardless of the operating systems and platforms on which they run. Orbix is used to integrate applications written in languages such as Java, C++, C, Visual Basic, and COBOL, and runs on PCs, UNIX hosts, and mainframes.

Orbix is built on IONA's Adaptive Runtime Technology (ART). ART is a plug-in framework that delivers scalability, high performance, and flexibility to distributed enterprise applications, as illustrated in Figure 1.

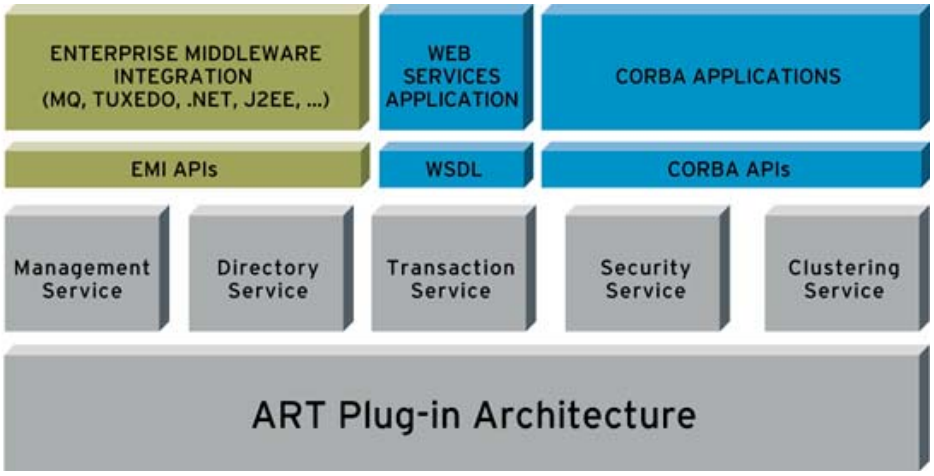

Figure 1: ART Plug-In Architecture

Enterprise qualities of service are provided by way of the Management Service, Directory Service (Naming Service), Transaction Service, Security Service and Clustering Service, all of which are built as ART plug-ins. Plug-ins can be linked directly with an application, loaded when an application starts up, or loaded on demand while the application is running. (Note that IONA's Artix products provide Enterprise Middleware Integration based on the same ART plug-in framework as Orbix).

Orbix is an Object Request Broker (ORB) that presents an abstraction layer that relieves the programmer of dealing with many of the complexities of network programming, such as thread pool control, request dispatch, connection management, and so on. The ORB creates the illusion that all objects are local objects and accessible by way of an application's native programming language.

Orbix was designed from the ground up to support enterprise-class distributed systems. Orbix provides major advances in scalability, performance, and deployment flexibility, thanks in part to its plug-in framework, which delivers greater flexibility and scalability to distributed applications than has ever before been possible.

2.1 Summary Of Features

- Fault tolerance and dynamic load balancing
- Data security for distributed applications by ensuring authentication, privacy, and integrity for communication over TCP/IP connections
- Firewall Proxy Service
- Support for the Object Transaction Service (OTS), with an integrated, distributed transaction engine
- Management Console providing mechanisms to set thresholds on critical system attributes in order to alert devices, system operators, or other software components of problems without requiring human intervention
- Both synchronous and asynchronous messaging, with event storage and playback, providing highly reliable message delivery
- Implementation of the CORBA Events, Naming, Notification, Trader, and Telecom Log Services
- Java Messaging Service and bridge between the Notification Service and JMS
- Orbix COMet, a high-performance, dynamic bridge that enables transparent communication between COM clients and CORBA servers, and which gives Visual Basic and COM developers easy access to CORBA applications running on NT, UNIX and OS/390
- CORBA objects can easily be exposed as Web Services
- *Active connection management* which automatically reclaims idle connections and reallocating them for active use

- IONA's implementation of the CORBA *Portable Object Adapter* (POA) provides high scalability for servers that contain very large numbers of objects
- Sophisticated memory-management techniques to avoid duplicating common components of object references
- Internationalization support, including support for multi-byte character sets and international operating systems
- Broad operating platform support, including Solaris, Compaq Tru64, Linux, Windows 2000/NT, OS/390, AIX, and HP-UX
- Language support for C++, Java, COBOL, PL/I, and .NET programmers
- 5-year customer support available up-front

2.2 Orbix 6.1 New Features.

2.2.1 Native .NET to CORBA Connector

- Enterprise CORBA scalability for .NET
- VB, C#, J# and managed C++ clients can call CORBA and J2EE Servers
- Security, transaction, load balancing and fault tolerance features available to a normal Orbix C++ client in Orbix Enterprise
- Standard .NET API-no CORBA knowledge required
- Non-intrusive to existing CORBA systems
- No additional hardware or software resources required

2.2.2 Security

- Single Sign On-after authenticating to a secure service: clients receive a token that can be used for subsequent access to other services, giving extra security because passwords are sent only to the SSO server.
- Smart card and Smart Token support using the Microsoft Crypto API (MS CAPI) through the Schannel toolkit. Orbix TLS C++ now also supports the Windows certificate store, making it easier to manage than certificate files.
- SSL toolkit replaceability
- IONA Security Service now secures the Configuration Repository (CFR)

2.2.3 Management

- Integration with the IBM Tivoli Enterprise Management System, allowing you to use all the features of Tivoli to manage your CORBA infrastructure
- Performance Data Logging—Orbix produces logs that record management data including server throughput and response times

2.2.4 General

- Bi-directional GIOP is compatible with GIOP 1.2 and supports bi-directional invocations on Orbix 3.x callback references
- CORBA reflection, which is the ability to discover interfaces at runtime, and to call these discovered interfaces dynamically
- Improved Java connection scalability through exploitation of Java NIO
- Notification Service performance improvements
- Support for Linux

3 Orbix Standard

Orbix is available in Standard and Enterprise versions. In both versions, Orbix provides an implementation of the CORBA 2.5 specification, support for C++ and Java language bindings, and the features and benefits described in the sections below. See Section 4 for information about the additional features provided in Orbix Enterprise.

3.1 .NET Connector

IONA's .NET Connector is the successor to IONA's COMet, a COM/CORBA Interworking offering (COMet is still bundled with Orbix 6.1). .NET Connector allows .NET applications to connect to and use CORBA servers as if they were native .NET objects. Like COMet, .NET Connector can take advantage of all the CORBA enterprise features and services that are available to an ordinary C++ client, as illustrated in Figure 2.

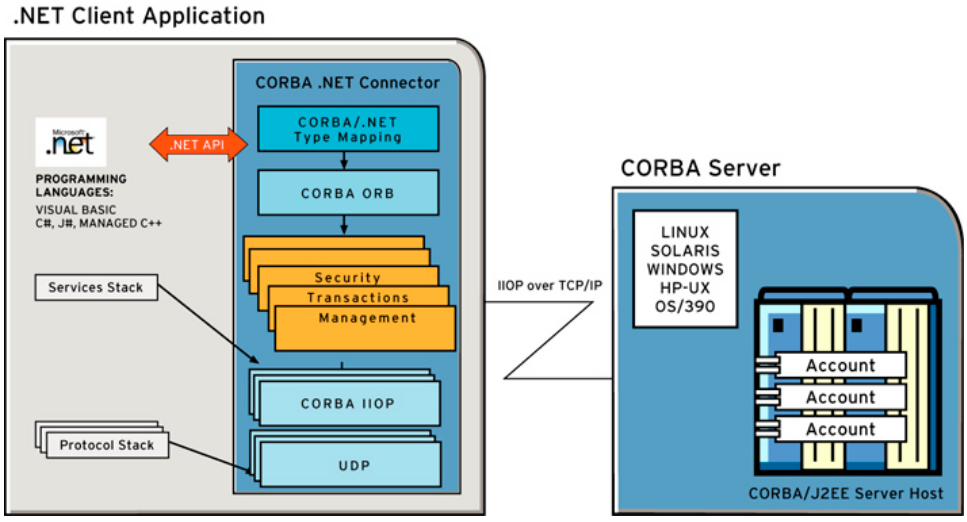

Figure 2: IONA's .NET Connector

.NET Connector is a dynamic bridge that resides on the client side and works seamlessly with the .NET client. It appears to work just like any other .NET remoting channel.

To use .NET Connector, a client first registers the remoting channel that it wishes to use. The custom remoting channel provided by the .NET Connector is registered in the same way as any other .NET remoting channel. Once the remoting channel is in place, the client creates a remote proxy object, and invokes methods on it.

The marshalling code in .NET Connector then marshals this message into a Dynamic Invocation Interface (DII) CORBA request object, which is invoked on the remote CORBA server. In the marshalling process, the out parameters are converted into .NET types and a return message is created and passed back to the caller with any `out`, `inout` or `return` parameters.

In order to use CORBA interfaces defined using OMG IDL in .NET applications, a mapping must be defined between the various CORBA types and their .NET counterparts as defined in the Common Type System. Mappings are provided for:

- CORBA Interfaces
- Interface Inheritance
- CORBA Structs, Enums, Unions, Arrays, and Sequences
- CORBA System Exceptions
- CORBA User Exceptions
- CORBA Object References
- CORBA Modules
- IDL Constants

3.2 Web Services Technology in Orbix

Web services support is provided with Orbix, and delivers the most complete and easy-to-use platform for deployment of CORBA applications as Web services.

Orbix Web services technology is built on a foundation of open standards including XML, SOAP, WSDL, UDDI, and HTTP(S). It provides a non-programmatic, graphical environment for defining, assembling, and integrating Web services from existing resources (Java, J2EE, and CORBA). It includes graphical tools that let you create new Web services.

Orbix Web services technology provides secure communication through TLS/SSL, as well as full interoperability with Microsoft's .NET architecture. It even lets you generate J2ME-ready Web service client applications for embedded systems (for example, wireless applications on hand-held devices).

Orbix's features greatly enhanced Web service support for CORBA. C++ or Java-based CORBA applications can be extended and re-purposed by exposing CORBA application functions as Web Services, without application modification. Additional CORBA-related features include:

- Complete CORBA Datatype support, including CORBA Any
- Support for CORBA exceptions
- Support for CORBA Principals
- Support for Orbix 3 format Interoperable Object References (IORs)
- Support for IONA Orbix 3.x and IONA Orbacus 4.1
- CORBA Security Support: TLS/SSL dispatching to CORBA systems, and the propagation of CORBA Security Contexts from SOAP clients to backend CORBA systems

In addition, Orbix Web service support provides:

- Improved Data Type Support, including multi-dimensional arrays, Java and EJB Exceptions, and Soap Encoding (soapenc)
- Internationalization, including support for non-ASCII data encodings and for localized operating systems (for example, the Japanese version of Windows)
- Improved Usability, including enhanced GUIs, clipboard support, and automatic UDDI deployment

3.3 Bi-Directional GIOP

The Orbix 6.1 GIOP plug-in supports the CORBA standard for connection establishment between client and server. Typically, a client can open a connection to a server through a firewall, but it is not possible for the server to open a new connection back to the client in order to send a "callback."

Bi-directional GIOP is a simple and efficient solution to this problem, and allows connections from the client to the server to be reused for callbacks from the server to the client. It is applicable over any connection-oriented transport, such as IIOP, IIOP/TLS or SHMIOP.

Orbix bi-directional GIOP is compatible with GIOP 1.2 (that is, it is not dependent on a GIOP 1.4 NegotiateSession message) and supports bidirectional invocations on Orbix 3 callback references.

In bi-directional communications secured by TLS, the connections are validated by mutual authentication. In a future release, anti-spoofing challenges will be supported (this requires the completion of the GIOP 1.4 specification).

3.4 Asynchronous Messaging Interfaces

Orbix implements some key features of the CORBA messaging specification from CORBA 3.0. Asynchronous messaging interfaces (AMI) allow clients to make type-safe, asynchronous invocations of normal CORBA operations. The IDL compiler generates an AMI stub from a normal synchronous interface, and clients can use the AMI stub to make a request and then do other work until the reply is ready. The ORB delivers the reply by invoking a client-supplied callback object. The clients' use of the AMI does not require any special support by the server that implements the original synchronous interface.

3.5 CORBA Reflection and Dynamic Type Support

Orbix 6.1 supports CORBA reflection, which is the ability to discover interfaces at runtime, and to call these discovered interfaces dynamically. CORBA provides reflection by way of its Dynamic Invocation Interface (DII), Dynamic Skeleton Interface (DSI), DynAny Interface, and the Interface Repository. All these features

except DynAny have been available since the early releases of the CORBA specification; DynAny completes the set of features needed to support CORBA reflection.

Thus Orbix 6.1 provides full support for handling data types that are not known at compile time. The *Interface Repository* (IFR) stores information about all CORBA types known to the system and can be queried at runtime. Clients can construct requests based on this runtime type information using the Dynamic Invocation Interface (DII), and servers can implement "universal" objects that can implement any interface at runtime, with the Dynamic Skeleton Interface (DSI).

The DynAny interface allows clients and servers to interpret or construct values based purely on runtime information, without any compiled-in data types. Together, these features are ideal for building generic object browsers, type repositories, or protocol gateways that map CORBA requests into some other object protocol. Interfaces discovered by way of DynAny can also be dynamically modified, for example, by adding or removing interface methods.

In addition, Orbix provides the Reflection module, which is used to contain all definitions related to object metadata queries. This module implements an XMLProvider interface, which has a single `get_description` operation that returns a string.

Using the interface described above, a CORBA client application can query an object for its metadata, and the client can use regular XML parsing tools such as DOM or SAX to parse the returned XML-formatted metadata.

3.6 Portable Object Adapter

Orbix servers use the Portable Object Adapter (POA). The POA is the CORBA-standard way to write portable server code. The POA provides a flexible framework for mapping abstract CORBA objects to concrete programming language objects. You can select from a variety of POA policies that control the memory/speed trade-offs in highly scalable servers. For example, using POA policies you can:

- Create server objects on demand
- Maintain a bounded cache of most-recently-used server objects
- Implement many CORBA objects of the same type with a single C++ or Java object in memory

3.7 Secure Socket Layer/Transport Level Security (SSL/TLS)

The Secure Sockets Layer (SSL) provides data security for applications that communicate across networks. SSL/TLS (Transport Layer Security) is a security protocol that sits in between various application protocols and TCP/IP.

CORBA applications communicate using the CORBA standard Internet Inter-ORB Protocol (IIOP). This application-level protocol is layered above the transport level

protocol, TCP/IP. SSL/TLS applications communicate using IIOP layered above SSL/TLS.

The SSL/TLS protocol provides connection security that has three basic properties:

- **Authentication:** The connection can be authenticated using asymmetric (public key) cryptography. SSL/TLS supports authentication based on RSA and DSS algorithms with X.509 certificates.

- **Privacy:** The connection is private. During the initial handshake, public key encryption is used to define a symmetric secret session key. Symmetric cryptography is used for data encryption.

- **Integrity:** The connection is reliable. Message transport includes a message integrity check, using a keyed Hashed Message Authentication Code (HMAC). Secure hash functions (for example, SHA-1 and MD5) are used for HMAC computations.

Security policies are set up using a single configuration file, and a set of SSL/TLS utilities are used to define the policy. Programmers can enable SSL/TLS by adding just a few lines of code to a standard CORBA application. As with standard CORBA communication, the details of the SSL/TLS protocol are hidden from programmers. However, SSL/TLS also provides an API that allows programmers to adjust the use of SSL/TLS in their own applications.

Specific Orbix 6.1 TLS features include:

- Sophisticated, mechanism-neutral API based on CORBASEC Level 2 interfaces

- Support for CSIv2 Level 0 username/password authentication, and identity propagation control fully integrated with the new IONA Security Server (iS2) Single Sign On CORBA login service

- Replaceable SSL/TLS toolkits; Orbix Java applications can use the Orbix JSSE toolkit option and a custom toolkit plug-in for C++ applications can be used (contact IONA Professional Services for more information on this option)

- Orbix C++ applications can use the following SSL/TLS toolkits: Baltimore Toolkit for C++ and Java, and Schannel Toolkit for C++

- Integration of TLS X.509 authentication with the new iS2 Single Sign On CORBA login service

- Separate Key Distribution Mechanism (KDM) component. The ORB can distribute pass-phrases to automatically-launched server applications. The server uses these pass-phrases to decrypt the relevant private key. KDM communications are fully TLS secure (encryption, privacy and integrity are guaranteed).

- An extensive X.509 C++ parsing API is supported, providing a complete IDL wrapping of X.509v3 certificates, and including X.509v3 extension

support (note that use of this API is only required for advanced applications)

- PKCS#12 container format support
- OpenSSL Command Line Utilities. These open-source tools are shipped with the product to allow customers to generate X.509 certificates

The Orbix TLS implementation also provides a framework for new cryptographic algorithms (for example, AES support) to be easily incorporated into the specification. The Orbix TLS out-of-the-box solution uses LDAP for authentication and authorization purposes.

Orbix's ORB infrastructure is itself secured using TLS, CSIv2, and integration with iS2 that provides full ACL support, which allows the system administrator to limit access to core CORBA services. By default, the Configuration Repository, Information Repository, and node daemon, as well as the Activator, Locator, Security, Management, and Naming Services, can be accessed only by a strongly authenticated user.

3.8 Event Service

Orbix implements the CORBA Event Service specification, which defines an asynchronous model to supplement the synchronous request/response model normally used for communication between CORBA client and server applications.

The CORBA Event Service introduces the concept of *events* to CORBA communications. An event originates at an event *supplier* and is transferred to any number of event *consumers*. Suppliers and consumers are completely decoupled: a supplier has no knowledge of the number of consumers or their identities and consumers have no knowledge of which supplier generated a given event.

3.9 Multi-Threading Support

Orbix provides excellent support for multi-threaded applications. Orbix libraries are multi-threaded and thread-safe, and the Orbix ORB provides standard POA policies to enable multi-threading in servers. The ORB creates a thread pool, which automatically grows or shrinks depending on the demand load. Thread pool size, growth, and request queuing can be controlled by configuration settings, without any coding.

Of course, multi-threaded applications must themselves be thread-safe. Although most platforms provide similar thread-synchronization facilities, the interfaces vary widely. Orbix includes an object-oriented thread-synchronization portability library that allows you to write portable multi-threaded code. This library is simply a layer over the native threading interfaces on each platform, so there is no clash with other portability libraries or applications that use native interfaces directly.

3.10 Configuration and Logging Interfaces

Applications can store their own configuration information in configuration domains, and access them via IDL interfaces—thus taking advantage of the infrastructure Orbix provides for ORB configuration.

Applications can also use IDL interfaces to log diagnostic messages to the logging subsystem. These messages are logged to whatever log-stream objects are registered with the ORB. Log streams for local output, file logging, and system logging (UNIX syslogd or Windows NT event logging service) are provided with Orbix. Application developers can also implement their own log streams to capture ORB and application diagnostics and send them to any desired destination.

3.11 Interoperable Naming Service and Load Balancing Extensions

Orbix supports the Interoperable Naming Service specification, a superset of the original CORBA Naming Service. The Interoperable Naming Service adds some ease-of-use features and provides a standard URL format for CORBA object references, in order to simplify configuration and administration of CORBA services.

3.12 Persistent State Service

Orbix includes an implementation of the Persistent State Service (PSS), a CORBA service that interposes a CORBA-based abstraction layer between a server and persistent data. The implementation of PSS is integrated with Berkeley DB, an efficient embedded database that is included with the ORB. Orbix uses PSS extensively in its internal operation. In addition, the PSS interface is available for use by applications.

3.13 Code Generation Toolkit

The Orbix developer kit includes a Code Generation Toolkit for rapid application development. At the heart of this toolkit is an IDL compiler integrated with the Tcl scripting language. Out-of-the-box scripts can generate a complete and operational client/server application automatically from an IDL file. The toolkit also provides a useful debugging tool: you can use an auto-generated server to debug your client, and vice versa. Advanced users can write their own code generation scripts to automate repetitive coding in a large application.

The Code Generation Toolkit is supplied with a set of ready-to-use code generation scripts. By simply running these scripts, you can do common tasks such as creating a complete client-server application from an IDL file, creating functions that print your IDL data types, and so on. The toolkit also includes libraries of Tcl procedures that allow you to create your own customized code-generation scripts.

3.14 System Management

System administration includes the installation, configuration, deployment and ongoing management of a distributed computing system. System management capabilities in Orbix Standard support test and debug, fine-tuning of applications, and operational control and support.

Orbix management facilities provide mechanisms to set thresholds on critical system attributes in order to alert devices, system operators, or other software components of problems without requiring human intervention.

Orbix provides a management framework that supports a web-based console, allowing an administrator to explore a managed application, examine and set attributes of managed entities, and invoke operations. The Orbix Management Framework also provides for integration with the major third-party enterprise management systems, such as those provided by HP, IBM, CA, and BMC. See Section 4.1 for more details.

4 Orbix Enterprise

Available in both C++ and Java for a variety of NT and UNIX platforms (including Linux), Orbix Enterprise is the most advanced CORBA product available today. In addition to all the features and benefits described above for Orbix Standard, Orbix Enterprise offers the features described in the following sections.

4.1 Orbix Management And Administration Tools

Orbix 6.1 provides administrative tools that address the biggest system management problem facing enterprises that run large-scale, mission-critical systems. The problem is that in such enterprises, operations groups must deal with hundreds or thousands of servers, many different operating systems, and an even greater number of different software vendors and products—while in a “must not fail” system environment.

Customers need an integrated management system, and the IT industry is doing its best to provide such “Enterprise Management Systems” (EMSs). The EMS market is dominated by four players: IBM Tivoli, HP OpenView, CA Unicenter and BMC Patrol.

The benefits of an EMS—integrated with the customers’ applications and infrastructure—are obvious. An integrated management system means that fault reports can be organized and correlated so that operators can find the cause of a problem, rather than being swamped by the symptoms.

Having a single management console clearly reduces the learning curve for the operations staff, and typical EMS offerings help here as well, by providing the automatic triggering of recovery actions when problems occur. In addition, an integrated EMS allows monitoring of Service Level Agreement (SLA) compliance as well as analysis of the business impact of system problems.

The IONA Management Architecture allows customers to exploit existing investment in Enterprise Management Systems. Integration with EMS is based on

a generic architecture for EMS systems integration, which provides abstract interfaces for the functions provided by third-party EMS systems, as shown in Figure 3.

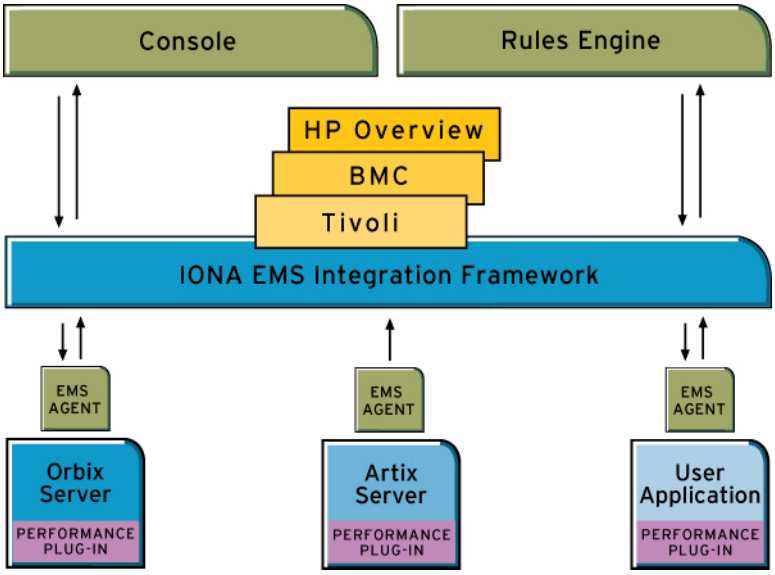

Figure 3: Orbix Management Architecture

In Figure 3, EMS components are shown in green. IONA-supplied plug-ins (shown in purple) use abstract interfaces to monitor and log key system measurements. This information is then mapped to the requirements of specific EMS systems, for example Tivoli Resource Models. (The IONA Management Architecture also supports integration with other IONA products such as Artix Relay and Artix Encompass.)

System management information for Orbix systems and applications is transferred to the Tivoli EMS (by way of the Java APIs in Tivoli) to inform the operations staff of significant events, including:

- Whether a server is alive or dead (an event is posted when a server becomes inoperative)
- Server metrics, including the number of invocations received, and average, maximum, and minimum response times (events can be generated when any of these parameters go out of specified bounds)
- Perform actions on servers; by default, these actions include start, stop, and restart, but the list of possible actions is extensible

Orbix 6.1 supports the Tivoli Management Framework (versions 4.1 and later) and IBM Tivoli Monitoring 5.1.1 (Fixpack 4). The capabilities of Orbix system management tools integrated with Tivoli in this way include:

- Setting thresholds
- Monitoring endpoints
- Detecting server crashes

- Detecting response time problems
- Viewing historical data
- Manual start and automatic restart of servers

IONA provides Tivoli installation utilities that help users go from an installed Orbix 6.1 to full integration with Tivoli in a matter of hours. These installation aids:

- Create directories for configuration files, logs, and scripts
- Generate the configuration "glue" between Orbix and Tivoli
- Generate a package to install on the Tivoli server
- Instantiate sample monitoring profiles and task libraries with pre-filled arguments

The IONA Tivoli integration can also be used as the basis for integration with other Tivoli products, including Tivoli Enterprise Console (TEC). TEC is an Enterprise Console and rules engine that correlates events and takes actions. IONA provides definitions for events and actions; the customer's production environment determines how TEC rules are applied. For example, rules can be set up such that, if the locator fails, it is restarted and a warning is issued; if it fails twice within 10 minutes an alarm is issued.

Integration with the Tivoli EMS is only the first step for Orbix. Integration with other EMS vendor offerings is an ongoing effort.

4.2 New Security Features

The Orbix Security Architecture, shown in Figure 4, supports several new Orbix security features that increase the security of the enterprise—and make life easier for developers, integrators, and users. User authentication is easier thanks to single sign on and Smart Card or Smart Token support. Developers can implement application security more easily, thanks to new CORBA security features such parameterized ACLs and CSIv2. Integrators benefit from the IONA Security Service (iS2), which makes it easy to integrate multiple Public Key Infrastructure (PKI) solutions.

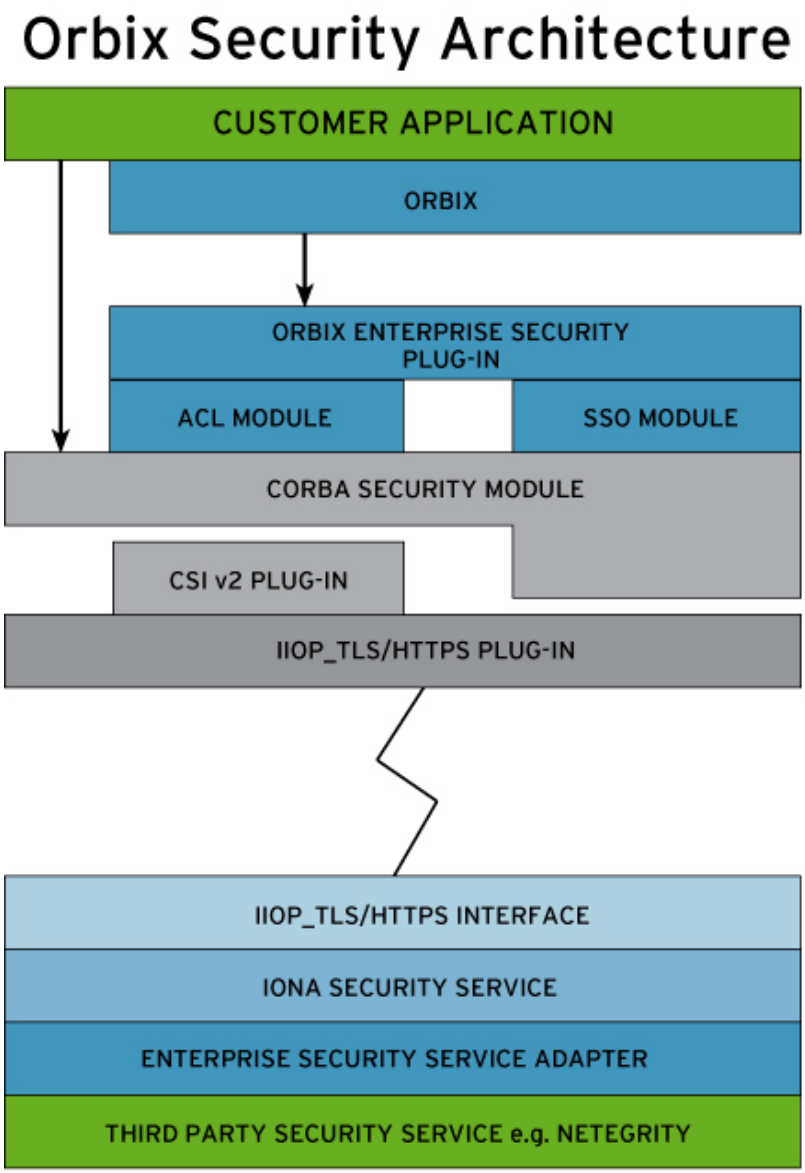

Figure 4: Orbix Security Architecture

In addition, Orbix 6.1 features enhance the security of the entire enterprise by providing support for strong authentication within the Orbix infrastructure itself, and by providing greater control over the enterprise firewall.

The IONA Security Service (iS2), shown near the bottom of Figure 4, is a scalable, standards-based security implementation with the following features:

- Pluggable integration with third-party enterprise security systems
- Out-of-the-box integration with flat file, LDAP, or Netegrity SiteMinder security systems
- Centralized management of user accounts

- Role-Based Access Control
- Role-to-permission mapping supported by access control lists
- Unified security platform works across CORBA, J2EE, and Web services
- Logging

The iS2 server provides an authentication service, an authorization service and a repository of user information and credentials. When the iS2 server is deployed in standalone mode, all kinds of application, including J2EE and CORBA applications, can call it remotely.

The following security standards are supported by iS2:

- HTTP login mechanisms—that is, HTTP basic authentication and HTTP form-based authentication
- Secure Sockets Layer / Transport Layer Security (SSL/TLS)
- CCITT X.509, which governs the form of security certificates based on public (asymmetric) key systems)
- OMG Common Secure Interoperability specification (CSIv2)
- The XML Key management Specification (XKMS), which is composed of the XML Key Information Service Specification (X-KISS), and the XML Key Registration Service Specification (X-KRSS)
- Security Assertion Markup Language (SAML)
- Secure Multipurpose Internet Mail Extensions (S/MIME)
- WS-Security
- Java Authentication and Authorization Service (JAAS)

iS2 is designed to be integrated with other authorization servers, in order to support customers who have already made a product choice or created their own authorization service.

There are two iS2-related software development kits available for separate purchase, iS2 Client SDK and iS2 Adapter SDK. The iS2 client SDK supports the connection of third-party applications to iS2, and the iS2 Adapter SDK supports the creation of new iS2 adapters.

The following new security features are available in Orbix Enterprise 6.1:

- Single Sign on support for C++ and Java applications
- Smart Card, Smart Token, and SChannel support
- Configuration Repository Authorization support
- CORBA iS2 X.509 based authentication

- Sophisticated parameterized ACL support
- CSIv2 for C++
- Securing of additional Orbix Infrastructure (Configuration Repository, Information Repository, and node daemon, as well as the Activator, Locator, Security, Management, and Naming Services)
- Pluggable C++ TLS toolkit provider support

These features are described in more detail in the following sections.

4.2.1 Single Sign-On

Single sign-on (SSO) means that an application user needs to authenticate himself only once, even if the application spans technologies and security domains. SSO significantly increases the security of password deployment and usage, because the username/password information is visible only to the CORBA Login service.

Orbix SSO support is layered above CSIv2, and supports both username/password and TLS x.509 client authentication. Clients can use SSO tokens to authentication to server applications, and TLS X.509 clients can authenticate themselves to servers that demand CSIv2 username/password authentication.

The new Orbix 6.1 CORBA Login server shares the same process as existing security service implementations by default.

CORBA clients can simply be configured to use single sign on with no code changes. In addition, SSO token expiry and automatic refreshing of SSO tokens is handled in a manner transparent to the application, which avoids the need for new exception handling code.

4.2.2 Configuration Repository Authorization

Configuration Repository (CFR) Authorization allows administrators to define access control lists (ACLs) for configuration scopes, thus preventing unauthorized users from reading/writing sensitive scopes. Replication of the CFR is supported as well.

The hierarchical nature of the Orbix configuration scoping mechanism makes it easy to define ACLs for groups of related applications that share a parent configuration scope.

4.2.3 ACL Extensions

Orbix 6.1 provides sophisticated new parameterized ACL support, which simply means that access control can be based on the values of a parameter in a method call. For example, consider a function having a currency parameter—access to the function can be permitted or denied based on the amount specified in the parameter.

The Orbix Software Development Kit (SDK) provides support for writing custom “Action Analyser” plug-ins. These plug-ins determine what resources are actually

being accessed and by what attempted actions, and the ACL enforcement mechanism ensures that the user has the required access rights. (Note that this SDK was used to create the CFR ACL model described in the previous section.)

Orbix 6.1 ACLs provide extensive new wildcarding support for server names, interfaces, and operations. In addition, shared credentials are supported, which is useful in order to simplify credential gathering for applications that use multiple ORBs. An application's credentials can be shared across all ORBs that are so configured.

Applications can use the Orbix 6.1 enhanced ACL support without changes to code.

4.2.4 CSIv2 Support

The Common Secure Interoperability Specification, version 2 (CSIv2) enables interoperable authentication, delegation, and privileges in the service context of General Interoperability Protocol (GIOP) request/reply messages, over a connection-based transport. It can also be used in environments where transport layer security mechanisms, such as SSL/TLS or SECIOP, provide message protection and authentication of a server to a client.

Orbix 6.1 provides powerful control over all aspects of CSIv2, including the acceptance and transmission of propagated identities.

4.2.5 Smart Card Support

Smart Cards are used to store user credentials more securely than on a user's machine. It is a stronger authentication mechanism because users must be in possession of a Smart Card rather than a certificate stored on a computer disk (the Smart Card plugs into a USB port or is read through a reader attached to the serial port). When employees leave a company, they must return their Smart Cards, whereas a file-based private key could be copied and transferred to a home machine.

A Java Orbix application can obtain credentials from a Smart Card as well as from a file. The API used is the standard JDK 1.4 JSSE API with cryptographic support provided via the JDK JCE interface. There are a number of implementations of the JCE modules that support cryptographic hardware devices available from 3rd parties. This is designed for replaceability and allows customers to plug in any Smart Card that provides a JDK 1.3 or 1.4 JCE API-compliant wrapper.

For C++ Orbix applications use of the Microsoft Crypto API (MS CAPI) is supported, through the Schannel toolkit. Microsoft Windows provides a Security Support Provider Interface (SSPI) for plugging security providers into the system. One of the standard Windows security providers, called Schannel, is a software implementation of the SSL/TLS security protocol.

Schannel uses MS CAPI to implement cryptographic functionality required by the SSL/TLS protocol. Since practically all cryptographic hardware vendors make their devices available as a MS CAPI Cryptographic Service Provider (CSP), the use of Schannel allows access to a great many existing cryptographic devices, Smart Cards, cryptographic tokens, and so on.

4.3 Server Clustering

One of the most powerful features in Orbix Enterprise is its server clustering architecture. With server clustering, it is possible to group together multiple physical servers—each of which may be running on a different machine—into a single logical server. To clients using the server, the appearance is that of a single server process, but the Orbix infrastructure actually distributes invocations across the set of server processes in the cluster.

Server clustering greatly improves the reliability of the system because the failure of one server in a cluster does not result in a loss of service—there is always another server available to take its place. The clustering mechanism is transparent to the application; when one of the servers in the cluster fails, the infrastructure detects the failure and automatically re-routes clients to another functioning server. As long as there are no transactions in progress on the server when it fails, the entire failover process is invisible to the application.

Server clustering also dramatically improves the overall performance of the system because it provides *load balancing*. This ensures that no single server becomes a bottleneck in the system, and lets you take advantage of the processing power of multiple machines.

Orbix provides several out-of-the box strategies that can be used to distribute client requests across the different servers in the cluster, including round robin, random distribution, and a feedback-driven method that routes new clients to the least-loaded server. These strategies can be applied on a per-service basis, which lets you employ different strategies for different applications.

In addition to supporting the development of clustered applications, the ORB infrastructure itself uses clustering to achieve high availability. Each of the CORBA services, including the Locator, Naming Service, Trading Service, and Configuration Repository, can be grouped into a server cluster.

4.4 Advanced Messaging Architecture

As noted above, Orbix Standard includes the CORBA Event Service. Orbix Enterprise includes both the CORBA Event Service and the CORBA Notification Service. IONA's implementation of these services provides both connection and event reliability. In addition, IONA provides an implementation of the CORBA Telecom Log Service. Finally, Orbix Enterprise also provides a bridge between the Java Messaging Service (JMS) and the CORBA Notification Service.

These features are described in the following sections.

4.4.1 CORBA Notification Service

The Orbix 6.1 CORBA Notification Service supports event reliability and connection reliability, thus enabling channels to be configured to recover their state after process restart, and to provide highly reliable event delivery. The Notification Service has a built-in database (Berkeley DB), allowing persistence of the Notification messages. Open Data Base Connectivity (ODBC) is not required. The

Orbix 6.1 Notification Service implementation provides twice the throughput of the ASP 6.0 version.

- **Connection Reliability:** Specifies whether or not the channel retains persistent information about all of its clients, including their filters and configured quality-of-service properties. By retaining this information, the channel's state can be dynamically recreated upon restart.

- **Event Reliability:** Specifies whether or not the channel stores a persistent copy of the event, so that it can guarantee delivery to all consumers even if the channel server fails.

4.4.2 Multicast

The Notification Service supports User Datagram Protocol (UDP) based IP multicast as an alternate plug-in transport layer (thus subsuming the functionality of OrbixTalk).

A multicast group is the set of hosts that receive messages on a particular IP multicast address. A multicast group can span multiple networks. Hosts. can join or leave multicast groups at any time. Adding hosts to a multicast group does not affect the number of messages sent over the network—a single message is sent regardless of the number of hosts in the multicast group. Thus the multicast transport service reduces the load on network resources and scales easily to large numbers of receivers. Because Orbix implements multicast as a plug-in, a single channel can support both IIOP and multicast messages.

4.4.3 CORBA Telecom Log Service

IONA's implementation of the CORBA Telecom Log Service is a high-performance, distributed logging system that provides support for long-term event storage and playback of historical events (for example, by clients that connect to the service after events they are interested in have already been emitted). Telecom logging provides a mechanism for creating a persistent log of events in a distributed computing environment and allows for the recovery of events in the face of a catastrophic failure.

When combined with the Notification Service, the Telecom Log Service lets you simply exchange an existing event or notification channel with a fully compatible logging version of the channel.

4.4.4 Java Messaging Service and CORBA Notification Bridge

The Java Message Service (JMS) is an enterprise-capable middleware component that provides the fundamental functions of message-oriented middleware (MOM). JMS support is built into Orbix, giving developers the ability to write applications that deliver high volumes of messages in a secure and reliable manner.

Orbix provides a bridging mechanism between the CORBA Notification service clients and JMS clients. This bridge allows Notification Service clients and JMS clients to exchange messages. Using the bridge, JMS publishers can forward messages to CORBA Notification consumers and CORBA Notification suppliers can forward messages to JMS subscribers.

4.5 CORBA Object Transaction Service

Transactions are an important programming paradigm for simplifying the construction of reliable and available applications, especially those that require concurrent access to shared data. Today it is widely accepted that transactions are the key to constructing reliable distributed applications.

Orbix includes an implementation of the Object Transaction Service (OTS). IONA provides two varieties of the OTS, supporting single and multiple resources respectively. The single-resource OTS version supports transactional semantics for transactions that involve only one resource, for example, a single database. This version of OTS is bundled with Orbix Standard and Orbix Enterprise. The multiple-resource version supports transactions that span multiple resources and therefore require two-phase commit. The multiple-resource version of OTS is a separately-orderable component.

4.6 CORBA Trading Object Service

Orbix Enterprise Edition provides an efficient, robust, and complete implementation of the CORBA Trading Object Service specification. The Trading Object Service is the mechanism by which instances of a particular kind of service advertise themselves, and by which other objects can discover these services. In this way, the Trading Object Service supports the dynamic discovery of services as well as the ability to bind to these services at runtime.

4.7 IONA Firewall Proxy Service (FPS)

J2EE application servers, and CORBA servers in the middle tier, use IIOP to communicate with services and resources, which means that IIOP-based messages need to traverse firewalls securely. Unfortunately, most TCP firewalls do not support IIOP traffic at the protocol proxy level. IONA's Firewall Proxy Service (FPS) is a configurable proxy that is placed on a "bastion" host between the server and its clients, as shown in Figure 5.

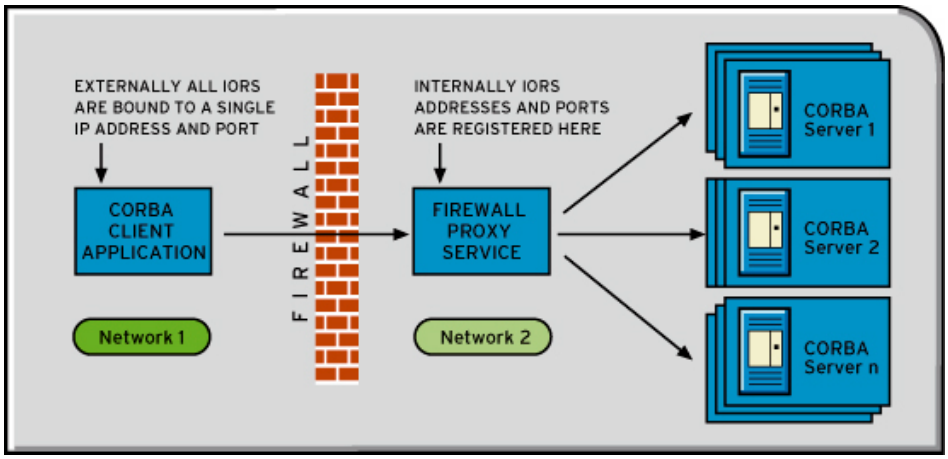

Figure 5: IONA Firewall Proxy Service

Rather than opening a large range of dynamically allocated ports, the FPS allows an administrator to open a limited number of specific ports.

FPS works by mapping interoperable object references (IORs) exposed to the external clients to those of the real CORBA servers. Only Portable Object Adapter (POA) based servers can be accessed through FPS. A server that uses FPS exchanges IOR template information with the service during a registration process that is initiated by the creation of a POA.

Once a server has registered with FPS, all IORs generated by that service point clients to proxies managed by FPS. FPS maintains a persistent store of registration information, such that when FPS initializes, it recreates the bindings for any server that registered with it during a previous execution. This assures that server registration is persistent across many executions of FPS.

For performance reasons, the Firewall Proxy Service does not attempt any authorization or filtering of the messages. The Firewall Proxy Service supports IIOP traffic only. It does not support IIOP over SSL/TLS.

5 Building Out From Orbix

For more than 10 years, Orbix has been the most prevalent high-performance application integration standard, embedded in thousands of telecom, finance, manufacturing and government applications. Orbix is an enterprise architecture solution that goes beyond integration to solve the harder problems of interoperability, security, and transaction monitoring among CORBA, Web Services, J2EE, .Net, CICS, and IMS.

The ability to “build out” from Orbix systems to other platforms is an important capability for our customers—as is the ability to “build toward” CORBA systems from other technology platforms. Here is how IONA is providing the ability to build out and build toward this proven CORBA technology:

- **Enterprise Qualities of Service:** Orbix provides a single, shared set of enterprise qualities of service for security, asynchronous messaging, management, transactions, load balancing, and fault tolerance, all of which can be leveraged by any component operating on the Orbix platform.

- **Web Services:** CORBA objects can be easily exposed as Web Services with Orbix 6.1. In fact, you can create Web services clients for CORBA objects in minutes.

- **.NET Interoperability:** Orbix provides a native .Net to CORBA Connector (as described in Section 3.1), which brings enterprise CORBA scalability to .NET and allows clients written in VB, C#, J#, and managed C++ to access CORBA and J2EE Servers.

  Clients can use the security, transaction, load balancing and fault tolerance features available to a normal Orbix C++ client in Orbix Enterprise. No additional hardware or software resources are required for this high-performance solution, which is completely non-intrusive to

existing CORBA systems. Best of all, .NET developers don't need to know anything about CORBA to use this feature.

- **J2EE to CORBA Interoperability:** Orbix Connect reduces the cost and complexity of achieving transparent and seamless J2EE-to-CORBA connectivity. Orbix Connect is based on the Java Connection Architecture, and works with JBoss, BEA's Weblogic, and IBM's WebSphere. Orbix Connect lets Java developers exploit CORBA investments without requiring vendor-specific CORBA knowledge. The Orbix Connect architecture encompasses all the details of packaging, deployment, and configuration, which allows these CORBA elements to be hidden from J2EE developers.

- **Mainframe Interoperability:** Orbix Mainframe turns CICS and IMS transactions into CORBA services. Middle-tier applications running on Unix or Windows can access mainframe application logic, while Orbix maintains transactional integrity, enables mutual access control, and provides common infrastructure (security, transactions, and communication protocols) across platforms.

- **Mainframe Web Services**: Artix Mainframe lets you expose mainframe application functions—including CICS and IMS transactions—as Web services, and manages data conversions between XML and CICS/IMS formats. It is an efficient and high-performance implementation that can run entirely on the host, which means that the mainframe's system administrators maintain control over CICS/IMS applications, including the authentication of off-host clients with host-based security and access control. Web services enabled by Artix Mainframe do not require any code changes to production applications, do not require any supplementary system services (such as CICS Web Support, CICS Business Transaction Services, etc.), and do not require Java on the mainframe.

Orbix Mainframe, Artix Mainframe, and Orbix Connect are separately-orderable products.

IONA covers everything CORBA—and beyond. IONA's solutions extend CORBA to work with J2EE, .NET, CICS, IMS, and more. From desktop applications to the most massive mainframe systems, IONA makes it easy to get distributed applications working together.

6 Support From IONA Professional Services and Partners

IONA now offers up-front support contracts with a duration ranging from one to five years for Orbix 6.1 at Standard, Silver, and Gold levels contracts. With more customers and more systems in production than any other CORBA vendor, IONA offers the highest quality CORBA support in the industry. We provide technical support for Orbix 6.1 at three levels:

- Standard Support: Expert technical help is available to from IONA's Technical Support Center (8) hours a day, five (5) days a week.
- Silver Support: Expert technical help is available to from IONA's Technical Support Center twelve (12) hours a day, five (5) days a week.

- Gold Support: Expert technical help is available to from IONA's Technical Support Center twenty-four (24) hours a day, seven (7) days a week.

In addition, IONA Professional Services and our Consulting Partners provide experienced Developers, Architect and Project Managers to assist with the architecture, design, development, integration, rollout, support and optimization of your Orbix 6.1 applications. No matter what your integration challenges are, our consultants can play any number of roles on both short and long-term engagements to help you get the most out of Orbix 6.1.

Whether you're building a new application from the ground up or working to improve an existing application, our consultants will keep your project on track by sharing their world class expertise in areas such as:

- Architecture
- Security
- Scalability/Performance/Throughput
- Systems Management
- High Availability/Fault Tolerance
- Server Consolidation
- Interoperability with other middleware such as .NET, COM and J2EE
- Integration with the mainframe
- Migration from earlier versions of Orbix and from 3rd party ORBs

For an immediate impact on your project, we offer one-week assessments in any of the areas above that provide you with a specific set of steps to improve your application.

7 Conclusion

Orbix 6.1 adds many desirable enterprise-class features to IONA's CORBA implementation—which is already the world's most popular and prevalent development/integration platform for large-scale mission-critical systems. These enhancements include an adapter for integration with .NET, a variety of important new security features, bi-directional GIOP, enhanced performance, and integration with Tivoli Enterprise Management System.

Orbix provides all of these capabilities through modern, open, and entirely standards-based technologies. Orbix 6.1 continues to excel at meeting the needs of integration/development teams that need to integrate CORBA with other technologies, while meeting the most demanding scalability, reliability, and performance requirements.

See www.iona.com for the current list of platforms supported by Orbix 6.1.

Contact Details

IONA Technologies PLC
The IONA Building
Shelbourne Road
Dublin 4
Ireland
Phone: .....+353 1 637 2000
Fax: .....+353 1 637 2888

IONA Technologies Inc.
200 West St
Waltham, MA 02451
USA
Phone: .....+1 781 902 8000
Fax: .....+1 781 902 8001

IONA Technologies Japan Ltd
Akasaka Sanchome Bldg 7/F
3-21-16 Akasaka
Minato-ku, Tokyo
Japan 107-0052
Phone: .....+813 3560 5611
Fax: .....+813 3560 5612

Support: .....support@iona.com
Training: .....training@iona.com
Orbix Sales: .....sales@iona.com
IONA's FTP site .....ftp.iona.com

World Wide Web: www.iona.com