OBJECTIVE
INTERFACE

ORBexpress

Core ORB Product Line

What Is ORBexpress?

Typically, engineers create custom communications infrastructures for each project. When the underlying technologies change, the software must be rebuilt. The ORB*express* high performance approach to comunications architectures insulates the team from these changes. Developers using ORB*express* deliver product quicker and accommodate changes easier.

Key Values

- **Simplicity:** Developers write less code. And the code they write contains just simple method calls. The ORB takes care of the details of messaging. *More capability, less time.*
- **Transparency**: Developers write the same code independent of object location. CORBA's transparent architecture means no recoding when objects are made local or made distributed. *More flexible systems.*
- **Independency**: All calls are written the same regardless of the media. Details of differing media APIs are hidden. The ORB correctly sends calls to their destination via the chosen media. *Less platform dependence.*
- **Heterogeneity**: CORBA is built to bridge differing processors, operating systems, programming languages and transport media. The ORB provides significant portability and uniformity for a heterogeneous environment. *Greater system interoperability.*
- **Reliability**: Errors are detected earlier in the project lifecycle. Typical messaging approaches detect errors only at integration time – when the system crashes. Using strongly type checked IDL interfaces, errors are detected earlier: at compile time. *More reliable systems.*
- **Rapidity**: Developers build software faster – with less work to do. Alternatively, more capabilities can be added since the team can focus on the application design rather than the infrastructure "plumbing." *Faster time to market.*
- **Flexibility**: Late life-cycle optimization (LALO) becomes possible. Rather than determining the location of objects a priori, the location can be determined based on empirical data. Changing the location later doesn't require major changes to the software. *More efficient systems.*
- **Scalability**: ORB speed, size and capabilities can be optimized based on the scale of the project (up or down). ORB*express* can be tuned for deeply embedded systems and for large-scale efforts. *Long-lived systems.*
- **Fast, Lean and Predictable**: Built for Real-time applications. All of these ORB*express* benefits come with minimal overhead. The ORB*express* architecture is built from the ground up to be predictable and have minimal jitter. *Systems that solve tomorrow's problems.*

The bottom line is cost and schedule: ORB*express saves projects money and time without the overhead of other approaches.*

Key Architecture Concepts

The ORB*express* product family is Objective Interface's high-performance implementation of the latest CORBA technology. All members of the ORB*express* family are related by a common architecture. ORB*express* is designed from the ground up for the high-performance, real-time, and embedded market. The architecture's goals are to be:

- Reliable
- Fast
- Lean
- Predictable

Significant effort has been taken to optimize all layers of the product. The result is a system that is tuned much more thoroughly than any one project could afford to do. The architecture features:

- **Real-time**: Predictable behavior in an end-to-end round trip with predictable ORB internals
- **Interoperability**: Over 50% of our customers have tested ORB*express* with another ORB
- **Multiplexed Connections**: Multiple threads in a program share one transport connection to minimize resource usage
- **Multi-threaded**: Servers process requests using multiple threads to ensure minimal latency and predictable response
- **Consistent**: A single shared code base means that features and fixes added to one platform are reflected in the others.
- **Fault Resilient**: ORB*express* is designed to ensure that communication faults are detected and managed. Developers are provided with tools to help achieve robust and reliable software.

Performance

Introduction

ORBs designed from the ground up for the high-performance, real-time and embedded domain – such as ORB*express* – are fundamentally different. In crucial performance metrics, such as latency and footprint, ORB*express* is an order of magnitude faster and leaner than its desktop counterparts.

ORB*express*	Desktop ORBs
Fast (Microseconds)	Slow (Milliseconds)
Lean (Kilobytes)	Fat (Megabytes)
Predictable	Jittery

Latency

Independent benchmarks prove that ORB*express* is the fastest ORB available. ORB*express* clients and servers add minimal overhead to the network transport. For small amounts of data passed as parameters, the ORB adds less than 10% to the network or inter-process transport time. For large amounts of data (more than 2K), the overhead drops to 3%. A sample comparison of ORB*express* versus socket times between two processors is shown at right.

Bandwidth

Programs transferring larger amounts of data are concerned more with the amount of data per second than with the latency. A large bandwidth transport is of no use if the ORB limits the program to a fraction of the total potential. ORB*express* has been carefully engineered to ensure that it's overhead proportionally decreases as the amount of data increases. Other ORBs loose performance when more and more data is sent.

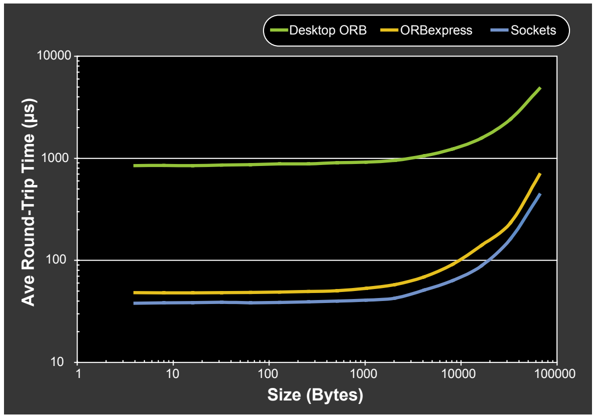
Plug-in transports eliminate the overhead of TCP/IP by directly accessing the underlying media APIs. ORB*express* supports a shared memory plug-in transport that eliminates the TCP/IP stack. Examining the second chart on the right shows that the slope of the curve for the shared memory transport is lower. The shared memory transport scales more effectively and yields better bandwidth. Our *zero copy technology* yields more significant bandwidth improvements.

High Performance Advantage

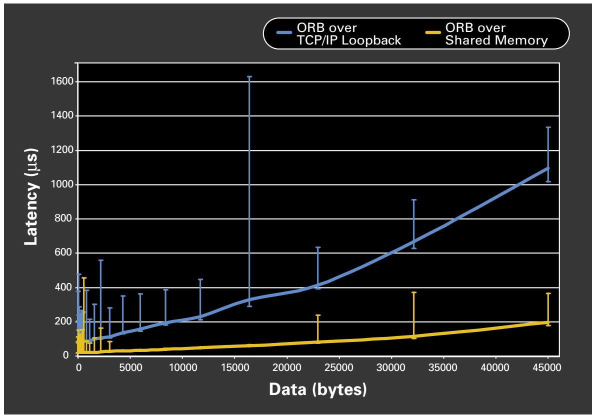
Comparing ORB*express* performance to the network transport illustrates what is maximally possible. Comparisons with other ORBs reveal what vendors accomplish. Data for the top chart at right is based on published results from tests conducted by Lockheed Martin ATL. Note: the other ORBs *overhead* exceeds ORB*express* total round-trip time.

Visit www.ois.com to review independent studies including ones from Lockheed Martin ATL and Boeing Phantom works (for the RT DII COE) and an article from IEEE Communications on CORBA in Software Defined Radio.

Adds Minimal Overhead



Easy Access to Better Transports



CORBA Overview

CORBA – The Common Object Request Broker Architecture – provides for simplified distributed applications. Over the past decade, the 800+ member organizations of the Object Management Group (OMG) have created an open standard for heterogeneous, distributed object-oriented Middleware for communications. The Object Request Broker (ORB) is the product implementing the CORBA standard. A focused subgroup of these members has created additional standards specifically to support the high-performance, real-time and embedded communities.

Distributed Real-time Concepts and Issues

Background

Hard and soft real-time systems are concerned with the timeliness of responses in addition to the correctness of the response. Having a fast ORB is important. This alone will not ensure a predictable end-to-end connection. Combining an enterprise or desktop ORB with an RTOS and predictable transport won't work either (see the *Distributed Priority Inheritance* sidebar at right). To support the needs of hard and soft real-time systems, in 1999 the OMG adopted the Real-time CORBA standard.

Ingredients of a Real-time ORB

ORB*express* RT exceeds the OMG's Real-time CORBA standard. Capabilities include:

- **Distributed Priority Inheritance and Priority Propagation**: See sidebar at right
- **Bounded Priority Inversions**: Limits the time that low priority activities suspend high priority activities
- **Predictable ORB Internals**: Ensure that the data structures and algorithms execute in a bounded manner
- **Priority Ceiling Locking**: Bounds priority inversion using ceiling locking mutexes
- **Real-time POA**: Standard API for server configuration
- **Plug-In Transports**: For media independent real-time objects. See sidebar on following page
- **Quality of Service (QoS)**: Take advantage of media specific options on plug-in transports

Distributed Priority Inheritance

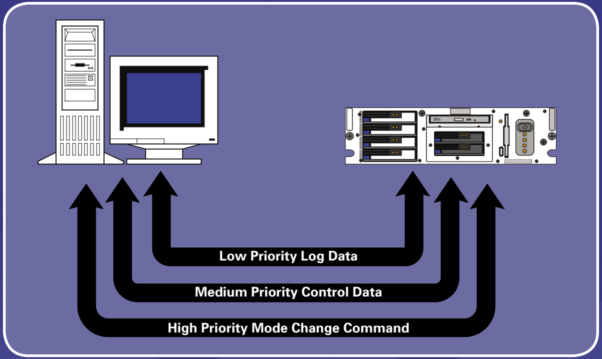
Priority Propagation

The primary goal of Real-time CORBA is end-to-end predictability. Real-time systems achieve this through priority based scheduling. Because the remote system doesn't know about the originating priority the system is indeterminate. Real-time ORBs solve this problem by propagating the priority to the server. The result is a distributed priority inheritance and the ability to correctly schedule distributed processes.

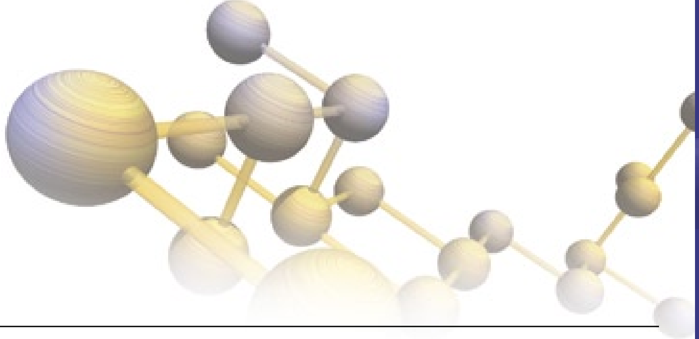
The heterogeneous nature of CORBA is important in priority propagation. Most RTOS have differing ranges of priorities. The Real-time CORBA standard defines a universal priority range. ORBs supply the ability to map between the native and universal. This is just one more way that CORBA provides a portable and transparent approach to distributed systems building.

Priority Banded Connections

Priority propagation ensures that the RTOS can schedule correctly end-to-end. However, the network can still be a source of priority inversions. A high priority message may have to await the completion of a large, low priority, message. That represents a message based priority inversion. To avoid this bottleneck, Real-time CORBA supports Priority Banded Connections. The ORB sends messages over user specified bands based on the priorities and QoS involved. Developers can trade-off network and OS resource usage against priority inversions.



Low Priority Log Data
Medium Priority Control Data
High Priority Mode Change Command



Distributed Embedded Concepts and Issues

Background

The embedded market is heterogeneous. It is fragmented – no single CPU type, RTOS or communications media dominates. Many teams create project specific solutions. The physical environment for these systems often imposes constraints on the ORB.

Ingredients of an Embedded ORB

- **Wide Availability**: Available a variety of CPUs and RTOS – from the common to the unusual
- **Host Development and Test**: Develop and test before limited target hardware is available
- **Mass Market Affordable**: No ORB run-time royalties *See sidebar at right*
- **Plug-in Transports**: *See sidebar at right*
- **Minimal Footprint**: Fit deeply embedded systems constraints
- **Scalable Architecture**: Address both small and large systems' constrains (time vs. space)
- **Professional Technical Support**: Partners ready to ensure your project success

Footprint

Many deeply embedded systems have limited RAM and flash memory available for the ORB. The ORB library must minimize memory use to leave room for the application and RTOS. Yet, systems with more generous resource limits may have issues with ORB scalability.

ORB*express* has been carefully tuned to provide minimal footprint configurations. Developers can configure ORB*express* GT for C++ as small as 126 KB for the ORB library. The full ORB*express* RT for C++ ORB library is less than 250 KB. Careful attention to the generated code size ensures that clients and servers are embeddable. Developer choice of ORB feature configurations also controls the size. No ORB need be multiple megabytes in size.

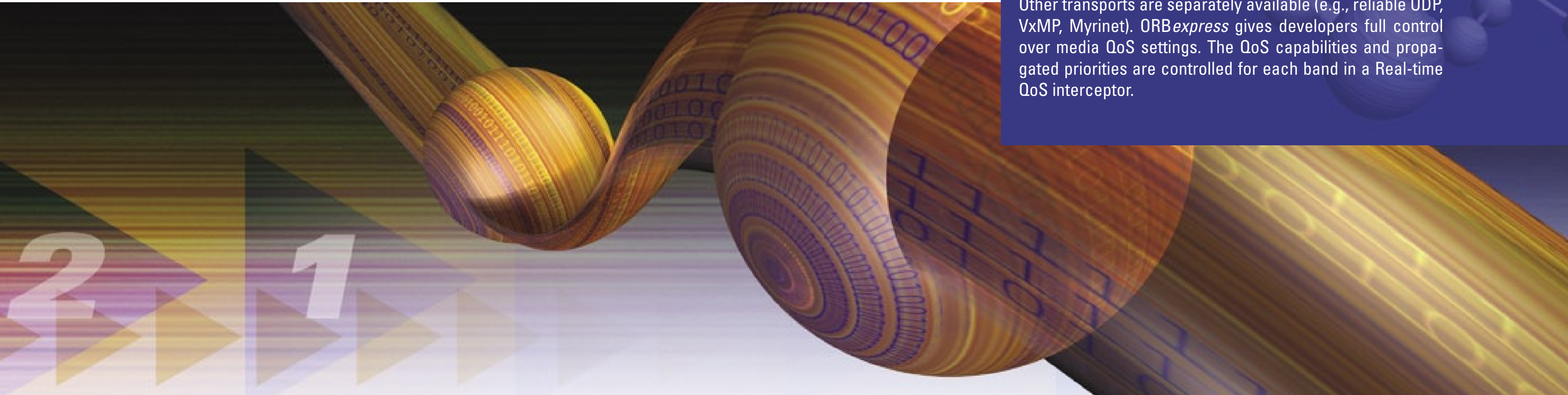
Keep Unit Costs Low

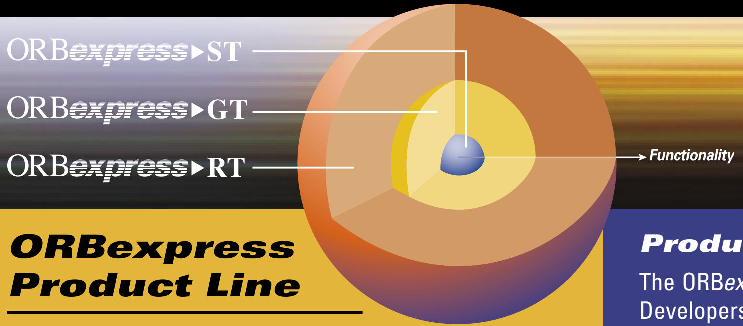
It is important that the ORB does not increase your product's unit costs. That's why there is no run-time license cost for ORB*express*. It is also very difficult for many organizations to track these costs. Tracking often costs more than the run-times do. The ORB characteristics can affect your cost of goods. ORB*express* has low latency and a lean footprint. That way the ORB doesn't force the use of a faster CPU (cost and cooling) or require more RAM (costs or board layout). The impact of the ORB on the project's bottom line is much more than just the cost of the product license.

Plug-In Transports

Real-time systems require a predictable end-to-end client to server call chain: Real-time ORB, RTOS, and Transport. Embedded environments have unique hardware needs beyond Ethernet. A plug-in transport mechanism is needed.

ORB*express* developers can create their own transports. ORB*express* is bundled with additional transports (e.g., shared memory, unreliable IP multicast, RACE++, etc.). Other transports are separately available (e.g., reliable UDP, VxMP, Myrinet). ORB*express* gives developers full control over media QoS settings. The QoS capabilities and propagated priorities are controlled for each band in a Real-time QoS interceptor.





ORB*express* Product Line

There are three products in the ORB*express* family:

ORB*express* ST:

The high-performance core of the product family. A robust, reliable implementation aimed at self-hosted systems. Available for C++, Java and Ada 95, each ORB is implemented in its native language. Platforms include Windows, Solaris and Linux. Fully multi-threaded, ORB*express* always multiplexes connections. This ensures that resource usage scales efficiently. ORB*express* ST also supports fault resilient connections to allow failover for connections to alternative transports or servers.

ORB*express* GT:

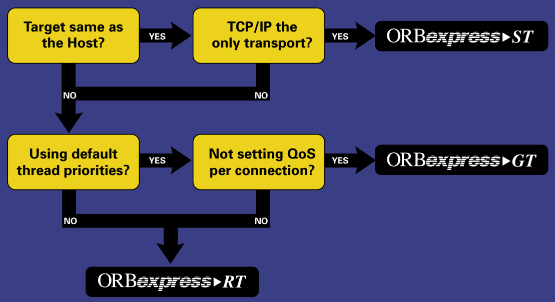
Designed for high-performance embedded systems. ORB*express* GT adds to the capabilities of the ST product. ORB*express* GT is available only for C++. It has been ported to a variety of embedded targets from the common to the unusual. A scalable architecture provides flexibility in all size environments. It's plug-in transport mechanism lets developers use unique embedded transports. ORB*express* GT is fast, and lean.

ORB*express* RT:

The flagship of the product line. ORB*express* RT adds to the capabilities of the ST and GT products. ORB*express* RT implements (and exceeds) the Real-time CORBA standard. It provides support for both hard and soft real-time systems. ORB*express* RT is designed to deliver predictable time behavior, plug-in transports and transport Quality of Service. It is available for C++ and Ada 95. Each ORB is implemented in its native language. ORB*express* RT has been ported to a variety of Real-time OS's and embedded targets from the common to the unusual.

Product Selection

The ORB*express* product line covers all types of applications. Developers can choose the appropriate product for their application. Projects starting with one ORB can transparently move their code to the more advanced products.


Target same as the Host?
YES
TCP/IP the only transport?
YES
ORBexpress▸ST
NO
NO
Using default thread priorities?
YES
Not setting QoS per connection?
YES
ORBexpress▸GT
NO
NO
ORBexpress▸RT

Complete Language Coverage

ORB*express* supports all major Object-Oriented languages. Each ORB is written in its native language, based on a common ORB*express* architecture. Each implementation contains an linkable library and an IDL translator which generates code according to the CORBA specified language mapping. In addition, each comes with a full suite of documentation and demo programs.

C++

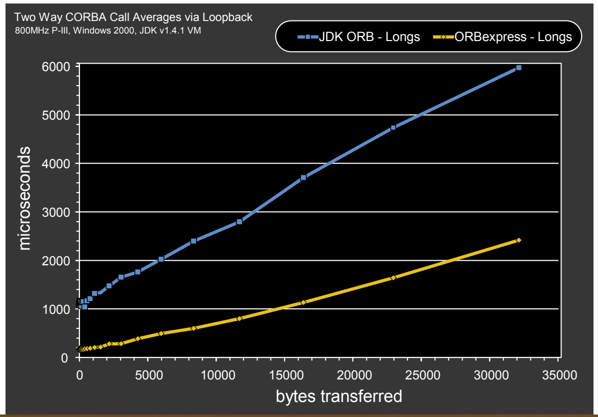
- **Available**: On a variety of platforms from the pedestrian to the unusual
- **Fast and Lean**: Optimized for high-performance, real-time and embedded systems (the charts on the performance page are for the C++ ORB)
- **Configurable**: ORB*express* GT can scale for systems large to small

Ada 95

- **Heterogeneous**: Connect Ada to other languages
- **Simple**: Easier than writing direct bindings to C++
- **Real-time**: Integrated with Ada's multitasking and the Ada Real-time Systems Annex
- **Reconfigurable**: Incrementally add new components in any language

Java

- **High Performance**: Combining the ORB*express* architecture with pure Java. (See the chart below)
- **Interoperable**: With ORB*express* for Ada and C++
- **Low Risk**: Multiple languages supported by a single ORB vendor
- **Simplified Support**: Common support team for all sides of the program



Available Platforms

Processors	Operating Systems	Compilers
Alpha	AIX	Apex
ARM	HP/UX	Concurrent
HP	INTEGRITY	Dec CC
Mercury	Linux	Eclipse Java
MIPS	LynxOS	EGCS
PowerPC	Mercury OS	GCC
SH	OSE	GNAT
SPARC	PowerMax	GreenHills Multi
x86	QnX Neutrino	HP ACC
XScale	SGI Irix	ObjectAda
	Sun JVM	SGI CC
	Sun Solaris	Sun CC
	TimeSys RT JVM	Sun Java
	Tru64	Visual C++
	VxWorks	
	Windows	

Vertical Markets

ORB*express* has been in use since 1997. Projects using it have reached every stage of development: from initial definition to full-scale release, deployment, and maintenance. Projects encompass many different vertical domains including:

- **Telecom/Datacom**: Optical switches, routers, cell phones, communications equipment, etc.
- **Defense & Aerospace**: Embedded weapons, command, control, communications and intelligence systems, avionics, software radio, radar, etc.
- **Process Control**: Laser powered nuclear fusion research, machine vision, industrial microscopes, etc.
- **Transportation**: Shipboard command, control, monitoring and alarm systems; metro fare card collection, etc.
- **Consumer Electronics**: Set top boxes, internet infrastructure, etc.

The cross cutting requirements of *safety* and *security* are being addressed in a new multi-year initiative by Objective Interface. These new capabilities will be applicable across all the domains of ORB*express* use.

Project Highlights

Many companies and organizations have selected ORB*express.* Noteable projects that use ORB*express* include:

- **Kingcat Yacht Monitoring, Control and Alarm**: Paranor AG's onboard ship command and control system. This system is a 5 computer bridge control system for the Kingcat M270 70ft. catamaran. It monitors the engine, fuel, electrical, water, etc.
- **Lawrence Livermore Labs National Ignition Facility**: This extremely large application controls 192 lasers (and associated actuators, sensors, motors, etc.) that send a 2 Megajoule 25ns laser pulse into a deuterium pellet. A combination of workstations (for user interface and scientific control) and embedded device controllers are connected by real-time CORBA.
- **Software Defined Radio / Joint Tactical Radio System (JTRS)**: A hard real-time, embedded application that transmits incoming and outgoing voice, video and data radio signals via D/A, A/D and software defined waveforms. It features a common software architecture (the Software Communications Architecture or SCA) defined using IDL and standardized via the OMG. ORB*express* has been used in more than 90% of the JTRS and SDR applications to date.

Training and Consulting

To complement the ORB*express* product line, Objective Interface offers professional services, to support ORB*express* users. Training is available either at our facility or on-site at customer locations. Two 3-day courses are available which take developers from initial introduction to ORB*express* proficiency and project readiness. Hands-on training provides 50% lecture and 50% lab.

Project specific mentoring and consulting is also available. Objective Interface supplies a variety of services to help ensure customer's projects success. Going beyond tech support, mentoring can help with tasks such as:

- Working with developers to create a CORBA based object-oriented design
- Evaluating trade-offs in the architecture
- Determining how to obtain the best performance from ORB*express*
- Review and support during project reviews

Mentoring and consulting can take place at on-site customer locations or remotely via phone, fax, internet and e-mail.

Customer Service

Solid tech support is essential to ensuring successful product use. Objective Interface is committed to first-rate tech support. When Boeing's Phantom Works group studied Real-time CORBA ORBs for the DII COE, they surveyed users and compared tech support experiences. ORB*express* tech support was ranked the best, scoring between 1.25 and 1.75 on a scale of -2.0 to +2.0.

The first year of ORB*express* tech support is included at no extra cost with each ORB*express* license. Our dedicated tech support group is available via phone, fax, internet and e-mail to answer questions, help reproduce bugs and determine workarounds and solutions. ORB*express* licensees current on maintenance receive all updates and upgrades issued for their platform.

ORBexpress

Objective Interface Systems, Inc.

Objective Interface Systems, Inc. is a leader in real-time embedded and high-performance communications software. Objective Interface provides customers with advanced real-time connectivity software development tools for use in products thought the telecom/datacom, defense, aerospace, consumer electronics, process control, and transportation industries. A privately held company located in Herndon, Virginia, Objective Interface has been developing high-performance software applications since 1989.

A pioneer in the development and adoption of advanced real-time CORBA technology, Objective Interface helps different customers understand the uses, applications, and advantages of using CORBA for seamless application connectivity. Widely recognized as experts within the real-time community, Objective Interface continues to develop cutting edge, real-time commercial applications through continuous research and development efforts. Committed to customer success, Objective Interface provides unprecedented customer support and market leading technologies.

Contact Info

Corporate Headquarters:
Objective Interface Systems, Inc.
13873 Park Center Road, Suite 360
Herndon, VA 20171-3247

Toll Free: (800) 800-OIS7

Phone: (703) 295-6500

Fax: (703) 295-6501

E-mail: info@ois.com

Web: http://www.ois.com/

Additional satellite sales offices are located in California, Texas, Maryland and Tokyo, Japan.

OBJECTIVE INTERFACE