introduction

ORBexpress delivers product quicker and accommodates changes easier.

Typically, engineers create custom communications infrastructures for each project. What the under-
taking brings with it is long lead time, complexity, leaps, and bounds.

What is ORBexpress?

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

- Reliability
- Fast
- Predictable

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

- Real-time: Predictable behavior in an end-to-end round trip with predictable ORB internals.
- Multi-threaded: Connections Multiple threads in a process share one transport connection to minimize resource usage and provide optimal performance.
- Consistent: A single shared code base means that features and fixes added to one platform are reflected in the others.
- Fault-Isolation: ORBexpress is designed to ensure that communication links are detected and managed. Developers are provided with tools to help achieve robust and reliable software.

Performance

LATENCY

Independent benchmarks prove that ORBexpress is the fastest ORB on the market today, both in terms of the overall inter-process transport and the network inter-process transport. For small amounts of data passed as parameters, the ORB adds less than 10% to the overhead to the network transport. For large amounts of data passed as parameters, the ORB adds overhead from 5% to 10%. A good comparison of ORBexpress versus actual times between two processes 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 this total bandwidth. ORBexpress has been carefully engineered to ensure that it’s overhead proportionally decreases as the amount of data increases. Other ORBs lose performance when more and more data is sent.

High Performance Advantage

Comparing ORBexpress performance to the network transport illustrates what is really possible. Comparisons with other ORBs reveal what vendors accomplish. Data for this graph was collected at Lockheed Martin ATL. Note: the other ORBs overlaid exceed ORBexpress's benchmark in some regions.

Corban Overview

ORBexpress – The Common Object Request Broker Architecture – provides for scalable distributed applications. Over the past decade, the ORB membership organization of the Object Management Group (OMG) have created an open standard for interoperable distributed applications. The Object Request Broker (ORB) is the protocol that runs on top of CORBA, ensuring the CORBA standard. A focused subset of those members has created additional standards specifically to support real-time, high-performance, real-time and embedded communities.
What Is ORBExpress?

Typically, engineers create custom communications infrastructures for each project. What the underly ing technology, the software, or the hardware, the ORBExpress high-performance approach to communications architectures insulates the team from these changes. Developing ORBExpress delivers predict and quicker accommodate changes easier.

Key Values

- **Reliability:** ORBExpress builds on industry standards, not on proprietary code. And the code they write consists of just simple methods calls. The ORBExpress avoidance of proprietary code means no recoding when objects are made local or node distributed. More flexible systems.

- **Multi-threaded:** ORBExpress creates two threads to ensure minimal latency and predictable response time.

- **Heterogeneity:** ORBExpress is built on bridging software processes, operating systems, programming languages and transport mediums. The ORBExpress provides significant portability and flexibility for a heterogeneous environment. Drivers are transparent.

- **Scalability:** ORBExpress is designed to ensure that scaling from one CORBA machine to the next is done at a fraction of the total potential. ORBExpress adds minimal overhead.

- **Predictable:** ORBExpress offers the assurance that the real-time and embedded domain – such as ORBExpress – are fundamentally different. In crucial performance metrics, such as latency and footprint, ORBExpress is in the range of magnitude faster than other reality computer counterparts.

- **Lean:** A single shared code base means that features can be tuned for deeply embedded processors is shown at right. Note: the other ORBs exceed ORBExpress bandwidth scales more effectively and predictably than the infrastructure “plumbing.”

- **Easy Access to Better Transports:** ORBExpress supports a zero copy technology that yields more significant bandwidth improvements.

Performance

**Introduction**

ORBExpress designed from the ground up for the high-performance, real-time, and embedded domain, the product is currently under development. All of the ORBExpress product family are related by a common architecture. ORBExpress is designed from the ground up for the high-performance, real-time, and embedded market. The architecture's goals are to:

- Be lean
- Be fast
- Be predictable

**Key Architecture Concepts**

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

- Be lean
- Be fast
- Be predictable

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

- Real-time: Predictable behavior in an end-to-end round trip with predictable ORB internals.
- Multi-threaded: Servers process requests using multiple threads to ensure minimal latency and predictable response time.
- Flexibility: ORBExpress allows a program to be tuned for deeply embedded processors is shown at right.
- Consistent: A single shared code base means that features and fixes added to one platform are reflected in the others.
- Fast, Lean, and Predictable: Built for Real-Time applications. All of these ORBExpress benefits come with minimal overhead. The ORBExpress architecture is built from the ground up to be predictable and have minimal overhead. Systems that solve tomorrow’s problems.

**Rapidity:** Developers build software faster — with less work to do. Alternatively, more capabilities can be added since the team has the time to plan and plan and plan the application design rather than the infrastructure “plumbing.” Faster, faster, faster.

**Flexibility:** Real-time cycle optimization (SALSO) becomes possible. Rather than determining the location of objects a priori, the location can be determined based on emer gence data. Changing the location later doesn’t require major changes to the software. More efficient systems.

**Reliability:** Fast (Megabytes) Slow (Milliseconds)

**Easy Access to Better Transports**

**Latency**

Independent benchmarks prove that ORBExpress is the fastest ORB as of September, 1997. ORBExpress offers the advantages of over 80% of our customers have tested ORBExpress and other ORB. Multiple platforms and multiple ORBs, the ORBExpress is compared to the shared memory transport and the TCP/IP transport. For small amounts of data passing the ORB, the overhead is less than 10% to the network or inter-process transport time. For large amounts of data passing the ORB, the overhead is less than 5%. A sample of ORBExpress versus other architectures shows that the shared memory transport scales more effectively and predictably than the TCP/IP transport. ORBExpress offers the advantage of zero copy technology yields more significant bandwidth improvements.

**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 bandwidth. The result is a system that is tuned much more than any one project could afford to do. The differences include:

- Real-time:Predictable behavior in an end-to-end round trip with predictable ORB internals.
- Multi-threaded: Servers process requests using multiple threads to ensure minimal latency and predictable response time.
- Consistent: A single shared code base means that features and fixes added to one platform are reflected in the others.
- Fast, Lean, and Predictable: Built for Real-Time applications. All of these ORBExpress benefits come with minimal overhead. The ORBExpress architecture is built from the ground up to be predictable and have minimal overhead.
- Flexibility: ORBExpress is designed to ensure that communication latencies are detected and managed. Developers are given more tools to help achieve robust and reliable software.

CORBA Overview

**CORBA** – The Common Object Request Broker Architecture is designed to enable applications to communicate over all platforms and operating systems. The Object Request Broker (ORB) is the generic component running on the CORBA standard. A focused subset of these members has created additional standards specifically to support high-performance, real-time and embedded environments.

**Introduction**

ORBExpress is a product of Object Interchange’s high-performance implementation of the latest CORBA technology. All members of the ORBExpress family are related by a common architecture. ORBExpress is designed from the ground up for the high-performance, real-time, and embedded domain. The architecture's goals are to:

- Be lean
- Be fast
- Be predictable

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

- Real-time: Predictable behavior in an end-to-end round trip with predictable ORB internals.
- Multi-threaded: Servers process requests using multiple threads to ensure minimal latency and predictable response time.
- Flexibility: ORBExpress allows a program to be tuned for deeply embedded processors is shown at right.
- Consistent: A single shared code base means that features and fixes added to one platform are reflected in the others.
- Fast, Lean, and Predictable: Built for Real-Time applications. All of these ORBExpress benefits come with minimal overhead. The ORBExpress architecture is built from the ground up to be predictable and have minimal overhead. Systems that solve tomorrow’s problems.

**Rapidity:** Developers build software faster — with less work to do. Alternatively, more capabilities can be added since the team has the time to plan and plan and plan the application design rather than the infrastructure “plumbing.” Faster, faster, faster.

**Flexibility:** Real-time cycle optimization (SALSO) becomes possible. Rather than determining the location of objects a priori, the location can be determined based on emer gence data. Changing the location later doesn’t require major changes to the software. More efficient systems.

**Reliability:** Fast (Megabytes) Slow (Milliseconds)

**Easy Access to Better Transports**

**Latency**

Independent benchmarks prove that ORBExpress is the fastest ORB as of September, 1997. ORBExpress offers the advantages of over 80% of our customers have tested ORBExpress and other ORB. Multiple platforms and multiple ORBs, the ORBExpress is compared to the shared memory transport and the TCP/IP transport. For small amounts of data passing the ORB, the overhead is less than 10% to the network or inter-process transport time. For large amounts of data passing the ORB, the overhead is less than 5%. A sample of ORBExpress versus other architectures shows that the shared memory transport scales more effectively and predictably than the TCP/IP transport. ORBExpress offers the advantage of zero copy technology yields more significant bandwidth improvements.

**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 bandwidth. The result is a system that is tuned much more than any one project could afford to do. The differences include:

- Real-time:Predictable behavior in an end-to-end round trip with predictable ORB internals.
- Multi-threaded: Servers process requests using multiple threads to ensure minimal latency and predictable response time.
- Consistent: A single shared code base means that features and fixes added to one platform are reflected in the others.
- Fast, Lean, and Predictable: Built for Real-Time applications. All of these ORBExpress benefits come with minimal overhead. The ORBExpress architecture is built from the ground up to be predictable and have minimal overhead.
- Flexibility: ORBExpress is designed to ensure that communication latencies are detected and managed. Developers are given more tools to help achieve robust and reliable software.

CORBA Overview

**CORBA** – The Common Object Request Broker Architecture is designed to enable applications to communicate over all platforms and operating systems. The Object Request Broker (ORB) is the generic component running on the CORBA standard. A focused subset of these members has created additional standards specifically to support high-performance, real-time and embedded environments.

**Introduction**

ORBExpress is a product of Object Interchange’s high-performance implementation of the latest CORBA technology. All members of the ORBExpress family are related by a common architecture. ORBExpress is designed from the ground up for the high-performance, real-time, and embedded domain. The architecture's goals are to:

- Be lean
- Be fast
- Be predictable

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

- Real-time: Predictable behavior in an end-to-end round trip with predictable ORB internals.
- Multi-threaded: Servers process requests using multiple threads to ensure minimal latency and predictable response time.
- Flexibility: ORBExpress allows a program to be tuned for deeply embedded processors is shown at right.
- Consistent: A single shared code base means that features and fixes added to one platform are reflected in the others.
- Fast, Lean, and Predictable: Built for Real-Time applications. All of these ORBExpress benefits come with minimal overhead. The ORBExpress architecture is built from the ground up to be predictable and have minimal overhead.
Distributed Real-time Concepts and Issues

Background
Hard and soft real-time systems are concerned with the timeliness of response to actions or the correctness of the response. Having a fast ORB is important. This alone will not result in a good real-time system. Combining an enterprise or desktop ORB with a RTOS and predictable transport will result in a true Distributed Priority Interface suitable for real-time systems.

In order 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
Distributed Real-time CORBA is a plug-in transport mechanism that allows developers to create their own transports. It also allows developers to choose between hard and soft real-time systems.

- Distributed Priority Inheritance and Priority Propagation
  - Hard real-time CORBA provides the means to propagate priorities to objects across the network. It ensures that the priority of the sender is carried through the network to the recipient.
  - Soft real-time CORBA provides a mechanism for objects to inherit priorities from other objects.

- Priority Banded Connections
  - Priority propagation ensures that the RTOS can schedule correctly.
  - Priority ceiling locking mutexes limits the time that low priority activities spend in high priority activities.

- Predictable QoS
  - The Real-time CORBA standard defines a universal interface for setting QoS parameters.

- Scalable Architecture
  - The ORB characteristics can affect your project success. It is important that the ORB does not increase your product's unit cost.

Distributed Priority Inheritance
The primary goal of Real-time CORBA is end-to-end predictable real-time systems allowing its through priority-based scheduling. Because the remote system doesn’t know about the initiating priority, the request is sent to an intermediate node, called the Receiving ORB (ROR). The ROR resolves this problem by propagating the priority to the server. This results in a distributed priority inheritance and the ability to correctly schedule distributed processes.

The heterogeneous nature of CORBA is important in priority propagation. Most RTOSs have different ranges of priorities. The Real-time CORBA standard defines a restricted priority range. CORBA supplies the ability to map between the range and medium. This is a new concern for the CORBA designer. It provides a portable and transparent approach to distributed real-time scheduling.

- Priority Propagation
  - Propagation ensures that the RTOS can schedule correctly.
  - Priority ceiling locking mutexes limits the time that low priority activities spend in high priority activities.

- Priority Banded Connections
  - Priority propagation ensures that the RTOS can schedule correctly.
  - Priority ceiling locking mutexes limits the time that low priority activities spend in high priority activities.

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.

- 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. New systems with more processor resources limit may have issues with ORB scalability.

- Incentives of an Embedded ORB
  - Wide Availability: Available a variety of CPUs and RTOS from the real-time or the embedded platform.
  - Real-time Operation: Design real-time OSes to support any applications.
  - Designability: Developers can trade-off network and CPU resource usage against priority inversions.

- Predictive QoS
  - The embedded environment must have low latency and a predictable network.

- Minimal Footprint
  - The ORB footprint is less than 250 KB. Careful planning and tuning have resulted in this size.

Keep Unit Costs Low
It is important that the ORB does not increase your product’s unit cost. Your project success is dependent on the cost of goods. It is also very difficult for many organizations to scale their ORB solutions. How much more does the ORB consumption cost than the cost of goods? It is important that the ORB footprint is less than 126 KB to ensure that the ORB does not increase your product’s unit cost.

Plug-In Transports
Real-time systems require a predictable end-to-end client-server call chain: Real-time ORB, RTOS, and Transport. Embedded environments have unique hardware needs. For example, an embedded system is limited to a small set of transports.

- Real-time Systems: Real-time systems require a predictable end-to-end client-server call chain: Real-time ORB, RTOS, and Transport. Embedded environments have unique hardware needs.

- ORB Configuration: Developers can create their own transports. ORBs are developed with additional features, such as secure messaging, multimedia (e.g., VoIP), etc.

- ORB Configuration: Developers can create their own transports. ORBs are developed with additional features, such as secure messaging, multimedia (e.g., VoIP), etc.

- ORB Configuration: Developers can create their own transports. ORBs are developed with additional features, such as secure messaging, multimedia (e.g., VoIP), etc.
Distributed Real-time Concepts and Issues

Background
Hard and soft real-time systems are concerned with the correctness of response in addition to the correctness of the response. Having a fast ORB is important. This alone will not ensure a predictable, real-time system. Combining an enterprise or desktop ORB with an RTOS and predictable network stack will create a real-time distributed system architecture that supports hard and soft real-time systems, in 1999 the OMG adopted the Real-time CORBA standard.

Ingredients of a Real-time ORB

Distributed Priority Inheritance

The primary goal of Real-time CORBA is end-to-end predictable real-time systems architecture. Achieving this through priority based scheduling. Because the remote system doesn't know about the priority of the remote ORB, it needs to implement distributed priority inheritance. Priority of requests initialize remote systems. To avoid this, a message-based priority inheritance is used. A high priority message must have enough time to complete its task before the low priority message. However, a message-based priority inheritance is too slow to be predictable. A distributed priority inheritance architecture solves this problem. Developers can use a distributed priority inheritance and the ability to correctly schedule distributed processes.

Distributed Priority Inheritance

The heterogeneous nature of CORBA is important in priority propagation. Most RTOS have different ranges of priorities. The Real-time CORBA standard defines a prioritized priority range. ORBS supply the ability to map between the range and external. This is true for all systems the CORBA provides a portable and transparent approach to distributed systems building.

Priority Propagation

The primary goal of Real-time CORBA is end-to-end predictable real-time systems architecture. Achieving this through priority based scheduling. Because the remote system doesn’t know about the priority of the remote ORB, it needs to implement distributed priority inheritance. Priority of requests initialize remote systems. To avoid this, a message-based priority inheritance is used. A high priority message must have enough time to complete its task before the low priority message. However, a message-based priority inheritance is too slow to be predictable. A distributed priority inheritance architecture solves this problem.

Priority Banded Connections

Priority propagation ensures that the RTOS can schedule correctly based on the priority of the request. The primary goal of Real-time CORBA is end-to-end predictable real-time systems architecture. Achieving this through priority based scheduling. Because the remote system doesn’t know about the priority of the remote ORB, it needs to implement distributed priority inheritance. Priority propagation ensures that the RTOS can schedule correctly based on the priority of the request.

Distributed Embedded Concepts and Issues

Background

The embedded market is heterogeneous. It is fragmented – separate CPU type, RTOS, or communications media dominates. Many teams create project specific solutions. The physical environment for these systems often impose constraints on the ORB.

Ingredients of an Embedded ORB

Wide Availability

Available a variety of CPUs and RTOS from the common to the unusual

Heat Development

Develop and test before limit overhead system is available

Mass Market Affordable

No ORB run-time royalties

Plug-In Transports

Real-time systems require a predictable end-to-end system to ensure real-time correctness. ORB, RTOS, and Transport. Embedded environments often require plug-in transports to provide real-time services. ORB developers can create their own transports. ORBs are highly flexible, developers can add new transports (e.g., MobiORB, Myrinet). ORB developers have embedded full control over their products. The CORBA specifications and supported transports are controlled for each vendor in a Real-time full interoperable.
**Product Selection**

The ORBExpress product line covers all types of applications. Developers can choose the appropriate product for their application by considering the following criteria:

- **Performance:**/ORBExpress ensures that ORBExpress ORBs are scalable and reliable. ORBExpress ORBs can scale for systems ranging from small to large.
- **Hardware:**/ORBExpress ORBs are available for a variety of embedded targets from the common to the unusual.
- **Software:**/ORBExpress ORBs are available for a variety of Real-time OS's and embedded targets.

**Java**

- **High-Performance:** Combining the ORBExpress architecture with Java (See the charted)
- **Interoperable:** With ORBExpress ORBs for Ada and C++
- **New-Multilanguage support:** supported by a single ORB vendor

**Complete Language Coverage**

ORBExpress supports all major Object-Oriented languages. Each ORB is written in its native language, based on a common ORBExpress architecture. Each implementation contains an inline library and an IDL translator which generates 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.

**Ada 55**

- **Interoperable:** Connect Ada to other languages
- **Simple:** Easier than writing direct bindings to C++
- **Real-time:** Integared with Ada's multi threading and Ada Real-Time Systems Auras
- **Reconfigurable:** In addition add new components in any language

**Oracle**

code according to the CORBA specific language mapping. In addition, each comes with a full suite of documentation and demos programs.

**Vertical Markets**

ORBExpress has been in use since 1987. Projects using it have reached every stage of development from initial definition to high-volume release, deployment, and maintenance. Projects encompass many different vertical domains including:

- **Telemetry/Defense:** Optical switches, radar, cell phones, communications equipment, command, control, communications and intelligence systems, software defined RF
- **Process Control:** Nuclear power plant, laser 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 infotainment, etc.

**Project Highlights**

Many companies and organizations have selected ORBExpress products that use ORBExpress include:

- **Kapost**
- **Laser Processing**
- **LightWave**
- **Fusion Research**
- **Climate Control**
- **Media Distribution**
- **Transportation**
- **Military & Aerospace**
- **Oil & Gas**
- **Healthcare**

**Training and Consulting**

To complement the ORBExpress product line, Objective Interface offers professional services to support ORBExpress users. Training is available either at our offices or on-site at customer locations. Two-3 day courses are available which take developers from initial introduction to ORBExpress proficiency and project readiness. Training on specific ORBs provides 95% inclusive and 90% lab.

**Customer Service**

Solid tech support is essential to ensuring successful product use. Objective Interface is committed to the highest level of technical service available for ORBExpress ORBs. Our tech support is available 24 hours a day, 7 days a week to assist ORBExpress user. Our tech support is available 24 hours a day, 7 days a week to assist ORBExpress users. Training is available either at our offices or on-site at customer locations. Two-3 day courses are available which take developers from initial introduction to ORBExpress proficiency and project readiness. Training on specific ORBs provides 95% inclusive and 90% lab.

**Noteable projects that use ORBExpress include:**

- **Kapost**
- **Laser Processing**
- **LightWave**
- **Fusion Research**
- **Climate Control**
- **Media Distribution**
- **Transportation**
- **Military & Aerospace**
- **Oil & Gas**
- **Healthcare**

**Available Platforms**

<table>
<thead>
<tr>
<th>Product</th>
<th>Operating Systems</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORBExpress ST</td>
<td>Windows, Tru64, Solaris</td>
<td>All major Object-Oriented languages</td>
</tr>
<tr>
<td>ORBExpress GT</td>
<td>Windows, Tru64, Solaris</td>
<td>All major Object-Oriented languages</td>
</tr>
<tr>
<td>ORBExpress RT</td>
<td>Windows, Tru64, Solaris</td>
<td>All major Object-Oriented languages</td>
</tr>
</tbody>
</table>

**Project specific mentoring and consulting is also available.**

- **Developers**
- **Mentoring**
- **Consulting**

**Ease of use:** CorBA has been taught at universities worldwide. Our developers can use the DevKit in on the fly, then develop and compiled tech support experience. ORBExpress includes full source code, high-performance code, scoring between 2.15 and 7.35 on a scale of 2.0-7.5.

The first ORBExpress release is a support is included at no extra cost with each ORBExpress license. Our technical support is available online at our web site, email, and telephone. Our tech support is available at on-site locations. Two-3 day courses are available which take developers from initial introduction to ORBExpress proficiency and project readiness. Training on specific ORBs provides 95% inclusive and 90% lab.

**ORBExpress Product Line**

There are three products in the ORBExpress family

**ORBExpress ST:**

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

**ORBExpress GT:**

Designed for high-performance embedded systems. ORBExpress GT adds to the capabilities of the ST product. ORBExpress GT is available only to C++. It has been ported to a variety of embedded targets from the common to the unusual. A scalable system design provides flexibility for all environments. It’s plug in transport mechanism lets developers compose the transport or embedded ORBExpress ST to new, and lean.

**ORBExpress RT:**

The flag ship of the product line. ORBExpress RT adds to the capabilities of the ST and GT products. ORBExpress RT implements (and extends) the Realtime CORBA standard. It provides support for both hard and soft real-time systems. ORBExpress RT is designed to deliver predictable low latency, plug in transport and transport quality of Service. It’s available to C and Ada only. Each ORB implements in its native language. ORBExpress RT has been ported to a variety of Real-time OS’s and embedded targets from the common to the unusual.
Available Products

<table>
<thead>
<tr>
<th>Platform</th>
<th>Operating System</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>Visual C++</td>
<td>VS Code</td>
</tr>
<tr>
<td>Windows</td>
<td>Visual C#</td>
<td>.NET</td>
</tr>
<tr>
<td>Linux</td>
<td>GCC</td>
<td>GCC</td>
</tr>
<tr>
<td>Solaris</td>
<td>SGI CC</td>
<td>SGI CC</td>
</tr>
<tr>
<td>Solaris</td>
<td>ObjectAda</td>
<td>GNAT</td>
</tr>
<tr>
<td>Solaris</td>
<td>Dec CC</td>
<td>Dec CC</td>
</tr>
<tr>
<td>LINPACK</td>
<td>Apex</td>
<td>Apex</td>
</tr>
<tr>
<td>QNX Neutrino</td>
<td>LynxOS</td>
<td>LynxOS</td>
</tr>
<tr>
<td>Linux</td>
<td>INTEGRITY</td>
<td>INTEGRITY</td>
</tr>
<tr>
<td>Solaris</td>
<td>QNX Neutrino</td>
<td>QNX Neutrino</td>
</tr>
<tr>
<td>Solaris</td>
<td>PowerMax</td>
<td>PowerMax</td>
</tr>
<tr>
<td>Linux</td>
<td>TimeSys RT JVM</td>
<td>TimeSys RT JVM</td>
</tr>
<tr>
<td>Solaris</td>
<td>PowerMax</td>
<td>PowerMax</td>
</tr>
<tr>
<td>Linux</td>
<td>LWP</td>
<td>LWP</td>
</tr>
<tr>
<td>Solaris</td>
<td>NetBSD</td>
<td>NetBSD</td>
</tr>
<tr>
<td>LVFS</td>
<td>Linux</td>
<td>Linux</td>
</tr>
<tr>
<td>Solaris</td>
<td>VxWorks</td>
<td>VxWorks</td>
</tr>
<tr>
<td>Windows</td>
<td>VS Code</td>
<td>VS Code</td>
</tr>
<tr>
<td>Solaris</td>
<td>TimeSys RT JVM</td>
<td>TimeSys RT JVM</td>
</tr>
<tr>
<td>Solaris</td>
<td>PowerMax</td>
<td>PowerMax</td>
</tr>
<tr>
<td>Linux</td>
<td>LWP</td>
<td>LWP</td>
</tr>
<tr>
<td>Solaris</td>
<td>NetBSD</td>
<td>NetBSD</td>
</tr>
<tr>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
</tr>
<tr>
<td>Solaris</td>
<td>VxWorks</td>
<td>VxWorks</td>
</tr>
</tbody>
</table>

Vertical Markets

ORBexpress 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:

- Telephony/Security: Optical switches, radio, phone, video and voice equipment.
- IIoT: Sensors and actuators, wireless, etc.
- Embedded system: Automotive, aerospace, medical, etc.
- Communication: Security, industrial, etc.
- Telecommunication: Satellite, sub-sat, etc.
- Telecom/Datacom: Telecom, internet service, etc.
- Real-time Systems: Real-time systems.

Complete Language Coverage

ORBexpress supports all major Object-Oriented languages. Each ORB is written in its native language, based on a common ORBexpress architecture. Each implementation contains an interfacing library and an IDL translator which generates 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 previous page are for the ORB4).

- Codegen: ORBexpress GT is available only for C++.

Simple Support: Common support team for all sides of the product.

Java

High Performance: Combining the ORBexpress architecture with Java (see the chart below).

- Interoperable: With ORBexpress for Ada and C++.

- Low-level: Application languages supported by a single ORB vendor.

- Simplified Support: Common support team for all sides of the product.

Java

- 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 previous page are for the ORB4).

- Codegen: ORBexpress GT is available only for C++.

- Simple: Easier than writing direct bindings to C++.


- Reconfigurable: Incrementally add new components in any language.

Product Selection

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

Training and Consulting

To complement the ORBexpress product line, Objective In- terface offers professional services to support ORBexpress users. Training is available either at our facilities or on-site at customer locations. Two-day courses are available which take developers from an introduction to ORBexpress products and project readiness. Hands-on training provides 50% lecture and 50% lab.

- Developing with developers to create a CORBA-based OS.

- Evaluating trade-offs in the architecture.

- Determining how to obtain the least performance from ORBexpress.

- Review and support during project reviews.

Customer Service

Solid tech support is essential to ensuring successful product use. Objective Interface is committed to reliable tech support. When a problem occurs, a trained support specialist is available 24/7. Our support specialists are experienced ORBexpress users and can provide fast support on a very wide variety of technology issues. ORBexpress product support is available on a scale of -2.0 to +2.0.

Training and Consulting

To complement the ORBexpress product line, Objective Inter- face offers professional services to support ORBexpress users. Training is available either at our facilities or on-site at customer locations. Two-day courses are available which take developers from an introduction to ORBexpress products and project readiness. Hands-on training provides 50% lecture and 50% lab.

- Developing with developers to create a CORBA-based OS.

- Evaluating trade-offs in the architecture.

- Determining how to obtain the least performance from ORBexpress.

- Review and support during project reviews.

Customer Service

Solid tech support is essential to ensuring successful product use. Objective Interface is committed to reliable tech support. When a problem occurs, a trained support specialist is available 24/7. Our support specialists are experienced ORBexpress users and can provide fast support on a very wide variety of technology issues. ORBexpress product support is available on a scale of -2.0 to +2.0.

The first year of ORBexpress support is included at no extra cost with each ORBexpress license. After the first year, an extended support package is available. Improved ORBexpress support will also be available in the future.
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.