VOYAGER

Core Technology

Version 2.0 Now Available. Download Now!

Voyager Core Technology[tm] 2.0 let's you quickly and easily build advanced high-performance distributed Java solutions that fully satisfy enterprise needs. It is a standards-neutral development platform and object request broker that gives you unmatched ease of use, unique features and capabilities, far-reaching innovations and pure seamless Java integration for most distributed object models.

For our existing Voyager Core users who want to know more about why they should upgrade to VCT 2.0 and the differences between VCT 2.0 and prior releases, click here.

VCT 2.0 is the next-generation of ObjectSpace's flagship technology and offers users a multitude of new capabilities, along with enhancements to its existing code base. These include:

- **CORBA integration**. Voyager has integrated native CORBA support, and can communicate with other CORBA systems regardless of their implementation language, plus full native support for all IDL types and IIOP with bi-directional IDL/Java conversion. Proxy classes are generated at runtime so no stub generators are required. Java objects are automatically CORBA enabled without modification and the runtime system requires no IDL repository or helper files.
- **Proxy Generation**. Voyager's dynamic proxy generation utility can eliminate manual proxy class generation, otherwise known as vcc, along with the associated class synchronization requirements. This utility can easily generate executable Java .class files or Java source code.
- **Dynamic Aggregation**[tm]. This advanced feature allows you to add secondary objects, referred to as facets, to a primary object at runtime. Dynamic Aggregation lets you add behavior to a third party component whose source is not available or can extend an object's behavior at runtime. This feature represents a significant step forward for object modeling and complements the traditional mechanisms of inheritance and polymorphism.
- **Activation**. The activation framework allows objects to be persisted to any kind of database and automatically

re-activated in the case that the VM is restarted. This allows an object to survive VM restarts, where the object continues to receive messages as if the VM had never shut down. An object does not have to be modified in any way to be activated.

- **Timer**. Voyager offers Timer Services, Stopwatch and Timer classes, to facilitate common timing chores. The Stopwatch allows you to clock time intervals and print time measurement statistics. Use of the Timer allows you to generate timer events and add listeners to timers.

Also, in keeping with ObjectSpace's philosophy of providing technology tools to help make your development efforts as productive and effective as possible, ObjectSpace will provide a Voyager 2.0 compatible version of VoyagerDB. VoyagerDB will be available, including source, on the ObjectSpace Web site in 4th quarter 1998.

Technical Highlights

- What Is ObjectSpace Voyager?
- Write Once, Use Remotely Anywhere
- Fundamental ORB Features
- Mobile Objects
- Autonomous Agents
- CORBA Integration
- DCOM Integration
- Distributed JavaBeans
- Security
- Applets
- Activation
- Space[tm] - Group Communications
- Federated Directory Service
- Core Technology Training
- QuickStart Program
- Platinum Partners Program
- Voyager Technology Partner Program
- Voyager Hardware Distribution Licensing
- The Business Model
- Year 2000 Compliance

Compact - 100% Java
"Write Once, Use Remotely Anywhere"

What Is ObjectSpace Voyager?

ObjectSpace Voyager is the ObjectSpace product line designed to help developers produce high-impact distributed systems quickly. Voyager is 100% Java and is designed to use the Java language object model. Voyager allows you to use regular message syntax to construct remote objects, send them messages, and move them between programs. This reduces learning curves, minimizes maintenance, and, most importantly, speeds your time-to-market for new advanced systems. Voyager's architecture is designed to provide developers full flexibility and powerful expansion paths.

The root of the Voyager product line is the ObjectSpace Voyager

Core Technology. This product contains the core features and architecture of the platform, including a full-featured, intuitive ORB with support for mobile objects and autonomous agents. Also in the core package are services for persistence, scalable group communication, and basic directory services. The ObjectSpace Voyager Core Technology is everything you need to get started building high-impact systems in Java today.

As the industry evolves, other companies providing distributed technologies struggle with how to adapt to the new Java language. These companies are required to adapt earlier object models to fit the Java language. This results in a series of compromises that together have a dramatic impact on time-to-market and development costs. Voyager, on the other hand, is developed to use the Java language as its fundamental interface.

One of Java's primary differentiations is the ability to load classes into a virtual machine at run time. This capability enables infrastructures to use mobile objects and autonomous agents as another tool for building distributed systems. Adding this capability to older distributed technologies is often impractical and results in difficult-to-use infrastructures. Voyager provides seamless support for mobile objects and autonomous agents.

Top

Write Once, Use Remotely Anywhere

ObjectSpace Voyager is written 100% in Java. This approach provides developers the "Write Once Run Anywhere" portability of the Java language. Unfortunately when it comes to allowing other remote clients to use your class, there is still an ongoing debate over protocol and language binding standards in the industry.

With Voyager, programmers can take existing unmodified Java classes and publish them for use by CORBA. With the future addition of DCOM in 1QTR1999, developers will be able to write network services once as pure portable Java code, and publish these services for use by Voyager, CORBA, or DCOM. This allows the use of a single code base for all platforms and all distributed object models.

Top

Fundamental ORB Features

The following is a summary of the fundamental ORB features within Voyager. For additional information and details, read this white paper.

- **Nonintrusive Remote Enabling.** Voyager can remotely construct and communicate with any Java class, even a third-party library class, without requiring source. Other technologies typically require the use of IDL files, interface definitions, and modifications to the original class, which consume development time and tightly couple your domain classes to a particular ORB technology.

- **Java-Based Remote Construction and Messaging.** Objects can be constructed remotely using the regular Java construction syntax, static methods can be executed remotely, remote exceptions are automatically rethrown to the caller, and serializable objects can be passed and returned by value. All of these features result in simpler and quicker development for a Java programmer.
- **Advanced Messaging.** Version 1.0.0 includes support for synchronous, oneway, future, group multicast, and publish/subscribe group multicast messages using TCP communications. Voyager also supports full dynamic method invocation at run time and programmable message timeouts.
- **Dynamic Class Loading.** Voyager provides remote class loading capabilities from a remote class repository. Voyager can load classes from remote servers or directories, and serve classes to remote clients.

Top

Mobile Objects

Voyager was designed from the start to allow full object mobility. With one message, you can move an object at run time from one virtual machine to another, even if the object is being used by other clients at the same time. Voyager uses an innovative message forward and backfeed notification architecture to ensure message traffic efficiently tracks the object as it moves.

Mobile objects allow developers to quickly and easily build software distribution and management systems as well as dynamic run-time load balancing. Perhaps most beneficial is the ability of an object to move into another virtual machine and communicate at native Java speed with another object. Often, a single move followed by many small message invocations can be faster than performing the messaging across a network. Voyager's mobile objects allow programmers to choose from several approaches when optimizing the performance, reliability, and scalability of distributed systems.

Top

Autonomous Agents

Many distributed applications can benefit from active, mobile objects. Voyager allows agents, autonomous objects, to move themselves and continue executing as they move. In this way, agents can act independently on behalf of a client, even if the client is disconnected or unavailable. This approach is particularly valuable in any type of workflow or resource automation.

Top

CORBA Integration

Customers can take any existing Java class, even without access to the source code, and use it as a CORBA server, automatically creating CORBA IDL files from Java. Customers can also process existing IDL for CORBA services, automatically create Java interfaces, and then use these interfaces to communicate with the CORBA services using natural Java language syntax. The CORBA integration works with leading C++ and Java CORBA ORBs. Click here for details on the Version 2.0 release. For a complete white paper with excellent examples, click here.

Top

DCOM Integration

Support for Microsoft's DCOM communication model will be added to ObjectSpace Voyager Core Technology is currently scheduled for initial availability in 1QTR1999. Because of Voyager's innovative nonintrusive architecture, developers will be able to write code once and use it remotely without modification using Voyager, CORBA, and DCOM. This protects investments from the ongoing struggle for market leadership between CORBA and Microsoft DCOM.

Top

Distributed JavaBeans

Because Voyager is designed to allow remote-enabling of any existing Java class without modification, users can remote-enable any JavaBeans Listener. A remote-enabled Listener can be added to any JavaBeans event source, allowing seamless distribution of JavaBeans events without any additional coding. Using Space, you can even multicast bean events to many Listeners scattered throughout a system.

Top

Security

The Voyager Core Technology comes complete with an optional Voyager Security Manager. The Voyager Security Manager is a pluggable security manager that restricts the operations of foreign objects. A foreign object is an object whose class was loaded across the network from another program.

Top

Applets

Some ORBs prevent objects in browsers from communicating with objects that are not located in the browser's server. Voyager does not impose this restriction and includes an integrated software router that allows objects to communicate with each other anywhere in the world, firewalls permitting.

Top

Activation

Voyager Core Technology's activation framework allows objects to be persisted to any kind of database and automatically re-activated in case the VM is restarted. An object does not have to be modified in any way to be activated.

Top

Space[tm] - Scalable Group Communications

Space[tm] is an innovative architecture designed to allow programmers to build distributed "rooms" in which objects reside. By connecting together smaller building blocks called subspaces, the programmer builds a virtual Space. Then users can connect to this virtual Space as if it were a single object. When you send messages into the Space, they are cloned to adjacent Spaces and then multicast in parallel to the objects in the Space.

Because a Space can contain many different types of often unrelated objects, all objects in a Space might not understand every type of message. Special type-checking is performed so that each message is delivered only to objects that understand the particular message.

Top

Federated Directory Service

The Voyager Federated Directory Service allows programmers to build and link together hierarchies for the management of objects in a distributed system. Using naming strategies, programmers can locate and then connect to an existing object without knowledge of its true location.

Top

Core Technology Training

ObjectSpace offers several training and education classes that aid developers in the adoption and use of Java and Voyager

Top

QuickStart Program

Commitment to the early adoption of new technology represents a significant step for any organization. The adoption process involves much more than development of an application. In addition to development of a solid fundamental understanding of the technology itself by the entire team, a working knowledge of the applications and implications within the client organization’s business must be developed. Also, the cultural readiness of the client organization must be considered in order to position the all-important first project and the associated development team appropriately to lead the continued evolution of the overall client organization.

The QuickStart Program accomplishes this through close interaction between ObjectSpace and the client team. Aspects of the program include: selected training classes executed at the beginning of the project - just as the knowledge is needed, on-site mentoring through the first two weeks of the project in order to establish a solid program foundation, and recurrent on-site consulting to provide checkpoints during the life of the project. In some cases the QuickStart program is used to establish both technology adoption and proof of concept for companies exploring new technologies such as Java and Voyager. For additional information or to speak with a high-level consultant and explore how ObjectSpace can QuickStart your organization into high-impact rapid distributed system development with Java, contact us now.

Top

Platinum Partners Program

The platinum partner program is designed to form a close relationship between ObjectSpace and a small number of nonconflicting strategic partners to advance the Voyager platform. Platinum members receive the following benefits:

- Full source code and design documentation for select ObjectSpace Voyager technology
- Binary and source code support with direct second-tier or developer contacts
- Onsite training and consultation
- Participation in the Voyager Advisory Board
- Participation in all alpha and beta early access programs
- Source code escrow

Platinum members participate under nondisclosure in the feature definition and design of Voyager technology. They bring their expertise together with ours to ensure the Voyager platform evolves to satisfy the industrial commercial needs in the industry today.

To become a platinum member, a company must apply for membership and be accepted. Terms of membership are communicated on a case-by-case basis. We cannot guarantee this offering because we must avoid creating conflict of interest between ObjectSpace customers. For additional information or to apply for membership, contact ObjectSpace now.

Top

Voyager Technology Partner Program

The Voyager Technology Partner Program is designed for system integrators and small consulting and product businesses that provide advanced technology solutions to their clients. Many opportunities are available to promote your company, provide technical support, and put you ahead of the competition. Technology Partners receive the following services:

- **Business:** These services will help you develop contacts with other companies, form strategic relationships with other members, and promote your company all at the same time.
- **Educational:** Members are eligible for discounts on all

ObjectSpace open-enrollment educational courses.
- **Sales/Marketing:** We've included several options to help you co-market with ObjectSpace and to expand your existing client base.
- **Technical:** These services grant you access to technical support, early access to the Voyager Core Technology and provide us with feedback on future releases.

Top

Voyager Hardware Distribution Licensing

Voyager is designed to be very powerful but also very compact. It is ideal for use in many embedded Java devices. Special licensing is required if you are planning to deploy your Java application containing Voyager embedded in or packaged with hardware. These licenses are negotiated on a case-by-case basis. For additional information, contact ObjectSpace now.

Top

The Business Model

Years from now, companies will not make much money selling basic middleware. DCOM will be embedded and distributed everywhere. CORBA price points are already plummeting. In the not-too-distant future, the bulk of the features currently in the Voyager Core Technology will be available in several forms and locations. Your cost is in development time, and your revenues are increasingly dependent upon time-to-market. ObjectSpace believes Voyager's Java-centric binding, advanced mobile object features, and innovative services provide the best basis for rapid development of distributed systems in Java.

As the industry changes, ObjectSpace will continue to develop and sell partnerships, support, and other services, but will continue to unveil more and more next-generation add-on features for the ObjectSpace Voyager platform. These add-ons will progress in the areas of security, group communication, persistence, and concurrency and will deliver the same time-to-market and rapid development advantages found in the Voyager Core Technology today. These add-ons will be available for purchase from ObjectSpace in the coming months.

ObjectSpace is also pursuing several partnerships for the creation of technology integrations and enhanced development tools. Our relationships, based on the deployment of JGL, will enable the rapid distribution, adoption, and integration of the ObjectSpace Voyager platform.

As you look further into the future, you will see ObjectSpace using the Voyager technology base as the platform for its own next-generation product lines. As definite product-release dates approach, we will announce these longer-term projects. Be assured that we will use the advantages of Voyager, such as agent technology, to deliver products that, until now, you have only speculated about.

Top

ObjectSpace Voyager Year 2000 Compliance Statement

ObjectSpace warrants that ObjectSpace Voyager does not contain any code that is dependent on, or is encumbered by, the Year 2000 boundary. Any limitations incurred by the operating system, virtual machine, JDK, or other software not written by ObjectSpace may appear as limitations of ObjectSpace Voyager. ObjectSpace has no control over dependent third-party software, and cannot guarantee its correct behavior when used directly or with ObjectSpace Voyager. We do not test third party software for year 2000 compliance, as that is the responsibility of the respective vendors. Similarly, we cannot guarantee user code written using ObjectSpace Voyager will itself be free of defects causing year 2000 problems.

The IEEE is currently preparing standards and guidelines for Year 2000 compliance. We intend to follow those guidelines as appropriate for our line of business.

Top