digital™

Software Product Description

PRODUCT NAME: ObjectBroker for Win32, Version 2.7 **SPD 50.73.04**

DESCRIPTION

ObjectBroker for Win32 (Windows NT™ (Intel™ and Alpha) and Windows 95 ®), Version 2.7 provides application developers and system integrators with the development tools and a run-time environment to integrate applications and services distributed across these environments: Digital UNIX™, SunOS™, Solaris™, HP-UX®, IBM® AIX®, Microsoft® Windows™ and OpenVMS™ (VAX and Alpha).

ObjectBroker Version 2.7 is Digital's implementation of the Object Management Group's (OMG®) Common Object Request Broker Architecture (CORBA™) specification V1.2 and V2.0 for enterprise wide production use. ObjectBroker also extends Microsoft OLE technology to a multiplatform distributed computing environment.

ObjectBroker simplifies distributed application development by providing platform-independent, client/server, programming interfaces. It offers an object-oriented (O-O) approach to dynamic linking of independently developed applications and services. ObjectBroker allows application processes to transparently invoke, control, and interact with each other across multiple platforms.

Features

- A high-level, standards-based application programming interface (API) across 15 platforms.
- Compliance with OMG's CORBA specification, protecting users' software development investments.
- C++ language bindings provide C++ programmers a natural interface as defined in CORBA Specification Version 2.0. C programming language and Visual Basic (Win16) bindings continue to be supported.

November 1996
AE-QDFJE-TE

- Integration with Kerberos/DCE Security for optional integration of secure client and server applications on Windows NT and Digital UNIX.
- Support for the CORBA V2.0 Initialization Service.
- ObjectBroker COSS Naming implementation has become invocable, allowing users to replace it with one of their choosing.
- OLE Integration (known as the OLE Portal) and DDE Integration (known as the DDE Listener) provide the ability for shrink-wrapped and custom desktop applications, that comply with the Microsoft OLE and DDE interfaces, to communicate with certain remote ObjectBroker servers. Developers can use the OLE Portal and DDE Listener features today to begin moving toward open client/server environments.
- Script servers allow client access to existing applications using command line interfaces, with no source changes necessary, facilitating the migration to client/server computing.
- O-O abstraction provides insulation of clients and servers from changes in network topology or application code through the use of implementation-independent brokering services.
- QuickStart makes it easier for new users to get an application up and running, as well as making experienced users more productive.
- A Server Management API to enables system managers to manage servers through a callable API.

ObjectBroker for Win32 Capabilities

ObjectBroker V2.7 is a software product that enables client and server applications, on different computer systems, to communicate over a network. Each message between the client and server is called a request and the process of sending a request is called invocation.

Client Application

The client application sends a request, possibly over a network, for an operation to be performed. The request is performed by an implementation within a server application.

Server Application

The server application contains implementations that satisfy client requests when they are received. The server application returns information back to the client application.

ORB

The Object Request Broker (ORB) provides functions specific to the client and to the server. A client request first goes to the ORB, which is the primary mechanism that ObjectBroker uses to locate ObjectBroker implementations on the network.

The server-side ORB answers requests from clients and dispatches them to the appropriate server application.

Agent

The agent is the entity provided as part of ObjectBroker V2.7 that performs services on behalf of a user in the context of ObjectBroker. An agent must be present on any computer that supports applications that provide services using ObjectBroker. The agent maintains a list of active servers on the computer and can automatically start new servers if needed to respond to client requests.

A new function, Proxy Failure Log, has been added to ObjectBroker V2.7. This function provides the ObjectBroker agent with the ability to record certain types of security failures in the agent log file.

Repository

The ObjectBroker repository contains definitions of ObjectBroker interfaces and their associated implementations. An ObjectBroker interface is the set of operations and attributes that can be performed on a object. An implementation contains the code that satisfies a client request for operations on a specific object.

The interface repository also contains method maps, which describe the criteria by which to select implementations. The interface definitions in the Interface Repository are written in OMG IDL, which is described in the CORBA specification and in the ObjectBroker documentation.

Context Object

A context object contains information about a user's preferences, which are taken into account during the resolution process, and additional properties and values which are propagated to the server side during an invocation request. In addition, applications can store profile information in the context object that is not used as resolution criteria.

Registry

The ObjectBroker registry stores information outside of user applications, that is used by all of the components of ObjectBroker.

The configuration partition of the registry contains information about the configuration of ObjectBroker on a given computer, including the list of available transports and the authentication package to be used.

The security partition of the registry contains information about a user's authorization to access specific implementations and methods. Along with authorization information for users, the security registry also contains information that describes the list of remote users that are allowed to access ObjectBroker on the current computer

The advertisement partition of the registry stores information about servers, where to find them, and what they provide.

The implementation partition of the registry contains information about object implementations that have been installed on the computer. At a minimum, the implementation registry contains the unique identifier of the implementation, the operating system specific command for starting the server process that contains the implementation code, and the list of attributes to be defined as environment variables when the process that executes the implementation is started.

Developer's Features

The ObjectBroker product contains many functions to aid in program development. There are utilities to generate code for the client and server applications, utilities to define and manage security and environmental settings, and utilities to support other products and their interface to ObjectBroker.

For code generation, ObjectBroker utilizes a language to describe the interfaces of your distributed application. ObjectBroker utilities use this language to generate code and other definitions used at run-time.

Developer's features include the following:

- ObjectBroker V2.7 implements the CORBA 2.0 standard for allowing an application to initialize itself into and ORB environment.
- CORBAservices naming service is implemented as an invokable server. This makes it possible for customers to replace the CORBAservices Naming Service implementation supplied with ObjectBroker without relinking the client.
- OMG Interface Definition Language (IDL) for describing classes and messages
- Support for User Defined Types (UDT)
- Generation of client-side stubs for operations defined on one or more interfaces:
  - Static interfaces use code skeletons (stubs) containing routines that cannot be changed. Stubs provide a linear interface and reduce the complexity of the method resolution.
  - Dynamic interfaces use routines that the client defines and builds as it is running. The dynamic interface provides client with more flexibility to use deferred synchronous operations, different method maps, and new interfaces.
- Generation of OMG IDL skeletons (dispatchers), registration routines, and method routine stubs for servers
- Support for the Dynamic Invocation Interface (DII) routines

Additionally, ObjectBroker provides utilities to manage the security settings and the overall environment. These utility functions are available as command-line functions as well as through graphical user interface utilities (GUIs).

Digital Extensions to the CORBA specification provide enhanced capabilities. These extensions include:

Languages

- The Implementation Language (IML) describes methods and their implementations.
- The Method Mapping Language (MML) maps operations to the methods that implement them.

QuickStart is a new code generation facility. It is a prototyping tool that allows a user to take an OMG IDL file and generate code for the client, server, methods, make files and more. QuickStart reduces the complexity and time to create a running ObjectBroker client/server application and is particularly useful for evaluators to test their IDL and showcase ORB functionality.

Improved System Management makes it possible for a system manager to monitor the state of server application within a distributed production environment through a callable API. This facilitates integration with other management tools or products.

ObjectBroker Graphical User Interface Utilities provide a fully supported, windowed set of utilities that help system integrators manage the CORBA environment and programmers develop new distributed applications. The utilities provide interfaces that:

- View and manipulate context objects, repositories, and remote servers
- Generate code for programmers
- Set up the environment (security, proxies, transport)

The user interface of each utility is based on CORBA concepts. The utility's interface is native to the platform on which it is executing (Motif®, Windows).

Security Enhancements

ObjectBroker V2.7 provides Kerberos DCE authentication via DCE's Generic Security Services interface (GSSAPI). Kerberos provides authentication, mutual authentication, and protection against replay and sequencing attacks. Implementing security enhancements through GSSAPI on ObjectBroker V2.7 on Digital Unix and Windows NT, provides more options in choosing and integrating third-party authentication packages. In addition, support for DCE security facilitates integration with DCE-based software systems.

Network Tester

The network tester utility provides the ability to determine whether the network is properly configured for use with ObjectBroker. A user can specify the amount of data to be sent between a network tester client and server. This tool also provides detailed error messages when failures occur. The user can test ObjectBroker and the network between two computers without having to write an ObjectBroker test program.

Multiple Simultaneous Transports

ObjectBroker V2.7 supports multiple network transports on a single system; each transport can run concurrently on the system. A client running TCP/IP and DECnet can communicate with a server that is running either TCP/IP, DECnet, or both. The system administrator controls which transports are available to client and server applications on each system. The platforms that support multiple simultaneous transports are OpenVMS (VAX and Alpha), Microsoft Windows, and Windows NT (Intel and Alpha).

Windows 95 Shell Extensions

In ObjectBroker V2.7, you can use the ObjectBroker Windows 95 shell extensions to develop and manage ObjectBroker applications and resources with the click of a mouse button. Online help is available from the ObjectBroker shell extensions.

Load Balancing

ObjectBroker allows you to associate descriptive attributes to server applications so that clients can select a particular server application based on those attributes. ObjectBroker also provides a means of balancing the load among multiple copies of a server application on a given computer.

HARDWARE REQUIREMENTS

Processors Supported — Alpha Processors for Development and Run-time Only:

Alpha: DEC 2000 Model 300S,
DEC 2000 Model 500

Processors Supported — Intel Processors for Development and Run-time Only:

Intel IBM-compatible 386 or 486, or a Pentium® PC system with CD–ROM drive

Disk Space Requirements

Development		
Platform	**To Install**	**After Installation**
Windows NT Alpha	18 MB	13MB
Windows NT Intel	17 MB	16 MB
Windows 95	11.4 MB	11.4 MB

Runtime		
Platform	**To Install**	**After Installation**
Windows NT Alpha	7 MB	7 MB
Windows NT Intel	6 MB	6 MB
Windows 95	4.6 MB	4.6 MB

SOFTWARE REQUIREMENTS

Supported Alpha and Intel configurations running the following software versions:

- Windows NT 3.51
- Windows 95
- One of the following communications protocol interfaces:
  - DECnet - PATHWORKS for Windows NT, Version 4.1
  - TCP/IP (included with O/S)

OPTIONAL SOFTWARE

Visual C ++ Version 4.0 (for program development)

Shell Exentions:

- Windows NT Shell Preview

Security:

- NT DCE V1.1A

ObjectBroker development and run-time licenses for the following platforms can be used in conjunction with these product.

- ObjectBroker for OpenVMS VAX, Version 2.7
- ObjectBroker for OpenVMS Alpha, Version 2.7
- ObjectBroker for Microsoft Windows, Version 2.7
- ObjectBroker for HP-UX, Version 2.7
- ObjectBroker for IBM AIX, Version 2.7
- ObjectBroker for Digital UNIX, Version 2.7
- ObjectBroker for SunOS, Version 2.7
- ObjectBroker for Solaris, Version 2.7

GROWTH CONSIDERATIONS

The hardware/software requirements for any future version of this product may be different from the requirements for the current version.

ObjectBroker for Win32, Version 2.7 **SPD 50.73.04**

DISTRIBUTION MEDIA

CD–ROM

ORDERING INFORMATION

Windows NT Development Option

Software License:
QL-23UAY-**

Software Product Services:
QT-23UA*-**

Windows NT Run-time Option

Software License:
QL-23VAY-**

Software Product Services:
QT-23VA*-**

Windows 95 Development Option

Software License:
QL-4ZDA*-**

Software Product Services:
QT-4ZDA*-**

Windows 95 Runtime Option

Software License:
QL-4ZEA*-**

Software Product Services:
QT-4ZEA*-**

ObjectBroker Media and Documentation

ObjectBroker Consolidated Development CD–ROM:
QA-0PKAA-H8
ObjectBroker Pilot Package (Media/License combination):
QB-0PKAA-AA
ObjectBroker Documentation only:
QA-0PKAA-GZ

* Denotes variant fields. For additional information on available licenses, services, and media, refer to the appropriate price book.

SOFTWARE LICENSING

One platform-specific ObjectBroker Development license is required per concurrent user for developing, compiling, and/or linking applications. At least one license is required per operating system.
To deploy ObjectBroker, the appropriate platform-specific runtime license is required for each concurrent user included in the deployment.

For more information about Digital's licensing terms and policies, contact your local Digital office.

SOFTWARE PRODUCT SERVICES

A variety of service options are available from Digital. For more information, contact your local Digital office.

SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum of this SPD.

The above information is valid at time of release. Please contact your local Digital office for the most up-to-date information.