digital™

Software Product Description

PRODUCT NAME: ObjectBroker for UNIX, Version 2.7 **SPD 47.07.04**

DESCRIPTION

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

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 bindings continue to be supported.

November 1996
AE-QC4LE-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 invokable, 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 enables system managers to manage servers through a callable API.

ObjectBroker Version 2.7 Capabilities

ObjectBroker 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 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.

Script Server

The script server is a special ObjectBroker server process which allows legacy applications to be encapsulated as object implementations. The script server allows applications that export a command line interface to be used as object implementations. Client applications have no knowledge that an object's implementation is actually implemented as a script.

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 an ORB environment.
- CORBAservices naming service is implemented as an invocable 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 applications 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 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 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. Subsequent to authentication, ObjectBroker Security Authorization checks are performed based on information in the security partition of the ObjectBroker Registry. Authorization checking ranges from security model mapping to object Access Control Lists (ACL's) offering method-level granularity.

OLE and DDE

ObjectBroker works cooperatively with Microsoft's Object Linking and Embedding (OLE) to provide a network "portal" from Microsoft Windows platforms to ObjectBroker server platforms. ObjectBroker applications on UNIX and OpenVMS systems can be made OLE-aware (by implementing a set of pre-defined OLE methods shipped with ObjectBroker) and will respond to requests from OLE applications on Microsoft Windows.

The ObjectBroker OLE Portal processes OLE API calls on Microsoft Windows and maps them to messages. These messages are then sent to an appropriate ObjectBroker server on the network for resolution. From a user's perspective, the mapping process is transparent. Within an OLE application, the OLE Portal appears as another local OLE server application, which can be expanded to a window to display OLE-aware applications/servers on the network. Information from sources across the network can be linked or embedded into desktop applications through ObjectBroker. The ObjectBroker OLE portal employs OLE Version 2.0 interfaces and also provides support for OLE Version 1.0 applications.

ObjectBroker extends the Microsoft Dynamic Data Exchange (DDE) communication protocol to allow Microsoft Windows applications on a networked PC to interact using DDE with applications running on, Solaris, SunOS, IBM AIX, HP-UX, OpenVMS and Digital UNIX. Applications can function as DDE clients or servers. DDE support in ObjectBroker allows two Microsoft applications to communicate on different PCs.

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.

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 Digital UNIX Development and Run-Time Only:

Alpha:	DEC 2000 Model 300, DEC 2000 Model 500
	DEC 3000 Model 300,

DEC 3000 Model 300L,
DEC 3000 Model 300LX,
DEC 3000 Model 300X,
DEC 3000 Model 400,
DEC 3000 Model 400S,
DEC 3000 Model 500,
DEC 3000 Model 500S,
DEC 3000 Model 500X,
DEC 3000 Model 600,
DEC 3000 Model 600S,
DEC 3000 Model 700,
DEC 3000 Model 800,
DEC 3000 Model 800S
DEC 3000 Model 900,
DEC 4000 Model 6xx, 7xx Alpha Series,
DEC 7000 Model 7xx, 7xx Alpha Series,
DEC 10000 Model 6xx, 7xx Alpha Series
AlphaServer 1000 4/200, 4/233, 4/266
AlphaServer 2000 4/200, 4/233, 4/275, 5/250, 5/300
AlphaServer 2100 4/200, 4/233, 4/275, 5/250, 5/300
AlphaServer 2100A 4/275, 5/250, 5/300
AlphaServer 8200 5/300
AlphaServer 8400 5/300
AlphaStation 200 4/100, 4/166, 4/233
AlphaStation 250 4/266
AlphaStation 400 4/233
AlphaStation 600 5/300

Processors Supported - HP-UX-based Processors for Development and Run-time Only:
HP® 9000-7xx and HP9000-8xx Family of RISC Processors

Processors Supported - AIX-based Processors for Development and Run-time Only:
RISC System/6000

Processors Supported - SunOS and Solaris Development and Run-time Only:

- Sun/SPARC® Classic
- Sun/SPARC ELC, IPC, IPX

- Sun/SPARC 2, 3xx, 4xx, 6xx, 1000xx, 2000

Disk Space Requirements

Development		
Platform	**To Install**	**After Installation**
Digital UNIX	14.2 MB	12.7 MB
SunOS	8.9 MB	8.9 MB
Solaris	19.6 MB	9.6 MB
HP-UX	11.6 MB	11.4 MB
IBM AIX	9.3 MB	7.8 MB

Runtime		
Platform	**To Install**	**After Installation**
Digital UNIX	8.0 MB	6.6 MB
SunOS	4.7 MB	4.7 MB
Solaris	10.1 MB	5.1 MB
HP-UX	6.2 MB	6.0 MB
IBM AIX	6.7 MB	6.2 MB

This value refers to the disk space required on the user file system. This size is an approximate; actual size may vary depending on the user's system environment, configuration, and software options.

SOFTWARE REQUIREMENTS

Supported UNIX configurations with the following software versions:

Operating Systems

- Digital UNIX Operating System Version 3.2C through Version 4.0
    - Digital UNIX V4.0 requires patch OSF400-019
- HP-UX HP9000-7xx, 8xx Operating System, Version 9.04 through Version 9.x, Version 10.01
- IBM AIX Operating System Version 4.1.2
- SunOS Operating System Version 4.1.3

- Solaris Operating System Version 2.4-5 and Version 2.5

Windowing Software

- Digital UNIX Operating System - Motif for Digital UNIX 1.2
- HP-UX Operating System - Motif HP Vue V2.01 A.01.03
- IBM AIX Operating System - IBM AIXwindows Environment/6000 V1.2
- SunOS Operating System - SunOS IXI premier Motif 1.2.4(b)
- Solaris Operating System - Included with operating system

Transports

- TCP/IP included with Digital UNIX, SunOS, Solaris, HP-UX, and IBM-AIX

OPTIONAL SOFTWARE

DCE Security
Digital UNIX 3.2C-3.2x - DCE Version 1.3

Compilers		
Platform	**C Language**	**C ++ Language**
Digital UNIX	Digital UNIX C Developers Extension V1.2	DEC C++ V5.0
SunOS	C ANSI SC 1.0	SPARCompiler C++ 4.0.1
Solaris	C ANSI SC 3.01	SPARCompiler C++ 4.0.1
IBM AIX	C set for AIX V3	C Set ++ for AIX V3
HP-UX	HP C Compiler A.09.77	HP C++ A.03.50 with patch PHSS_6468

ObjectBroker development and run-time licenses for the following platforms can be used in conjunction with the UNIX platforms:

- ObjectBroker for OpenVMS VAX Version 2.7
- ObjectBroker for OpenVMS Alpha Version 2.7
- ObjectBroker for Windows NT Intel Version 2.7
- ObjectBroker for Windows NT Alpha Version 2.7
- ObjectBroker for Microsoft Windows Version 2.7

- ObjectBroker for Windows 95 Version 2.7

GROWTH CONSIDERATIONS

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

DISTRIBUTION MEDIA

- CD–ROM

ORDERING INFORMATION

ObjectBroker for UNIX Software Licenses

Orderable for Specific UNIX Operating Systems:

ObjectBroker for Digital UNIX Development:
QL-0TSA9-**

ObjectBroker for Digital UNIX Run-Time Only:
QL-0TTA9-**

ObjectBroker for HP-UX Development:
QL-06WAX-**

ObjectBroker for HP-UX Run-Time Only:
QL-06XAX-**

ObjectBroker for IBM AIX Development:
QL-06YAX-**

ObjectBroker for IBM AIX Run-Time Only:
QL-06ZAX-**

ObjectBroker for SunOS Development:
QL-MGCAX-**

ObjectBroker for SunOS Run-Time Only:
QL-MGDAX-**

ObjectBroker for Solaris Development:
QL-4DDAX-**

ObjectBroker for Solaris Run-Time Only:
QL-4CAAX-**

ObjectBroker for UNIX Software Media and Documentation Kits

The ObjectBroker Development and Run-Time CD–ROMs include all supported platforms:

ObjectBroker Development and Run-time CD–ROM kit:
QA-0PKAA-H8

ObjectBroker Pilot Package (media/license combination):
QB-0PKAA-AA

ObjectBroker Software Documentation

Order the Same Documentation Kit for all ObjectBroker Products:
QA-0PKAA-GZ

ObjectBroker for UNIX Software Product Services

Orderable for Specific UNIX Operating Systems:

ObjectBroker for Digital UNIX Development:
QT-0TSA*-**

ObjectBroker for Digital UNIX Run-Time Only:
QT-0TTA*-**

ObjectBroker for HP-UX Development:
QT-06WA*-**

ObjectBroker for HP-UX Run-Time Only:
QT-06XA*-**

ObjectBroker for IBM AIX Development:
QT-06YA*-**

ObjectBroker for IBM AIX Run-Time Only:
QT-06ZA*-**

ObjectBroker for SunOS Development:
QT-MGCA*-**

ObjectBroker for SunOS Run-Time Only:
QT-MGDA*-**

ObjectBroker for Solaris Development:
QT-4DDA*-**

ObjectBroker for Solaris Run-Time Only:
QT-4CAA*-**

* 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.

License Management Facility Support

The Digital UNIX layered product supports the Digital UNIX License Management Facilities.

License units for these products are allocated on an Unlimited System Use basis.

For more information on the License Management Facility, refer to the Software Product Descriptions for the Digital UNIX Operating System (SPD 41.87.xx).

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.

® UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

™ CORBA is a trademark of Object Management Group.

™ Intel is a trademark of Intel Corporation.

™ SunOS is a trademark of Sun Microsystems, Inc.

™ Windows and Windows NT are trademarks of Microsoft Corporation.

™ The DIGITAL Logo, Alpha AXP, AXP, ObjectBroker, DECnet, DECstation, DECsystem, Digital, OpenVMS, PATHWORKS, TK, and VAX are trademarks of Digital Equipment Corporation.