

# Do you want your embedded system to Listen and Talk! ?<sup>TM</sup>

**SANKHYA Varadhi<sup>TM</sup>**  
Object Middleware for Embedded Systems

SANKHYA Varadhi is an Object Request Broker<sup>TM</sup> compliant with OMG's CORBA 2.2 (Minimum CORBA). Varadhi provides embedded systems developers a standards-based framework for building and deploying distributed systems. The tools and components provided by Varadhi manage the complexities of language, location and system/processor dependencies enabling the developers to concentrate on implementing the application logic in the client and server.

With its **small footprint, high performance and easy configurability**, Varadhi is ideally suited for systems where size and performance are important. Some of the applications for Varadhi include :

- **Telecommunications**
- **Intelligent Devices**
- **Data Networking**
- **Industrial Automation**
- **Internet Appliances**
- **Instrumentation and Control**

# SANKHYA Varadhi™

## Key Features

- Compliant with CORBA 2.2 (Minimum CORBA), including POA and supports interoperability with CORBA IOP (GIOP 1.1) protocol
- Small footprint
- Supports standard ORB options: -ORBid, -ORBInitRef
- Customizable ORB library for :
  - Static Allocation of Memory
  - Debug/profile Support
  - Including/excluding C++ exceptions
  - POA conforming to Minimum CORBA specification
- Supports co-location of the client and server

## Development Tools

- Support for OMG's Interface Definition Language (CORBA IDL) and provides an IDL compiler (*idlc*) for C++ binding
- Custom ORB and POA images can be generated using *vconf*, Varadhi's configurator
- Supports Name Server Manager, *nsmn* for bootstrapping Name Server from command line or script
- Support for IOR Tools to view IOP based stringified IOR's

## Services

- Includes Varadhi Names, Varadhi's Name Service conforming to OMG's CORBA 2.2 COSNaming Services Specification

## General

- Comes with complete PDF Documentation
- Quick to install and ready to use
- Includes samples that demonstrate IDL functions, Varadhi configuration, POA functionality, exception usage and Name Service usage
- Create custom applications quickly by modifying samples
- On-site training and consulting services available

## Supported Platforms

Varadhi is currently supported on the following development environments -

Hosts : Solaris 2.7\*, Linux 2.0.x, Windows NT\*.

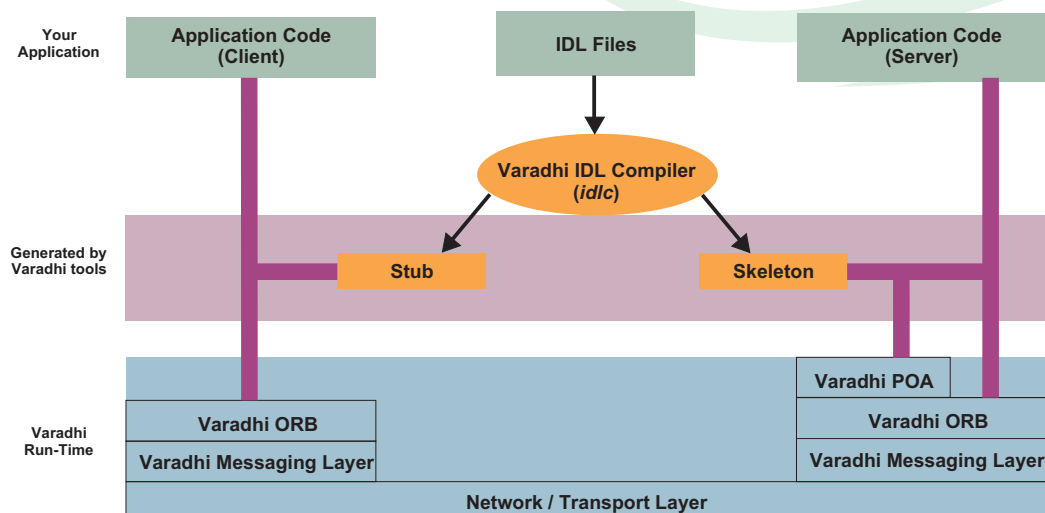
Compilers : g++ 2.95.2, VC++ 6.0\*

Targets : Varadhi can be easily ported to the following target platforms :

- ARM\*
- MIPS\*
- PowerPC\*
- 68K\* and x86\*.

For a simple demo program, typical code size for client and server on Linux 2.0.x running on Intel\* Pentium\* is of the order of 107 KB

Evaluate SANKHYA Varadhi : <http://www.sankhya.com/info/varadhi.html>.



## SANKHYA Varadhi - Technical Overview

Varadhi provides the following **development tools**:

- *idlc*, the IDL compiler that generates C++ stubs and skeletons
- *vconf*, interactive configuration tool
- IOR tool, DUMPIOR to view stringified IOR files

*idlc* generates C++ server skeletons and client stubs from the IDL specification provided by the user. It can also be used to just check the syntax of IDL files. *vconf* can be used to interactively configure and generate custom configurations of the ORB and POA.

Varadhi provides the following **run time components** :

- Object Request Broker (Varadhi ORB)
- Portable Object Adapter (POA)
- Client Stubs (generated by Varadhi IDL compiler from IDL)
- Server Skeletons (generated by Varadhi IDL compiler from IDL)

Varadhi has a modular and layered architecture. This provides easy **portability, extensibility and configurability** of the ORB. Extensibility and portability are achieved by abstraction of the transport and OS layer respectively. The abstraction of the OS layer allows easy portability to any embedded operating environment or real-time OS. The protocol layer is also abstracted so that environment specific Inter-operable protocols can be supported. Varadhi can be configured to use static memory allocation for various ORB objects. This enables the generation of an ORB image that has a statically determined foot-print, **essential for embedded systems**.

## SANKHYA Varadhi - Developing a Distributed Application

- Requirements Specification and Design
  - Describe object interface using IDL syntax. Verify using Varadhi IDL compiler, *idlc*.
- Implementation
  - Use *idlc*, Varadhi IDL compiler to generate necessary client stub and server skeleton
  - Use the Varadhi sample programs provided as templates to develop the client and server
- Test and Debug
  - Use the ORB trace and debug facility provided by Varadhi to test and debug your applications
  - Use IOR tools to view stringified IOR files
- Build and Deploy
  - Use *vconf*, Varadhi's configurator to create an appropriate image for your application

## SANKHYA Varadhi - Usage Scenarios

Here is an example of how Varadhi can be used in a building control application. A large number of Varadhi Building Automation Processing Elements (VPE) can be connected in a grid or any other topology. The VPEs export appropriate Control, Activation and Sensor objects as CORBA IDL interfaces to the neighboring VPEs using SANKHYA Varadhi.

The VPEs can communicate with each other to provide intelligent control of various facilities in the building. For example, when a person walks in to the building the VPE nearest to the person can detect and turn on the lamps in the room. These neighboring VPEs can then react suitably.

In a more complex scenario, the VPEs can use more sophisticated algorithms to `learn` the most likely route that will be taken by the person moving around the facility. Such a system when deployed in a large shopping mall can be used to predict the number of persons likely to visit a shop after a particular time.

This shows the ease with which CORBA can be used to build Intelligent Distributed Embedded Systems. Varadhi has been optimized for use in precisely such environments.

## About CORBA®

CORBA (Common Object Request Broker Architecture) is an open **distributed object computing infrastructure** standardized by OMG, the Object Management Group. The CORBA specification is based on OMG's Object Management Architecture (OMA) for object technology. CORBA specification is the standardized solution for reusability, portability and interoperability of object based software in distributed, heterogeneous environments. The ORB provides the mechanisms by which objects transparently make requests and receive responses, providing interoperability between applications on different machines in heterogeneous distributed environments.

## About Sankhya

Sankhya Technologies is one of India's leading System Software Development Companies. Sankhya is focused on research and development activities in the areas of software development tools and application platforms for embedded and enterprise application development. Sankhya is a Platform member of Object Management Group (OMG). Sankhya is a member of Software Technology Parks of India (STPI), National Association of Software and Service Companies (NASSCOM). Sankhya Technologies Private Limited is ISO 9001:2000 certified for "provision of system software products and development services".

## For More Information

<b>Varadhi Evaluation</b>	<a href="http://www.sankhya.com/info/varadhi.html">http://www.sankhya.com/info/varadhi.html</a>
<b>Varadhi Download</b>	<a href="http://www.sankhya.com/info/products/varadhi/download.html">http://www.sankhya.com/info/products/varadhi/download.html</a>
<b>Varadhi Road Map</b>	<a href="http://www.sankhya.com/info/products/varadhi/roadmap.html">http://www.sankhya.com/info/products/varadhi/roadmap.html</a>
<b>Varadhi Sales &amp; Support</b>	<a href="mailto:sales@sankhya.com">sales@sankhya.com</a> <a href="mailto:varadhi-support@sankhya.com">varadhi-support@sankhya.com</a>

**SANKHYA**<sup>TM</sup>

Sankhya Technologies Private Limited  
#30-15-58, Third Floor, "Silver Willow",  
DabaGardens, Visakhapatnam 530 020, INDIA  
Tel : +91 891 5542 666  
Fax : +91 891 5542 665  
<http://www.sankhya.com>

