

Introduction to RPC Programming

This chapter covers the following topics:

- RPC Technology
 - RPC-based Components
 - Reliable RPC
-

RPC Technology

A Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details. (A procedure call is also sometimes known as a function call or a subroutine call.)

RPC uses the client/server model. The requesting program is a client and the service-providing program is the server. Like a regular or local procedure call, an RPC is a synchronous operation requiring the requesting program to be suspended until the results of the remote procedure are returned. However, the use of lightweight processes or threads that share the same address space allows multiple RPCs to be performed concurrently.

When program statements that use RPC are compiled into an executable program, an interface object is included in the compiled code that acts as the representative of the remote procedure code. When the program is run and the procedure call is issued, the interface object receives the request and forwards it to a client runtime program in the local computer. The client runtime program has the knowledge of how to address the remote computer and server application and sends the message across the network that requests the remote procedure. Similarly, the server includes a runtime program and interface object that interface with the remote procedure itself. Results are returned the same way.

Some examples of RPC technology are Software AG's EntireX/Natural RPC, Microsoft RPC, and DCE RPC.

RPC-based Components

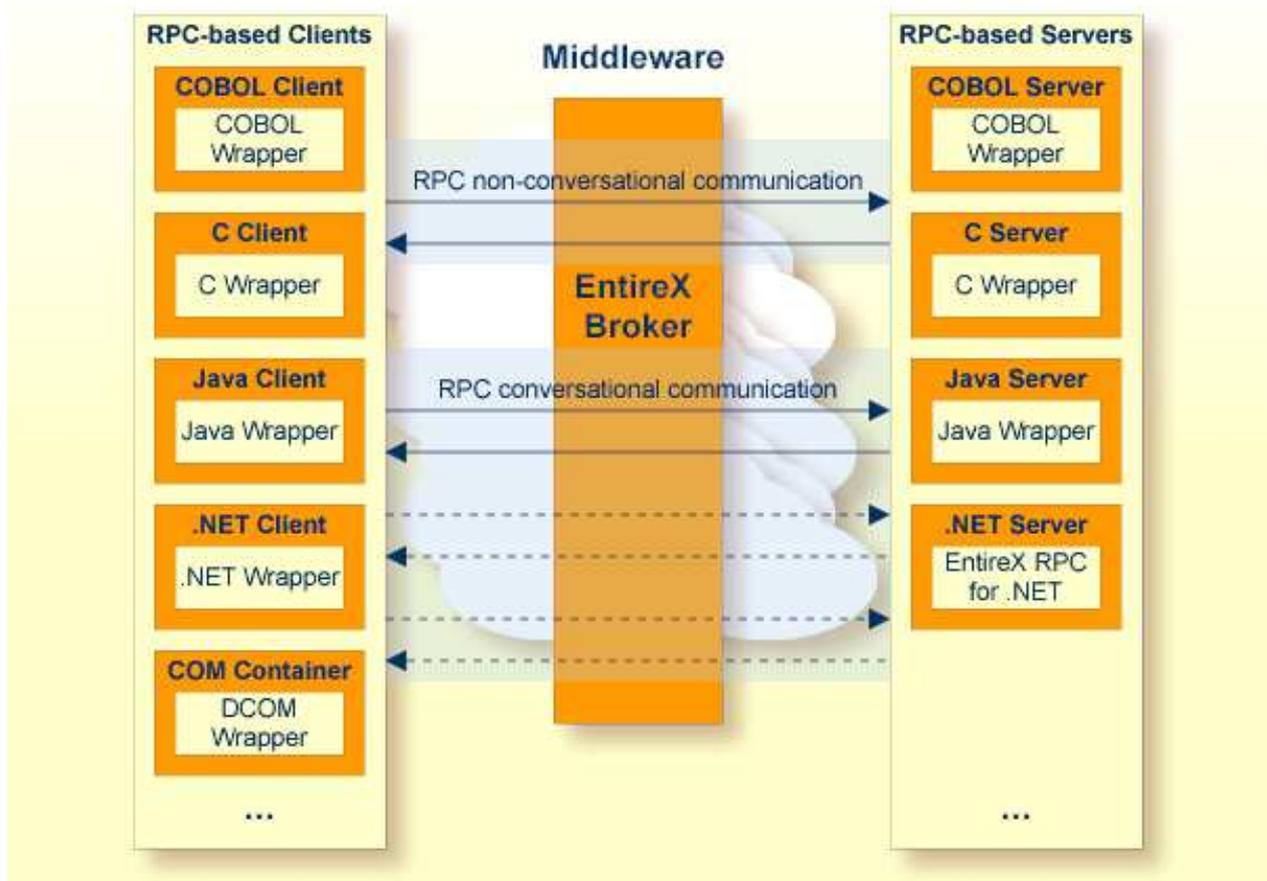
This section covers the following topics:

- Introduction
- Advantages of RPC-based Components
- Connectivity Matrix

Introduction

The production of RPC-based components is called "wrapping" (Java Wrapper, XML/SOAP Wrapper, DCOM Wrapper, .NET Wrapper etc.). The wrapped components are perfectly embedded in their environments, for example:

- in C as functions and procedures
- in Java and .NET as classes and methods
- in COBOL and Natural as subprograms
- in COM container-enabled applications as COM/DCOM objects



Advantages of RPC-based Components

- The programmer can work with familiar data types without having to worry about their representation on different hardware platforms, including conversion of codepages, etc.
- RPC-based components use the *EntireX Interface Definition Language (IDL)* to create programming-language-independent interfaces between client and server components. See *Software AG IDL File*.
- The Software AG IDL file can be automatically created from Natural subprograms, COBOL and PL/I. See *IDL Extractor for Natural*, *IDL Extractor for COBOL*, *IDL Extractor for PL/I*.
- There are generation tools for the RPC-based client and server components (stubs, skeletons, etc.). See the *EntireX Workbench* and the *IDL Compiler*.

- RPC-based components support non-conversational and conversational RPC communication.
- EntireX RPC-based components are compatible with Natural RPC. For Natural RPC servers, see *Setting Up a Natural RPC Environment* in your Natural documentation.

Connectivity Matrix

Connection	Feature					
	Client Extraction	Client Generation	Server Extraction	Server Generation	Client Connectivity	Server Connectivity
Web Service	x	x	x	x	x	x
XML Service	x	x	x	x	x	x
Java		x		x	x	x
.NET		x		x	x	x
PHP, Perl, Ruby	via WSDL/XSD				x	
COBOL		x	x	x	x	x
PL/I		x	x	x	x	x
Natural	x	x	x	x	x	x
RPG						x
CL						x
C		x		x	x	x
Assembler						
WebSphere MQ		x		x	x	x
webMethods Integration Server		x	x	x	x	x

Reliable RPC

In the architecture of modern e-business applications (such as SOA), loosely coupled systems are becoming more and more important. Reliable messaging is one important technology for this type of system.

Reliable RPC is the EntireX implementation of a reliable messaging system. It combines EntireX RPC technology and persistence, which is implemented with units of work (UOWs).

- Reliable RPC allows asynchronous calls ("fire and forget")
- Reliable RPC is supported by most EntireX wrappers

- Reliable RPC messages are stored in the Broker’s persistent store until a server is available
- Reliable RPC clients are able to request the status of the messages they have sent

