The DCOM Connector

HELP.BCMIDDCOM

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
SYN	Syntax
💡	Tip

Contents

The DCOM Connector

DCOM Connector: Overview

Purpose

DCOM provides a standardized basis for creating and integrating distributed software components. External client applications can use DCOM as infrastructure for their communication with SAP servers, to access R/3 components.
With this technology, R/3 components such as BAPIs or remote function modules appear as COM objects and use the runtime environment of the **M**icrosoft **T**ransaction **S**erver (**MTS**).
The DCOM Connector provides a C++ Template Library developed by SAP, which you can use to create COM objects from business objects of the R/3 system and administer them.

The two scenarios below describe the different uses of the DCOM Connector:

PC Client Applications

Client applications on a frontend use DCOM to access COM objects installed on one or more MTS servers. In such a scenario, all you need to do is distribute the client application software to different PCs. It is not necessary to install any additional SAP software on the PC, since MTS components are accessed remotely.

The DCOM Connector allows you to

- Concentrate on programming the client application without having to consider the RFC protocol.
- Simplify table handling with ADO using standard tools (for example, Data Bound Control).
- Use comfortable development environments (VB 5/6) for developing the clients.

WEB Scenario

You use **A**ctive **S**erver **P**ages (ASP) to create WEB pages whose HTML coding contains executable scripts on the server side. ASP communicates with the MTS via DCOM to access MTS components. The MTS can run on the same machine as the web server (IIS), but this is no requirement.

In this scenario, the advantages of resource pooling in the DCOM Connector apply:

- High performance of the R/3 accesses results in shorter response times.
- Reducing the number of user contexts in R/3 reduces the storage required and the rollin/rollout activities for the R/3 access.

Implementation Considerations

More Documentation

An up-to-date documentation is installed together with the DCOM Connector, or you can copy the required files from the **NT RFC SDK** of the **presentation CD**. The documentation contains these topics [Page 9].

For an introduction to programming client applications based on the DCOM technology, refer to the tutorial Programming DCOM Clients in Visual Basic [Ext.].

Features

The Template Library offers these features:

- **Strongly typed compile time object**:
  The binary description of the interfaces an R/3 component opens up is stored in a **type library**. The type library is part of the generated proxy objects, which allows you to access meta information from the R/3 system during the design phase as well as duing the client runtime. You can diplay the interface definition in the development environment. In addition, during runtime no access to the SAP system is required. This increases performance as compared to solutions that must retrieve their meta data at runtime.
- **ABAP data types** are automatically mapped onto the respective OLE data types, and vice versa.
- **Tables and structures** from R/3 are represented by ADO Recordsets.
- **COM versus RFC:**
  Using DCOM between the client program and the MTS server allows you to distribute intranet applications without having to provide the SAP DLL and the corresponding ActiveX controls. In addition, the net-wide maintenance of *services, samsg.ini* and *saprout.ini* files is not required.
- **Dual Interface:**
  VB5 (and higher) or C++ can use the dual interface to access methods directly, that is, to bypass the Automation interfaces IDispatch.
  VB4 or the Scripting languages access via IDispatch.
- **Resource pooling** allows you to administer resources perfectly and thus to achieve a very high performance of the R/3 accesses.
- There is **no BOR runtime**, since the BAPIs are called directly via calls of the function modules that define them.
- DCOM Connector uses the security services of the MTS to guarantee **access security**.
- **BOR names** for methods, key fields, and parameters are copied completely and case-sensitively.
- Generating DCOM proxies and installing them on the MTS (as host for COM objects) is controlled via a user interface called **DCOM Object Builder**. This allows you to minimize the effort required to create DCOM proxies and concentrate mainly on programming the clients.
- Generating and installing a **session object** using the Object Builder.

Supporting Programming Environments

The generated objects can be used by all programming languages that support COM.

SAP tested these environments for developing client applications:

- Visual C++
- Visual Basic and VBA
- Java (Microsoft Virtual Machine)
- Scripting environments:
  - Active Server Pages (ASP)
  - Java Script

- VB Script

Contraints

At present, you can not yet generate proxy objects from ABAP objects of the R/3 class library.

Topics of the Documentation

To find these documentations, refer to the **NT RFC SDK** of the **SAP Presentation CD**.

DCOM Connector: Concepts

Overview

Resource Pooling

Transactions

The Session Objekt

Type Conversion

Security Concept

Installation

Installation of the DCOM Connector

Administration and Destination Management

Maintaining Destinations

Displaying Meta Data of R/3 Components

DCOM Object Builder

Object Builder: Overview

Architecture

Generation Process

Installation Process

Generating and Installing DCOM Objects

Programming Clients

Basic Steps

Using Visual Basic

Tips and Notes

Formatting Dates and Times

Error Handling in the Client Program

Reference to the Type Library

Basic Structure of a Program

Sample Program: "SalesOrder"

Tutorial: Programming DCOM Clients in VB

Glossary