

## Installation

### Archives

RSrv250.exe

setup program for server and sample code (including runtime libraries)

RSrv250S.zip

VC++ 6.0 and VB sources for server and controls, CVS access is in work

### Installation

**R (D)COM Server** installs COM-servers and Active X controls. On most Windows installations, this requires the user running setup to be a member of at least the *Power Users* or *Administrators* group. Please contact your system administrator if your user account is not a member of one of these groups. (More specifically, the setup requires write access in the computer's registry database under `HKEY_CLASSES_ROOT`.)

- Install the latest **R** release ( $\geq$  **R 2.2.0**).
- If a previous version of **R (D)COM Server** is currently installed, please uninstall this version first
  - close all client applications using the COM server
  - uninstall the previous version using "Add/Remove Programs"
- Start the setup program
- **R (D)COM Server** installation

- Which Package shall I Install?

`RSrv200.exe` is a full installer and provides all required components including runtime libraries for Visual Basic and Visual C++. Previous versions provided an installer without the runtime libraries, too. This isn't supported any more.

If you don't want to make changes to the server software you won't have to download and install the sources either. All information required for creating a client application using the COM server is included in the documentation and the sample code installed with the COM server.

- install `RSrv200.exe`.

Installation allows you to choose between various default types of installation.

- If you want to develop client applications, you should choose a "Development installation". This will install binaries for running local and remote servers, test and sample applications, including source code. An installation of **R** is required on this machine, too.

When deploying your own client application to some machine, you have to install **R (D)COM Server** on this machine, too.

- If running **R** locally on this machine, choose "Local Server Installation". This will install binaries for the local COM servers and a simple test application used for testing the basic functionality of **R** and the **R (D)COM Server** installation. A local installation of **R** is required, too. This machine can be used for running client applications using a local **R** server and for providing **R** server objects for remote clients, too.
- For running a client application locally and accessing **R** objects installed on a remote server, choose "Remote Server Files" as your setup option. This will set up necessary binaries for accessing an installation of **R** and **R (D)COM server** on a remote machine. **R** is not required on this machine

- start 01 - Basic Test from the **R (D)COM Server** program group in the start menu and press "Start".

## Testing the (D)COM Server

Lots of example and test programs are provided with the **R (D)COM Server**.

One of the test programs is always installed with the **R (D)COM Server**. It can be found in the Start Menu. The default program group is called `R\ (D)COM Server`, the test program found there is called `01 - Basic Test`.

Applications using the **R (D)COM Server** always perform very similar steps:

- create an instance of the `IStatConnector` interface
- call `Init` on the COM object to start up **R**
- perform the application's work
- shut down the **R** by calling `Close`
- release the COM object

The test program just shows some diagnostic information and performs some tests:

- data transfer to **R**
- perform some computations in **R**
- data transfer from **R**

## Troubleshooting

If anything goes wrong when calling `.Init()` of the COM object for the first time, retrieve error information using `.GetErrorText()`:

"installation problem: unable to load connector"

**R** proxy library could not be loaded. The cause of this common failure can be:

- the environment variable `%R_HOME%` points to a directory where **R** is not installed *and*
- the registry key corresponding to your **R** installation does not point to the installation folder of **R** *and*
- `rproxy.dll` and `R.dll` cannot be found in the `%PATH%`

In case you have downloaded precompiled binaries from CRAN and installed them, something went wrong there. Please try to reinstall **R** using the setup program. Download "dbgview" (see section "Resources") to find out what's going wrong.

If you have compiled **R** yourself, check `%R_HOME%\bin\rproxy.dll` and the registry key `HKEY_LOCAL_MACHINE\Software\R-core\R\InstallPath` if this one points to your **R** installation folder. Starting with **R** 2.2.0, the registry key `HKEY_CURRENT_USER\Software\R-core\R\InstallPath` can also be used. The **R (D)COM Server** first looks at the key under `HKEY_CURRENT_USER`.

If the registry key is missing or wrong, re-install **R** and do not forget to check the box asking about putting information about **R** into the registry. Alternatively, you can call `RSetReg.exe` from a command window.

"installation problem: invalid connector library"

possibly damaged **R** proxy library. This version of the COM server requires **R**  $\geq$  2.2.

"installation problem: interpreter interface version mismatch"

wrong version of **R** proxy library. This version of the COM server requires **R**  $\geq$  2.2.

"installation problem: interpreter version mismatch"

wrong version of **R**. This version of the COM server requires **R**  $\geq$  2.2.

If a message box shows up on your call to `.Init()`, which tells you that the base library could not be found, you probably should set `R_HOME` to your installation folder. This should rarely be necessary.

If you cannot solve the installation problem by yourself, please try to find help on the **R** COM mailing list (see "Resources"). Subscribing to this list is recommended if you want to use/are using the **R** COM server (or any other part of **R**'s COM connectivity modules).

All functions will return an error code corresponding to the texts shown above. This error code is the return value of the functions when called e.g. from C or C++ or can be retrieved by checking `Err.Number` in VB or VBA. The following table will list the error codes (defines for C/C++ and numbers)

Define	Decimal	Hexadecimal
SCN_E_INVALIDARG	-2147221503	0x80040001
SCN_E_INVALIDFORMAT	-2147221502	0x80040002
SCN_E_NOTIMPL	-2147221501	0x80040003
SCN_E_UNKNOWN	-2147221500	0x80040004
SCN_E_INITIALIZED	-2147221499	0x80040005
SCN_E_NOTINITIALIZED	-2147221498	0x80040006
SCN_E_INVALIDSYMBOL	-2147221497	0x80040007
SCN_E_PARSE_INVALID	-2147221496	0x80040008
SCN_E_PARSE_INCOMPLETE	-2147221495	0x80040009
SCN_E_UNSUPPORTEDTYPE	-2147221494	0x8004000A
SCN_E_EVALUATE_STOP	-2147221493	0x8004000B
SCN_E_INVALIDINTERFACEVERSION	-2147221488	0x80040010
SCN_E_INVALIDINTERPRETERVERSION	-2147221487	0x80040011
SCN_E_INTERFACENOTFOUND	-2147221486	0x80040012
SCN_E_LIBRARYNOTFOUND	-2147221485	0x80040013
SCN_E_INVALIDLIBRARY	-2147221484	0x80040014
SCN_E_INITIALIZATIONFAILED	-2147221483	0x80040015
SCN_E_INVALIDCONNECTORNAME	-2147221482	0x80040016
SCN_E_INVALIDINTERPRETERSTATE	-2147221481	0x80040017
SCN_E_FATALBACKEND	-2147221472	0x80040020

### Registry Information

The setup program will add information to the registry. This information can be found in

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\R (D)COM Server_is1.`

In addition to that, various COM servers are registered.

### Re-Packaging and Silent installation

Taking files from the setup (e.g. only the (D)COM Server executable) and creating your own installer (or adding the files to a 3rd party setup program) is **not allowed**. One of the reasons for this is that using the "official" setup application guarantees that everything is installed correctly.

If you want to integrate the (D)COM Server and/or RExcel into your own application you can freely distribute the (D)COM Server setup with your setup program. To automatically install the (D)COM Server and RExcel with your application, you can call the setup program in "silent" mode (see the next section for more information).

To choose the set of components to be installed for silent installation, use the following component names:

`localserver`

Local Server Files

`simpletest`

Simple Server Test Files

`remoteserver`

Remote Server Files

`development`

## Development Files

samples

(D)COM Server Sample Files

excel

RExcel: Excel Add-In

excelsamples

RExcel Sample Files

## Setup Command Line Parameters

**This section is quoted ``as is'' from the documentation of Inno Setup version 5.0.8:**

The Setup program accepts optional command line parameters. These can be useful to system administrators, and to other programs calling the Setup program.

**/SP-**

Disables the *This will install... Do you wish to continue?* prompt at the beginning of Setup. Of course, this will have no effect if the DisableStartupPrompt [Setup] section directive was set to yes.

**/SILENT, /VERYSILENT**

nstructs Setup to be silent or very silent. When Setup is silent the wizard and the background window are not displayed but the installation progress window is. When a setup is very silent this installation progress window is not displayed. Everything else is normal so for example error messages during installation are displayed and the startup prompt is (if you haven't disabled it with DisableStartupPrompt or the '/SP-' command line option explained above)

If a restart is necessary and the '/NORESTART' command isn't used (see below) and Setup is silent, it will display a Reboot now? message box. If it's very silent it will reboot without asking.

**/SUPPRESSMSGBOXES**

Instructs Setup to suppress message boxes. Only has an effect when combined with '/SILENT' and '/VERYSILENT'. The default response in situations where there's a choice is:

Yes in a 'Keep newer file?' situation.

No in a 'File exists, confirm overwrite.' situation.

Abort in Abort/Retry situations.

Cancel in Retry/Cancel situations.

Yes (=continue) in a

DiskSpaceWarning/DirExists/DirDoesntExist/NoUninstallWarning/ExitSetupMessage/ConfirmUninstall situation.

Yes (=restart) in a FinishedRestartMessage/UninstalledAndNeedsRestart situation.

5 message boxes are not suppressible:

The About Setup message box.

The Exit Setup? message box.

The FileNotInDir2 message box displayed when Setup requires a new disk to be inserted and the disk was not found.

Any (error) message box displayed before Setup (or Uninstall) could read the command line parameters.

Any message box displayed by [Code] support function MsgBox.

**/LOG**

Causes Setup to create a log file in the user's TEMP directory detailing file installation and [Run] actions taken during the installation process. This can be a helpful debugging aid. For example, if you suspect a file isn't being replaced when you believe it should be (or vice versa), the log file will tell you if the file was really

skipped, and why.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be machine-parseable; the format of the file is subject to change without notice.

`/LOG="filename"`

Same as `/LOG`, except it allows you to specify a fixed path/filename to use for the log file. If a file with the specified name already exists it will be overwritten. If the file cannot be created, Setup will abort with an error message.

`/NOCANCEL`

Prevents the user from cancelling during the installation process, by disabling the Cancel button and ignoring clicks on the close button. Useful along with `'/SILENT'` or `'/VERYSILENT'`.

`/NORESTART`

Instructs Setup not to reboot even if it's necessary.

`/RESTARTEXITCODE=exit code`

Specifies the custom exit code that Setup is to return when a restart is needed. Useful along with `'/NORESTART'`. Also see Setup Exit Codes.

`/LOADINF="filename"`

Instructs Setup to load the settings from the specified file after having checked the command line. This file can be prepared using the `'/SAVEINF='` command as explained below.

Don't forget to use quotes if the filename contains spaces.

`/SAVEINF="filename"`

Instructs Setup to save installation settings to the specified file.

Don't forget to use quotes if the filename contains spaces.

`/LANG=language`

Specifies the language to use. language specifies the internal name of the language as specified in a [Languages] section entry.

When a valid `/LANG` parameter is used, the Select Language dialog will be suppressed.

`/DIR="x:\dirname"`

Overrides the default directory name displayed on the Select Destination Location wizard page. A fully qualified pathname must be specified.

`/GROUP="folder name"`

Overrides the default folder name displayed on the Select Start Menu Folder wizard page. If the [Setup] section directive `DisableProgramGroupPage` was set to yes, this command line parameter is ignored.

`/NOICONS`

Instructs Setup to initially check the Don't create any icons check box on the Select Start Menu Folder wizard page.

`/COMPONENTS="comma separated list of component names"`

Overrides the default components settings. Using this command line parameter causes Setup to automatically

select a custom type

`/PASSWORD=password`

Specifies the password to use. If the [Setup] section directive Password was not set, this command line parameter is ignored.

When an invalid password is specified, this command line parameter is also ignored.

---

[Package [Index](#)]